实时音视频

客户端 API



腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

客户端 API
简介
iOS&Mac
API 概览
TRTCCloud
TRTCCloudDelegate
TRTCStatistics
TXAudioEffectManager
TXBeautyManager
TXDeviceManager
类型定义
废弃接口
错误码表
Android
API 概览
TRTCCloud
TRTCCloudListener
TRTCStatistics
TXAudioEffectManager
TXBeautyManager
TXDeviceManager
类型定义
废弃接口
错误码表
全平台（C++）
API 概览
ITRTCCloud
TRTCCloudCallback
ITRTCStatistics
ITXAudioEffectManager
ITXDeviceManager
类型定义
废弃接口
错误码表
Windows（C#）

版权所有：腾讯云计算（北京）有限责任公司

API 概览
错误码
Windows（ActiveX）
API 概览
错误码
Web
API 概览
错误码
Electron
API 概览
错误码
小程序
trtc-wx
错误码
Flutter
API 概览
错误码
Uni-App
API 概览
错误码

客户端 API
简介

最近更新时间：2023-06-29 16:36:52

欢迎使用实时音视频客户端无 UI 集成方案，来实现您的业务需求。腾讯实时音视频（Tencent Real-Time Communication，TRTC），将腾讯多年来在网络与音视频技术上的深度积累，以多人音视频通话和低延时互动直播两大场景化方案，通过腾讯云服务向开发者开放，致力于帮助开发者快速搭建低成本、低延时、高品质的音视频互动解决方案。
目前无 UI 集成方案，主要面向需要自定义 UI、业务高度灵活的客户。您可以通过无 UI 文档的介绍，来快速了解如何使用 TRTC SDK 实现实时音视频房间相关的基础功能。

> **注意：**
> 目前 TRTC SDK 仅提供媒体服务相关的能力和 API，如果您想使用 组件介绍 (TUIRoomKit) 产品，想要基于 TUIRoomKit 的能力通过自定义 UI 来实现对应的业务，又或者您的业务期望能快速实现有类似媒体控制（如禁言、禁画），麦位管理（如上下麦、锁麦），以及房间成员管理（邀请、移人）等能力支持，您可以参考我们的多人音视频 SDK（TUIRoomEngine）自定义 UI 集成 方案来实现。

iOS&Mac API 概览

最近更新时间：2024-03-06 14:51:31

API OVERVIEW

创建实例和事件回调

函数列表	描述
sharedInstance	创建 TRTCCloud 实例（单例模式）
destroySharedInstance	销毁 TRTCCloud 实例（单例模式）
addDelegate:	添加 TRTC 事件回调
removeDelegate:	移除 TRTC 事件回调
delegateQueue	设置驱动 TRTCCloudDelegate 事件回调的队列

房间相关接口函数

函数列表	描述
enterRoom:appScene:	进入房间
exitRoom	离开房间
switchRole:	切换角色
switchRole:privateMapKey:	切换角色（支持设置权限位）
switchRoom:	切换房间
connectOtherRoom:	请求跨房通话
disconnectOtherRoom	退出跨房通话
setDefaultStreamRecvMode:video:	设置订阅模式（需要在进入房前设置才能生效）

createSubCloud	创建子房间实例（用于多房间并发观看）
destroySubCloud:	销毁子房间实例
updateOtherRoomForwardMode:	更改跨房主播在本房间的上行能力

CDN 相关接口函数

函数列表	描述
startPublishing:type:	开始向腾讯云直播 CDN 上发布音视频流
stopPublishing	停止向腾讯云直播 CDN 上发布音视频流
startPublishCDNStream:	开始向非腾讯云 CDN 上发布音视频流
stopPublishCDNStream	停止向非腾讯云 CDN 上发布音视频流
setMixTranscodingConfig:	设置云端混流的排版布局和转码参数
startPublishMediaStream:encoderParam:mixingConfig:	开始发布媒体流
updatePublishMediaStream:publishTarget:encoderParam:mixingConfig:	更新发布媒体流
stopPublishMediaStream:	停止发布媒体流

视频相关接口函数

函数列表	描述
startLocalPreview:view:	开启本地摄像头的预览画面（移动端）
startLocalPreview:	开启本地摄像头的预览画面（桌面端）
updateLocalView:	更新本地摄像头的预览画面
stopLocalPreview	停止摄像头预览
muteLocalVideo:mute:	暂停/恢复发布本地的视频流
setVideoMuteImage:fps:	设置本地画面被暂停期间的替代图片
startRemoteView:streamType:view:	订阅远端用户的视频流，并绑定视频渲染控件
updateRemoteView:streamType:forUs	更新远端用户的视频渲染控件

er:	
stopRemoteView:streamType:	停止订阅远端用户的视频流，并释放渲染控件
stopAllRemoteView	停止订阅所有远端用户的视频流，并释放全部渲染资源
muteRemoteVideoStream:streamType:mute:	暂停/恢复订阅远端用户的视频流
muteAllRemoteVideoStreams:	暂停/恢复订阅所有远端用户的视频流
setVideoEncoderParam:	设置视频编码器的编码参数
setNetworkQosParam:	设置网络质量控制的相关参数
setLocalRenderParams:	设置本地画面的渲染参数
setRemoteRenderParams:streamType:params:	设置远端画面的渲染模式
enableEncSmallVideoStream:withQuality:	开启大小画面双路编码模式
setRemoteVideoStreamType:type:	切换指定远端用户的大小画面
snapshotVideo:type:sourceType:	视频画面截图
setPerspectiveCorrectionWithUser:srcPoints:dstPoints:	视频画面透视校正坐标设置
setGravitySensorAdaptiveMode:	设置重力感应的适配模式（11.7 及以上版本）

音频相关接口函数

函数列表	描述
startLocalAudio:	开启本地音频的采集和发布
stopLocalAudio	停止本地音频的采集和发布
muteLocalAudio:	暂停/恢复发布本地的音频流
muteRemoteAudio:mute:	暂停/恢复播放远端的音频流
muteAllRemoteAudio:	暂停/恢复播放所有远端用户的音频流
setAudioRoute:	设置音频路由

setRemoteAudioVolume:volume:	设定某一个远端用户的声音播放音量
setAudioCaptureVolume:	设定本地音频的采集音量
getAudioCaptureVolume	获取本地音频的采集音量
setAudioPlayoutVolume:	设定远端音频的播放音量
getAudioPlayoutVolume	获取远端音频的播放音量
enableAudioVolumeEvaluation:withParams:	启用音量大小提示
startAudioRecording:	开始录音
stopAudioRecording	停止录音
startLocalRecording:	开启本地媒体录制
stopLocalRecording	停止本地媒体录制
setRemoteAudioParallelParams:	设置远端音频流智能并发播放策略
enable3DSpatialAudioEffect:	启用 3D 音效
updateSelf3DSpatialPosition	设置 3D 音效中自身坐标及朝向信息
updateRemote3DSpatialPosition:	设置 3D 音效中远端用户坐标信息
set3DSpatialReceivingRange:range:	设置指定用户所发出声音的可被接收范围

设备管理相关接口

函数列表	描述
getDeviceManager	获取设备管理类（TXDeviceManager）

美颜特效和图像水印

函数列表	描述
getBeautyManager	获取美颜管理类（TXBeautyManager）

setWatermark:streamType:rect:	添加水印

背景音乐和声音特效

函数列表	描述
getAudioEffectManager	获取音效管理类（TXAudioEffectManager）
startSystemAudioLoopback	开启系统声音采集（iOS 端暂未支持）
stopSystemAudioLoopback	停止系统声音采集（iOS 端暂未支持）
setSystemAudioLoopbackVolume:	设置系统声音的采集音量

屏幕分享相关接口

函数列表	描述
startScreenCaptureInApp:encParam:	开始应用内的屏幕分享（仅支持 iOS 13.0 及以上系统）
startScreenCaptureByReplaykit:encParam:appGroup:	开始全系统的屏幕分享（仅支持 iOS 11.0 及以上系统）
startScreenCapture:streamType:encParam:	启动屏幕分享
stopScreenCapture	停止屏幕分享
pauseScreenCapture	暂停屏幕分享
resumeScreenCapture	恢复屏幕分享
getScreenCaptureSourcesWithThumbnailSize:iconSize:	枚举可分享的屏幕和窗口（该接口仅支持 Mac OS 系统）
selectScreenCaptureTarget:rect:capturesCursor:highlight:	选取要分享的屏幕或窗口（该接口仅支持 Mac OS 系统）
setSubStreamEncoderParam:	设置屏幕分享（即辅路）的视频编码参数（桌面系统和移动系统均已支持）
setSubStreamMixVolume:	设置屏幕分享时的混音音量大小（该接口仅支持

	桌面系统）
addExcludedShareWindow:	将指定窗口加入屏幕分享的排除列表中（该接口仅支持桌面系统）
removeExcludedShareWindow:	将指定窗口从屏幕分享的排除列表中移除（该接口仅支持桌面系统）
removeAllExcludedShareWindows	将所有窗口从屏幕分享的排除列表中移除（该接口仅支持桌面系统）
addIncludedShareWindow:	将指定窗口加入屏幕分享的包含列表中（该接口仅支持桌面系统）
removeIncludedShareWindow:	将指定窗口从屏幕分享的包含列表中移除（该接口仅支持桌面系统）
removeAllIncludedShareWindows	将全部窗口从屏幕分享的包含列表中移除（该接口仅支持桌面系统）

自定义采集和自定义渲染

函数列表	描述
enableCustomVideoCapture:enable:	启用/关闭视频自定义采集模式
sendCustomVideoData:frame:	向 SDK 投送自己采集的视频帧
enableCustomAudioCapture:	启用音频自定义采集模式
sendCustomAudioData:	向 SDK 投送自己采集的音频数据
enableMixExternalAudioFrame:playout:	启用/关闭自定义音轨
mixExternalAudioFrame:	向 SDK 混入自定义音轨
setMixExternalAudioVolume:playoutVolume:	设置推流时混入外部音频的推流音量和播放音量
generateCustomPTS	生成自定义采集时的时间戳
setLocalVideoProcessDelegete:pixelFormat:bufferType:	设置第三方美颜的视频数据回调
setLocalVideoRenderDelegate:pixelFormat:bufferType:	设置本地视频自定义渲染回调
setRemoteVideoRenderDelegate:deleg	设置远端视频自定义渲染回调

ate:pixelFormat:bufferType:	
setAudioFrameDelegate:	设置音频数据自定义回调
setCapturedAudioFrameDelegateFormat:	设置本地麦克风采集出的音频帧回调格式
setLocalProcessedAudioFrameDelegateFormat:	设置经过前处理后的本地音频帧回调格式
setMixedPlayAudioFrameDelegateFormat:	设置最终要由系统播放出的音频帧回调格式
enableCustomAudioRendering:	开启音频自定义播放
getCustomAudioRenderingFrame:	获取可播放的音频数据

自定义消息发送接口

函数列表	描述
sendCustomCmdMsg:data:reliable:ordered:	使用 UDP 通道发送自定义消息给房间内所有用户
sendSEIMsg:repeatCount:	使用 SEI 通道发送自定义消息给房间内所有用户

网络测试接口

函数列表	描述
startSpeedTest:	开始进行网速测试（进入房间前使用）
stopSpeedTest	停止网络测速

调试相关接口

函数列表	描述
getSDKVersion	获取 SDK 版本信息
setLogLevel:	设置 Log 输出级别
setConsoleEnabled:	启用/禁用控制台日志打印
setLogCompressEna	启用/禁用日志的本地压缩

bled:	
setLogDirPath:	设置本地日志的保存路径
setLogDelegate:	设置日志回调
showDebugView:	显示仪表盘
setDebugViewMargin:margin:	设置仪表盘的边距
callExperimentalAPI:	调用实验性接口

加密接口

函数列表	描述
enablePayloadPrivateEncryption:params:	开启或关闭媒体流私有加密

错误和警告事件

函数列表	描述
onError:errMsg:extInfo:	错误事件回调
onWarning:warningMsg:extInfo:	警告事件回调

房间相关事件回调

函数列表	描述
onEnterRoom:	进入房间成功与否的事件回调
onExitRoom:	离开房间的事件回调
onSwitchRole:errMsg:	切换角色的事件回调
onSwitchRoom:errMsg:	切换房间的结果回调
onConnectOtherRoom:errCode:errMsg:	请求跨房通话的结果回调
onDisconnectOtherRoom:errMsg:	结束跨房通话的结果回调

版权所有：腾讯云计算（北京）有限责任公司

onUpdateOtherRoomForwardMode:errMsg:	更改跨房主播上行能力的结果回调

用户相关事件回调

函数列表	描述
onRemoteUserEnterRoom:	有用户加入当前房间
onRemoteUserLeaveRoom:reason:	有用户离开当前房间
onUserVideoAvailable:available:	某远端用户发布/取消了主路视频画面
onUserSubStreamAvailable:available:	某远端用户发布/取消了辅路视频画面
onUserAudioAvailable:available:	某远端用户发布/取消了自己的音频
onFirstVideoFrame:streamType:width:height:	SDK 开始渲染自己本地或远端用户的首帧画面
onFirstAudioFrame:	SDK 开始播放远端用户的首帧音频
onSendFirstLocalVideoFrame:	自己本地的首个视频帧已被发布出去
onSendFirstLocalAudioFrame	自己本地的首个音频帧已被发布出去
onRemoteVideoStatusUpdated:streamType:streamStatus:reason:extrainfo:	远端视频状态变化的事件回调
onRemoteAudioStatusUpdated:streamStatus:reason:extrainfo:	远端音频状态变化的事件回调
onUserVideoSizeChanged:streamType:newWidth:newHeight:	用户视频大小发生改变回调

网络和技术指标统计回调

函数列表	描述
onNetworkQuality:remoteQuality:	网络质量的实时统计回调
onStatistics:	音视频技术指标的实时统计回调
onSpeedTestResult:	网速测试的结果回调

与云端连接情况的事件回调

函数列表	描述
onConnectionLost	SDK 与云端的连接已经断开
onTryToReconnect	SDK 正在尝试重新连接到云端
onConnectionRecovery	SDK 与云端的连接已经恢复

硬件设备相关事件回调

函数列表	描述
onCameraDidReady	摄像头准备就绪
onMicDidReady	麦克风准备就绪
onAudioRouteChanged:fromRoute:	当前音频路由发生变化（仅适用于移动设备）
onUserVoiceVolume:totalVolume:	音量大小的反馈回调
onDevice:type:stateChanged:	本地设备的通断状态发生变化（仅适用于桌面系统）
onAudioDeviceCaptureVolumeChanged:muted:	当前麦克风的系统采集音量发生变化
onAudioDevicePlayoutVolumeChanged:muted:	当前系统的播放音量发生变化
onSystemAudioLoopbackError:	系统声音采集是否被成功开启的事件回调（仅适用于 Mac 系统）

自定义消息的接收事件回调

函数列表	描述
onRecvCustomCmdMsgUserId:cmdID:seq:message:	收到自定义消息的事件回调
onMissCustomCmdMsgUserId:cmdID:errCode:missed:	自定义消息丢失的事件回调
onRecvSEIMsg:message:	收到 SEI 消息的回调

CDN 相关事件回调

函数列表	描述
onStartPublishing:errMsg:	开始向腾讯云直播 CDN 上发布音视频流的事件回调
onStopPublishing:errMsg:	停止向腾讯云直播 CDN 上发布音视频流的事件回调
onStartPublishCDNStream:errMsg:	开始向非腾讯云 CDN 上发布音视频流的事件回调
onStopPublishCDNStream:errMsg:	停止向非腾讯云 CDN 上发布音视频流的事件回调
onSetMixTranscodingConfig:errMsg:	设置云端混流的排版布局和转码参数的事件回调
onStartPublishMediaStream:code:message:extraInfo:	开始发布媒体流的事件回调
onUpdatePublishMediaStream:code:message:extraInfo:	更新媒体流的事件回调
onStopPublishMediaStream:code:message:extraInfo:	停止媒体流的事件回调
onCdnStreamStateChanged:status:code:msg:extraInfo:	RTMP/RTMPS 推流状态发生改变回调

屏幕分享相关事件回调

函数列表	描述
onScreenCaptureStarted	屏幕分享开启的事件回调
onScreenCapturePaused:	屏幕分享暂停的事件回调
onScreenCaptureResumed:	屏幕分享恢复的事件回调
onScreenCaptureStoped:	屏幕分享停止的事件回调

本地录制和本地截图的事件回调

函数列表	描述
onLocalRecordBegin:storagePath:	本地录制任务已经开始的事件回调
onLocalRecording:storagePath:	本地录制任务正在进行中的进展事件回调
onLocalRecordFragment:	本地录制分片的事件回调
onLocalRecordComplete:storagePath:	本地录制任务已经结束的事件回调

废弃的事件回调

函数列表	描述
onUserEnter:	有主播加入当前房间（已废弃）
onUserExit:reason:	有主播离开当前房间（已废弃）
onAudioEffectFinished:code:	音效播放已结束（已废弃）

视频数据自定义回调

函数列表	描述
onRenderVideoFrame:userId:streamType:	自定义视频渲染回调
onProcessVideoFrame:dstFrame:	用于对接第三方美颜组件的视频处理回调
onGLContextDestory	SDK 内部 OpenGL 环境被销毁的通知

音频数据自定义回调

函数列表	描述
onCapturedAudioFrame:	本地采集并经过音频模块前处理后的音频数据回调
onLocalProcessedAudioFrame:	本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调

版权所有：腾讯云计算（北京）有限责任公司

onRemoteUserAudioFrame:userId:	混音前的每一路远程用户的音频数据
onMixedPlayAudioFrame:	将各路待播放音频混合之后并在最终提交系统播放之前的数据回调
onMixedAllAudioFrame:	SDK 所有音频混合后的音频数据（包括采集到的和待播放的）
onVoiceEarMonitorAudioFrame:	耳返的音频数据

更多事件回调接口

函数列表	描述
onLog:LogLevel:WhichModule:	本地 LOG 的打印回调

人声相关的特效接口

函数列表	描述
enableVoiceEarMonitor:	开启耳返
setVoiceEarMonitorVolume:	设置耳返音量
setVoiceReverbType:	设置人声的混响效果
setVoiceChangerType:	设置人声的变声特效
setVoiceVolume:	设置语音音量
setVoicePitch:	设置语音音调

背景音乐的相关接口

函数列表	描述
startPlayMusic:onStart:onProgress:onComplete:	开始播放背景音乐
stopPlayMusic:	停止播放背景音乐

pausePlayMusic:	暂停播放背景音乐
resumePlayMusic:	恢复播放背景音乐
setAllMusicVolume:	设置所有背景音乐的本地音量和远端音量的大小
setMusicPublishVolume:volume:	设置某一首背景音乐的远端音量的大小
setMusicPlayoutVolume:volume:	设置某一首背景音乐的本地音量的大小
setMusicPitch:pitch:	调整背景音乐的音调高低
setMusicSpeedRate:speedRate:	调整背景音乐的变速效果
getMusicCurrentPosInMS:	获取背景音乐的播放进度（单位：毫秒）
getMusicDurationInMS:	获取背景音乐的总时长（单位：毫秒）
seekMusicToPosInMS:pts:	设置背景音乐的播放进度（单位：毫秒）
setMusicScratchSpeedRate:speedRate:	调整搓碟的变速效果
preloadMusic:onProgress:onError:	预加载背景音乐
getMusicTrackCount:	获取背景音乐的音轨数量
setMusicTrack:track:	指定背景音乐的播放音轨

美颜相关接口

函数列表	描述
setBeautyStyle:	设置美颜（磨皮）算法
setBeautyLevel:	设置美颜级别
setWhitenessLevel:	设置美白级别
enableSharpnessEnhancement:	开启清晰度增强
setRuddyLevel:	设置红润级别
setFilter:	设置色彩滤镜效果
setFilterStrength:	设置色彩滤镜的强度
setGreenScreenFile:	设置绿幕背景视频

setEyeScaleLevel:	设置大眼级别
setFaceSlimLevel:	设置瘦脸级别
setFaceVLevel:	设置 V 脸级别
setChinLevel:	设置下巴拉伸或收缩
setFaceShortLevel:	设置短脸级别
setFaceNarrowLevel:	设置窄脸级别
setNoseSlimLevel:	设置瘦鼻级别
setEyeLightenLevel:	设置亮眼级别
setToothWhitenLevel:	设置牙齿美白级别
setWrinkleRemoveLevel:	设置祛皱级别
setPounchRemoveLevel:	设置祛眼袋级别
setSmileLinesRemoveLevel:	设置法令纹去除级别
setForeheadLevel:	设置发际线调整级别
setEyeDistanceLevel:	设置眼距
setEyeAngleLevel:	设置眼角调整级别
setMouthShapeLevel:	设置嘴型调整级别
setNoseWingLevel:	设置鼻翼调整级别
setNosePositionLevel:	设置鼻子位置
setLipsThicknessLevel:	设置嘴唇厚度
setFaceBeautyLevel:	设置脸型
setMotionTmpl:inDir:	选择 AI 动效挂件
setMotionMute:	是否在动效素材播放时静音

音视频设备相关的类型定义

函数列表	描述
onDeviceChanged:type:state:	本地设备的通断状态发生变化（仅适用于桌面系统）

设备操作接口

函数列表	描述
isFrontCamera	判断当前是否为前置摄像头（仅适用于移动端）
switchCamera:	切换前置或后置摄像头（仅适用于移动端）
isCameraZoomSupported	查询当前摄像头是否支持缩放（仅适用于移动端）
getCameraZoomMaxRatio	获取摄像头的最大缩放倍数（仅适用于移动端）
setCameraZoomRatio:	设置摄像头的缩放倍数（仅适用于移动端）
isAutoFocusEnabled	查询是否支持自动识别人脸位置（仅适用于移动端）
enableCameraAutoFocus:	开启自动对焦功能（仅适用于移动端）
setCameraFocusPosition:	设置摄像头的对焦位置（仅适用于移动端）
isCameraTorchSupported	查询是否支持开启闪光灯（仅适用于移动端）
enableCameraTorch:	开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）
setAudioRoute:	设置音频路由（仅适用于移动端）
setExposureCompensation:	设置摄像头的曝光参数，取值范围从-1到1
getDevicesList:	获取设备列表（仅适用于桌面端）
setCurrentDevice:deviceId:	设置当前要使用的设备（仅适用于桌面端）
getCurrentDevice:	获取当前正在使用的设备（仅适用于桌面端）
setCurrentDeviceVolume:deviceType:	设置当前设备的音量（仅适用于桌面端）
getCurrentDeviceVolume:	获取当前设备的音量（仅适用于桌面端）
setCurrentDeviceMute:deviceTyp	设置当前设备的静音状态（仅适用于桌面端）

e:	
getCurrentDeviceMute:	获取当前设备的静音状态（仅适用于桌面端）
enableFollowingDefaultAudioDevice:enable:	设置 SDK 使用的音频设备根据跟随系统默认设备（仅适用于桌面端）
startCameraDeviceTest:	开始摄像头测试（仅适用于桌面端）
stopCameraDeviceTest	结束摄像头测试（仅适用于桌面端）
startMicDeviceTest:	开始麦克风测试（仅适用于桌面端）
startMicDeviceTest:playback:	开始麦克风测试（仅适用于桌面端）
stopMicDeviceTest	结束麦克风测试（仅适用于桌面端）
startSpeakerDeviceTest:	开始扬声器测试（仅适用于桌面端）
stopSpeakerDeviceTest	结束扬声器测试（仅适用于桌面端）
setObserver:	设备热插拔回调（仅适用于 Mac 系统）

弃用接口

函数列表	描述
setSystemVolumeType:	设置系统音量类型（仅适用于移动端）

废弃接口

函数列表	描述
destroySharedIntance	销毁 TRTCCloud 实例（单例模式）
delegate	设置 TRTC 事件回调
setBeautyStyle:beautyLevel:whitenessLevel:ruddinessLevel:	设置美颜、美白以及红润效果级别
setEyeScaleLevel:	设置大眼级别
setFaceScaleLevel:	设置瘦脸级别
setFaceVLevel:	设置 V 脸级别
setChinLevel:	设置下巴拉伸或收缩幅度

setFaceShortLevel:	设置短脸级别
setNoseSlimLevel:	设置瘦鼻级别
selectMotionTmpl:	设置动效贴纸
setMotionMute:	设置动效静音
setFilter:	设置色彩滤镜效果
setFilterConcentration:	设置色彩滤镜浓度
setGreenScreenFile:	设置绿幕背景视频
setReverbType:	设置混响效果
setVoiceChangerType:	设置变声类型
enableAudioEarMonitoring:	开启（或关闭）耳返
enableAudioVolumeEvaluation:	启用音量大小提示
enableAudioVolumeEvaluation:enable_vad:	启用音量大小提示
switchCamera	切换摄像头
isCameraZoomSupported	查询当前摄像头是否支持缩放
setZoom:	设置摄像头缩放倍数（焦距）
isCameraTorchSupported	查询是否支持开关闪光灯
enbaleTorch:	开关/关闭闪光灯
isCameraFocusPositionInPreviewSupported	查询摄像头是否支持设置焦点
setFocusPosition:	设置摄像头焦点坐标位置
isCameraAutoFocusFaceModeSupported	查询是否支持自动识别人脸位置
enableAutoFaceFoucs:	开启/关闭人脸跟踪对焦
setSystemVolumeType:	设置系统音量类型
snapshotVideo:type:	视频截图

startScreenCaptureByReplaykit:appGroup:	开始全系统的屏幕分享（iOS）
startLocalAudio	设置音频质量
startRemoteView:view:	开始显示远端视频画面
stopRemoteView:	停止显示远端视频画面，同时不再拉取该远端用户的视频数据流
setLocalViewFillMode:	设置本地图像的渲染模式
setLocalViewRotation:	设置本地图像的顺时针旋转角度
setLocalViewMirror:	设置本地摄像头预览画面的镜像模式
setRemoteViewFillMode:mode:	设置远端图像的渲染模式
setRemoteViewRotation:rotation:	设置远端图像的顺时针旋转角度
startRemoteSubStreamView:view:	开始显示远端用户的辅路画面
stopRemoteSubStreamView:	停止显示远端用户的辅路画面
setRemoteSubStreamViewFillMode:mode:	设置辅路画面的填充模式
setRemoteSubStreamViewRotation:rotation:	设置辅路画面的顺时针旋转角度
setAudioQuality:	设置音频质量
setPriorRemoteVideoStreamType:	设定优先观看大画面还是小画面
setMicVolumeOnMixing:	设置麦克风音量大小
playBGM:	启动播放背景音乐
stopBGM	停止播放背景音乐
pauseBGM	停止播放背景音乐
resumeBGM	停止播放背景音乐
getBGMDuration:	获取背景音乐总时长（单位：毫秒）
setBGMPosition:	设置背景音乐的播放进度
setBGMVolume:	设置背景音乐的音量大小

setBGMPlayoutVolume:	设置背景音乐的本地播放音量
setBGMPublishVolume:	设置背景音乐的远端播放音量
playAudioEffect:	播放音效
setAudioEffectVolume:volume:	设置音效音量
stopAudioEffect:	停止播放音效
stopAllAudioEffects	停止所有音效
setAllAudioEffectsVolume:	设置所有音效音量
pauseAudioEffect:	暂停音效
resumeAudioEffect:	暂停音效
enableCustomVideoCapture:	启用视频自定义采集模式
sendCustomVideoData:	投送自己采集的视频数据
muteLocalVideo:	暂停/恢复发布本地的视频流
muteRemoteVideoStream:mute:	暂停 / 恢复订阅远端用户的视频流
startSpeedTest:userId:userSig:	开始进行网络测速（进入房间前使用）
startScreenCapture:	启动屏幕分享
getCameraDevicesList	获取摄像头设备列表
setCurrentCameraDevice:	选定当前要使用的摄像头
getCurrentCameraDevice	获取当前使用的摄像头
getMicDevicesList	获取麦克风设备列表
getCurrentMicDevice	获取当前的麦克风设备
setCurrentMicDevice:	选定当前使用的麦克风
getCurrentMicDeviceVolume	获取当前麦克风的设备音量
setCurrentMicDeviceVolume:	设置当前麦克风的设备音量
setCurrentMicDeviceMute:	设置系统当前麦克风设备的静音状态
getCurrentMicDeviceMute	获取系统当前麦克风设备是否静音

版权所有：腾讯云计算（北京）有限责任公司

getSpeakerDevicesList	获取扬声器设备列表
getCurrentSpeakerDevice	获取当前的扬声器设备
setCurrentSpeakerDevice:	设置要使用的扬声器
getCurrentSpeakerDeviceVolume	获取当前扬声器的设备音量
setCurrentSpeakerDeviceVolume:	设置当前扬声器的设备音量
getCurrentSpeakerDeviceMute	获取系统当前扬声器设备是否静音
setCurrentSpeakerDeviceMute:	设置系统当前扬声器设备的静音状态
startCameraDeviceTestInView:	开始进行摄像头测试
stopCameraDeviceTest	停止进行摄像头测试
startMicDeviceTest:	开始进行麦克风测试
stopMicDeviceTest	开始进行麦克风测试
startSpeakerDeviceTest:	开始进行扬声器测试
stopSpeakerDeviceTest	停止进行扬声器测试
startScreenCaptureInApp:	开始应用内的屏幕分享（iOS）
setVideoEncoderRotation:	设置视频编码器输出的画面方向
setVideoEncoderMirror:	设置编码器输出的画面镜像模式
setGSensorMode:	设置重力感应的适配模式

版权所有：腾讯云计算（北京）有限责任公司

TRTCCloud

最近更新时间：2024-03-06 14:51:31

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTCCloud @ TXLiteAVSDK
Function: 腾讯云 TRTC 主功能接口
Version: 11.7

TRTCCloud

TRTCCloud

函数列表	描述
sharedInstance	创建 TRTCCloud 实例（单例模式）
destroySharedInstance	销毁 TRTCCloud 实例（单例模式）
addDelegate:	添加 TRTC 事件回调
removeDelegate:	移除 TRTC 事件回调
delegateQueue	设置驱动 TRTCCloudDelegate 事件回调的队列
enterRoom:appScene:	进入房间
exitRoom	离开房间
switchRole:	切换角色
switchRole:privateMapKey:	切换角色（支持设置权限位）
switchRoom:	切换房间
connectOtherRoom:	请求跨房通话
disconnectOtherRoom	退出跨房通话
setDefaultStreamRecvMode:video:	设置订阅模式（需要在进入房前设置才能生效）
createSubCloud	创建子房间实例（用于多房间并发观看）

destroySubCloud:	销毁子房间实例
updateOtherRoomForwardMode:	更改跨房主播在本房间的上行能力
startPublishing:type:	开始向腾讯云直播 CDN 上发布音视频流
stopPublishing	停止向腾讯云直播 CDN 上发布音视频流
startPublishCDNStream:	开始向非腾讯云 CDN 上发布音视频流
stopPublishCDNStream	停止向非腾讯云 CDN 上发布音视频流
setMixTranscodingConfig:	设置云端混流的排版布局和转码参数
startPublishMediaStream:encoderParam:mixingConfig:	开始发布媒体流
updatePublishMediaStream:publishTarget:encoderParam:mixingConfig:	更新发布媒体流
stopPublishMediaStream:	停止发布媒体流
startLocalPreview:view:	开启本地摄像头的预览画面（移动端）
startLocalPreview:	开启本地摄像头的预览画面（桌面端）
updateLocalView:	更新本地摄像头的预览画面
stopLocalPreview	停止摄像头预览
muteLocalVideo:mute:	暂停/恢复发布本地的视频流
setVideoMuteImage:fps:	设置本地画面被暂停期间的替代图片
startRemoteView:streamType:view:	订阅远端用户的视频流，并绑定视频渲染控件
updateRemoteView:streamType:forUser:	更新远端用户的视频渲染控件
stopRemoteView:streamType:	停止订阅远端用户的视频流，并释放渲染控件
stopAllRemoteView	停止订阅所有远端用户的视频流，并释放全部渲染资源
muteRemoteVideoStream:streamType:mute:	暂停/恢复订阅远端用户的视频流
muteAllRemoteVideoStreams:	暂停/恢复订阅所有远端用户的视频流

setVideoEncoderParam:	设置视频编码器的编码参数
setNetworkQosParam:	设置网络质量控制的相关参数
setLocalRenderParams:	设置本地画面的渲染参数
setRemoteRenderParams:streamType:params:	设置远端画面的渲染模式
enableEncSmallVideoStream:withQuality:	开启大小画面双路编码模式
setRemoteVideoStreamType:type:	切换指定远端用户的大小画面
snapshotVideo:type:sourceType:	视频画面截图
setPerspectiveCorrectionWithUser:srcPoints:dstPoints:	视频画面透视校正坐标设置
setGravitySensorAdaptiveMode:	设置重力感应的适配模式（11.7 及以上版本）
startLocalAudio:	开启本地音频的采集和发布
stopLocalAudio	停止本地音频的采集和发布
muteLocalAudio:	暂停/恢复发布本地的音频流
muteRemoteAudio:mute:	暂停/恢复播放远端的音频流
muteAllRemoteAudio:	暂停/恢复播放所有远端用户的音频流
setAudioRoute:	设置音频路由
setRemoteAudioVolume:volume:	设定某一个远端用户的声音播放音量
setAudioCaptureVolume:	设定本地音频的采集音量
getAudioCaptureVolume	获取本地音频的采集音量
setAudioPlayoutVolume:	设定远端音频的播放音量
getAudioPlayoutVolume	获取远端音频的播放音量
enableAudioVolumeEvaluation:withParams:	启用音量大小提示
startAudioRecording:	开始录音
stopAudioRecording	停止录音

版权所有：腾讯云计算（北京）有限责任公司

startLocalRecording:	开启本地媒体录制
stopLocalRecording	停止本地媒体录制
setRemoteAudioParallelParams:	设置远端音频流智能并发播放策略
enable3DSpatialAudioEffect:	启用 3D 音效
updateSelf3DSpatialPosition	设置 3D 音效中自身坐标及朝向信息
updateRemote3DSpatialPosition:	设置 3D 音效中远端用户坐标信息
set3DSpatialReceivingRange:range:	设置指定用户所发出声音的可被接收范围
getDeviceManager	获取设备管理类（TXDeviceManager）
getBeautyManager	获取美颜管理类（TXBeautyManager）
setWatermark:streamType:rect:	添加水印
getAudioEffectManager	获取音效管理类（TXAudioEffectManager）
startSystemAudioLoopback	开启系统声音采集（iOS 端暂未支持）
stopSystemAudioLoopback	停止系统声音采集（iOS 端暂未支持）
setSystemAudioLoopbackVolume:	设置系统声音的采集音量
startScreenCaptureInApp:encParam:	开始应用内的屏幕分享（仅支持 iOS 13.0 及以上系统）
startScreenCaptureByReplaykit:encParam:appGroup:	开始全系统的屏幕分享（仅支持 iOS 11.0 及以上系统）
startScreenCapture:streamType:encParam:	启动屏幕分享
stopScreenCapture	停止屏幕分享
pauseScreenCapture	暂停屏幕分享
resumeScreenCapture	恢复屏幕分享
getScreenCaptureSourcesWithThumbnailSize:iconSize:	枚举可分享的屏幕和窗口（该接口仅支持 Mac OS 系统）
selectScreenCaptureTarget:rect:capturesCursor:highlight:	选取要分享的屏幕或窗口（该接口仅支持 Mac OS 系统）
setSubStreamEncoderParam:	设置屏幕分享（即辅路）的视频编码参数（桌面

	系统和移动系统均已支持）
setSubStreamMixVolume:	设置屏幕分享时的混音音量大小（该接口仅支持桌面系统）
addExcludedShareWindow:	将指定窗口加入屏幕分享的排除列表中（该接口仅支持桌面系统）
removeExcludedShareWindow:	将指定窗口从屏幕分享的排除列表中移除（该接口仅支持桌面系统）
removeAllExcludedShareWindows	将所有窗口从屏幕分享的排除列表中移除（该接口仅支持桌面系统）
addIncludedShareWindow:	将指定窗口加入屏幕分享的包含列表中（该接口仅支持桌面系统）
removeIncludedShareWindow:	将指定窗口从屏幕分享的包含列表中移除（该接口仅支持桌面系统）
removeAllIncludedShareWindows	将全部窗口从屏幕分享的包含列表中移除（该接口仅支持桌面系统）
enableCustomVideoCapture:enable:	启用/关闭视频自定义采集模式
sendCustomVideoData:frame:	向 SDK 投送自己采集的视频帧
enableCustomAudioCapture:	启用音频自定义采集模式
sendCustomAudioData:	向 SDK 投送自己采集的音频数据
enableMixExternalAudioFrame:playout:	启用/关闭自定义音轨
mixExternalAudioFrame:	向 SDK 混入自定义音轨
setMixExternalAudioVolume:playoutVolume:	设置推流时混入外部音频的推流音量和播放音量
generateCustomPTS	生成自定义采集时的时间戳
setLocalVideoProcessDelegete:pixelFormat:bufferType:	设置第三方美颜的视频数据回调
setLocalVideoRenderDelegate:pixelFormat:bufferType:	设置本地视频自定义渲染回调
setRemoteVideoRenderDelegate:delegate:pixelFormat:bufferType:	设置远端视频自定义渲染回调

setAudioFrameDelegate:	设置音频数据自定义回调
setCapturedAudioFrameDelegateFormat:	设置本地麦克风采集出的音频帧回调格式
setLocalProcessedAudioFrameDelegateFormat:	设置经过前处理后的本地音频帧回调格式
setMixedPlayAudioFrameDelegateFormat:	设置最终要由系统播放出的音频帧回调格式
enableCustomAudioRendering:	开启音频自定义播放
getCustomAudioRenderingFrame:	获取可播放的音频数据
sendCustomCmdMsg:data:reliable:ordered:	使用 UDP 通道发送自定义消息给房间内所有用户
sendSEIMsg:repeatCount:	使用 SEI 通道发送自定义消息给房间内所有用户
startSpeedTest:	开始进行网速测试（进入房间前使用）
stopSpeedTest	停止网络测速
getSDKVersion	获取 SDK 版本信息
setLogLevel:	设置 Log 输出级别
setConsoleEnabled:	启用/禁用控制台日志打印
setLogCompressEnabled:	启用/禁用日志的本地压缩
setLogDirPath:	设置本地日志的保存路径
setLogDelegate:	设置日志回调
showDebugView:	显示仪表盘
setDebugViewMargin:margin:	设置仪表盘的边距
callExperimentalAPI:	调用实验性接口
enablePayloadPrivateEncryption:params:	开启或关闭媒体流私有加密

sharedInstance

sharedInstance

创建 TRTCCloud 实例（单例模式）

参数	描述
context	仅适用于 Android 平台，SDK 内部会将其转化为 Android 平台的 ApplicationContext 用于调用 Android System API。 如果传入的 context 参数为空，SDK 内部会自动获取当前进程的 ApplicationContext。

注意

1. 如果您使用 delete ITRTCCloud* 会导致编译错误，请使用 destroyTRTCCloud 释放对象指针。
2. 在 Windows、Mac 和 iOS 平台上，请调用 getTRTCShareInstance() 接口。
3. 在 Android 平台上，请调用 getTRTCShareInstance(void *context) 接口。

destroySharedInstance

destroySharedInstance

销毁 TRTCCloud 实例（单例模式）

addDelegate:

addDelegate:

```
- (void)addDelegate:    (id<TRTCCloudDelegate>)delegate
```

添加 TRTC 事件回调

您可以通过 TRTCCloudDelegate 获得来自 SDK 的各类事件通知（比如：错误码，警告码，音视频状态参数等）。

removeDelegate:

removeDelegate:

```
- (void)removeDelegate:    (id<TRTCCloudDelegate>)delegate
```

移除 TRTC 事件回调

delegateQueue

delegateQueue

设置驱动 TRTCCloudDelegate 事件回调的队列

如果您不指定自己的 delegateQueue，SDK 默认会采用 MainQueue 作为驱动 TRTCCloudDelegate 事件回调的队列。
也就是当您不设置 delegateQueue 属性时，TRTCCloudDelegate 中的所有回调函数都是由 MainQueue 来驱动的。

> **注意**
> 如果您指定了自己的 delegateQueue，请不要在 TRTCCloudDelegate 回调函数中操作 UI，否则会引发线程安全问题。

enterRoom:appScene:

enterRoom:appScene:

```
- (void)enterRoom:    (TRTCParams *)param
         appScene:    (TRTCAppScene)scene
```

进入房间

TRTC 的所有用户都需要进入房间才能“发布”或“订阅”音视频流，“发布”是指将自己的音频和视频推送到云端，“订阅”是指从云端拉取房间里其他用户的音视频流。
调用该接口时，您需要指定您的应用场景 TRTCAppScene 以获取最佳的音视频传输体验，这些场景可以分成两大类：

实时通话：
包括 TRTCAppSceneVideoCall 和 TRTCAppSceneAudioCall 两个可选项，分别是视频通话和语音通话，该模式适合 1对1 的音视频通话，或者参会人数在 300 人以内的在线会议。

在线直播：
包括 TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom 两个可选项，分别是视频直播和语音直播，该模式适合十万人以内的在线直播场景，但需要您在接下来介绍的 TRTCParams 参数中指定 **角色(role)** 这个字段，也就是将房间中的用户区分为 **主播**（TRTCRoleAnchor）和 **观众**（TRTCRoleAudience）两种不同的角色。

调用该接口后，您会收到来自 TRTCCloudDelegate 中的 onEnterRoom(result) 回调：

- 如果进房成功，参数 result 会是一个正数（result > 0），表示从函数调用到进入房间所花费的时间，单位是毫秒（ms）。
- 如果进房失败，参数 result 会是一个负数（result < 0），表示进房失败的错误码。

参数	描述
param	进房参数，用于指定用户的身份、角色和安全票据等信息，详情请参见 TRTCParams 。
scene	应用场景，用于指定您的业务场景，同一个房间内的所有用户需要设定相同的 TRTCAppScene 。

注意
1. 同一个房间内的所有用户需要设定相同的 scene。不同的 scene 会导致偶现的异常问题。
2. 当您指定参数 scene 为 TRTCAppSceneLIVE 或 TRTCAppSceneVoiceChatRoom 时，您必须通过 TRTCParams 中的 “role” 字段为当前用户设定他/她在房间中的角色。
3. 请您尽量保证 enterRoom 与 exitRoom 前后配对使用，即保证”先退出前一个房间再进入下一个房间”，否则会导致很多异常问题。

exitRoom

exitRoom

离开房间

调用该接口会让用户离开自己所在的音视频房间，并释放摄像头、麦克风、扬声器等设备资源。
等资源释放完毕之后，SDK 会通过 TRTCCloudDelegate 中的 onExitRoom 回调向您通知。

如果您要再次调用 enterRoom 或者切换到其他的供应商的 SDK，建议等待 onExitRoom 回调到来之后再执行之后的操作，以避免摄像头或麦克风被占用的问题。

switchRole:

switchRole:

-(void)switchRole: (TRTCRoleType)role

切换角色

调用本接口可以实现用户在 主播 和 观众 两种角色之间来回切换。

由于视频直播和语音聊天室需要支持多达10万名观众同时观看，所以设定了 **只有主播才能发布自己的音视频** 的规则。因此，当有些观众希望发布自己的音视频流（以便能跟主播互动）时，就需要先把自己的角色切换成 **主播**。

您可以在进入房间时通过 TRTCParams 中的 role 字段事先确定用户的角色，也可以在进入房间后通过 switchRole 接口动态切换角色。

参数	描述
role	角色，默认为 **主播**。 • TRTCRoleAnchor ：主播，可以发布自己的音视频，同一个房间里最多支持50个主播同时发布音视频。 • TRTCRoleAudience ：观众，不能发布自己的音视频流，只能观看房间中其他主播的音视频。如果要发布自己的音视频，需要先通过 switchRole 切换成 **主播**，同一个房间内同时最多可以容纳 10 万名观众。

注意

1. 该接口仅适用于视频直播（TRTCAppSceneLIVE）和语音聊天室（TRTCAppSceneVoiceChatRoom）这两个场景。
2. 如果您在 enterRoom 时指定的 scene 为 TRTCAppSceneVideoCall 或 TRTCAppSceneAudioCall，请不要调用这个接口。

switchRole:privateMapKey:

switchRole:privateMapKey:

```
-(void)switchRole:      (TRTCRoleType)role
privateMapKey:          (NSString*)privateMapKey
```

切换角色（支持设置权限位）

调用本接口可以实现用户在 `主播` 和 `观众` 两种角色之间来回切换。

由于视频直播和语音聊天室需要支持多达10万名观众同时观看，所以设定了“只有主播才能发布自己的音视频”的规则。
因此，当有些观众希望发布自己的音视频流（以便能跟主播互动）时，就需要先把自己的角色切换成“主播”。
您可以在进入房间时通过 TRTCParams 中的 role 字段事先确定用户的角色，也可以在进入房间后通过 switchRole 接口动态切换角色。

参数	描述
privateMapKey	用于权限控制的权限票据，当您希望某个房间只能让特定的 userId 进入或者上行视频时，需要使用 privateMapKey 进行权限保护。 • 仅建议有高级别安全需求的客户使用，更多详情请参见 开启高级权限控制 。
role	角色，默认为“主播”： • TRTCRoleAnchor ：主播，可以发布自己的音视频，同一个房间里最多支持50个主播同时发布音视频。 • TRTCRoleAudience ：观众，不能发布自己的音视频流，只能观看房间中其他主播的音视频。如果要发布自己的音视频，需要先通过 switchRole 切换成“主播”，同一个房间内同时最多可以容纳 10 万名观众。

注意
1. 该接口仅适用于视频直播（TRTCAppSceneLIVE）和语音聊天室（TRTCAppSceneVoiceChatRoom）这两个场景。
2. 如果您在 enterRoom 时指定的 scene 为 TRTCAppSceneVideoCall 或 TRTCAppSceneAudioCall，请不要调用这个接口。

switchRoom:

switchRoom:

```
- (void)switchRoom:     (TRTCSwitchRoomConfig *)config
```

切换房间

使用该接口可以让用户可以快速从一个房间切换到另一个房间。

- 如果用户的身份是“观众”，该接口的调用效果等同于 exitRoom(当前房间) + enterRoom（新的房间）。
- 如果用户的身份是“主播”，该接口在切换房间的同时还会保持自己的音视频发布状态，因此在房间切换过程中，摄像头的预览和声音的采集都不会中断。

该接口适用于在线教育场景中，监课老师在多个房间中进行快速切换的场景。在该场景下使用 switchRoom 可以获得比 exitRoom+enterRoom 更好的流畅性和更少的代码量。
接口调用结果会通过 TRTCCloudDelegate 中的 onSwitchRoom(errCode, errMsg) 回调。

参数	描述
config	房间参数，详情请参见 TRTCSwitchRoomConfig 。

注意
由于对老版本 SDK 兼容的需求，参数 config 中同时包含 roomId 与 strRoomId 两个参数，这两个参数的填写格外讲究，请注意如下事项：
1. 若您选用 strRoomId，则 roomId 需要填写为0。若两者都填，将优先选用 roomId。
2. 所有房间需要同时使用 strRoomId 或同时使用 roomId，不可混用，否则将会出现很多预期之外的 bug。

connectOtherRoom:

connectOtherRoom:

```
- (void)connectOtherRoom:      (NSString *)param
```

请求跨房通话

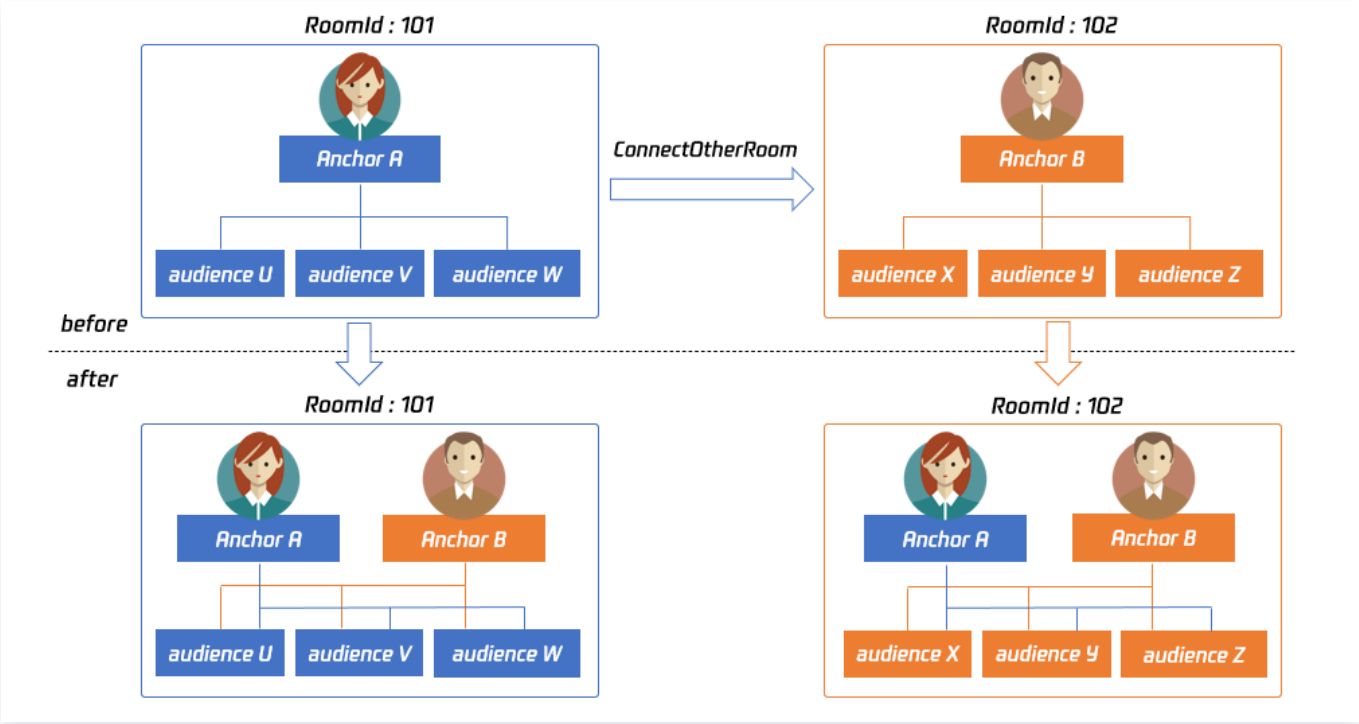
默认情况下，只有同一个房间中的用户之间可以进行音视频通话，不同的房间之间的音视频流是相互隔离的。但您可以通过调用该接口，将另一个房间中某个主播音视频流发布到自己所在的房间中，与此同时，该接口也会将自己的音视频流发布到目标主播的房间中。

也就是说，您可以使用该接口让身处两个不同房间中的主播进行跨房间的音视频流分享，从而让每个房间中的观众都能观看到这两个主播的音视频。该功能可以用来实现主播之间的 PK 功能。

跨房通话的请求结果会通过 TRTCCloudDelegate 中的 onConnectOtherRoom 回调通知给您。

例如：当房间“101”中的主播 A 通过 connectOtherRoom() 跟房间“102”中的主播 B 建立跨房通话后，

- 房间“101”中的用户都会收到主播 B 的 onRemoteUserEnterRoom(B) 和 onUserVideoAvailable(B,YES) 这两个事件回调，即房间“101”中的用户都可以订阅主播 B 的音视频。
- 房间“102”中的用户都会收到主播 A 的 onRemoteUserEnterRoom(A) 和 onUserVideoAvailable(A,YES) 这两个事件回调，即房间“102”中的用户都可以订阅主播 A 的音视频。



跨房通话的参数考虑到后续扩展字段的兼容性问题，暂时采用了 JSON 格式的参数：

情况一：数字房间号

如果房间“101”中的主播 A 要跟房间“102”中的主播 B 连麦，那么主播 A 调用该接口时需要传入：{"roomId": 102, "userId": "userB"}

示例代码如下：

```
NSMutableDictionaryjsonDict = [[NSMutableDictionary alloc] init];
[jsonDict setObject:@(102) forKey:@"roomId"];
[jsonDict setObject:@"userB" forKey:@"userId"];
NSData* jsonData = [NSJSONSerialization dataWithJSONObject:jsonDict
options:NSJSONWritingPrettyPrinted error:nil];
```

```
  NSString* jsonString = [[NSString alloc] initWithData:jsonData
encoding:NSUTF8StringEncoding];
  [trtc connectOtherRoom:jsonString];
```

情况二：字符串房间号

如果您使用的是字符串房间号，务必请将 json 中的 “roomId” 替换成 “strRoomId”：{"strRoomId": "102", "userId": "userB"}

示例代码如下：

```
  NSMutableDictionaryjsonDict = [[NSMutableDictionary alloc] init];
  [jsonDict setObject:@"102" forKey:@"strRoomId"];
  [jsonDict setObject:@"userB" forKey:@"userId"];
  NSData* jsonData = [NSJSONSerialization dataWithJSONObject:jsonDict
options:NSJSONWritingPrettyPrinted error:nil];
  NSString* jsonString = [[NSString alloc] initWithData:jsonData
encoding:NSUTF8StringEncoding];
  [trtc connectOtherRoom:jsonString];
```

参数	描述
param	需要你传入 JSON 格式的字符串参数，roomId 代表数字格式的房间号，strRoomId 代表字符串格式的房间号，userId 代表目标主播的用户 ID。

disconnectOtherRoom

disconnectOtherRoom

退出跨房通话

退出结果会通过 TRTCCloudDelegate 中的 onDisconnectOtherRoom 回调通知给您。

setDefaultStreamRecvMode:video:

setDefaultStreamRecvMode:video:

```
- (void)setDefaultStreamRecvMode:    (BOOL)autoRecvAudio
                           video:    (BOOL)autoRecvVideo
```

设置订阅模式（需要在进入房前设置才能生效）

您可以通过该接口在“自动订阅”和“手动订阅”两种模式下进行切换：

- 自动订阅：默认模式，用户在进入房间后会立刻接收到该房间中的音视频流，音频会自动播放，视频会自动开始解码（依然需要您通过 startRemoteView 接口绑定渲染控件）。
- 手动订阅：在用户进入房间后，需要手动调用 startRemoteView 接口才能启动视频流的订阅和解码，需要手动调用 muteRemoteAudio (NO) 接口才能启动声音的播放。

在绝大多数场景下，用户进入房间后都会订阅房间中所有主播的音视频流，因此 TRTC 默认采用了自动订阅模式，以求得最佳的“秒开体验”。
如果您的应用场景中每个房间同时会有很多路音视频流在发布，而每个用户只想选择性地订阅其中的 1–2 路，则推荐使用“手动订阅”模式以节省流量费用。

参数	描述
autoRecvAudio	YES：自动订阅音频；NO：需手动调用 muteRemoteAudio(NO) 订阅音频。默认值：YES。
autoRecvVideo	YES：自动订阅视频；NO：需手动调用 startRemoteView 订阅视频。默认值：YES。

注意

1. 需要在进入房间前调用该接口进行设置才能生效。
2. 在自动订阅模式下，如果用户在进入房间后没有调用 startRemoteView 订阅视频流，SDK 会自动停止订阅视频流，以便达到节省流量的目的。

createSubCloud

createSubCloud

创建子房间实例（用于多房间并发观看）

TRTCCloud 一开始被设计成单例模式，限制了多房间并发观看的能力。
通过调用该接口，您可以创建出多个 TRTCCloud 实例，以便同时进入多个不同的房间观看音视频流。

但需要注意的是，您在多个 TRTCCloud 实例中发布音视频流的能力会受到一定限制。

该功能主要用于在线教育场景中一种被称为“超级小班课”的业务场景中，用于解决“每个 TRTC 的房间中最多只能有 50 人同时发布自己音视频流”的限制。

示例代码如下：

```
    //In the small room that needs interaction, enter the room as an anchor and
push audio and video streams
    TRTCCloud *mainCloud = [TRTCCloud sharedInstance];
    TRTCParams *mainParams = [[TRTCParams alloc] init];
    //Fill your params
    mainParams.role = TRTCRoleAnchor;
    [mainCloud enterRoom:mainParams appScene:TRTCAppSceneLIVE)];
    //...
    [mainCloud startLocalPreview:YES view:videoView];
    [mainCloud startLocalAudio:TRTCAudioQualityDefault];

    //In the large room that only needs to watch, enter the room as an audience
and pull audio and video streams
    TRTCCloud *subCloud = [mainCloud createSubCloud];
    TRTCParams *subParams = [[TRTCParams alloc] init];
    //Fill your params
    subParams.role = TRTCRoleAudience;
    [subCloud enterRoom:subParams appScene:TRTCAppSceneLIVE)];
    //...
    [subCloud startRemoteView:userId streamType:TRTCVideoStreamTypeBig
view:videoView];
    //...
    //Exit from new room and release it.
    [subCloud exitRoom];
    [mainCloud destroySubCloud:subCloud];
```

注意

- 同一个用户，可以使用同一个 userId 进入多个不同 roomId 的房间。
- 两台不同的终端设备不可以同时使用同一个 userId 进入同一个 roomId 的房间。
- 您可以分别为不同实例分别设置 TRTCCloudDelegate 获取各自的事件通知。
- 同一个用户可以在多个 TRTCCloud 实例中推流，也可以调用子实例中与本地音视频相关的接口。但需要注意：
- 多实例的音频需要同时为麦克风采集或自定义采集，而且与音频设备相关的接口调用会以最后一次为准；
- 与摄像头相关的调用会以最后一次为准：startLocalPreview 。

返回值说明：

子 TRTCCloud 实例

destroySubCloud:

destroySubCloud:

```
- (void)destroySubCloud:(TRTCCloud *)subCloud
```

销毁子房间实例

参数	描述
subCloud	子房间实例

updateOtherRoomForwardMode:

updateOtherRoomForwardMode:

```
- (void)updateOtherRoomForwardMode:(NSString *)param
```

更改跨房主播在本房间的上行能力

通常情况下，在调用 connectOtherRoom 接口与另一个房间的主播进行跨房通话后，本房间内的所有观众都将收到该主播发布的音视频流。
您可以通过调用该接口，限制跨房主播在本房间内的上行能力，禁止或允许跨房主播发布音频/主路视频/辅路视频，该行为会影响房间内的所有用户。
在禁用跨房主播某种上行能力后，本房间内所有用户将无法收到对应音视频流，且无法再订阅对应的音视频。

需要注意的是，该接口只能由进行跨房通话的主播调用，且通过该接口设置的限制会因为跨房通话的中断或对应主播退房重置。

该接口的调用结果会通过 TRTCCloudDelegate 中的 onUpdateOtherRoomForwardMode 回调通知给您。

例如：

房间“101”中有主播 A 与观众 B，房间“102”中有主播 C，主播 C 正常发布音视频流。主播 A 通过 connectOtherRoom() 跟主播 C 建立跨房通话。

- 此时，主播 A 与观众 B 都会收到主播 C 的 onRemoteUserEnterRoom(C) 和 onUserVideoAvailable(C,YES)、onUserAudioAvailable(C,YES) 这三个事件回调，可以订阅主播 C 的音视频。

之后，主播 A 调用该接口禁用主播 C 在本房间发布音频的能力。

- 此后，主播 C 的音频流将无法发布至房间“101”，主播 A 与观众 B 都将收到 onUserAudioAvailable(C,NO) 事件回调，且无法再通过 muteRemoteAudio(C,NO) 订阅主播 C 的音频。
- 主播 C 的视频流将不受影响。房间“102”中的其他观众也不受影响，可以正常订阅主播 C 的音频。

该接口的参数考虑到后续扩展字段的兼容性问题，暂时采用了 JSON 格式的参数，示例如下：

情况一：数字房间号

```
{
 "roomId":102,
 "userId":"userC",
 "muteAudio":false,
 "muteVideo":true,
 "muteSubStream":false
}
```

情况二：字符串房间号

```
{
 "strRoomId":"102",
 "userId":"userC",
 "muteAudio":false,
 "muteVideo":true,
 "muteSubStream":false
}
```

参数	描述
param	需要你传入 JSON 格式的字符串参数，roomId 代表数字格式的房间号，strRoomId 代表字符串格式的房间号，userId 代表目标主播的用户 ID，muteAudio、muteVideo、muteSubStream均为可选项，分表代表禁止或允许跨房主播发布音频/主路视频/辅路视频的能力。

startPublishing:type:

startPublishing:type:

```
- (void)startPublishing:    (NSString *)streamId
                  type:     (TRTCVideoStreamType)streamType
```

开始向腾讯云直播 CDN 上发布音视频流

该接口会向 TRTC 服务器发送指令，要求其将当前用户的音视频流旁路到直播 CDN 上。您可以通过参数 streamId 设定直播流的 StreamId，从而可以指定该用户的音视频流对应在直播 CDN 上的播放地址。

例如：您可以通过该接口将当前用户的直播流 ID 指定为 user_stream_001，那么该用户音视频流对应的 CDN 播放地址为：
“http://yourdomain/live/user_stream_001.flv”，其中 yourdomain 为您自己备案的播放域名，

您可以在直播控制台 配置您的播放域名，腾讯云不提供默认的播放域名。

```
TRTCCloud *trtcCloud = [TRTCCloud sharedInstance];
[trtcCloud enterRoom:params appScene:TRTCAppSceneLIVE];
[trtcCloud startLocalPreview:frontCamera view:localView];
[trtcCloud startLocalAudio];
[trtcCloud startPublishing: @"user_stream_001" type:TRTCVideoStreamTypeBig];
```

您也可以在设置 enterRoom 的参数 TRTCParams 时指定 streamId, 而且我们更推荐您采用这种方案。

参数	描述
streamId	自定义流 ID。
streamType	仅支持 TRTCVideoStreamTypeBig 和 TRTCVideoStreamTypeSub 。

注意
您需要提前在 实时音视频控制台 中的功能配置页面上开启“启用旁路推流”才能生效。
- 若您选择“指定流旁路”，则您可以通过该接口将对应音视频流推送到腾讯云 CDN 且指定为填写的流 ID。

- 若您选择“全局自动旁路”，则您可以通过该接口调整默认的流 ID。

stopPublishing

stopPublishing

停止向腾讯云直播 CDN 上发布音视频流

startPublishCDNStream:

startPublishCDNStream:

```
- (void)startPublishCDNStream:(TRTCPublishCDNParam *)param
```

开始向非腾讯云 CDN 上发布音视频流

该接口跟 startPublishing 功能类似，不同之处在于，startPublishing 仅支持向腾讯云的 CDN 发布，而本接口支持向非腾讯云的直播 CDN 上转推音视频流。

参数	描述
param	CDN 转推参数，详情请参见 TRTCPublishCDNParam

注意

- 使用 startPublishing 接口向腾讯云的直播 CDN 上发布音视频流不会收取额外费用。
- 使用 startPublishCDNStream 接口向非腾讯云的直播 CDN 上发布音视频流，需要收取额外的转推带宽费用。

stopPublishCDNStream

stopPublishCDNStream

停止向非腾讯云 CDN 上发布音视频流

setMixTranscodingConfig:

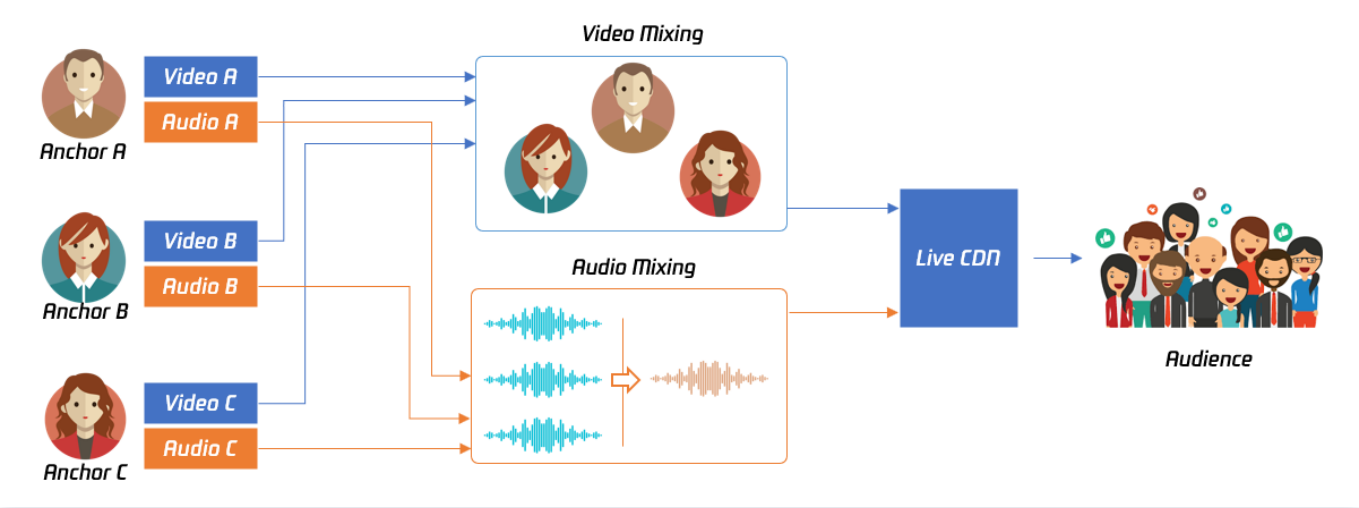
setMixTranscodingConfig:

- (void)setMixTranscodingConfig: (nullable TRTCTranscodingConfig *)config

设置云端混流的排版布局和转码参数

在一个直播间中可能同时会有多个主播发布自己的音视频流，但对于直播 CDN 上的观众而言，只需要观看一条 HTTP-FLV 或 HLS 格式的视频流即可。
当您调用本接口函数时，SDK 会向腾讯云的 TRTC 混流服务器发送一条指令，混流服务器会将房间里的多路音视频流混合成一路。
您可以通过 TRTCTranscodingConfig 参数来调整每一路画面的排版布局，也可以设置混合后的音视频流的各项编码参数。

参考文档：云端混流转码 。



参数	描述
config	如果 config 不为空，则开启云端混流，如果 config 为空则停止云端混流。详情请参见 TRTCTranscodingConfig 。

注意

关于云端混流的注意事项：

- 混流转码为收费功能，调用接口将产生云端混流转码费用，详见 云端混流转码计费说明 。
- 调用该接口的用户，如果没设定 config 参数中的 streamId 字段，TRTC 会将房间中的多路画面混合到当前用户所对应的音视频流上，即 A + B => A。

- 调用该接口的用户，如果设定了 config 参数中的 streamId 字段，TRTC 会将房间中的多路画面混合到您指定的 streamId 上，即 A + B => streamId。
- 请注意，若您还在房间中且不再需要混流，请务必再次调用本接口并将 config 设置为空以进行取消，不及时取消混流可能会引起不必要的计费损失。
- 请放心，当您退房时 TRTC 会自动取消混流状态。

startPublishMediaStream:encoderParam:mixingConfig:

startPublishMediaStream:encoderParam:mixingConfig:

```
- (void)startPublishMediaStream:    (TRTCPublishTarget*)target
                   encoderParam:    (nullable TRTCStreamEncoderParam*)pa
                   mixingConfig:    (nullable TRTCStreamMixingConfig*)conf
```

开始发布媒体流

该接口会向 TRTC 服务器发送指令，要求其将当前用户的音视频流转推/转码到直播 CDN 或者回推到 TRTC 房间中您可以通过 TRTCPublishTarget 配置中的 TRTCPublishMode 指定具体的发布模式

参数	描述
config	媒体流转码配置参数。具体配置参考 TRTCStreamMixingConfig 。转码和回推到 TRTC 房间中时为必填项，您需要指定您预期的转码配置参数。转推模式下则无效。
params	媒体流编码输出参数，具体配置参考 TRTCStreamEncoderParam 。转码和回推到 TRTC 房间中时为必填项，您需要指定您预期的转码输出参数。在转推时，为了更好的转推稳定性和 CDN 兼容性，也建议您进行配置。
target	媒体流发布的目标地址，具体配置参考 TRTCPublishTarget 。支持转推/转码到腾讯或者第三方 CDN，也支持转码回推到 TRTC 房间中。

注意

1. SDK 会通过回调 onStartPublishMediaStream 带给您后台启动的任务标识（即 taskId）。
2. 同一个任务（TRTCPublishMode 与 TRTCPublishCdnUrl 均相同）仅支持启动一次。若您后续需要更新或者停止该项任务，需要记录并使用返回的 taskId，通过 updatePublishMediaStream 或者 stopPublishMediaStream 来操作。
3. target 支持同时配置多个 CDN URL（最多同时 10 个）。若您的同一个转推/转码任务需要发布至多路 CDN，则仅需要在 target 中配置多个 CDN URL 即可。同一个转码任务即使有多个转推地

址，对应的转码计费仍只收取一份。
4. 使用时需要注意不要多个任务同时往相同的 URL 地址推送，以免引起异常推流状态。一种推荐的方案是 URL 中使用 “sdkappid_roomid_userid_main” 作为区分标识，这种命名方式容易辨认且不会在您的多个应用中发生冲突。

updatePublishMediaStream:publishTarget:encoderParam:mixingConfig:

updatePublishMediaStream:publishTarget:encoderParam:mixingConfig:

```
- (void)updatePublishMediaStream:    (NSString *)taskId
                   publishTarget:    (TRTCPublishTarget *)target
                   encoderParam:     (nullable TRTCStreamEncoderParam *)pa
                   mixingConfig:     (nullable TRTCStreamMixingConfig *)conf
```

更新发布媒体流

该接口会向 TRTC 服务器发送指令，更新通过 startPublishMediaStream 启动的媒体流

参数	描述
config	媒体流转码配置参数。具体配置参考 TRTCStreamMixingConfig 。转码和回推到 TRTC 房间中时为必填项，您需要指定您预期的转码配置参数。转推模式下则无效。
params	媒体流编码输出参数，具体配置参考 TRTCStreamEncoderParam 。转码和回推到 TRTC 房间中时为必填项，您需要指定您预期的转码输出参数。在转推时，为了更好的转推稳定性和 CDN 兼容性，也建议您进行配置。
target	媒体流发布的目标地址，具体配置参考 TRTCPublishTarget 。支持转推/转码到腾讯或者第三方 CDN，也支持转码回推到 TRTC 房间中。
taskId	通过回调 onStartPublishMediaStream 带给您后台启动的任务标识（即 taskId）

注意
1. 您可以通过本接口来更新发布的 CDN URL（支持增删，最多同时 10 个），但您使用时需要注意不要多个任务同时往相同的 URL 地址推送，以免引起异常推流状态。

2. 您可以通过 taskId 来更新调整转推/转码任务。例如在 pk 业务中，您可以先通过 startPublishMediaStream 发起转推，接着在主播发起 pk 时，通过 taskId 和本接口将转推更新为转码任务。此时，CDN 播放将连续并且不会发生断流（您需要保持媒体流编码输出参数 param 一致）。
3. 同一个任务不支持纯音频、音视频、纯视频之间的切换。

stopPublishMediaStream:

stopPublishMediaStream:

```
- (void)stopPublishMediaStream:    (NSString *)taskId
```

停止发布媒体流

该接口会向 TRTC 服务器发送指令，停止通过 startPublishMediaStream 启动的媒体流

参数	描述
taskId	通过回调 onStartPublishMediaStream 带给您后台启动的任务标识（即 taskId）

注意
1. 若您的业务后台并没有保存该 taskId，在您的主播异常退房重进后，如果您需要重新获取 taskId，您可以再次调用 startPublishMediaStream 启动任务。此时 TRTC 后台会返回任务启动失败，同时带给您上一次启动的 taskId
2. 若 taskId 填空字符串，将会停止该用户所有通过 startPublishMediaStream 启动的媒体流，如果您只启动了一个媒体流或者想停止所有通过您启动的媒体流，推荐使用这种方式。

startLocalPreview:view:

startLocalPreview:view:

```
- (void)startLocalPreview:    (BOOL)frontCamera
                     view:    (nullable TXView *)view
```

开启本地摄像头的预览画面（移动端）

在 enterRoom 之前调用此函数，SDK 只会开启摄像头，并一直等到您调用 enterRoom 之后才开始推流。
在 enterRoom 之后调用此函数，SDK 会开启摄像头并自动开始视频推流。当开始渲染首帧摄像头画面时，您会收到 TRTCCloudDelegate 中的 onCameraDidReady 回调通知。

参数	描述
frontCamera	YES：前置摄像头；NO：后置摄像头。
view	承载视频画面的控件。

注意
如果希望开播前预览摄像头画面并通过 BeautyManager 调节美颜参数，您可以：
- 方案一：在调用 enterRoom 之前调用 startLocalPreview。
- 方案二：在调用 enterRoom 之后调用 startLocalPreview + muteLocalVideo(YES)。

startLocalPreview:

startLocalPreview:

```
- (void)startLocalPreview:      (nullable TXView *)view
```

开启本地摄像头的预览画面（桌面端）

在调用该接口之前，您可以先调用 setCurrentCameraDevice 选择使用 Mac 自带摄像头或外接摄像头。
在 enterRoom 之前调用此函数，SDK 只会开启摄像头，并一直等到您调用 enterRoom 之后才开始推流。
在 enterRoom 之后调用此函数，SDK 会开启摄像头并自动开始视频推流。
当开始渲染首帧摄像头画面时，您会收到 TRTCCloudDelegate 中的 onCameraDidReady 回调通知。

参数	描述
view	承载视频画面的控件。

注意
如果希望开播前预览摄像头画面并通过 BeautyManager 调节美颜参数，您可以：

- 方案一：在调用 enterRoom 之前调用 startLocalPreview。
- 方案二：在调用 enterRoom 之后调用 startLocalPreview + muteLocalVideo(YES)。

updateLocalView:

updateLocalView:

```
- (void)updateLocalView:    (nullable TXView *)view
```

更新本地摄像头的预览画面

stopLocalPreview

stopLocalPreview

停止摄像头预览

muteLocalVideo:mute:

muteLocalVideo:mute:

```
- (void)muteLocalVideo:    (TRTCVideoStreamType)streamType
                  mute:    (BOOL)mute
```

暂停/恢复发布本地的视频流

该接口可以暂停（或恢复）发布本地的视频画面，暂停之后，同一房间中的其他用户将无法继续看到自己画面。
该接口在指定 TRTCVideoStreamTypeBig 时等效于 start/stopLocalPreview 这两个接口，但具有更好的响应速度。
因为 start/stopLocalPreview 需要打开和关闭摄像头，而打开和关闭摄像头都是硬件设备相关的操作，非常耗时。
相比之下，muteLocalVideo 只需要在软件层面对数据流进行暂停或者放行即可，因此效率更高，也更适合需要频繁打开关闭的场景。

当暂停/恢复发布指定 TRTCVideoStreamTypeBig 后，同一房间中的其他用户将会收到 onUserVideoAvailable 回调通知。
当暂停/恢复发布指定 TRTCVideoStreamTypeSub 后，同一房间中的其他用户将会收到 onUserSubStreamAvailable 回调通知。

参数	描述
mute	YES：暂停；NO：恢复。
streamType	要暂停/恢复的视频流类型（仅支持 TRTCVideoStreamTypeBig 和 TRTCVideoStreamTypeSub ）。

setVideoMuteImage:fps:

setVideoMuteImage:fps:

```
- (void)setVideoMuteImage:    (nullable TXImage *)image
                      fps:    (NSInteger)fps
```

设置本地画面被暂停期间的替代图片

当您调用 muteLocalVideo(YES) 暂停本地画面时，您可以通过调用本接口设置一张替代图片，设置后，房间中的其他用户会看到这张替代图片，而不是黑屏画面。

参数	描述
fps	设置替代图片帧率，最小值为5，最大值为10，默认5。
image	设置替代图片，空值代表在 muteLocalVideo 之后不再发送视频流数据，默认值为空。

startRemoteView:streamType:view:

startRemoteView:streamType:view:

```
- (void)startRemoteView:    (NSString *)userId
            streamType:     (TRTCVideoStreamType)streamType
```

view: (nullable TXView *)view

订阅远端用户的视频流，并绑定视频渲染控件

调用该接口可以让 SDK 拉取指定 userid 的视频流，并渲染到参数 view 指定的渲染控件上。您可以通过 setRemoteRenderParams 设置画面的显示模式。

- 如果您已经知道房间中有视频流的用户的 userid，可以直接调用 startRemoteView 订阅该用户的画面。
- 如果您不知道房间中有哪些用户在发布视频，您可以在 enterRoom 之后等待来自 onUserVideoAvailable 的通知。

调用本接口只是启动视频流的拉取，此时画面还需要加载和缓冲，当缓冲完毕后您会收到来自 onFirstVideoFrame 的通知。

参数	描述
streamType	指定要观看 userId 的视频流类型。 • 高清大画面：TRTCVideoStreamTypeBig 。 • 低清小画面：TRTCVideoStreamTypeSmall（需要远端用户通过 enableEncSmallVideoStream 开启双路编码后才有效果）。 • 辅流画面（常用于屏幕分享）：TRTCVideoStreamTypeSub 。
userId	指定远端用户的 ID。
view	用于承载视频画面的渲染控件。

注意

注意几点规则需要您关注：

1. SDK 支持同时观看某 userid 的大画面和辅路画面，或者同时观看某 userid 的小画面和辅路画面，但不支持同时观看大画面和小画面。
2. 只有当指定的 userid 通过 enableEncSmallVideoStream 开启双路编码后，才能观看该用户的小画面。
3. 当指定的 userid 的小画面不存在时，SDK 默认切换到该用户的大画面。

updateRemoteView:streamType:forUser:

updateRemoteView:streamType:forUser:

- (void)updateRemoteView: (nullable TXView *)view

streamType:	(TRTCVideoStreamType)streamType
forUser:	(NSString *)userId

更新远端用户的视频渲染控件

该接口可用于更新远端视频画面的渲染控件，常被用于切换显示区域的交互场景中。

参数	描述
streamType	要设置预览窗口的流类型（仅支持 TRTCVideoStreamTypeBig 和 TRTCVideoStreamTypeSub）。
userId	指定远端用户的 ID。
view	承载视频画面的控件。

stopRemoteView:streamType:

stopRemoteView:streamType:

- (void)stopRemoteView:	(NSString *)userId
streamType:	(TRTCVideoStreamType)streamType

停止订阅远端用户的视频流，并释放渲染控件

调用此接口会让 SDK 停止接收该用户的视频流，并释放该路视频流的解码和渲染资源。

参数	描述
streamType	指定要观看 userId 的视频流类型。 • 高清大画面：TRTCVideoStreamTypeBig。 • 低清小画面：TRTCVideoStreamTypeSmall。 • 辅流画面（常用于屏幕分享）：TRTCVideoStreamTypeSub。
userId	指定远端用户的 ID。

stopAllRemoteView

stopAllRemoteView

停止订阅所有远端用户的视频流，并释放全部渲染资源

调用此接口会让 SDK 停止接收所有来自远端的视频流，并释放全部的解码和渲染资源。

> **注意**
> 如果当前有正在显示的辅路画面（屏幕分享）也会一并被停止。

muteRemoteVideoStream:streamType:mute:

muteRemoteVideoStream:streamType:mute:

```
- (void)muteRemoteVideoStream:    (NSString*)userId
                    streamType:   (TRTCVideoStreamType)streamType
                          mute:   (BOOL)mute
```

暂停/恢复订阅远端用户的视频流

该接口仅暂停/恢复接收指定用户的视频流，但并不释放显示资源，视频画面会被冻屏在接口调用时的最后一帧。

参数	描述
mute	是否暂停接收。
streamType	要暂停/恢复的视频流类型。 • 高清大画面：TRTCVideoStreamTypeBig 。 • 低清小画面：TRTCVideoStreamTypeSmall 。 • 辅流画面（常用于屏幕分享）：TRTCVideoStreamTypeSub 。
userId	指定远端用户的 ID。

> **注意**
> 该接口支持您在进入房间（enterRoom）前调用，暂停状态在退出房间（exitRoom）之后会被重置。

调用该接口暂停接收指定用户的视频流后，只调用 startRemoteView 接口无法播放指定用户的视频，需要调用 muteRemoteVideoStream(NO) 或 muteAllRemoteVideoStreams(NO) 才能恢复。

muteAllRemoteVideoStreams:

muteAllRemoteVideoStreams:

- (void)muteAllRemoteVideoStreams: (BOOL)mute

暂停/恢复订阅所有远端用户的视频流

该接口仅暂停/恢复接收所有用户的视频流，但并不释放显示资源，视频画面会被冻屏在接口调用时的最后一帧。

参数	描述
mute	是否暂停接收。

注意

该接口支持您在进入房间（enterRoom）前调用，暂停状态在退出房间（exitRoom）之后会被重置。

调用该接口暂停接收所有用户的视频流后，只调用 startRemoteView 接口无法播放指定用户的视频，需要调用 muteRemoteVideoStream(NO) 或 muteAllRemoteVideoStreams(NO) 才能恢复。

setVideoEncoderParam:

setVideoEncoderParam:

- (void)setVideoEncoderParam: (TRTCVideoEncParam *)param

设置视频编码器的编码参数

该设置能够决定远端用户看到的画面质量，同时也能决定云端录制出的视频文件的画面质量。

参数	描述
param	用于设置视频编码器的相关参数，详情请参见 TRTCVideoEncParam 。

注意
从v11.5版本开始，编码输出分辨率会按照宽8高2字节对齐，并且是向下调整，eg:输入分辨率540x960，实际编码输出分辨率536x960。

setNetworkQosParam:

setNetworkQosParam:

```
- (void)setNetworkQosParam:(TRTCNetworkQosParam *)param
```

设置网络质量控制的相关参数

该设置决定在差网络环境下的质量调控策略，如“画质优先”或“流畅优先”等策略。

参数	描述
param	用于设置网络质量控制的相关参数，详情请参见 TRTCNetworkQosParam 。

setLocalRenderParams:

setLocalRenderParams:

```
- (void)setLocalRenderParams:(TRTCRenderParams *)params
```

设置本地画面的渲染参数

可设置的参数包括有：画面的旋转角度、填充模式以及左右镜像等。

参数	描述
params	画面渲染参数，详情请参见 TRTCRenderParams 。

setRemoteRenderParams:streamType:params:

setRemoteRenderParams:streamType:params:

```
- (void)setRemoteRenderParams:  (NSString *)userId
                   streamType:  (TRTCVideoStreamType)streamType
                       params:  (TRTCRenderParams *)params
```

设置远端画面的渲染模式

可设置的参数包括有：画面的旋转角度、填充模式以及左右镜像等。

参数	描述
params	画面渲染参数，详情请参见 TRTCRenderParams 。
streamType	可以设置为主路画面（ TRTCVideoStreamTypeBig ）或辅路画面（ TRTCVideoStreamTypeSub ）。
userId	指定远端用户的 ID。

enableEncSmallVideoStream:withQuality:

enableEncSmallVideoStream:withQuality:

```
- (int)enableEncSmallVideoStream:  (BOOL)enable
                     withQuality:  (TRTCVideoEncParam *)smallVideoEncPa
```

开启大小画面双路编码模式

开启双路编码模式后，当前用户的编码器会同时输出【高清大画面】和【低清小画面】两路视频流（但只有一路音频流）。
如此以来，房间中的其他用户就可以根据自身的网络情况或屏幕大小选择订阅【高清大画面】或是【低清小画面】。

参数	描述
enable	是否开启小画面编码，默认值：NO。

smallVideoEncParam	小流的视频参数。

注意
双路编码开启后，会消耗更多的 CPU 和 网络带宽，所以 Mac、Windows 或者高性能 Pad 可以考虑开启，不建议手机端开启。

返回值说明：

0：成功；-1：当前大画面已被设置为较低画质，开启双路编码已无必要。

setRemoteVideoStreamType:type:

setRemoteVideoStreamType:type:

```
- (void)setRemoteVideoStreamType:    (NSString*)userId
                             type:   (TRTCVideoStreamType)streamType
```

切换指定远端用户的大小画面

当房间中某个主播开启了双路编码之后，房间中其他用户通过 startRemoteView 订阅到的画面默认会是【高清大画面】。
您可以通过此接口选定希望订阅的画面是大画面还是小画面，该接口在 startRemoteView 之前和之后调用均可生效。

参数	描述
streamType	视频流类型，即选择看大画面还是小画面，默认为大画面。
userId	指定远端用户的 ID。

注意
此功能需要目标用户已经通过 enableEncSmallVideoStream 提前开启了双路编码模式，否则此调用无实际效果。

snapshotVideo:type:sourceType:

snapshotVideo:type:sourceType:

```
- (void)snapshotVideo:    (nullable NSString *)userId
                 type:    (TRTCVideoStreamType)streamType
           sourceType:    (TRTCSnapshotSourceType)sourceType
```

视频画面截图

您可以通过本接口截取本地的视频画面，远端用户的主路画面以及远端用户的辅路（屏幕分享）画面。

参数	描述
sourceType	画面来源，可选择截取视频流画面（TRTCSnapshotSourceTypeStream）或视频渲染画面（TRTCSnapshotSourceTypeView），前者一般更清晰。
streamType	视频流类型，可选择截取主路画面（TRTCVideoStreamTypeBig，常用于摄像头）或辅路画面（TRTCVideoStreamTypeSub，常用于屏幕分享）。
userId	用户 ID，如指定空置表示截取本地的视频画面。

注意
Windows 平台目前仅支持截取 TRTCSnapshotSourceTypeStream 来源的视频画面。

setPerspectiveCorrectionWithUser:srcPoints:dstPoints:

setPerspectiveCorrectionWithUser:srcPoints:dstPoints:

```
- (void)setPerspectiveCorrectionWithUser:    (nullable NSString *)userId
                                 srcPoints:    (nullable NSArray *)srcPoints
                                 dstPoints:    (nullable NSArray *)dstPoints
```

视频画面透视校正坐标设置

您可以通过本接口指定渲染画面透视校正的坐标区域。

参数	描述

dstPoints	期望校正到的目标坐标区域4个顶点坐标，需按照左上、左下、右上、右下的顺序传入，所有的坐标需要根据画面宽高做 [0,1] 区间的归一化；传 null 则停止透视校正。
srcPoints	原始画面坐标区域4个顶点坐标，需按照左上、左下、右上、右下的顺序传入，所有的坐标需要根据画面宽高做 [0,1] 区间的归一化; 传 null 则停止透视校正。
userId	用户 ID，如指定空值表示校正本地的视频画面。

setGravitySensorAdaptiveMode:

setGravitySensorAdaptiveMode:

```
- (void)setGravitySensorAdaptiveMode:(TRTCGravitySensorAdaptiveMode)mo
```

设置重力感应的适配模式（11.7 及以上版本）

开启重力感应后，如果采集端的设备发生旋转，采集端和观众端的画面都会进行相应地渲染以确保视野中的画面始终朝上。
只在sdk内部摄像头采集场景生效，并且只在移动端生效。
1. 该接口仅对采集端起作用，如果只是观看房间中的画面，开启此接口是无效的
2. 当采集端设备发生 90 度或 270 度旋转时，采集端或者观众看到的画面可能会被裁剪以保持比例的协调

参数	描述
mode	重力感应模式，详情请参见 TRTCGravitySensorAdaptiveMode_Disable 、TRTCGravitySensorAdaptiveMode_FillByCenterCrop 和 TRTCGravitySensorAdaptiveMode_FitWithBlackBorder 默认值： TRTCGravitySensorAdaptiveMode_Disable 。

startLocalAudio:

startLocalAudio:

```
- (void)startLocalAudio:(TRTCAudioQuality)quality
```

开启本地音频的采集和发布

SDK 默认不开启麦克风，当用户需要发布本地音频时，需要调用该接口开启麦克风采集，并将音频编码并发布到当前的房间中。
开启本地音频的采集和发布后，房间中的其他用户会收到 onUserAudioAvailable(userId, YES) 的通知。

参数	描述
quality	声音音质 • TRTCAudioQualitySpeech，流畅：采样率：16k；单声道；音频裸码率：16kbps；适合语音通话为主的场景，比如在线会议，语音通话。 • TRTCAudioQualityDefault，默认：采样率：48k；单声道；音频裸码率：50kbps；SDK 默认的音频质量，如无特殊需求推荐选择之。 • TRTCAudioQualityMusic，高音质：采样率：48k；双声道 + 全频带；音频裸码率：128kbps；适合需要高保真传输音乐的场景，比如在线K歌、音乐直播等。

注意
该函数会检查麦克风的使用权限，如果当前 App 没有麦克风权限，SDK 会自动向用户申请麦克风使用权限。

stopLocalAudio

stopLocalAudio

停止本地音频的采集和发布

停止本地音频的采集和发布后，房间中的其他用户会收到 onUserAudioAvailable(userId, NO) 的通知。

muteLocalAudio:

muteLocalAudio:

```
- (void)muteLocalAudio:    (BOOL)mute
```

暂停/恢复发布本地的音频流

当您暂停发布本地音频流之后，房间中的其他用户会收到 onUserAudioAvailable(userId, NO) 的通知。
当您恢复发布本地音频流之后，房间中的其他用户会收到 onUserAudioAvailable(userId, YES) 的通知。

与 stopLocalAudio 的不同之处在于，muteLocalAudio(YES) 并不会释放麦克风权限，而是继续发送码率极低的静音包。
这对于需要云端录制的场景非常适用，因为 MP4 等格式的视频文件，对于音频数据的连续性要求很高，使用 stopLocalAudio 会导致录制出的 MP4 文件不易播放。
因此在对录制文件的质量要求较高的场景中，建议选择 muteLocalAudio 而不建议使用 stopLocalAudio。

参数	描述
mute	YES：静音；NO：恢复。

muteRemoteAudio:mute:

muteRemoteAudio:mute:

```
- (void)muteRemoteAudio:    (NSString *)userId
                   mute:    (BOOL)mute
```

暂停/恢复播放远端的音频流

当您静音某用户的远端音频时，SDK 会停止播放指定用户的声音，同时也会停止拉取该用户的音频数据。

参数	描述
mute	YES：静音；NO：取消静音。
userId	用于指定远端用户的 ID。

注意
在进入房间（enterRoom）之前或之后调用本接口均生效，静音状态在退出房间（exitRoom）之后会被重置为 NO。

muteAllRemoteAudio:

muteAllRemoteAudio:

- (void)muteAllRemoteAudio: (BOOL)mute

暂停/恢复播放所有远端用户的音频流

当您静音所有用户的远端音频时，SDK 会停止播放所有来自远端的音频流，同时也会停止拉取所有用户的音频数据。

参数	描述
mute	YES：静音；NO：取消静音。

注意
在进入房间（enterRoom）之前或之后调用本接口均生效，静音状态在退出房间（exitRoom）之后会被重置为 NO。

setAudioRoute:

setAudioRoute:

- (void)setAudioRoute: (TRTCAudioRoute)route

设置音频路由

设置“音频路由”，即设置声音是从手机的扬声器还是从听筒中播放出来，因此该接口仅适用于手机等移动端设备。

手机有两个扬声器：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。
设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。

参数	描述
route	音频路由，即声音由哪里输出（扬声器、听筒），默认值：TRTCAudioModeSpeakerphone。

setRemoteAudioVolume:volume:

setRemoteAudioVolume:volume:

```
- (void)setRemoteAudioVolume:    (NSString *)userId
                      volume:    (int)volume
```

设定某一个远端用户的声音播放音量

您可以通过 setRemoteAudioVolume(userId, 0) 将某一个远端用户的声音静音。

参数	描述
userId	用于指定远端用户的 ID。
volume	音量大小，取值范围为0 - 100，默认值：100。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setAudioCaptureVolume:

setAudioCaptureVolume:

```
- (void)setAudioCaptureVolume:    (NSInteger)volume
```

设定本地音频的采集音量

参数	描述
volume	音量大小，取值范围为0 - 100；默认值：100。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

getAudioCaptureVolume

getAudioCaptureVolume

获取本地音频的采集音量

setAudioPlayoutVolume:

setAudioPlayoutVolume:

```
- (void)setAudioPlayoutVolume:      (NSInteger)volume
```

设定远端音频的播放音量

该接口会控制 SDK 最终交给系统播放的声音音量，调节效果会影响到本地音频录制文件的音量大小，但不会影响到耳返的音量大小。

参数	描述
volume	音量大小，取值范围为0 - 100，默认值：100。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

getAudioPlayoutVolume

getAudioPlayoutVolume

获取远端音频的播放音量

enableAudioVolumeEvaluation:withParams:

enableAudioVolumeEvaluation:withParams:

```
- (void)enableAudioVolumeEvaluation:    (BOOL)enable
                         withParams:    (TRTCAudioVolumeEvaluateParams *)pa
```

启用音量大小提示

开启此功能后，SDK 会在 TRTCCloudDelegate 中的 onUserVoiceVolume 回调中反馈本地或远端用户的音频音量评估信息，包括音量大小、人声检测、音频频谱等。

参数	描述
enable	是否启用音量提示，默认为关闭状态。
params	音量评估等相关参数，请参见 TRTCAudioVolumeEvaluateParams 。

startAudioRecording:

startAudioRecording:

```
- (int)startAudioRecording:    (TRTCAudioRecordingParams *) param
```

开始录音

当您调用该接口后， SDK 会将本地和远端的所有音频（包括本地音频，远端音频，背景音乐和音效等）混合并录制到一个本地文件中。
该接口在进入房间前后调用均可生效，如果录制任务在退出房间前尚未通过 stopAudioRecording 停止，则退出房间后录制任务会自动被停止。
本次录制的启动、完成状态会通过本地录制相关回调进行通知。参见 TRTCCloud 相关回调。

参数	描述
param	录音参数，请参见 TRTCAudioRecordingParams 。

> **注意**
> 自 v11.5 版本，音频录制的状态结果由返回值统一调整为异步回调进行通知。参见 TRTCCloud 相关回调。

返回值说明：

0：成功；-1：录音已开始；-2：文件或目录创建失败；-3：后缀指定的音频格式不支持。

stopAudioRecording

stopAudioRecording

停止录音

如果录制任务在退出房间前尚未通过本接口停止，则退出房间后录音任务会自动被停止。

startLocalRecording:

startLocalRecording:

```
- (void)startLocalRecording:(TRTCLocalRecordingParams *)params
```

开启本地媒体录制

开启后把直播过程中的音视频内容录制到本地的一个文件中。

参数	描述
params	录制参数，请参见 TRTCLocalRecordingParams 。

stopLocalRecording

stopLocalRecording

停止本地媒体录制

如果录制任务在退出房间前尚未通过本接口停止，则退出房间后录音任务会自动被停止。

版权所有：腾讯云计算（北京）有限责任公司

setRemoteAudioParallelParams:

setRemoteAudioParallelParams:

```
- (void)setRemoteAudioParallelParams:(TRTCAudioParallelParams *)params
```

设置远端音频流智能并发播放策略

设置远端音频流智能并发播放策略，适用于上麦人数比较多的场景。

参数	描述
params	音频并发参数，请参见 TRTCAudioParallelParams 。

enable3DSpatialAudioEffect:

enable3DSpatialAudioEffect:

```
- (void)enable3DSpatialAudioEffect:(BOOL)enabled
```

启用 3D 音效

启用 3D 音效。注意需使用流畅音质 TRTCAudioQualitySpeech 或默认音质 TRTCAudioQualityDefault 。

参数	描述
enabled	是否启用 3D 音效，默认为关闭状态。

updateSelf3DSpatialPosition

updateSelf3DSpatialPosition

设置 3D 音效中自身坐标及朝向信息

更新自身在世界坐标系中的位置和朝向， SDK 会根据该方法参数计算自身和远端用户之间的相对位置，进而渲染出空间音效。注意各参数应分别传入长度为 3 的数组。

参数	描述
axisForward	自身坐标系前轴在世界坐标系中的单位向量，三个值依次表示前、右、上坐标值。
axisRight	自身坐标系右轴在世界坐标系中的单位向量，三个值依次表示前、右、上坐标值。
axisUp	自身坐标系上轴在世界坐标系中的单位向量，三个值依次表示前、右、上坐标值。
position	自身在世界坐标系中的坐标，三个值依次表示前、右、上坐标值。

注意
请适当限制调用频率，推荐两次坐标设置至少间隔 100ms。

updateRemote3DSpatialPosition:

updateRemote3DSpatialPosition:

```
- (void)updateRemote3DSpatialPosition:      (NSString *)userId
```

设置 3D 音效中远端用户坐标信息

更新远端用户在世界坐标系中的位置，SDK 会根据自身和远端用户之间的相对位置，进而渲染出空间音效。注意参数为长度等于 3 的数组。

参数	描述
position	该远端用户在世界坐标系中的坐标，三个值依次表示前、右、上坐标值。
userId	指定远端用户的 ID。

注意
请适当限制调用频率，推荐同一远端用户两次坐标设置至少间隔 100ms。

set3DSpatialReceivingRange:range:

set3DSpatialReceivingRange:range:

```
- (void)set3DSpatialReceivingRange:    (NSString *)userId
                             range:    (NSInteger)range
```

设置指定用户所发出声音的可被接收范围

设置该范围大小之后，该指定用户的声音将在该范围内可被听见，超出该范围将被衰减为 0。

参数	描述
range	声音最大可被接收范围。
userId	指定远端用户的 ID。

getDeviceManager

getDeviceManager

获取设备管理类（TXDeviceManager）

getBeautyManager

getBeautyManager

获取美颜管理类（TXBeautyManager）

通过美颜管理，您可以使用以下功能：

- 设置“磨皮”、“美白”、“红润”等美颜特效。
- 设置“大眼”、“瘦脸”、“V脸”、“下巴”、“短脸”、“小鼻”、“亮眼”、“白牙”、“祛眼袋”、“祛皱纹”、“祛法令纹”等修脸特效。
- 设置“发际线”、“眼间距”、“眼角”、“嘴形”、“鼻翼”、“鼻子位置”、“嘴唇厚度”、“脸型”等修脸特效。
- 设置“眼影”、“腮红”等美妆特效。

- 设置动态贴纸和人脸挂件等动画特效。

setWatermark:streamType:rect:

setWatermark:streamType:rect:

```
- (void)setWatermark:(nullable TXImage*)image
          streamType:(TRTCVideoStreamType)streamType
                rect:(CGRect)rect
```

添加水印

水印的位置是通过 rect 参数来指定的，rect 是一个四元组参数，其格式为 (x，y，width，height)

- x：水印的坐标，取值范围为0 – 1的浮点数。
- y：水印的坐标，取值范围为0 – 1的浮点数。
- width：水印的宽度，取值范围为0 – 1的浮点数。
- height：是不用设置的，SDK 内部会根据水印图片的宽高比自动计算一个合适的高度。

参数设置举例：
如果当前视频的编码分辨率是 540 × 960，且 rect 参数被您设置为（0.1，0.1，0.2，0.0），那么水印的左上坐标点就是（540 × 0.1，960 × 0.1）即（54，96），水印的宽度是 540 × 0.2 = 108px，水印的高度会根据水印图片的宽高比由 SDK 自动算出。

参数	描述
image	水印图片，**必须使用透明底色的 png 格式**。
rect	水印相对于编码分辨率的归一化坐标，x，y，width，height 取值范围0 – 1。
streamType	指定给哪一路画面设置水印，详情请参见TRTCVideoStreamType 。

注意
如果您要给主画面（一般为摄像头）和辅路画面（一般用作屏幕分享）同时设置水印，需要调用该接口两次，并设定不同的 streamType。

getAudioEffectManager

getAudioEffectManager

获取音效管理类（TXAudioEffectManager）

TXAudioEffectManager 是音效管理接口，您可以通过该接口实现如下功能：

- 背景音乐：支持在线音乐和本地音乐，支持变速、变调等特效、支持原生和伴奏并播放和循环播放。
- 耳机耳返：麦克风捕捉的声音实时通过耳机播放，常用于音乐直播。
- 混响效果：KTV、小房间、大会堂、低沉、洪亮...。
- 变声特效：萝莉、大叔、重金属...。
- 短音效：鼓掌声、欢笑声等简短的音效文件（对于小于10秒的文件，请将 isShortFile 参数设置为 YES）。

startSystemAudioLoopback

startSystemAudioLoopback

开启系统声音采集（iOS 端暂未支持）

该接口会从电脑的声卡中采集音频数据，并将其混入到 SDK 当前的音频数据流中，从而使房间中的其他用户也能听到主播的电脑所播放出的声音。
在线教育场景中，老师可以使用此功能让 SDK 采集教学影片中的声音，并广播给同房间中的学生。
音乐直播场景中，主播可以使用此功能让 SDK 采集音乐播放器中的音乐，从而为自己的直播间增加背景音乐。

> **注意**
> 1. 此功能需要为用户的 Mac 系统安装虚拟音频设备插件，安装完成后，SDK 会从已经安装的虚拟音频设备中采集声音。
> 2. SDK 会自动从网络上下载合适的插件进行安装，但是下载速度可能比较慢，如果您希望加速这个过程，可以将虚拟音频插件文件打包到您 App Bundle 的 Resources 目录下。

stopSystemAudioLoopback

stopSystemAudioLoopback

停止系统声音采集（iOS 端暂未支持）

setSystemAudioLoopbackVolume:

setSystemAudioLoopbackVolume:

```
- (void)setSystemAudioLoopbackVolume:(uint32_t)volume
```

设置系统声音的采集音量

参数	描述
volume	设置的音量大小，范围是：[0 ~ 150]，默认值为100。

startScreenCaptureInApp:encParam:

startScreenCaptureInApp:encParam:

```
- (void)startScreenCaptureInApp:(TRTCVideoStreamType)streamType
                       encParam:(TRTCVideoEncParam *)encParams
```

开始应用内的屏幕分享（仅支持 iOS 13.0 及以上系统）

该接口会抓取当前应用内的实时屏幕内容并将其分享给同房间中的其他用户，适用于 13.0 以上的 iOS 系统。
如果您希望抓取整个 iOS 系统的屏幕内容（而不受当前应用的限制），推荐使用 startScreenCaptureByReplaykit 。

iPhone 推荐的屏幕分享视频编码参数（ TRTCVideoEncParam ）：

- 分辨率(videoResolution): 1280 x 720。
- 帧率(videoFps): 10 FPS。
- 码率(videoBitrate): 1600 kbps。
- 分辨率自适应(enableAdjustRes): NO。

参数	描述
encParams	设置屏幕分享时的视频编码参数，推荐采用上述推荐配置。 如果您指定 encParams 为 nil，SDK 会使用您在调用 startScreenCapture 接口之前所设置的视频编码参数。

streamType	屏幕分享使用的线路，可以设置为主路（TRTCVideoStreamTypeBig）或者辅路（TRTCVideoStreamTypeSub）。

startScreenCaptureByReplaykit:encParam:appGroup:

startScreenCaptureByReplaykit:encParam:appGroup:

```
- (void)startScreenCaptureByReplaykit:(TRTCVideoStreamType)streamType
                             encParam:(TRTCVideoEncParam *)encParams
                             appGroup:(NSString *)appGroup
```

开始全系统的屏幕分享（仅支持 iOS 11.0 及以上系统）

该接口支持抓取整个 iOS 系统的屏幕，可以实现类似腾讯会议的全系统级的屏幕分享。
但对接步骤要比 startScreenCaptureInApp 略繁琐一些，需要为您的应用实现一个 Replaykit 扩展模块。

参考文档：实时屏幕分享(iOS)

iPhone 推荐的屏幕分享视频编码参数（TRTCVideoEncParam）：

- 分辨率(videoResolution)：1280 x 720。
- 帧率(videoFps)：10 FPS。
- 码率(videoBitrate)：1600 kbps。
- 分辨率自适应(enableAdjustRes)：NO。

参数	描述
appGroup	用于指定您的应用与录屏进程共享的 Application Group Identifier，您可以指定该参数为 nil，但推荐您按照文档指示进行设置，从而获得更好的可靠性。
encParams	设置屏幕分享时的视频编码参数，推荐采用上述推荐配置。如果您指定 encParams 为 nil，SDK 会使用您在调用 startScreenCapture 接口之前所设置的视频编码参数。
streamType	屏幕分享使用的线路，可以设置为主路（TRTCVideoStreamTypeBig）或者辅路（TRTCVideoStreamTypeSub）。

startScreenCapture:streamType:encParam:

startScreenCapture:streamType:encParam:

```
- (void)startScreenCapture:    (nullable NSView *)view
                streamType:    (TRTCVideoStreamType)streamType
                  encParam:    (nullable TRTCVideoEncParam *)encPara
```

启动屏幕分享

该接口可以抓取整个屏幕的内容，或抓取您指定的某个应用的窗口内容，并将其分享给同房间中的其他用户。

参数	描述
encParam	屏幕分享的画面编码参数，SDK 会优先使用您通过此接口设置的编码参数： • 如果您设置 encParam 为空值，且您已通过 setSubStreamEncoderParam 设置过辅路视频编码参数，SDK 将使用您设置过的辅路编码参数进行屏幕分享。 • 如果您设置 encParam 为空值，且您未通过 setSubStreamEncoderParam 设置过辅路视频编码参数，SDK 将自动选择一个最佳的编码参数进行屏幕分享。
streamType	屏幕分享使用的线路，可以设置为主路（TRTCVideoStreamTypeBig）或者辅路（TRTCVideoStreamTypeSub），推荐使用辅路。
view	渲染控件所在的父控件，可以设置为空值，表示不显示屏幕分享的预览效果。

注意

1. 同一个用户同时最多只能发布一路主路（TRTCVideoStreamTypeBig）画面和一路辅路（TRTCVideoStreamTypeSub）画面。
2. 默认情况下，屏幕分享使用辅路画面。如果使用主路做屏幕分享，您需要提前停止摄像头采集（stopLocalPreview）以避免相互冲突。
3. 同一个房间中同时只能有一个用户使用辅路做屏幕分享，也就是说，同一个房间中同时只允许一个用户开启辅路。
4. 当房间中已经有其他用户在使用辅路分享屏幕时，此时调用该接口会收到来自 TRTCCloudDelegate 的 onError(ERR_SERVER_CENTER_ANOTHER_USER_PUSH_SUB_VIDEO) 回调。

stopScreenCapture

stopScreenCapture

停止屏幕分享

pauseScreenCapture

pauseScreenCapture

暂停屏幕分享

> **注意**
> 从v11.5版本开始，暂停屏幕采集会使用最后一帧按照1fps帧率输出。

resumeScreenCapture

resumeScreenCapture

恢复屏幕分享

getScreenCaptureSourcesWithThumbnailSize:iconSize:

getScreenCaptureSourcesWithThumbnailSize:iconSize:

```
- (NSArray<TRTCScreenCaptureSourceInfo*>*)getScreenCaptureSourcesWithThumbnailSize:(
                                          iconSize:(
```

枚举可分享的屏幕和窗口（该接口仅支持 Mac OS 系统）

当您在对接桌面端系统的屏幕分享功能时，一般都需要展示一个选择分享目标的界面，这样用户能够使用这个界面选择是分享整个屏幕还是某个窗口。
通过本接口，您就可以查询到当前系统中可用于分享的窗口的 ID、名称以及缩略图。我们在 Demo 中提供了一份默认的界面实现供您参考。

参数	描述
iconSize	指定要获取的窗口图标大小。
thumbnailSize	指定要获取的窗口缩略图大小，缩略图可用于绘制在窗口选择界面上。

注意
返回的列表中包含屏幕和应用窗口，屏幕是列表中的第一个元素。如果用户有多个显示器，那么每个显示器都是一个分享目标。

返回值说明：

窗口列表包括屏幕。

selectScreenCaptureTarget:rect:capturesCursor:highlight:

selectScreenCaptureTarget:rect:capturesCursor:highlight:

```
- (void)selectScreenCaptureTarget:(TRTCScreenCaptureSourceInfo *)screen
                             rect:(CGRect)rect
                    capturesCursor:(BOOL)capturesCursor
                        highlight:(BOOL)highlight
```

选取要分享的屏幕或窗口（该接口仅支持 Mac OS 系统）

当您通过 getScreenCaptureSources 获取到可以分享的屏幕和窗口之后，您可以调用该接口选定期望分享的目标屏幕或目标窗口。
在屏幕分享的过程中，您也可以随时调用该接口以切换分享目标。

参数	描述
capturesCursor	是否捕获鼠标光标。

highlight	是否高亮正在分享的窗口。
rect	指定捕获的区域（设定该参数为 CGRectZero：当分享目标是某个窗口时则分享整个窗口，当分享目标是桌面时则分享整个桌面）。
screenSource	指定分享源。

setSubStreamEncoderParam:

setSubStreamEncoderParam:

```
- (void)setSubStreamEncoderParam:(TRTCVideoEncParam *)param
```

设置屏幕分享（即辅路）的视频编码参数（桌面系统和移动系统均已支持）

该接口可以设定远端用户所看到的屏幕分享（即辅路）的画面质量，同时也能决定云端录制出的视频文件中屏幕分享的画面质量。
请注意如下两个接口的差异：

- setVideoEncoderParam 用于设置主路画面（TRTCVideoStreamTypeBig，一般用于摄像头）的视频编码参数。
- setSubStreamEncoderParam 用于设置辅路画面（TRTCVideoStreamTypeSub，一般用于屏幕分享）的视频编码参数。

参数	描述
param	辅流编码参数，详情请参见 TRTCVideoEncParam。

setSubStreamMixVolume:

setSubStreamMixVolume:

```
- (void)setSubStreamMixVolume:(NSInteger)volume
```

设置屏幕分享时的混音音量大小（该接口仅支持桌面系统）

这个数值越高，屏幕分享音量的占比就越高，麦克风音量占比就越小，所以不推荐设置得太大，否则麦克风的声音就被压制了。

参数	描述
volume	设置的混音音量大小，范围0 - 100。

addExcludedShareWindow:

addExcludedShareWindow:

```
- (void)addExcludedShareWindow:      (NSInteger)windowID
```

将指定窗口加入屏幕分享的排除列表中（该接口仅支持桌面系统）

加入排除列表中的窗口不会被分享出去，常见的用法是将某个应用的窗口加入到排除列表中以避免隐私问题。支持启动屏幕分享前设置过滤窗口，也支持屏幕分享过程中动态添加过滤窗口。

参数	描述
windowID	不希望分享出去的窗口

注意

1. 该接口只有在 TRTCScreenCaptureSourceInfo 中的 type 指定为 TRTCScreenCaptureSourceTypeScreen 时生效，即只有在分享整个屏幕内容时，排除指定窗口的功能才生效。
2. 使用该接口添加到排除列表中的窗口会在退出房间后被 SDK 自动清除。
3. Mac 平台下请传入窗口 ID（即 CGWindowID），您可以通过 TRTCScreenCaptureSourceInfo 中的 sourceId 成员获得。

removeExcludedShareWindow:

removeExcludedShareWindow:

```
- (void)removeExcludedShareWindow:      (NSInteger)windowID
```

将指定窗口从屏幕分享的排除列表中移除（该接口仅支持桌面系统）

参数	描述

window ID	要排除的窗口 id。

removeAllExcludedShareWindows

removeAllExcludedShareWindows

将所有窗口从屏幕分享的排除列表中移除（该接口仅支持桌面系统）

addIncludedShareWindow:

addIncludedShareWindow:

```
- (void)addIncludedShareWindow:      (NSInteger)windowID
```

将指定窗口加入屏幕分享的包含列表中（该接口仅支持桌面系统）

该接口只有在 TRTCScreenCaptureSourceInfo 中的 type 指定为 TRTCScreenCaptureSourceTypeWindow 时生效。即只有在分享窗口内容时，额外包含指定窗口的功能才生效。
您在 startScreenCapture 之前和之后调用均可。

参数	描述
window ID	希望被分享出去的窗口（Windows 平台下为窗口句柄： HWND ）

注意
通过该方法添加到包含列表中的窗口，会在退出房间后被 SDK 自动清除。

removeIncludedShareWindow:

removeIncludedShareWindow:

```
- (void)removeIncludedShareWindow:      (NSInteger)windowID
```

将指定窗口从屏幕分享的包含列表中移除（该接口仅支持桌面系统）

该接口只有在 TRTCScreenCaptureSourceInfo 中的 type 指定为 TRTCScreenCaptureSourceTypeWindow 时生效。
即只有在分享窗口内容时，额外包含指定窗口的功能才生效。

参数	描述
window ID	希望被分享出去的窗口（Mac 平台：窗口 ID；Windows 平台：HWND）

removeAllIncludedShareWindows

removeAllIncludedShareWindows

将全部窗口从屏幕分享的包含列表中移除（该接口仅支持桌面系统）

该接口只有在 TRTCScreenCaptureSourceInfo 中的 type 指定为 TRTCScreenCaptureSourceTypeWindow 时生效。
即只有在分享窗口内容时，额外包含指定窗口的功能才生效。

enableCustomVideoCapture:enable:

enableCustomVideoCapture:enable:

```
- (void)enableCustomVideoCapture:    (TRTCVideoStreamType)streamType
                          enable:    (BOOL)enable
```

启用/关闭视频自定义采集模式

开启该模式后，SDK 不在运行原有的视频采集流程，即不再继续从摄像头采集数据和美颜，而是只保留视频编码和发送能力。
您需要通过 sendCustomVideoData 不断地向 SDK 塞入自己采集的视频画面。

参数	描述
enable	是否启用，默认值：NO。
streamTyp	用于指定视频流类型，TRTCVideoStreamTypeBig：高清大画

e	面；TRTCVideoStreamTypeSub：辅路画面。

sendCustomVideoData:frame:

sendCustomVideoData:frame:

```
- (void)sendCustomVideoData:    (TRTCVideoStreamType)streamType
                      frame:    (TRTCVideoFrame *)frame
```

向 SDK 投送自己采集的视频帧

使用此接口可以向 SDK 投送自己采集的视频帧，SDK 会将视频帧进行编码并通过自身的网络模块传输出去。

参数 TRTCVideoFrame 推荐下列填写方式（其他字段不需要填写）：

- pixelFormat：推荐选择 TRTCVideoPixelFormat_NV12 。
- bufferType：推荐选择 TRTCVideoBufferType_PixelBuffer 。
- pixelBuffer：iOS/Mac OS 平台上常用的视频数据格式。
- data：视频裸数据格式，bufferType 为 NSData 时使用。
- timestamp：时间戳，单位为毫秒（ms），请使用视频帧在采集时被记录下来的时间戳（可以在采集到一帧视频帧之后，通过调用 generateCustomPTS 获取时间戳）。
- width：视频图像长度，bufferType 为 NSData 时填写。
- height：视频图像宽度，bufferType 为 NSData 时填写。

参考文档：自定义采集和渲染 。

参数	描述
frame	视频数据，支持 PixelBuffer NV12，BGRA 以及 I420 格式数据。
streamType	用于指定视频流类型，TRTCVideoStreamTypeBig：高清大画面；TRTCVideoStreamTypeSub：辅路画面。

注意

1. 推荐您在采集到的一帧视频帧后，即调用 generateCustomPTS 接口获取该帧的 timestamp 数值，这样可以获得最佳的音画同步效果。

2. SDK 最终编码出的视频帧率并不是由您调用本接口的频率决定的，而是由您在 setVideoEncoderParam 中所设置的 FPS 决定的。
3. 请尽量保持本接口的调用间隔是均匀的，否则会导致编码器输出帧率不稳或者音画不同步等问题。

enableCustomAudioCapture:

enableCustomAudioCapture:

```
- (void)enableCustomAudioCapture:    (BOOL)enable
```

启用音频自定义采集模式

开启该模式后，SDK 不在运行原有的音频采集流程，即不再继续从麦克风采集音频数据，而是只保留音频编码和发送能力。
您需要通过 sendCustomAudioData 不断地向 SDK 塞入自己采集的音频数据。

参数	描述
enable	是否启用，默认值：NO。

注意
由于回声抵消（AEC）需要严格的控制声音采集和播放的时间，所以开启自定义音频采集后，AEC 能力可能会失效。

sendCustomAudioData:

sendCustomAudioData:

```
- (void)sendCustomAudioData:    (TRTCAudioFrame *)frame
```

向 SDK 投送自己采集的音频数据

参数 TRTCAudioFrame 推荐下列填写方式（其他字段不需要填写）：

- audioFormat：音频数据格式，仅支持 TRTCAudioFrameFormatPCM。
- data：音频帧 buffer。音频帧数据只支持 PCM 格式，支持[5ms ~ 100ms]帧长，推荐使用 20ms 帧长，长度计算方法：【48000采样率、单声道的帧长度：48000 × 0.02s × 1 × 16bit = 15360bit =

1920字节】。

- sampleRate：采样率，支持：16000、24000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- timestamp：时间戳，单位为毫秒（ms），请使用音频帧在采集时被记录下来的时间戳（可以在采集到一帧音频帧之后，通过调用 generateCustomPTS 获取时间戳）。

参考文档： 自定义采集和渲染 。

参数	描述
frame	音频数据

注意
请您精准地按每帧时长的间隔调用本接口，数据投送间隔不均匀时极易触发声音卡顿。

enableMixExternalAudioFrame:playout:

enableMixExternalAudioFrame:playout:

```
- (void)enableMixExternalAudioFrame:    (BOOL)enablePublish
                            playout:    (BOOL)enablePlayout
```

启用/关闭自定义音轨

开启后，您可以通过本接口向 SDK 混入一条自定义的音轨。通过两个布尔型参数，您可以控制该音轨是否要在远端和本地播放。

参数	描述
enablePlayout	控制混入的音轨是否要在本地播放，默认值：NO。
enablePublish	控制混入的音轨是否要在远端播放，默认值：NO。

注意
如果您指定参数 enablePublish 和 enablePlayout 均为 NO，代表完全关闭您的自定义音轨。

mixExternalAudioFrame:

mixExternalAudioFrame:

- (int)mixExternalAudioFrame: (TRTCAudioFrame *)frame

向 SDK 混入自定义音轨

调用该接口之前，您需要先通过 enableMixExternalAudioFrame 开启自定义音轨，之后就可以通过本接口将自己的音轨以 PCM 格式混入到 SDK 中。
理想情况下，我们期望您的代码能够以非常均匀的速度向 SDK 提供音轨数据。但我们也非常清楚，完美的调用间隔是一个巨大的挑战。
所以 SDK 内部会开启一个音轨数据的缓冲区，该缓冲区的作用类似一个“蓄水池”，它能够暂存您传入的音轨数据，平抑由于接口调用间隔的抖动问题。
本接口的返回值代表这个音轨缓冲区的大小，单位是毫秒（ms），比如：如果该接口返回 50，则代表当前的音轨缓冲区有 50ms 的音轨数据。因此只要您在 50ms 内再次调用本接口，SDK 就能保证您混入的音轨数据是连续的。
当您调用该接口后，如果发现返回值 > 100ms，则可以等待一帧音频帧的播放时间之后再次调用；如果返回值 < 100ms，则代表缓冲区比较小，您可以再次混入一些音轨数据以确保音轨缓冲区的大小维持在“安全水位”以上。

参数 TRTCAudioFrame 推荐下列填写方式（其他字段不需要填写）：

- data：音频帧 buffer。音频帧数据只支持 PCM 格式，支持[5ms ~ 100ms]帧长，推荐使用 20ms 帧长，长度计算方法：【48000采样率、单声道的帧长度：48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节】。
- sampleRate：采样率，支持：16000、24000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- timestamp：时间戳，单位为毫秒（ms），请使用音频帧在采集时被记录下来的时间戳（可以在获得一帧音频帧之后，通过调用 generateCustomPTS 获得时间戳）。

参数	描述
frame	音频数据

注意
请您精准地按每帧时长的间隔调用本接口，数据投送间隔不均匀时极易触发声音卡顿。

返回值说明：

>= 0 缓冲的长度，单位：ms。< 0 错误（-1 未启用 mixExternalAudioFrame）

setMixExternalAudioVolume:playoutVolume:

setMixExternalAudioVolume:playoutVolume:

```
- (void)setMixExternalAudioVolume:  (NSInteger)publishVolume
                     playoutVolume:  (NSInteger)playoutVolume
```

设置推流时混入外部音频的推流音量和播放音量

参数	描述
playoutVolume	设置的播放音量大小，范围0 - 100，-1表示不改变。
publishVolume	设置的推流音量大小，范围0 - 100，-1表示不改变。

generateCustomPTS

generateCustomPTS

生成自定义采集时的时间戳

本接口仅适用于自定义采集模式，用于解决音视频帧的采集时间（capture time）和投送时间（send time）不一致所导致的音画不同步问题。

当您通过 sendCustomVideoData 或 sendCustomAudioData 等接口进行自定义视频或音频采集时，请按照如下操作使用该接口：
1. 首先，在采集到一帧视频或音频帧时，通过调用本接口获得当时的 PTS 时间戳。
2. 之后可以将该视频或音频帧送入您使用的前处理模块（如第三方美颜组件，或第三方音效组件）。
3. 在真正调用 sendCustomVideoData 或 sendCustomAudioData 进行投送时，请将该帧在采集时记录的 PTS 时间戳赋值给 TRTCVideoFrame 或 TRTCAudioFrame 中的 timestamp 字段。

返回值说明：

时间戳（单位：ms）

setLocalVideoProcessDelegete:pixelFormat:bufferType:

setLocalVideoProcessDelegete:pixelFormat:bufferType:

```
- (int)setLocalVideoProcessDelegete:  (nullable id<TRTCVideoFrameDelegate>
                        pixelFormat:  (TRTCVideoPixelFormat)pixelFormat
                         bufferType:  (TRTCVideoBufferType)bufferType
```

设置第三方美颜的视频数据回调

设置该回调之后，SDK 会把采集到的视频帧通过您设置的 delegate 回调出来，用于第三方美颜组件进行二次处理，之后 SDK 会将处理后的视频帧进行编码和发送。

参数	描述
bufferType	指定回调的数据格式，目前仅支持 TRTCVideoBufferType_Texture
delegate	自定义预处理回调，详见 TRTCVideoFrameDelegate
pixelFormat	指定回调的像素格式，目前仅支持 TRTCVideoPixelFormat_Texture_2D

返回值说明：

0：成功；<0：错误

setLocalVideoRenderDelegate:pixelFormat:bufferType:

setLocalVideoRenderDelegate:pixelFormat:bufferType:

```
- (int)setLocalVideoRenderDelegate:  (nullable id<TRTCVideoRenderDelegate>
                        pixelFormat:  (TRTCVideoPixelFormat)pixelFormat
                         bufferType:  (TRTCVideoBufferType)bufferType
```

设置本地视频自定义渲染回调

设置该回调之后，SDK 内部会跳过原来的渲染流程，并把采集到的数据回调出来，您需要自己完成画面渲染。

- pixelFormat 指定回调的数据格式，例如 NV12、i420 以及 32BGRA。

- bufferType 指定 buffer 的类型，直接使用 PixelBuffer 效率最高；使用 NSData 相当于让 SDK 在内部做了一次内存转换，因此会有额外的性能损耗。

参考文档：自定义采集和渲染 。

参数	描述
bufferType	PixelBuffer：可以直接使用 imageWithCVImageBuffer 转成 UIImage；NSData：经过内存整理的视频数据。
delegate	自定义渲染回调。
pixelFormat	指定回调的像素格式。

返回值说明：

0：成功；< 0：错误。

setRemoteVideoRenderDelegate:delegate:pixelFormat:bufferType:

setRemoteVideoRenderDelegate:delegate:pixelFormat:bufferType:

```
- (int)setRemoteVideoRenderDelegate:  (NSString*)userId
                           delegate:  (nullable id<TRTCVideoRenderDelegate>)
                        pixelFormat:  (TRTCVideoPixelFormat)pixelFormat
                         bufferType:  (TRTCVideoBufferType)bufferType
```

设置远端视频自定义渲染回调

设置该回调之后，SDK 内部会跳过原来的渲染流程，并把采集到的数据回调出来，您需要自己完成画面渲染。

- pixelFormat 指定回调的数据格式，例如 NV12、i420 以及 32BGRA。
- bufferType 指定 buffer 的类型，直接使用 PixelBuffer 效率最高；使用 NSData 相当于让 SDK 在内部做了一次内存转换，因此会有额外的性能损耗。

参考文档：自定义采集和渲染 。

参数	描述

bufferType	PixelBuffer：可以直接使用 imageWithCVImageBuffer 转成 UIImage； NSData：经过内存整理的视频数据。
delegate	自定义渲染回调
pixelFormat	指定回调的像素格式
userId	指定远端用户的 ID

注意

调用此函数之前，需要先调用 startRemoteView(nil) 来获取远端用户的视频流（view 设置为 nil 即可），否则不会有数据回调出来。

返回值说明：

0：成功；<0：错误

setAudioFrameDelegate:

setAudioFrameDelegate:

```
- (void)setAudioFrameDelegate:    (nullable id<TRTCAudioFrameDelegate>)
```

设置音频数据自定义回调

设置该回调之后，SDK 内部会把音频数据（PCM 格式）回调出来，包括：

- onCapturedAudioFrame：本地麦克风采集到的音频数据回调
- onLocalProcessedAudioFrame：本地采集并经过音频模块前处理后的音频数据回调
- onRemoteUserAudioFrame：混音前的每一路远程用户的音频数据
- onMixedPlayAudioFrame：将各路音频混合之后并最终要由系统播放出的音频数据回调

注意

设置回调为空即代表停止自定义音频回调，反之，设置回调不为空则代表启动自定义音频回调。

setCapturedAudioFrameDelegateFormat:

setCapturedAudioFrameDelegateFormat:

- (int)setCapturedAudioFrameDelegateFormat: (TRTCAudioFrameDelegateFormat

设置本地麦克风采集出的音频帧回调格式

本接口用于设置 onCapturedAudioFrame 回调出来的 AudioFrame 的格式：

- sampleRate：采样率，支持：16000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- samplesPerCall：采样点数，定义回调数据帧长。帧长必须为 10ms 的整数倍。

如果希望用毫秒数计算回调帧长，则将毫秒数转换成采样点数的公式为：采样点数 = 毫秒数 * 采样率 / 1000。
举例：48000 采样率希望回调 20ms 帧长的数据，则采样点数应该填：960 = 20 * 48000 / 1000。

参数	描述
format	音频数据回调格式。

注意
最终回调的帧长度是以字节为单位，采样点数转换成字节数的计算公式为：字节数 = 采样点数 * channel * 2（位宽）举例：48000 采样率，双声道，20ms 帧长，采样点数为 960，字节数为 3840 = 960 * 2 * 2

返回值说明：

0：成功；<0：错误

setLocalProcessedAudioFrameDelegateFormat:

setLocalProcessedAudioFrameDelegateFormat:

- (int)setLocalProcessedAudioFrameDelegateFormat: (TRTCAudioFrameDelegateF

设置经过前处理后的本地音频帧回调格式

本接口用于设置 onLocalProcessedAudioFrame 回调出来的 AudioFrame 的格式：

- sampleRate：采样率，支持：16000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- samplesPerCall：采样点数，定义回调数据帧长。帧长必须为 10ms 的整数倍。

如果希望用毫秒数计算回调帧长，则将毫秒数转换成采样点数的公式为：采样点数 = 毫秒数 * 采样率 / 1000。
举例：48000 采样率希望回调20ms帧长的数据，则采样点数应该填: 960 = 20 * 48000 / 1000。

参数	描述
format	音频数据回调格式。

注意
最终回调的帧长度是以字节为单位，采样点数转换成字节数的计算公式为：字节数 = 采样点数 * channel * 2（位宽）举例：48000 采样率，双声道，20ms 帧长，采样点数为 960，字节数为 3840 = 960 * 2 * 2

返回值说明：

0：成功；<0：错误

setMixedPlayAudioFrameDelegateFormat:

setMixedPlayAudioFrameDelegateFormat:

```
- (int)setMixedPlayAudioFrameDelegateFormat:    (TRTCAudioFrameDelegateForma
```

设置最终要由系统播放出的音频帧回调格式

本接口用于设置 onMixedPlayAudioFrame 回调出来的 AudioFrame 的格式：

- sampleRate：采样率，支持：16000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- samplesPerCall：采样点数，定义回调数据帧长。帧长必须为 10ms 的整数倍。

如果希望用毫秒数计算回调帧长，则将毫秒数转换成采样点数的公式为：采样点数 = 毫秒数 * 采样率 / 1000。
举例：48000 采样率希望回调20ms帧长的数据，则采样点数应该填: 960 = 20 * 48000 / 1000。

参数	描述
format	音频数据回调格式。

> **注意**
> 最终回调的帧长度是以字节为单位，采样点数转换成字节数的计算公式为：字节数 = 采样点数 * channel * 2（位宽）举例：48000 采样率，双声道，20ms 帧长，采样点数为 960，字节数为 3840 = 960 * 2 * 2

返回值说明：

0：成功；<0：错误

enableCustomAudioRendering:

enableCustomAudioRendering:

```
- (void)enableCustomAudioRendering:    (BOOL)enable
```

开启音频自定义播放

如果您需要外接一些特定的音频设备，或者希望自己掌控音频的播放逻辑，您可以通过该接口启用音频自定义播放。
启用音频自定义播放后，SDK 将不再调用系统的音频接口播放数据，您需要通过 getCustomAudioRenderingFrame 获取 SDK 要播放的音频帧并自行播放。

参数	描述
enable	是否启用音频自定义播放，默认为关闭状态。

> **注意**
> 需要您在进入房间前设置才能生效，暂不支持进入房间后再设置。

getCustomAudioRenderingFrame:

getCustomAudioRenderingFrame:

```
- (void)getCustomAudioRenderingFrame:    (TRTCAudioFrame *)audioFrame
```

获取可播放的音频数据

调用该接口之前，您需要先通过 enableCustomAudioRendering 开启音频自定义播放。

参数 TRTCAudioFrame 推荐下列填写方式（其他字段不需要填写）：

- sampleRate：采样率，必填，支持 16000、24000、32000、44100、48000。
- channel：声道数，必填，单声道请填1，双声道请填2，双声道时数据是交叉的。
- data：用于获取音频数据的 buffer。需要您根据一帧音频帧的帧长度分配好 data 的内存大小。获取的 PCM 数据支持 10ms 或 20ms 两种帧长，推荐使用 20ms 的帧长。

计算公式为：采样率 x 播放时长 x 声道数量 x 2（每个样点的字节数），例如：48000采样率、单声道、且播放时长为 20ms 的一帧音频帧的 buffer 大小为 48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节。

参数	描述
audioFrame	音频数据帧。

注意

1. 参数 audioFrame 中的 sampleRate、channel 需提前设置好，同时分配好所需读取帧长的 data 空间。
2. SDK 内部会根据 sampleRate 和 channel 自动填充 data 数据。
3. 建议由系统的音频播放线程直接驱动该函数的调用，在播放完一帧音频之后，即调用该接口获取下一帧可播放的音频数据。

sendCustomCmdMsg:data:reliable:ordered:

sendCustomCmdMsg:data:reliable:ordered:

```
- (BOOL)sendCustomCmdMsg:    (NSInteger)cmdID
                    data:    (NSData *)data
                reliable:    (BOOL)reliable
                 ordered:    (BOOL)ordered
```

使用 UDP 通道发送自定义消息给房间内所有用户

该接口可以让您借助 TRTC 的 UDP 通道，向当前房间里的其他用户广播自定义数据，以达到传输信令的目的。
房间中的其他用户可以通过 TRTCCloudDelegate 中的 onRecvCustomCmdMsg 回调接收消息。

参数	描述
cmdID	消息 ID，取值范围为1 – 10。
data	待发送的消息，单个消息的最大长度被限制为 1KB。
ordered	是否要求有序，即是否要求接收端的数据包顺序和发送端的数据包顺序一致（这会带来一定的接收延时）。
reliable	是否可靠发送，可靠发送可以获得更高的发送成功率，但可靠发送比不可靠发送会带来更大的接收延迟。

注意

1. 发送消息到房间内所有用户（暂时不支持 Web/小程序端），每秒最多能发送30条消息。
2. 每个包最大为 1KB，超过则很有可能会被中间路由器或者服务器丢弃。
3. 每个客户端每秒最多能发送总计 8KB 数据。
4. 请将 reliable 和 ordered 同时设置为 YES 或同时设置为 NO，暂不支持交叉设置。
5. 强烈建议您将不同类型的消息设定为不同的 cmdID，这样可以在要求有序的情况下减小消息时延。
6. 目前仅支持主播身份。

返回值说明：

YES：消息已经发出；NO：消息发送失败。

sendSEIMsg:repeatCount:

sendSEIMsg:repeatCount:

```
- (BOOL)sendSEIMsg:    (NSData *)data
        repeatCount:    (int)repeatCount
```

使用 SEI 通道发送自定义消息给房间内所有用户

该接口可以让您借助 TRTC 的 SEI 通道，向当前房间里的其他用户广播自定义数据，已达到传输信令的目的。

视频帧的头部有一个叫做 SEI 的头部数据块，该接口的原理就是利用这个被称为 SEI 的头部数据块，将您要发送的自定义信令嵌入其中，使其同视频帧一并发送出去。
因此，与 sendCustomCmdMsg 相比，SEI 通道传输的信令具有更好的兼容性：信令可以伴随着视频帧一直传输到直播 CDN 上。
不过，由于视频帧头部的数据块不能太大，建议您使用该接口时，尽量将信令控制在几个字节的大小。

最常见的用法是把自定义的时间戳（timestamp）用本接口嵌入视频帧中，实现消息和画面的完美对齐（比如：教育场景下的课件和视频信号的对齐）。
房间中的其他用户可以通过 TRTCCloudDelegate 中的 onRecvSEIMsg 回调接收消息。

参数	描述
data	待发送的数据，最大支持 1KB（1000字节）的数据大小
repeatCount	发送数据次数

注意
本接口有以下限制：
1. 数据在接口调用完后不会被即时发送出去，而是从下一帧视频帧开始与视频帧一起发送。
2. 发送消息到房间内所有用户，每秒最多能发送 30 条消息（与 sendCustomCmdMsg 共享限制）。
3. 每个包最大为 1KB，若发送大量数据，会导致视频码率增大，可能导致视频画质下降甚至卡顿（与 sendCustomCmdMsg 共享限制）。
4. 每个客户端每秒最多能发送总计8KB数据（与 sendCustomCmdMsg 共享限制）。
5. 若指定多次发送（repeatCount > 1），则数据会被带在后续的连续 repeatCount 个视频帧中发送出去，同样会导致视频码率增大。
6. 如果 repeatCount > 1，多次发送，接收消息 onRecvSEIMsg 回调也可能会收到多次相同的消息，需要去重。

返回值说明：

YES：消息已通过限制，等待后续视频帧发送；NO：消息被限制发送

startSpeedTest:

startSpeedTest:

```
- (int)startSpeedTest:    (TRTCSpeedTestParams *)params
```

开始进行网速测试（进入房间前使用）

参数	描述
params	测速选项

> **注意**
> 1. 测速过程将产生少量的基础服务费用，详见 计费概述 > 基础服务 文档说明。
> 2. 请在进入房间前进行网速测试，在房间中网速测试会影响正常的音视频传输效果，而且由于干扰过多，网速测试结果也不准确。
> 3. 同一时间只允许一项网速测试任务运行。

返回值说明：

接口调用结果，< 0：失败

stopSpeedTest

stopSpeedTest

停止网络测速

getSDKVersion

getSDKVersion

获取 SDK 版本信息

setLogLevel:

setLogLevel:

```
+ (void)setLogLevel:    (TRTCLogLevel)level
```

设置 Log 输出级别

参数	描述
level	参见 TRTCLogLevel，默认值：TRTCLogLevelNone

setConsoleEnabled:

setConsoleEnabled:

```
+ (void)setConsoleEnabled:      (BOOL)enabled
```

启用/禁用控制台日志打印

参数	描述
enabled	指定是否启用，默认：禁止状态。

setLogCompressEnabled:

setLogCompressEnabled:

```
+ (void)setLogCompressEnabled:      (BOOL)enabled
```

启用/禁用日志的本地压缩

开启压缩后，Log 存储体积明显减小，但需要腾讯云提供的 Python 脚本解压后才能阅读。
禁用压缩后，Log 采用明文存储，可以直接用记事本打开阅读，但占用空间较大。

参数	描述
enabled	指定是否启用，默认为启动状态

setLogDirPath:

setLogDirPath:

+ (void)setLogDirPath: (NSString *)path

设置本地日志的保存路径

通过该接口您可以更改 SDK 本地日志的默认存储路径，SDK 默认的本地日志的存储位置：

- Windows 平台：在 C:/Users/[系统用户名]/AppData/Roaming/liteav/log，即 %appdata%/liteav/log 下。
- iOS 或 Mac 平台：在 sandbox Documents/log 下。
- Android 平台：在 /app私有目录/files/log/liteav/ 下。

参数	描述
path	存储日志的路径

注意
请务必在所有其他接口之前调用，并且保证您指定的目录是存在的，并且您的应用程序拥有对该目录的读写权限。

setLogDelegate:

setLogDelegate:

+ (void)setLogDelegate: (nullable id<TRTCLogDelegate>)logDeleg

设置日志回调

showDebugView:

showDebugView:

- (void)showDebugView: (NSInteger)showType

显示仪表盘

“仪表盘”是位于视频渲染控件之上的一个半透明的调试信息浮层，用于展示音视频信息和事件信息，便于对接和调试。

参数	描述
showType	0：不显示；1：显示精简版（仅显示音视频信息）；2：显示完整版（包含音视频信息和事件信息）。

setDebugViewMargin:margin:

setDebugViewMargin:margin:

```
- (void)setDebugViewMargin:    (NSString *)userId
                    margin:    (TXEdgeInsets)margin
```

设置仪表盘的边距

用于调整仪表盘在视频渲染控件中的位置，必须在 showDebugView 之前调用才能生效。

参数	描述
margin	仪表盘内边距，注意这里是基于 parentView 的百分比，margin 的取值范围是0 - 1。
userId	用户 ID。

callExperimentalAPI:

callExperimentalAPI:

```
- (NSString*)callExperimentalAPI:    (NSString*)jsonStr
```

调用实验性接口

enablePayloadPrivateEncryption:params:

enablePayloadPrivateEncryption:params:

```
- (int)enablePayloadPrivateEncryption:    (BOOL)enabled
                               params:    (TRTCPayloadPrivateEncryptionConfig
```

开启或关闭媒体流私有加密

在安全要求较高的场景下，TRTC 建议您在加入房间前，调用 enablePayloadPrivateEncryption 方法开启媒体流私有加密。
用户退出房间后，SDK 会自动关闭私有加密。如需重新开启私有加密，您需要在用户再次加入房间前调用该方法。

参数	描述
config	配置媒体流私有加密的算法和密钥，参见 TRTCPayloadPrivateEncryptionConfig 。
enabled	是否开启媒体流私有加密。

注意
TRTC 已经内置对媒体流进行加密后再传输，启用媒体流私有加密后将使用您传入的密匙与初始向量进行再次加密。

返回值说明：

接口调用结果，0: 方法调用成功， -1: 传入参数无效， -2: 功能已过期。若需解锁：请前往中国大陆站点开通 TRTC 旗舰版套餐 ，并填写 申请 ，审核通过后方可使用。

TRTCCloudDelegate

最近更新时间：2024-03-06 14:51:31

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTCCloudDelegate @ TXLiteAVSDK

Function: 腾讯云实时音视频的事件回调接口

TRTCCloudDelegate

TRTCCloudDelegate

函数列表	描述
onError:errMsg:extInfo:	错误事件回调
onWarning:warningMsg:extInfo:	警告事件回调
onEnterRoom:	进入房间成功与否的事件回调
onExitRoom:	离开房间的事件回调
onSwitchRole:errMsg:	切换角色的事件回调
onSwitchRoom:errMsg:	切换房间的结果回调
onConnectOtherRoom:errCode:errMsg:	请求跨房通话的结果回调
onDisconnectOtherRoom:errMsg:	结束跨房通话的结果回调
onUpdateOtherRoomForwardMode:errMsg:	更改跨房主播上行能力的结果回调
onRemoteUserEnterRoom:	有用户加入当前房间
onRemoteUserLeaveRoom:reason:	有用户离开当前房间
onUserVideoAvailable:available:	某远端用户发布/取消了主路视频画面
onUserSubStreamAvailable:available:	某远端用户发布/取消了辅路视频画面

onUserAudioAvailable:available:	某远端用户发布/取消了自己的音频
onFirstVideoFrame:streamType:width:height:	SDK 开始渲染自己本地或远端用户的首帧画面
onFirstAudioFrame:	SDK 开始播放远端用户的首帧音频
onSendFirstLocalVideoFrame:	自己本地的首个视频帧已被发布出去
onSendFirstLocalAudioFrame	自己本地的首个音频帧已被发布出去
onRemoteVideoStatusUpdated:streamType:streamStatus:reason:extrainfo:	远端视频状态变化的事件回调
onRemoteAudioStatusUpdated:streamStatus:reason:extrainfo:	远端音频状态变化的事件回调
onUserVideoSizeChanged:streamType:newWidth:newHeight:	用户视频大小发生改变回调
onNetworkQuality:remoteQuality:	网络质量的实时统计回调
onStatistics:	音视频技术指标的实时统计回调
onSpeedTestResult:	网速测试的结果回调
onConnectionLost	SDK 与云端的连接已经断开
onTryToReconnect	SDK 正在尝试重新连接到云端
onConnectionRecovery	SDK 与云端的连接已经恢复
onCameraDidReady	摄像头准备就绪
onMicDidReady	麦克风准备就绪
onAudioRouteChanged:fromRoute:	当前音频路由发生变化（仅适用于移动设备）
onUserVoiceVolume:totalVolume:	音量大小的反馈回调
onDevice:type:stateChanged:	本地设备的通断状态发生变化（仅适用于桌面系统）
onAudioDeviceCaptureVolumeChanged:muted:	当前麦克风的系统采集音量发生变化
onAudioDevicePlayoutVolumeChanged:muted:	当前系统的播放音量发生变化

onSystemAudioLoopbackError:	系统声音采集是否被成功开启的事件回调（仅适用于 Mac 系统）
onRecvCustomCmdMsgUserId:cmdID:seq:message:	收到自定义消息的事件回调
onMissCustomCmdMsgUserId:cmdID:errCode:missed:	自定义消息丢失的事件回调
onRecvSEIMsg:message:	收到 SEI 消息的回调
onStartPublishing:errMsg:	开始向腾讯云直播 CDN 上发布音视频流的事件回调
onStopPublishing:errMsg:	停止向腾讯云直播 CDN 上发布音视频流的事件回调
onStartPublishCDNStream:errMsg:	开始向非腾讯云 CDN 上发布音视频流的事件回调
onStopPublishCDNStream:errMsg:	停止向非腾讯云 CDN 上发布音视频流的事件回调
onSetMixTranscodingConfig:errMsg:	设置云端混流的排版布局和转码参数的事件回调
onStartPublishMediaStream:code:message:extraInfo:	开始发布媒体流的事件回调
onUpdatePublishMediaStream:code:message:extraInfo:	更新媒体流的事件回调
onStopPublishMediaStream:code:message:extraInfo:	停止媒体流的事件回调
onCdnStreamStateChanged:status:code:msg:extraInfo:	RTMP/RTMPS 推流状态发生改变回调
onScreenCaptureStarted	屏幕分享开启的事件回调
onScreenCapturePaused:	屏幕分享暂停的事件回调
onScreenCaptureResumed:	屏幕分享恢复的事件回调
onScreenCaptureStoped:	屏幕分享停止的事件回调
onLocalRecordBegin:storagePath:	本地录制任务已经开始的事件回调
onLocalRecording:storagePath:	本地录制任务正在进行中的进展事件回调

版权所有：腾讯云计算（北京）有限责任公司

onLocalRecordFragment:	本地录制分片的事件回调
onLocalRecordComplete:storagePath:	本地录制任务已经结束的事件回调
onUserEnter:	有主播加入当前房间（已废弃）
onUserExit:reason:	有主播离开当前房间（已废弃）
onAudioEffectFinished:code:	音效播放已结束（已废弃）

TRTCVideoRenderDelegate

函数列表	描述
onRenderVideoFrame:userId:streamType:	自定义视频渲染回调

TRTCVideoFrameDelegate

函数列表	描述
onProcessVideoFrame:dstFrame:	用于对接第三方美颜组件的视频处理回调
onGLContextDestory	SDK 内部 OpenGL 环境被销毁的通知

TRTCAudioFrameDelegate

函数列表	描述
onCapturedAudioFrame:	本地采集并经过音频模块前处理后的音频数据回调
onLocalProcessedAudioFrame:	本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调
onRemoteUserAudioFrame:userId:	混音前的每一路远程用户的音频数据
onMixedPlayAudioFrame:	将各路待播放音频混合之后并在最终提交系统播放之前的数据回调
onMixedAllAudioFrame:	SDK 所有音频混合后的音频数据（包括采集到的和待播放的）
onVoiceEarMonitorAudioFrame:	耳返的音频数据

TRTCLogDelegate

函数列表	描述
onLog:LogLevel:WhichModule:	本地 LOG 的打印回调

onError:errMsg:extInfo:

onError:errMsg:extInfo:

```
- (void)onError:    (TXLiteAVError)errCode
      errMsg:      (nullable NSString *)errMsg
      extInfo:     (nullable NSDictionary*)extInfo
```

错误事件回调

错误事件，表示 SDK 抛出的不可恢复的错误，比如进入房间失败或设备开启失败等。
参考文档：错误码表

参数	描述
errCode	错误码
errMsg	错误信息
extInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题

onWarning:warningMsg:extInfo:

onWarning:warningMsg:extInfo:

```
- (void)onWarning:    (TXLiteAVWarning)warningCode
      warningMsg:     (nullable NSString *)warningMsg
      extInfo:        (nullable NSDictionary*)extInfo
```

警告事件回调

警告事件，表示 SDK 抛出的提示性问题，比如视频出现卡顿或 CPU 使用率太高等。
参考文档：错误码表

参数	描述
extInfo	扩展信息字段，个别警告码可能会带额外的信息帮助定位问题
warningCode	警告码
warningMsg	警告信息

onEnterRoom:

onEnterRoom:

- (void)onEnterRoom: (NSInteger)result

进入房间成功与否的事件回调

调用 TRTCCloud 中的 enterRoom() 接口执行进房操作后，会收到来自 TRTCCloudDelegate 的 onEnterRoom(result) 回调：

- 如果加入成功，回调 result 会是一个正数（result > 0），代表进入房间所消耗的时间，单位是毫秒（ms）。
- 如果加入失败，回调 result 会是一个负数（result < 0），代表失败原因的错误码。

进房失败的错误码含义请参见错误码表 。

参数	描述
result	result > 0 时为进房耗时（ms），result < 0 时为进房错误码。

注意
1. 在 Ver6.6 之前的版本，只有进房成功会抛出 onEnterRoom(result) 回调，进房失败由 onError() 回调抛出。
2. 在 Ver6.6 之后的版本：无论进房成功或失败，均会抛出 onEnterRoom(result) 回调，同时进房失败也会有 onError() 回调抛出。

onExitRoom:

onExitRoom:

```
- (void)onExitRoom:    (NSInteger)reason
```

离开房间的事件回调

调用 TRTCCloud 中的 exitRoom 接口会执行退出房间的相关逻辑，例如释放音视频设备资源和编解码器资源等。
待 SDK 占用的所有资源释放完毕后，SDK 会抛出 onExitRoom() 回调通知到您。

如果您要再次调用 enterRoom 或者切换到其他的音视频 SDK，请等待 onExitRoom 回调到来后再执行相关操作。否则可能会遇到例如摄像头、麦克风设备被强占等各种异常问题。

参数	描述
reason	离开房间原因，0：主动调用 exitRoom 退出房间；1：被服务器踢出当前房间；2：当前房间整个被解散。

onSwitchRole:errMsg:

onSwitchRole:errMsg:

```
- (void)onSwitchRole:    (TXLiteAVError)errCode
                errMsg:    (nullable NSString *)errMsg
```

切换角色的事件回调

调用 TRTCCloud 中的 switchRole() 接口会切换主播和观众的角色，该操作会伴随一个线路切换的过程，待 SDK 切换完成后，会抛出 onSwitchRole() 事件回调。

参数	描述
errCode	错误码，ERR_NULL 代表切换成功，其他请参见 错误码 。
errMsg	错误信息。

onSwitchRoom:errMsg:

onSwitchRoom:errMsg:

```
- (void)onSwitchRoom:    (TXLiteAVError)errCode
             errMsg:     (nullable NSString *)errMsg
```

切换房间的结果回调

调用 TRTCCloud 中的 switchRoom() 接口可以让用户快速地从一个房间切换到另一个房间，待 SDK 切换完成后，会抛出 onSwitchRoom() 事件回调。

参数	描述
errCode	错误码，ERR_NULL 代表切换成功，其他请参见 错误码 。
errMsg	错误信息。

onConnectOtherRoom:errCode:errMsg:

onConnectOtherRoom:errCode:errMsg:

```
- (void)onConnectOtherRoom:    (NSString*)userId
                   errCode:    (TXLiteAVError)errCode
                   errMsg:     (nullable NSString *)errMsg
```

请求跨房通话的结果回调

调用 TRTCCloud 中的 connectOtherRoom() 接口会将两个不同房间中的主播拉通视频通话，也就是所谓的“主播PK”功能。
调用者会收到 onConnectOtherRoom() 回调来获知跨房通话是否成功，
如果成功，两个房间中的所有用户都会收到来自另一个房间中的 PK 主播的 onUserVideoAvailable() 回调。

参数	描述

errCode	错误码，ERR_NULL 代表切换成功，其他请参见 错误码 。
errMsg	错误信息。
userId	要跨房通话的另一个房间中的主播的用户 ID。

onDisconnectOtherRoom:errMsg:

onDisconnectOtherRoom:errMsg:

```
- (void)onDisconnectOtherRoom:    (TXLiteAVError)errCode
                       errMsg:    (nullable NSString *)errMsg
```

结束跨房通话的结果回调

onUpdateOtherRoomForwardMode:errMsg:

onUpdateOtherRoomForwardMode:errMsg:

```
- (void)onUpdateOtherRoomForwardMode:    (TXLiteAVError)errCode
                              errMsg:    (nullable NSString *)errMsg
```

更改跨房主播上行能力的结果回调

onRemoteUserEnterRoom:

onRemoteUserEnterRoom:

```
- (void)onRemoteUserEnterRoom:    (NSString *)userId
```

有用户加入当前房间

出于性能方面的考虑，在 TRTC 两种不同的应用场景（即 AppScene，在 enterRoom 时通过第二个参数指定）下，该通知的行为会有差别：

- 直播类场景（TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom）：该场景下的用户区分主播和观众两种角色，只有主播进入房间时才会触发该通知，观众进入房间时不会触发该通知。
- 通话类场景（TRTCAppSceneVideoCall 和 TRTCAppSceneAudioCall）：该场景下的用户没有角色的区分（可认为都是主播），任何用户进入房间都会触发该通知。

参数	描述
userId	远端用户的用户标识。

注意

1. 事件回调 onRemoteUserEnterRoom 和 onRemoteUserLeaveRoom 只适用于维护当前房间里的“用户列表”，有此事件回调不代表一定有视频画面。
2. 如果需要显示远程画面，请监听代表某个用户是否有视频画面的 onUserVideoAvailable() 事件回调。

onRemoteUserLeaveRoom:reason:

onRemoteUserLeaveRoom:reason:

```
- (void)onRemoteUserLeaveRoom:    (NSString *)userId
                       reason:    (NSInteger)reason
```

有用户离开当前房间

与 onRemoteUserEnterRoom 相对应，在两种不同的应用场景（即 AppScene，在 enterRoom 时通过第二个参数指定）下，该通知的行为会有差别：

- 直播类场景（TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom）：只有主播离开房间时才会触发该通知，观众离开房间不会触发该通知。
- 通话类场景（TRTCAppSceneVideoCall 和 TRTCAppSceneAudioCall）：该场景下用户没有角色的区别，任何用户的离开都会触发该通知。

参数	描述
reason	离开原因，0表示用户主动退出房间，1表示用户超时退出，2表示被踢出房间，3表示主播因切换到观众退出房间。
userId	远端用户的用户标识。

onUserVideoAvailable:available:

onUserVideoAvailable:available:

```
- (void)onUserVideoAvailable:    (NSString *)userId
                   available:    (BOOL)available
```

某远端用户发布/取消了主路视频画面

主路画面 一般被用于承载摄像头画面。当您收到 onUserVideoAvailable(userId, YES) 通知时，表示该路画面已经有可播放的视频帧到达。
此时，您需要调用 startRemoteView 接口订阅该用户的远程画面，订阅成功后，您会继续收到该用户的首帧画面渲染回调 onFirstVideoFrame(userid)。

当您收到 onUserVideoAvailable(userId, NO) 通知时，表示该路远程画面已经被关闭，关闭的原因可能是该用户调用了 muteLocalVideo 或 stopLocalPreview 。

参数	描述
available	该用户是否发布（或取消发布）了主路视频画面，YES：发布；NO：取消发布。
userId	远端用户的用户标识。

onUserSubStreamAvailable:available:

onUserSubStreamAvailable:available:

```
- (void)onUserSubStreamAvailable:    (NSString *)userId
                          available:    (BOOL)available
```

某远端用户发布/取消了辅路视频画面

“辅路画面”一般被用于承载屏幕分享的画面。当您收到 onUserSubStreamAvailable(userId, YES) 通知时，表示该路画面已经有可播放的视频帧到达。
此时，您需要调用 startRemoteView 接口订阅该用户的远程画面，订阅成功后，您会继续收到该用户的首帧画面渲染回调 onFirstVideoFrame(userid)。

参数	描述
available	该用户是否发布（或取消发布）了辅路视频画面，YES: 发布；NO：取消发布。
userId	远端用户的用户标识。

注意
显示辅路画面使用的函数是 startRemoteView 而非 startRemoteSubStreamView，startRemoteSubStreamView 已经被废弃。

onUserAudioAvailable:available:

onUserAudioAvailable:available:

```
- (void)onUserAudioAvailable:    (NSString *)userId
                   available:    (BOOL)available
```

某远端用户发布/取消了自己的音频

当您收到 onUserAudioAvailable(userId, YES) 通知时，表示该用户发布了自己的声音，此时 SDK 的表现为：

- 在自动订阅模式下，您无需做任何操作，SDK 会自动播放该用户的声音。
- 在手动订阅模式下，您可以通过 muteRemoteAudio(userid, NO) 来播放该用户的声音。

参数	描述
available	该用户是否发布（或取消发布）了自己的音频，YES: 发布；NO：取消发布。
userId	远端用户的用户标识。

注意
SDK 默认使用自动订阅模式，您可以通过 setDefaultStreamRecvMode 设置为手动订阅，但需要在您进入房间之前调用才生效。

onFirstVideoFrame:streamType:width:height:

onFirstVideoFrame:streamType:width:height:

```
- (void)onFirstVideoFrame:    (NSString*)userId
                streamType:    (TRTCVideoStreamType)streamType
                     width:    (int)width
                    height:    (int)height
```

SDK 开始渲染自己本地或远端用户的首帧画面

SDK 会在渲染自己本地或远端用户的首帧画面时抛出该事件，您可以通过回调事件中的 userId 参数来判断事件来自于“本地”还是来自于“远端”。

- 如果 userId 为空值，代表 SDK 已经开始渲染自己本地的视频画面，不过前提是您已经调用了 startLocalPreview 或 startScreenCapture。
- 如果 userId 不为空，代表 SDK 已经开始渲染远端用户的视频画面，不过前提是您已经调用了 startRemoteView 订阅了该用户的视频画面。

参数	描述
height	画面的高度。
streamType	视频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。
userId	本地或远端的用户标识，如果 userId 为空值代表自己本地的首帧画面已到来，userId 不为空则代表远端用户的首帧画面已到来。
width	画面的宽度。

注意

1. 只有当您调用了 startLocalPreview 或 startScreenCapture 之后，才会触发自己本地的首帧画面事件回调。
2. 只有当您调用了 startRemoteView 或 startRemoteSubStreamView 之后，才会触发远端用户的首帧画面事件回调。

onFirstAudioFrame:

onFirstAudioFrame:

```
- (void)onFirstAudioFrame:      (NSString*)userId
```

SDK 开始播放远端用户的首帧音频

SDK 会在播放远端用户的首帧音频时抛出该事件，本地音频的首帧事件暂不抛出。

参数	描述
userId	远端用户的用户标识。

onSendFirstLocalVideoFrame:

onSendFirstLocalVideoFrame:

```
- (void)onSendFirstLocalVideoFrame:      (TRTCVideoStreamType)streamType
```

自己本地的首个视频帧已被发布出去

当您成功进入房间并通过 startLocalPreview 或 startScreenCapture 开启本地视频采集之后（开启采集和进入房间的先后顺序无影响），SDK 就会开始进行视频编码，并通过自身的网络模块向云端发布自己本地的视频数据。当 SDK 成功地向云端送出自己的第一帧视频数据帧以后，就会抛出 onSendFirstLocalVideoFrame 事件回调。

参数	描述
streamType	视频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。

onSendFirstLocalAudioFrame

onSendFirstLocalAudioFrame

自己本地的首个音频帧已被发布出去

当您成功进入房间并通过 startLocalAudio 开启本地音频采集之后（开启采集和进入房间的先后顺序无影响），SDK 就会开始进行音频编码，并通过自身的网络模块向云端发布自己本地的音频数据。当 SDK 成功地向云端送出自己的第一帧音频数据帧以后，就会抛出 onSendFirstLocalAudioFrame 事件回调。

onRemoteVideoStatusUpdated:streamType:streamStatus:reason:extrainfo:

onRemoteVideoStatusUpdated:streamType:streamStatus:reason:extrainfo:

```
- (void)onRemoteVideoStatusUpdated:    (NSString *)userId
                         streamType:   (TRTCVideoStreamType)streamType
                         streamStatus: (TRTCAVStatusType)status
                         reason:       (TRTCAVStatusChangeReason)reason
                         extrainfo:    (nullable NSDictionary *)extrainfo
```

远端视频状态变化的事件回调

您可以通过此事件回调获取远端每一路画面的播放状态（包括 Playing、Loading 和 Stopped 三种状态），从而进行相应的 UI 展示。

参数	描述
extrainfo	额外信息。
reason	视频状态改变的原因。
status	视频状态：包括 Playing、Loading 和 Stopped 三种状态。
streamType	视频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。
userId	用户标识。

onRemoteAudioStatusUpdated:streamStatus:reason:extrainfo:

onRemoteAudioStatusUpdated:streamStatus:reason:extrainfo:

```
- (void)onRemoteAudioStatusUpdated:    (NSString *)userId
                         streamStatus: (TRTCAVStatusType)status
                         reason:       (TRTCAVStatusChangeReason)reason
```

```
extrainfo:    (nullable NSDictionary *)extrainfo
```

远端音频状态变化的事件回调

您可以通过此事件回调获取远端每一路音频的播放状态（包括 Playing、Loading 和 Stopped 三种状态），从而进行相应的 UI 展示。

参数	描述
extrainfo	额外信息。
reason	音频状态改变的原因。
status	音频状态：包括 Playing、Loading 和 Stopped 三种状态。
userId	用户标识。

onUserVideoSizeChanged:streamType:newWidth:newHeight:

onUserVideoSizeChanged:streamType:newWidth:newHeight:

```
- (void)onUserVideoSizeChanged:    (NSString *)userId
                     streamType:    (TRTCVideoStreamType)streamType
                       newWidth:    (int)newWidth
                      newHeight:    (int)newHeight
```

用户视频大小发生改变回调

当您收到 onUserVideoSizeChanged(userId, streamtype, newWidth, newHeight) 通知时，表示该路画面大小发生了调整，调整的原因可能是该用户调用了 setVideoEncoderParam 或者 setSubStreamEncoderParam 重新设置了画面尺寸。

参数	描述
newHeight	视频流的高度（像素）。
newWidth	视频流的宽度（像素）。
streamTyp	视频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏

e	幕分享画面。
userId	用户标识。

onNetworkQuality:remoteQuality:

onNetworkQuality:remoteQuality:

```
- (void)onNetworkQuality:    (TRTCQualityInfo *)localQuality
                remoteQuality:    (NSArray<TRTCQualityInfo *>*)remoteQu
```

网络质量的实时统计回调

该统计回调每间隔2秒抛出一次，用于通知 SDK 感知到的当前网络的上行和下行质量。
SDK 会使用一组内嵌的自研算法对当前网络的延迟高低、带宽大小以及稳定情况进行评估，并计算出一个评估结果：如果评估结果为 1（Excellent）代表当前的网络情况非常好，如果评估结果为 6（Down）代表当前网络无法支撑 TRTC 的正常通话。

参数	描述
localQuality	上行网络质量。
remoteQuality	下行网络质量，代表数据流历经“远端->云端->本端”这样一条完整的传输链路后，最终在本端统计到的数据质量。因此，这里的下行网络质量代表的是远端上行链路与本端下行链路共同影响的结果。

注意
暂时无法通过该接口单独判定远端用户的上行质量。

onStatistics:

onStatistics:

```
- (void)onStatistics:    (TRTCStatistics *)statistics
```

音视频技术指标的实时统计回调

该统计回调每间隔2秒抛出一次，用于通知 SDK 内部音频、视频以及网络相关的专业技术指标，这些信息在 TRTCStatistics 均有罗列：

- 视频统计信息：视频的分辨率（resolution）、帧率（FPS）和比特率（bitrate）等信息。
- 音频统计信息：音频的采样率（samplerate）、声道（channel）和比特率（bitrate）等信息。
- 网络统计信息：SDK 和云端一次往返（SDK > Cloud > SDK）的网络耗时（rtt）、丢包率（loss）、上行流量（sentBytes）和下行流量（receivedBytes）等信息。

参数	描述
statistics	统计数据，包括自己本地的统计信息和远端用户的统计信息，详情请参考 TRTCStatistics 。

注意
如果您只需要获知当前网络质量的好坏，并不需要花太多时间研究本统计回调，更推荐您使用 onNetworkQuality 来解决问题。

onSpeedTestResult:

onSpeedTestResult:

```
- (void)onSpeedTestResult:    (TRTCSpeedTestResult *)result
```

网速测试的结果回调

该统计回调由 startSpeedTest: 触发。

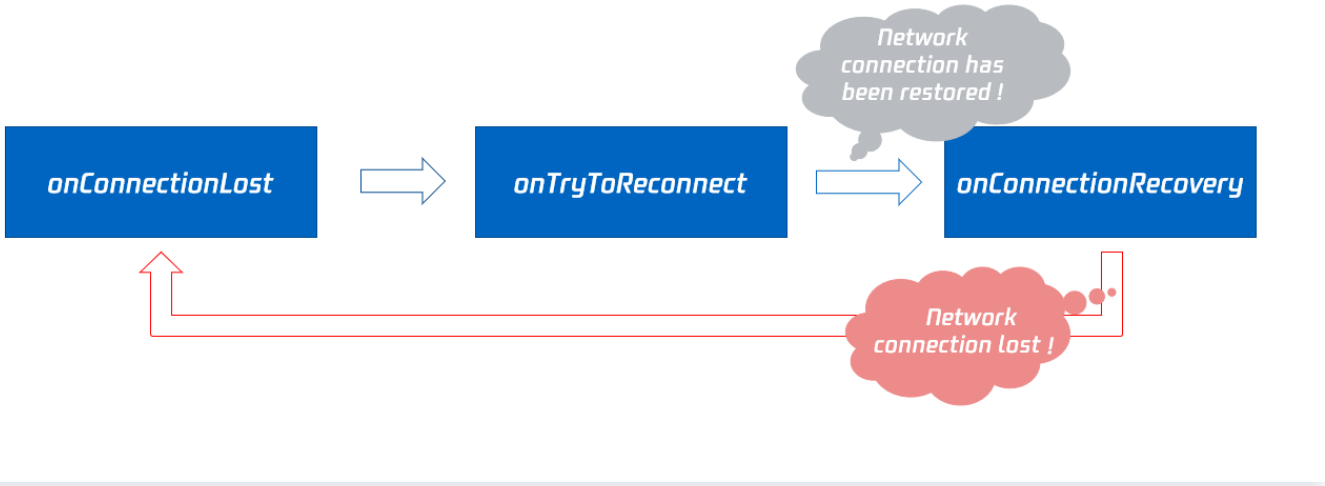
参数	描述
result	网速测试结果，包括丢包、往返延迟、上下行的带宽速率，详情请参见 TRTCSpeedTestResult 。

onConnectionLost

onConnectionLost

SDK 与云端的连接已经断开

SDK 会在跟云端的连接断开时抛出此事件回调，导致断开的原因大多是网络不可用或者网络切换所致，比如用户在通话中走进电梯时就可能会遇到此事件。
在抛出此事件之后，SDK 会努力跟云端重新建立连接，重连过程中会抛出 onTryToReconnect，连接恢复后会抛出 onConnectionRecovery 。
所以，SDK 会在如下三个连接相关的事件中按如下规律切换：



onTryToReconnect

onTryToReconnect

SDK 正在尝试重新连接到云端

SDK 会在跟云端的连接断开时抛出 onConnectionLost，之后会努力跟云端重新建立连接并抛出本事件，连接恢复后会抛出 onConnectionRecovery 。

onConnectionRecovery

onConnectionRecovery

SDK 与云端的连接已经恢复

SDK 会在跟云端的连接断开时抛出 onConnectionLost，之后会努力跟云端重新建立连接并抛出 onTryToReconnect，连接恢复后会抛出本事件回调。

onCameraDidReady

onCameraDidReady

摄像头准备就绪

当您调用 startLocalPreivew 之后，SDK 会尝试启动摄像头，如果摄像头能够启动成功就会抛出本事件。
如果启动失败，大概率是因为当前应用没有获得访问摄像头的权限，或者摄像头当前正在被其他程序独占使用中。
您可以通过捕获 onError 事件回调获知这些异常情况并通过 UI 界面提示用户。

onMicDidReady

onMicDidReady

麦克风准备就绪

当您调用 startLocalAudio 之后，SDK 会尝试启动麦克风，如果麦克风能够启动成功就会抛出本事件。
如果启动失败，大概率是因为当前应用没有获得访问麦克风的权限，或者麦克风当前正在被其他程序独占使用中。
您可以通过捕获 onError 事件回调获知这些异常情况并通过 UI 界面提示用户。

onAudioRouteChanged:fromRoute:

onAudioRouteChanged:fromRoute:

```
- (void)onAudioRouteChanged:    (TRTCAudioRoute)route
                  fromRoute:    (TRTCAudioRoute)fromRoute
```

当前音频路由发生变化（仅适用于移动设备）

所谓“音频路由”，是指声音是从手机的扬声器还是从听筒中播放出来，音频路由变化也就是声音的播放位置发生了变化。

- 当音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
- 当音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。
- 当音频路由为有线耳机。
- 当音频路由为蓝牙耳机。
- 当音频路由为USB专业声卡设备。

参数	描述
fromRoute	变更前的音频路由。
route	音频路由，即声音由哪里输出（扬声器、听筒）。

onUserVoiceVolume:totalVolume:

onUserVoiceVolume:totalVolume:

```
- (void)onUserVoiceVolume:    (NSArray<TRTCVolumeInfo *> *)userVolu
                totalVolume:  (NSInteger)totalVolume
```

音量大小的反馈回调

SDK 可以评估每一路音频的音量大小，并每隔一段时间抛出该事件回调，您可以根据音量大小在 UI 上做出相应的提示，比如 波形图 或 音量槽 。
要完成这个功能， 您需要先调用 enableAudioVolumeEvaluation 开启这个能力并设定事件抛出的时间间隔。
需要补充说明的是，无论当前房间中是否有人说话，SDK 都会按照您设定的时间间隔定时抛出此事件回调。

参数	描述
totalVolume	所有远端用户的总音量大小，取值范围 0 – 100。
userVolumes	是一个数组，用于承载所有正在说话的用户的音量大小，取值范围 0 – 100。

注意
userVolumes 为一个数组，对于数组中的每一个元素，当 userId 为空时表示本地麦克风采集的音量大小，当 userId 不为空时代表远端用户的音量大小。

onDevice:type:stateChanged:

onDevice:type:stateChanged:

```
- (void)onDevice:        (NSString *)deviceId
            type:        (TRTCMediaDeviceType)deviceType
    stateChanged:        (NSInteger)state
```

本地设备的通断状态发生变化（仅适用于桌面系统）

当本地设备（包括摄像头、麦克风以及扬声器）被插入或者拔出时，SDK 便会抛出此事件回调。

参数	描述
deviceId	设备 ID。
deviceType	设备类型。
state	通断状态，0：设备断开；1：设备连接。

onAudioDeviceCaptureVolumeChanged:muted:

onAudioDeviceCaptureVolumeChanged:muted:

```
- (void)onAudioDeviceCaptureVolumeChanged:    (NSInteger)volume
                                    muted:    (BOOL)muted
```

当前麦克风的系统采集音量发生变化

在 Mac 或 Windows 这样的桌面操作系统上，用户可以在设置中心找到声音相关的设置面板，并设置麦克风的采集音量大小。
用户将麦克风的采集音量设置得越大，麦克风采集到的声音的原始音量也就会越大，反之就会越小。
在有些型号的键盘以及笔记本电脑上，用户还可以通过按下 `禁用麦克风` 按钮（图标是一个 `话筒` 叠加了一道代表禁用的斜线）来将麦克风静音。

当用户通过系统设置界面或者通过键盘上的快捷键设定操作系统的麦克风采集音量时，SDK 便会抛出此事件。

参数	描述
muted	麦克风是否被用户禁用了：YES 被禁用，NO 被启用。

volume	系统采集音量，取值范围 0 – 100，用户可以在系统的声音设置面板上进行拖拽调整。

注意
您需要调用 enableAudioVolumeEvaluation 接口并设定 interval 大于 0 开启次事件回调，设定 interval 等于 0 关闭此事件回调。

onAudioDevicePlayoutVolumeChanged:muted:

onAudioDevicePlayoutVolumeChanged:muted:

```
- (void)onAudioDevicePlayoutVolumeChanged:    (NSInteger)volume
                                    muted:    (BOOL)muted
```

当前系统的播放音量发生变化

在 Mac 或 Windows 这样的桌面操作系统上，用户可以在设置中心找到声音相关的设置面板，并设置系统的播放音量大小。
在有些型号的键盘以及笔记本电脑上，用户还可以通过按下“静音”按钮（图标是一个喇叭上叠加了一道代表禁用的斜线）来将系统静音。

当用户通过系统设置界面或者通过键盘上的快捷键设定操作系统的播放音量时，SDK 便会抛出此事件。

参数	描述
muted	系统是否被用户静音了：YES 被静音，NO 已恢复。
volume	系统播放音量，取值范围 0 – 100，用户可以在系统的声音设置面板上进行拖拽调整。

注意
您需要调用 enableAudioVolumeEvaluation 接口并设定 interval 大于 0 开启次事件回调，设定 interval 等于 0 关闭此事件回调。

onSystemAudioLoopbackError:

onSystemAudioLoopbackError:

```
- (void)onSystemAudioLoopbackError:    (TXLiteAVError)err
```

系统声音采集是否被成功开启的事件回调（仅适用于 Mac 系统）

在 Mac 系统上，您可以通过调用 startSystemAudioLoopback 为当前系统安装一个音频驱动，并让 SDK 通过该音频驱动捕获当前 Mac 系统播放出的声音。
当用于播片教学或音乐直播中，比如老师端可以使用此功能，让 SDK 能够采集老师所播放的电影中的声音，使同房间的学生端也能听到电影中的声音。
SDK 会将系统声音采集是否被成功开启的结果，通过本事件回调抛出，需要您关注参数中的错误码。

参数	描述
err	ERR_NULL 表示成功，其余值表示失败。

onRecvCustomCmdMsgUserId:cmdID:seq:message:

onRecvCustomCmdMsgUserId:cmdID:seq:message:

```
- (void)onRecvCustomCmdMsgUserId:    (NSString *)userId
                           cmdID:    (NSInteger)cmdID
                             seq:    (UInt32)seq
                         message:    (NSData *)message
```

收到自定义消息的事件回调

当房间中的某个用户使用 sendCustomCmdMsg 发送自定义 UDP 消息时，房间中的其他用户可以通过 onRecvCustomCmdMsg 事件回调接收到该条消息。

参数	描述
cmdID	命令 ID。
message	消息数据。
seq	消息序号。

userId	用户标识。

onMissCustomCmdMsgUserId:cmdID:errCode:missed:

onMissCustomCmdMsgUserId:cmdID:errCode:missed:

```
- (void)onMissCustomCmdMsgUserId:    (NSString *)userId
                           cmdID:    (NSInteger)cmdID
                         errCode:    (NSInteger)errCode
                          missed:    (NSInteger)missed
```

自定义消息丢失的事件回调

当您使用 sendCustomCmdMsg 发送自定义 UDP 消息时，即使设置了可靠传输（reliable），也无法确 100% 不丢失，只是丢消息概率极低，能满足常规可靠性要求。
在发送端设置了可靠运输（reliable）后，SDK 都会通过此回调通知过去时间段内（通常为5s）传输途中丢失的自定义消息数量统计信息。

参数	描述
cmdID	命令 ID。
errCode	错误码。
missed	丢失的消息数量。
userId	用户标识。

注意
只有在发送端设置了可靠传输（reliable），接收方才能收到消息的丢失回调。

onRecvSEIMsg:message:

onRecvSEIMsg:message:

```
- (void)onRecvSEIMsg:    (NSString *)userId
```

```
message:    (NSData*)message
```

收到 SEI 消息的回调

当房间中的某个用户使用 sendSEIMsg 借助视频数据帧发送 SEI 消息时，房间中的其他用户可以通过 onRecvSEIMsg 事件回调接收到该条消息。

参数	描述
message	数据。
userId	用户标识。

onStartPublishing:errMsg:

onStartPublishing:errMsg:

```
- (void)onStartPublishing:    (int)err
                   errMsg:    (NSString*)errMsg
```

开始向腾讯云直播 CDN 上发布音视频流的事件回调

当您调用 startPublishing 开始向腾讯云直播 CDN 上发布音视频流时，SDK 会立刻将这一指令同步给云端服务器。
随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	0表示成功，其余值表示失败。
errMsg	具体错误原因。

onStopPublishing:errMsg:

onStopPublishing:errMsg:

```
- (void)onStopPublishing:    (int)err
                  errMsg:    (NSString*)errMsg
```

停止向腾讯云直播 CDN 上发布音视频流的事件回调

当您调用 stopPublishing 停止向腾讯云直播 CDN 上发布音视频流时，SDK 会立刻将这一指令同步给云端服务器。
随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	0表示成功，其余值表示失败。
errMsg	具体错误原因。

onStartPublishCDNStream:errMsg:

onStartPublishCDNStream:errMsg:

```
- (void)onStartPublishCDNStream:    (int)err
                         errMsg:    (NSString *)errMsg
```

开始向非腾讯云 CDN 上发布音视频流的事件回调

当您调用 startPublishCDNStream 开始向非腾讯云直播 CDN 上发布音视频流时，SDK 会立刻将这一指令同步给云端服务器。随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	0表示成功，其余值表示失败。
errMsg	具体错误原因。

注意
当您收到成功的事件回调时，只是说明您的发布指令已经同步到腾讯云后台服务器，但如果目标 CDN 厂商的服务器不接收该条视频流，依然可能导致发布失败。

onStopPublishCDNStream:errMsg:

onStopPublishCDNStream:errMsg:

```
- (void)onStopPublishCDNStream:    (int)err
                        errMsg:    (NSString *)errMsg
```

停止向非腾讯云 CDN 上发布音视频流的事件回调

当您调用 stopPublishCDNStream 开始向非腾讯云直播 CDN 上发布音视频流时，SDK 会立刻将这一指令同步给云端服务器。随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	0表示成功，其余值表示失败。
errMsg	具体错误原因。

onSetMixTranscodingConfig:errMsg:

onSetMixTranscodingConfig:errMsg:

```
- (void)onSetMixTranscodingConfig:    (int)err
                           errMsg:    (NSString*)errMsg
```

设置云端混流的排版布局和转码参数的事件回调

当您调用 setMixTranscodingConfig 调整云端混流的排版布局和转码参数时，SDK 会立刻将这一调整指令同步给云端服务器。随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	错误码：0表示成功，其余值表示失败。

errMsg	具体的错误原因。

onStartPublishMediaStream:code:message:extraInfo:

onStartPublishMediaStream:code:message:extraInfo:

```
- (void)onStartPublishMediaStream:    (NSString*)taskId
                             code:    (int)code
                          message:    (NSString*)message
                        extraInfo:    (nullable NSDictionary *)extraInfo
```

开始发布媒体流的事件回调

当您调用 startPublishMediaStream 开始向 TRTC 后台服务发布媒体流时，SDK 会立刻将这一指令同步给云端服务器，随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
code	回调结果，0 表示成功，其余值表示失败。
extraInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。
message	具体回调信息。
taskId	当请求成功时，TRTC 后台会在回调中提供给您这项任务的 taskId，后续您可以通过该 taskId 结合 updatePublishMediaStream 和 stopPublishMediaStream 进行更新和停止。

onUpdatePublishMediaStream:code:message:extraInfo:

onUpdatePublishMediaStream:code:message:extraInfo:

```
- (void)onUpdatePublishMediaStream:    (NSString*)taskId
                              code:    (int)code
```

```
                                    message:    (NSString*)message
                                    extraInfo:  (nullable NSDictionary *)extraInfo
```

更新媒体流的事件回调

当您调用媒体流发布接口（updatePublishMediaStream）开始向 TRTC 后台服务更新媒体流时，SDK 会立刻将这一指令同步给云端服务器，随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
code	回调结果，0 表示成功，其余值表示失败。
extraInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。
message	具体回调信息。
taskId	您调用媒体流发布接口（updatePublishMediaStream）时传入的 taskId，会通过此回调再带回给您，用于标识该回调属于哪一次更新请求。

onStopPublishMediaStream:code:message:extraInfo:

onStopPublishMediaStream:code:message:extraInfo:

```
- (void)onStopPublishMediaStream:    (NSString*)taskId
                             code:   (int)code
                          message:   (NSString*)message
                          extraInfo: (nullable NSDictionary *)extraInfo
```

停止媒体流的事件回调

当您调用停止发布媒体流（stopPublishMediaStream）开始向 TRTC 后台服务停止媒体流时，SDK 会立刻将这一指令同步给云端服务器，随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述

code	回调结果，0 表示成功，其余值表示失败。
extraInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。
message	具体回调信息。
taskId	您调用停止发布媒体流（stopPublishMediaStream）时传入的 taskId，会通过此回调再带回给您，用于标识该回调属于哪一次停止请求。

onCdnStreamStateChanged:status:code:msg:extraInfo:

onCdnStreamStateChanged:status:code:msg:extraInfo:

```
- (void)onCdnStreamStateChanged:    (NSString*)cdnUrl
                         status:    (int)status
                           code:    (int)code
                            msg:    (NSString*)msg
                      extraInfo:    (nullable NSDictionary *)info
```

RTMP/RTMPS 推流状态发生改变回调

当您调用 startPublishMediaStream 开始向 TRTC 后台服务发布媒体流时，SDK 会立刻将这一指令同步给云端服务。
若您在目标推流配置（TRTCPublishTarget）设置了向腾讯或者第三方 CDN 上发布音视频流的 URL 配置，则具体 RTMP 或者 RTMPS 推流状态将通过此回调同步给您。

参数	描述
cdnUrl	您调用 startPublishMediaStream 时通过目标推流配置（TRTCPublishTarget）传入的 url，在推流状态变更时，会通过此回调同步给您。
code	推流结果，0 表示成功，其余值表示出错。
extraInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。
message	具体推流信息。
status	推流状态。

- 0：推流未开始或者已结束。在您调用 stopPublishMediaStream 时会返回该状态。
- 1：正在连接 TRTC 服务器和 CDN 服务器。若无法立刻成功，TRTC 后台服务会多次重试尝试推流，并返回该状态（5s回调一次）。如成功进行推流，则进入状态 2；如服务器出错或 60 秒内未成功推流，则进入状态 4。
- 2：CDN 推流正在进行。在成功推流后，会返回该状态。
- 3：TRTC 服务器和 CDN 服务器推流中断，正在恢复。当 CDN 出现异常，或推流短暂中断时，TRTC 后台服务会自动尝试恢复推流，并返回该状态（5s回调一次）。如成功恢复推流，则进入状态 2；如服务器出错或 60 秒内未成功恢复，则进入状态 4。
- 4：TRTC 服务器和 CDN 服务器推流中断，且恢复或连接超时。即此时推流失败，你可以再次调用 updatePublishMediaStream 尝试推流。
- 5：正在断开 TRTC 服务器和 CDN 服务器。在您调用 stopPublishMediaStream 时，TRTC 后台服务会依次同步状态 5 和状态 0。

onScreenCaptureStarted

onScreenCaptureStarted

屏幕分享开启的事件回调

当您通过 startScreenCapture 等相关接口启动屏幕分享时，SDK 便会抛出此事件回调。

onScreenCapturePaused:

onScreenCapturePaused:

```
- (void)onScreenCapturePaused:    (int)reason
```

屏幕分享暂停的事件回调

当您通过 pauseScreenCapture 暂停屏幕分享时，SDK 便会抛出此事件回调。

参数	描述
reason	原因。 • 0：用户主动暂停。 • 1：注意此字段的含义在 MAC 和 Windows 平台有稍微差异。屏幕窗口不可见暂停（Mac）。表示设置屏幕分享参数导致的暂停（Windows）。

	• 2：表示屏幕分享窗口被最小化导致的暂停（仅 Windows）。 • 3：表示屏幕分享窗口被隐藏导致的暂停（仅 Windows）。

onScreenCaptureResumed:

onScreenCaptureResumed:

```
- (void)onScreenCaptureResumed:      (int)reason
```

屏幕分享恢复的事件回调

当您通过 resumeScreenCapture 恢复屏幕分享时，SDK 便会抛出此事件回调。

参数	描述
reason	恢复原因。 • 0：用户主动恢复。 • 1：注意此字段的含义在 MAC 和 Windows 平台有稍微差异。屏幕窗口恢复可见从而恢复分享（Mac）。屏幕分享参数设置完毕后自动恢复（Windows）。 • 2：表示屏幕分享窗口从最小化被恢复（仅 Windows）。 • 3：表示屏幕分享窗口从隐藏被恢复（仅 Windows）。

onScreenCaptureStoped:

onScreenCaptureStoped:

```
- (void)onScreenCaptureStoped:      (int)reason
```

屏幕分享停止的事件回调

当您通过 stopScreenCapture 停止屏幕分享时，SDK 便会抛出此事件回调。

参数	描述
reason	停止原因，0：用户主动停止；1：屏幕窗口关闭导致停止；2：表示屏幕分享的显示屏状态变更（如接口被拔出、投影模式变更等）。

onLocalRecordBegin:storagePath:

onLocalRecordBegin:storagePath:

```
- (void)onLocalRecordBegin:    (NSInteger)errCode
                storagePath:    (NSString *)storagePath
```

本地录制任务已经开始的事件回调

当您调用 startLocalRecording 启动本地媒体录制任务时，SDK 会抛出该事件回调，用于通知您录制任务是否已经顺利启动。

参数	描述
errCode	状态码。 • 0：录制任务启动成功。 • -1：内部错误导致录制任务启动失败。 • -2：文件后缀名有误（比如不支持的录制格式）。 • -6：录制已经启动，需要先停止录制。 • -7：录制文件已存在，需要先删除文件。 • -8：录制目录无写入权限，请检查目录权限问题。
storagePath	录制文件存储路径。

onLocalRecording:storagePath:

onLocalRecording:storagePath:

```
- (void)onLocalRecording:    (NSInteger)duration
             storagePath:    (NSString *)storagePath
```

本地录制任务正在进行中的进展事件回调

当您调用 startLocalRecording 成功启动本地媒体录制任务后，SDK 会定时地抛出本事件回调。
您可通过捕获该事件回调来获知录制任务的健康状况。
您可以在 startLocalRecording 时设定本事件回调的抛出间隔。

参数	描述
duration	已经录制的累计时长，单位毫秒。
storagePath	录制文件存储路径。

onLocalRecordFragment:

onLocalRecordFragment:

```
- (void)onLocalRecordFragment:    (NSString *)storagePath
```

本地录制分片的事件回调

当您开启分片录制时，每完成一个分片，都会通过此接口回调。

参数	描述
storagePath	分片文件存储路径。

onLocalRecordComplete:storagePath:

onLocalRecordComplete:storagePath:

```
- (void)onLocalRecordComplete:    (NSInteger)errCode
                  storagePath:    (NSString *)storagePath
```

本地录制任务已经结束的事件回调

当您调用 stopLocalRecording 停止本地媒体录制任务时，SDK 会抛出该事件回调，用于通知您录制任务的最终结果。

参数	描述
errCode	状态码。 • 0：结束录制任务成功。 • -1：录制失败。

	• -2：切换分辨率或横竖屏导致录制结束。 • -3：录制时间太短，或未采集到任何视频或音频数据，请检查录制时长，或是否已开启音、视频采集。
storagePath	录制文件存储路径。

onUserEnter:

onUserEnter:

```
- (void)onUserEnter:    (NSString *)userId
```

有主播加入当前房间（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 onRemoteUserEnterRoom 替代之。

onUserExit:reason:

onUserExit:reason:

```
- (void)onUserExit:    (NSString *)userId
           reason:     (NSInteger)reason
```

有主播离开当前房间（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 onRemoteUserLeaveRoom 替代之。

onAudioEffectFinished:code:

onAudioEffectFinished:code:

```
- (void)onAudioEffectFinished:    (int) effectId
                         code:    (int) code
```

音效播放已结束（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 ITXAudioEffectManager 接口替代之。
新的接口中不再区分背景音乐和音效，而是统一用 startPlayMusic 取代之。

onRenderVideoFrame:userId:streamType:

onRenderVideoFrame:userId:streamType:

```
- (void) onRenderVideoFrame:    (TRTCVideoFrame * _Nonnull)frame
                     userId:    (NSString* __nullable)userId
                 streamType:    (TRTCVideoStreamType)streamType
```

自定义视频渲染回调

当您设置了本地或者远端的视频自定义渲染回调之后，SDK 就会将原本要交给渲染控件进行渲染的视频帧通过此回调接口抛送给您，以便于您进行自定义渲染。

参数	描述
frame	待渲染的视频帧信息。
streamType	频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。
userId	视频源的 userId，如果是本地视频回调（setLocalVideoRenderDelegate），该参数可以忽略。

onProcessVideoFrame:dstFrame:

onProcessVideoFrame:dstFrame:

```
- (uint32_t)onProcessVideoFrame:    (TRTCVideoFrame * _Nonnull)srcFrame
                        dstFrame:    (TRTCVideoFrame * _Nonnull)dstFrame
```

用于对接第三方美颜组件的视频处理回调

如果您选购了第三方美颜组件，就需要在 TRTCCloud 中设置第三方美颜回调，之后 TRTC 就会将原本要进行预处理的视频帧通过此回调接口抛送给您。

之后您就可以将 TRTC 抛出的视频帧交给第三方美颜组件进行图像处理，由于抛出的数据是可读且可写的，因此第三方美颜的处理结果也可以同步给 TRTC 进行后续的编码和发送。

情况一：美颜组件自身会产生新的纹理。
如果您使用的美颜组件会在处理图像的过程中产生一帧全新的纹理（用于承载处理后的图像），那请您在回调函数中将 dstFrame.textureId 设置为新纹理的 ID：

```
uint32_t onProcessVideoFrame(TRTCVideoFrame * _Nonnull)srcFrame dstFrame:
(TRTCVideoFrame * _Nonnull)dstFrame{
    self.frameID += 1;
    dstFrame.pixelBuffer = [[FURenderer shareRenderer]
renderPixelBuffer:srcFrame.pixelBuffer
                                          withFrameId:self.frameID
                                                items:self.renderItems
                                            itemCount:self.renderItems.count];
    return 0;
}
```

情况二：美颜组件需要您提供目标纹理。
如果您使用的第三方美颜模块并不生成新的纹理，而是需要您设置给该模块一个输入纹理和一个输出纹理，则可以考虑如下方案：

```
uint32_t onProcessVideoFrame(TRTCVideoFrame * _Nonnull)srcFrame dstFrame:
(TRTCVideoFrame * _Nonnull)dstFrame{
    thirdparty_process(srcFrame.textureId, srcFrame.width, srcFrame.height,
dstFrame.textureId);
    return 0;
}
```

参数	描述
dstFrame	用于接收第三方美颜处理过的视频画面。
srcFrame	用于承载 TRTC 采集到的摄像头画面。

注意
目前仅支持 OpenGL 纹理方案（ PC 仅支持 TRTCVideoBufferType_Buffer 格式）。

onGLContextDestory

onGLContextDestory

SDK 内部 OpenGL 环境被销毁的通知

onCapturedAudioFrame:

onCapturedAudioFrame:

```
- (void) onCapturedAudioFrame:    (TRTCAudioFrame *)frame
```

本地采集并经过音频模块前处理后的音频数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把刚采集到并经过前处理(ANS、AEC、AGC）之后的数据，以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。
3. 此接口回调出的音频数据已经经过了前处理(ANS、AEC、AGC），但**不包含**背景音、音效、混响等前处理效果，延迟较低。

onLocalProcessedAudioFrame:

onLocalProcessedAudioFrame:

```
- (void) onLocalProcessedAudioFrame:    (TRTCAudioFrame *)frame
```

本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把刚采集到并经过前处理、音效处理和混 BGM 之后的数据，在最终进行网络编码之前，以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

特殊说明：
您可以通过设置接口中的 `TRTCAudioFrame.extraData` 字段，达到传输信令的目的。由于音频帧头部的数据块不能太大，建议您写入 `extraData` 时，尽量将信令控制在几个字节的大小，如果超过 100 个字节，写入的数据不会被发送。

房间内其他用户可以通过 TRTCAudioFrameDelegate 中的 `onRemoteUserAudioFrame` 中的 `TRTCAudioFrame.extraData` 字段回调接收数据。

参数	描述
frame	PCM 格式的音频数据帧。

注意
1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。
3. 此接口回调出的数据已经经过了前处理(ANS、AEC、AGC）、音效和混 BGM 处理，声音的延迟相比于 onCapturedAudioFrame 要高一些。

onRemoteUserAudioFrame:userId:

onRemoteUserAudioFrame:userId:

```
- (void) onRemoteUserAudioFrame:    (TRTCAudioFrame *)frame
                          userId:    (NSString *)userId
```

混音前的每一路远程用户的音频数据

当您设置完音频数据自定义回调之后，SDK 内部会把远端的每一路原始数据，在最终混音之前，以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。
userId	用户标识。

注意
此接口回调出的音频数据是只读的，不支持修改。

onMixedPlayAudioFrame:

onMixedPlayAudioFrame:

```
- (void) onMixedPlayAudioFrame:    (TRTCAudioFrame *)frame
```

将各路待播放音频混合之后并在最终提交系统播放之前的数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把各路待播放的音频混合之后的音频数据，在提交系统播放之前，以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。
3. 此接口回调出的是对各路待播放音频数据的混合，但其中并不包含耳返的音频数据。

onMixedAllAudioFrame:

onMixedAllAudioFrame:

```
- (void) onMixedAllAudioFrame:      (TRTCAudioFrame *)frame
```

SDK 所有音频混合后的音频数据（包括采集到的和待播放的）

当您设置完音频数据自定义回调之后，SDK 内部会把所有采集到的和待播放的音频数据混合起来，以 PCM 格式的形式通过本接口回调给您，便于您进行自定义录制。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 此接口回调出的是SDK所有音频数据的混合数据，包括：经过 3A 前处理、特效叠加以及背景音乐混音后的本地音频，所有远端音频，但不包括耳返音频。
2. 此接口回调出的音频数据不支持修改。

onVoiceEarMonitorAudioFrame:

onVoiceEarMonitorAudioFrame:

```
- (void) onVoiceEarMonitorAudioFrame:    (TRTCAudioFrame *)frame
```

耳返的音频数据

当您设置完音频数据自定义回调之后，SDK 内部会把耳返的音频数据在播放之前以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长不固定，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式 48000 采样率、单声道、16采样点位宽为例，0.02s 的音频数据字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，否则会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。

onLog:LogLevel:WhichModule:

onLog:LogLevel:WhichModule:

```
-(void) onLog:      (nullable NSString*)log
     LogLevel:      (TRTCLogLevel)level
  WhichModule:      (nullable NSString*)module
```

本地 LOG 的打印回调

如果您希望捕获 SDK 的本地日志打印行为，可以通过设置日志回调，让 SDK 将要打印的日志都通过本回调接口抛送给您。

参数	描述
level	日志等级请参见 TRTC_LOG_LEVEL。
log	日志内容。
module	保留字段，暂无具体意义，目前为固定值 TXLiteAVSDK。

TRTCStatistics

最近更新时间：2024-03-06 14:51:31

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTC 音视频统计指标（只读）

Function: TRTC SDK 会以两秒钟一次的频率向您汇报当前实时的音视频指标（帧率、码率、卡顿情况等）

TRTCStatistics

结构体类型

函数列表	描述
TRTCLocalStatistics	本地的音视频统计指标
TRTCRemoteStatistics	远端的音视频统计指标
TRTCStatistics	网络和性能的汇总统计指标

TRTCLocalStatistics

TRTCLocalStatistics

本地的音视频统计指标

枚举类型	描述
audioBitrate	【字段含义】本地音频的码率，即每秒钟新产生音频数据的多少，单位 Kbps。
audioCaptureState	【字段含义】音频设备采集状态（用于检测音频外设的健康度）。 0：采集设备状态正常；1：检测到长时间静音；2：检测到破音；3：检测到声音异常间断。

audioSampleRate	【字段含义】本地音频的采样率，单位 Hz。
frameRate	【字段含义】本地视频的帧率，即每秒钟会有多少视频帧，单位：FPS。
height	【字段含义】本地视频的高度，单位 px。
streamType	【字段含义】视频流类型（高清大画面|低清小画面|辅流画面）。
videoBitrate	【字段含义】本地视频的码率，即每秒钟新产生视频数据的多少，单位 Kbps。
width	【字段含义】本地视频的宽度，单位 px。

TRTCRemoteStatistics

TRTCRemoteStatistics

远端的音视频统计指标

枚举类型	描述
audioBitrate	【字段含义】本地音频的码率，单位 Kbps
audioBlockRate	【字段含义】音频播放卡顿率，单位 (%) 音频播放卡顿率（audioBlockRate）= 音频播放的累计卡顿时长（audioTotalBlockTime）/ 音频播放的总时长
audioPacketLoss	【字段含义】音频流的总丢包率（%）。 audioPacketLoss 代表音频流历经 `主播>云端>观众` 这样一条完整的传输链路后，最终在观众端统计到的丢包率。 audioPacketLoss 越小越好，丢包率为0即表示该路音频流的所有数据均已经完整地到达了观众端。 如果出现了 downLoss == 0 但 audioPacketLoss != 0 的情况，说明该路音频流在“云端=>观众”这一段链路上没有出现丢包，但是在 `主播>云端` 这一段链路上出现了不可恢复的丢包。
audioSampleRate	【字段含义】本地音频的采样率，单位 Hz。
audioTotalBlockTime	【字段含义】音频播放的累计卡顿时长，单位 ms
finalLoss	【字段含义】该路音视频流的总丢包率（%）。已废弃，不推荐使用；建议使用 audioPacketLoss、videoPacketLoss 替代。

frameRate	【字段含义】远端视频的帧率，单位：FPS。
height	【字段含义】远端视频的高度，单位 px。
jitterBufferDelay	【字段含义】播放延迟，单位 ms 为了避免网络抖动和网络包乱序导致的声音和画面卡顿，TRTC 会在播放端管理一个播放缓冲区，用于对接收到的网络数据包进行整理，该缓冲区的大小会根据当前的网络质量进行自适应调整，该缓冲区的大小折算成以毫秒为单位的时间长度，也就是 jitterBufferDelay。
point2PointDelay	【字段含义】端到端延迟，单位 ms point2PointDelay 代表 “主播=>云端=>观众” 的延迟，更准确地说，它代表了“采集=>编码=>网络传输=>接收=>缓冲=>解码=>播放” 全链路的延迟。 point2PointDelay 需要本地和远端的 SDK 均为 8.5 及以上的版本才生效，若远端用户为 8.5 以前的版本，此数值会一直为0，代表无意义。
remoteNetworkRTT	【字段含义】该用户从 SDK 到云端的往返延时，单位 ms 该数值代表从 SDK 发送一个网络包到云端，再从云端回送一个网络包到 SDK 的总计耗时，也就是一个网络包经历 “SDK=>云端=>SDK” 的总耗时。 该数值越小越好：如果 remoteNetworkRTT < 50ms，意味着较低的音视频通话延迟；如果 remoteNetworkRTT > 200ms，则意味着较高的音视频通话延迟。 需要特别解释的是，remoteNetworkRTT 代表 “SDK=>云端=>SDK” 的总耗时，所以不需要区分 remoteNetworkUpRTT 和 remoteNetworkDownRTT
remoteNetworkUplinkLoss	【字段含义】该用户从 SDK 到云端的上行丢包率，单位 (%) 该数值越小越好，如果 remoteNetworkUplinkLoss 为 0%，则意味着上行链路的网络质量很好，上传到云端的数据包基本不发生丢失。 如果 remoteNetworkUplinkLoss 为 30%，则意味着 SDK 向云端发送的音视频数据包中，会有 30% 丢失在传输链路中。
streamType	【字段含义】视频流类型（高清大画面|低清小画面|辅流画面）。
userId	【字段含义】用户 ID
videoBitrate	【字段含义】远端视频的码率，单位 Kbps。
videoBlockRate	【字段含义】视频播放卡顿率，单位 (%) 视频播放卡顿率（videoBlockRate）= 视频播放的累计卡顿时长（videoTotalBlockTime）/ 视频播放的总时长。
videoPacketLoss	【字段含义】该路视频流的总丢包率（%）。 videoPacketLoss 代表该路视频流历经 `主播>云端>观众` 这样一条完整的传输链路后，最终在观众端统计到的丢包率。

	videoPacketLoss 越小越好，丢包率为0即表示该路视频流的所有数据均已经完整地到达了观众端。 如果出现了 downLoss == 0 但 videoPacketLoss != 0 的情况，说明该路视频流在 `云端>观众` 这一段链路上没有出现丢包，但是在 `主播>云端` 这一段链路上出现了不可恢复的丢包。
videoTotalBlockTime	【字段含义】视频播放的累计卡顿时长，单位 ms
width	【字段含义】远端视频的宽度，单位 px。

TRTCStatistics

TRTCStatistics

网络和性能的汇总统计指标

枚举类型	描述
appCpu	【字段含义】当前应用的 CPU 使用率，单位 (%)，Android 8.0 以上不支持。
downLoss	【字段含义】从云端到 SDK 的下行丢包率，单位 (%) 该数值越小越好，如果 downLoss 为 0%，则意味着下行链路的网络质量很好，从云端接收的数据包基本不发生丢失。 如果 downLoss 为 30%，则意味着云端向 SDK 传输的音视频数据包中，会有 30% 丢失在传输链路中。
gatewayRtt	【字段含义】从 SDK 到本地路由器的往返时延，单位 ms 该数值代表从 SDK 发送一个网络包到本地路由器网关，再从网关回送一个网络包到 SDK 的总计耗时，也就是一个网络包经历【SDK>网关>SDK【 的总耗时。 该数值越小越好：如果 gatewayRtt < 50ms，意味着较低的音视频通话延迟；如果 gatewayRtt > 200ms，则意味着较高的音视频通话延迟。 当网络类型为蜂窝网时，该值无效。
localStatistics	【字段含义】本地的音视频统计信息 由于本地可能有三路音视频流（即高清大画面，低清小画面，以及辅流画面），因此本地的音视频统计信息是一个数组。
receivedBytes	【字段含义】总接收字节数（包含信令数据和音视频数据），单位：字节数（Bytes）。
remoteStatistics	【字段含义】远端的音视频统计信息

	因为同时可能有多个远端用户，而且每个远端用户同时可能有多路音视频流（即高清大画面，低清小画面，以及辅流画面），因此远端的音视频统计信息是一个数组。
rtt	【字段含义】从 SDK 到云端的往返延时，单位 ms 该数值代表从 SDK 发送一个网络包到云端，再从云端回送一个网络包到 SDK 的总计耗时，也就是一个网络包经历 “SDK=>云端=>SDK” 的总耗时。 该数值越小越好：如果 rtt < 50ms，意味着较低的音视频通话延迟；如果 rtt > 200ms，则意味着较高的音视频通话延迟。 需要特别解释的是，rtt 代表 “SDK=>云端=>SDK” 的总耗时，所以不需要区分 upRtt 和 downRtt。
sentBytes	【字段含义】总发送字节数（包含信令数据和音视频数据），单位：字节数（Bytes）。
systemCpu	【字段含义】当前系统的 CPU 使用率，单位 (%)，Android 8.0 以上不支持。
upLoss	【字段含义】从 SDK 到云端的上行丢包率，单位 (%) 该数值越小越好，如果 upLoss 为 0%，则意味着上行链路的网络质量很好，上传到云端的数据包基本不发生丢失。 如果 upLoss 为 30%，则意味着 SDK 向云端发送的音视频数据包中，会有 30% 丢失在传输链路中。

TXAudioEffectManager

最近更新时间：2024-03-06 14:51:31

Copyright (c) 2021 Tencent. All rights reserved.

Module: 背景音乐、短音效和人声特效的管理类

Function: 用于对背景音乐、短音效和人声特效进行设置的管理类

TXAudioEffectManager

TXAudioEffectManager

函数列表	描述
enableVoiceEarMonitor:	开启耳返
setVoiceEarMonitorVolume:	设置耳返音量
setVoiceReverbType:	设置人声的混响效果
setVoiceChangerType:	设置人声的变声特效
setVoiceVolume:	设置语音音量
setVoicePitch:	设置语音音调
startPlayMusic:onStart:onProgress:onComplete:	开始播放背景音乐
stopPlayMusic:	停止播放背景音乐
pausePlayMusic:	暂停播放背景音乐
resumePlayMusic:	恢复播放背景音乐
setAllMusicVolume:	设置所有背景音乐的本地音量和远端音量的大小
setMusicPublishVolume:volume:	设置某一首背景音乐的远端音量的大小
setMusicPlayoutVolume:volume:	设置某一首背景音乐的本地音量的大小
setMusicPitch:pitch:	调整背景音乐的音调高低

setMusicSpeedRate:speedRate:	调整背景音乐的变速效果
getMusicCurrentPosInMS:	获取背景音乐的播放进度（单位：毫秒）
getMusicDurationInMS:	获取背景音乐的总时长（单位：毫秒）
seekMusicToPosInMS:pts:	设置背景音乐的播放进度（单位：毫秒）
setMusicScratchSpeedRate:speedRate:	调整搓碟的变速效果
preloadMusic:onProgress:onError:	预加载背景音乐
getMusicTrackCount:	获取背景音乐的音轨数量
setMusicTrack:track:	指定背景音乐的播放音轨

结构体类型

函数列表	描述
TXAudioMusicParam	背景音乐的播放控制信息

枚举类型

枚举类型	描述
TXVoiceReverbType	混响特效
TXVoiceChangeType	变声特效

enableVoiceEarMonitor:

enableVoiceEarMonitor:

```
- (void)enableVoiceEarMonitor:      (BOOL)enable
```

开启耳返

主播开启耳返后，可以在耳机里听到麦克风采集到的自己发出的声音，该特效适用于主播唱歌的应用场景中。

需要您注意的是，由于蓝牙耳机的硬件延迟非常高，所以在主播佩戴蓝牙耳机时无法开启此特效，请尽量在用户界面上提示主播佩戴有线耳机。
同时也需要注意，并非所有的手机开启此特效后都能达到优秀的耳返效果，我们已经对部分耳返效果不佳的手机屏蔽了该特效。

参数	描述
enable	YES：开启；NO：关闭。

注意
仅在主播佩戴耳机时才能开启此特效，同时请您提示主播佩戴有线耳机。

setVoiceEarMonitorVolume:

setVoiceEarMonitorVolume:

```
- (void)setVoiceEarMonitorVolume:(NSInteger)volume
```

设置耳返音量

通过该接口您可以设置耳返特效中声音的音量大小。

参数	描述
volume	音量大小，取值范围为 0 - 100，默认值：100。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setVoiceReverbType:

setVoiceReverbType:

```
- (void)setVoiceReverbType:(TXVoiceReverbType)reverbType
```

设置人声的混响效果

通过该接口您可以设置人声的混响效果，具体特效请参见枚举定义 TXVoiceReverbType 。

> **注意**
> 设置的效果在退出房间后会自动失效，如果下次进房还需要对应特效，需要调用此接口再次进行设置。

setVoiceChangerType:

setVoiceChangerType:

```
- (void)setVoiceChangerType:(TXVoiceChangeType)changerType
```

设置人声的变声特效

通过该接口您可以设置人声的变声特效，具体特效请参见枚举定义 TXVoiceChangeType 。

> **注意**
> 设置的效果在退出房间后会自动失效，如果下次进房还需要对应特效，需要调用此接口再次进行设置。

setVoiceVolume:

setVoiceVolume:

```
- (void)setVoiceVolume:(NSInteger)volume
```

设置语音音量

该接口可以设置语音音量的大小，一般配合音乐音量的设置接口 setAllMusicVolume 协同使用，用于调谐语音和音乐在混音前各自的音量占比。

参数	描述
volu	音量大小，取值范围为0 - 100，默认值：100。

me	

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setVoicePitch:

setVoicePitch:

```
-(void)setVoicePitch:    (double)pitch
```

设置语音音调

该接口可以设置语音音调，用于实现变调不变速的目的。

参数	描述
pitch	音调，取值范围为-1.0f~1.0f，默认值：0.0f。

startPlayMusic:onStart:onProgress:onComplete:

startPlayMusic:onStart:onProgress:onComplete:

```
- (void)startPlayMusic:    (TXAudioMusicParam *)musicParam
onStart:    (TXAudioMusicStartBlock _Nullable)startI
onProgress:    (TXAudioMusicProgressBlock _Nullable)p
onComplete:    (TXAudioMusicCompleteBlock _Nullable)
```

开始播放背景音乐

每个音乐都需要您指定具体的 ID，您可以通过该 ID 对音乐的开始、停止、音量等进行设置。

参数	描述

completeBlock	播放结束回调。
musicParam	音乐参数。
progressBlock	播放进度回调。
startBlock	播放开始回调。

注意

1. 如果要多次播放同一首背景音乐，请不要每次播放都分配一个新的 ID，我们推荐使用相同的 ID。
2. 若您希望同时播放多首不同的音乐，请为不同的音乐分配不同的 ID 进行播放。
3. 如果使用同一个 ID 播放不同音乐，SDK 会先停止播放旧的音乐，再播放新的音乐。

stopPlayMusic:

stopPlayMusic:

```
- (void)stopPlayMusic:      (int32_t)id
```

停止播放背景音乐

参数	描述
id	音乐 ID。

pausePlayMusic:

pausePlayMusic:

```
- (void)pausePlayMusic:      (int32_t)id
```

暂停播放背景音乐

参数	描述
id	音乐 ID。

resumePlayMusic:

resumePlayMusic:

- (void)resumePlayMusic: (int32_t)id

恢复播放背景音乐

参数	描述
id	音乐 ID。

setAllMusicVolume:

setAllMusicVolume:

- (void)setAllMusicVolume: (NSInteger)volume

设置所有背景音乐的本地音量和远端音量的大小

该接口可以设置所有背景音乐的本地音量和远端音量。

- 本地音量：即主播本地可以听到的背景音乐的音量大小。
- 远端音量：即观众端可以听到的背景音乐的音量大小。

参数	描述
volume	音量大小，取值范围为0 - 100，默认值：60。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPublishVolume:volume:

setMusicPublishVolume:volume:

```
- (void)setMusicPublishVolume:    (int32_t)id
                       volume:    (NSInteger)volume
```

设置某一首背景音乐的远端音量的大小

该接口可以细粒度地控制每一首背景音乐的远端音量，也就是观众端可听到的背景音乐的音量大小。

参数	描述
id	音乐 ID。
volume	音量大小，取值范围为0 - 100；默认值：60。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPlayoutVolume:volume:

setMusicPlayoutVolume:volume:

```
- (void)setMusicPlayoutVolume:    (int32_t)id
                       volume:    (NSInteger)volume
```

设置某一首背景音乐的本地音量的大小

该接口可以细粒度地控制每一首背景音乐的本地音量，也就是主播本地可以听到的背景音乐的音量大小。

参数	描述
id	音乐 ID。
volume	音量大小，取值范围为0 - 100，默认值：60。

> **注意**
> 如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPitch:pitch:

setMusicPitch:pitch:

```
- (void)setMusicPitch:    (int32_t)id
pitch:    (double)pitch
```

调整背景音乐的音调高低

参数	描述
id	音乐 ID。
pitch	音调，默认值是0.0f，范围是：[-1 ~ 1] 之间的浮点数。

setMusicSpeedRate:speedRate:

setMusicSpeedRate:speedRate:

```
- (void)setMusicSpeedRate:    (int32_t)id
speedRate:    (double)speedRate
```

调整背景音乐的变速效果

参数	描述
id	音乐 ID。
speedRate	速度，默认值是1.0f，范围是：[0.5 ~ 2] 之间的浮点数。

getMusicCurrentPosInMS:

getMusicCurrentPosInMS:

```
- (NSInteger)getMusicCurrentPosInMS:    (int32_t)id
```

获取背景音乐的播放进度（单位：毫秒）

参数	描述
id	音乐 ID。

返回值说明：

成功返回当前播放时间，单位：毫秒，失败返回 -1。

getMusicDurationInMS:

getMusicDurationInMS:

```
- (NSInteger)getMusicDurationInMS:    (NSString *)path
```

获取背景音乐的总时长（单位：毫秒）

参数	描述
path	音乐文件路径。

返回值说明：

成功返回时长，失败返回 -1。

seekMusicToPosInMS:pts:

seekMusicToPosInMS:pts:

```
- (void)seekMusicToPosInMS:    (int32_t)id
                       pts:    (NSInteger)pts
```

设置背景音乐的播放进度（单位：毫秒）

参数	描述
id	音乐 ID。
pts	单位: 毫秒。

注意
请尽量避免过度频繁地调用该接口，因为该接口可能会再次读写音乐文件，耗时稍高。
因此，当用户拖拽音乐的播放进度条时，请在用户完成拖拽操作后再调用本接口。
因为 UI 上的进度条控件往往会以很高的频率反馈用户的拖拽进度，如不做频率限制，会导致较差的用户体验。

setMusicScratchSpeedRate:speedRate:

setMusicScratchSpeedRate:speedRate:

```
- (void)setMusicScratchSpeedRate:    (int32_t)id
                       speedRate:    (double)scratchSpeedRate
```

调整搓碟的变速效果

参数	描述
id	音乐 ID。
scratchSpeedRate	搓碟速度，默认值是1.0f，范围是：[-12.0 ~ 12.0] 之间的浮点数，速度值正/负表示方向正/反，绝对值大小表示速度快慢。

注意
前置条件 preloadMusic 成功。

preloadMusic:onProgress:onError:

preloadMusic:onProgress:onError:

```
- (void)preloadMusic:    (TXAudioMusicParam *)preloadParam
          onProgress:    (TXMusicPreloadProgressBlock _Nullable
             onError:    (TXMusicPreloadErrorBlock _Nullable)err
```

预加载背景音乐

每个音乐都需要您指定具体的 ID，您可以通过该 ID 对音乐的开始、停止、音量等进行设置。

参数	描述
preloadParam	预加载音乐参数。

注意

1. 预先加载最多同时支持2个不同 ID 的预加载，且预加载时长不超过10分钟，使用完需 stopPlayMusic，否则内存不释放。
2. 若该ID对应的音乐正在播放中，预加载会失败，需先调用 stopPlayMusic。
3. 当 musicParam 和传入 startPlayMusic 的 musicParam 完全相同时，预加载有效。

getMusicTrackCount:

getMusicTrackCount:

```
- (NSInteger)getMusicTrackCount:    (int32_t)id
```

获取背景音乐的音轨数量

参数	描述
id	音乐 ID。

setMusicTrack:track:

setMusicTrack:track:

```
- (void)setMusicTrack:    (int32_t)id
```

track: (NSInteger)track

指定背景音乐的播放音轨

参数	描述
id	音乐 ID。
index	默认播放第一个音轨。取值范围[0, 音轨总数)。

注意
音轨总数量可通过 getMusicTrackCount 接口获取。

TXVoiceReverbType

TXVoiceReverbType

混响特效

混响特效可以作用于人声之上，通过声学算法对声音进行叠加处理，模拟出各种不同环境下的临场感受，目前支持如下几种混响效果：
0：关闭；1：KTV；2：小房间；3：大会堂；4：低沉；5：洪亮；6：金属声；7：磁性；8：空灵；9：录音棚；10：悠扬 11：录音棚2。

枚举	取值	描述
TXVoiceReverbType_0	0	关闭特效
TXVoiceReverbType_1	1	KTV
TXVoiceReverbType_2	2	小房间
TXVoiceReverbType_3	3	大会堂
TXVoiceReverbType_4	4	低沉

TXVoiceReverbType_5	5	洪亮
TXVoiceReverbType_6	6	金属声
TXVoiceReverbType_7	7	磁性
TXVoiceReverbType_8	8	空灵
TXVoiceReverbType_9	9	录音棚
TXVoiceReverbType_10	10	悠扬
TXVoiceReverbType_11	11	录音棚2

TXVoiceChangeType

TXVoiceChangeType

变声特效

变声特效可以作用于人声之上，通过声学算法对人声进行二次处理，以获得与原始声音所不同的音色，目前支持如下几种变声特效：
0：关闭；1：熊孩子；2：萝莉；3：大叔；4：重金属；5：感冒；6：外语腔；7：困兽；8：肥宅；9：强电流；10：重机械；11：空灵。

枚举	取值	描述
TXVoiceChangeType_0	0	关闭
TXVoiceChangeType_1	1	熊孩子
TXVoiceChangeType_2	2	萝莉
TXVoiceChangeTy	3	大叔

pe_3		
TXVoiceChangeType_4	4	重金属
TXVoiceChangeType_5	5	感冒
TXVoiceChangeType_6	6	外语腔
TXVoiceChangeType_7	7	困兽
TXVoiceChangeType_8	8	肥宅
TXVoiceChangeType_9	9	强电流
TXVoiceChangeType_10	10	重机械
TXVoiceChangeType_11	11	空灵

TXAudioMusicParam

TXAudioMusicParam

背景音乐的播放控制信息

该信息用于在接口 startPlayMusic 中指定背景音乐的相关信息，包括播放 ID、文件路径和循环次数等：
1. 如果要多次播放同一首背景音乐，请不要每次播放都分配一个新的 ID，我们推荐使用相同的 ID。
2. 若您希望同时播放多首不同的音乐，请为不同的音乐分配不同的 ID 进行播放。
3. 如果使用同一个 ID 播放不同音乐，SDK 会先停止播放旧的音乐，再播放新的音乐。

枚举类型	描述
ID	【字段含义】音乐 ID。 【特殊说明】SDK 允许播放多路音乐，因此需要使用 ID 进行标记，用于控制音乐的开始、停止、音量等。
endTimeM	【字段含义】音乐结束播放时间点，单位毫秒，0表示播放至文件结尾。

S	
isShortFile	【字段含义】播放的是否为短音乐文件。 【推荐取值】YES：需要重复播放的短音乐文件；NO：正常的音乐文件。默认值：NO。
loopCount	【字段含义】音乐循环播放的次数。 【推荐取值】取值范围为0 - 任意正整数，默认值：0。0 表示播放音乐一次；1 表示播放音乐两次；以此类推。
path	【字段含义】音效文件的完整路径或 URL 地址。支持的音频格式包括 MP3、AAC、M4A、WAV。
publish	【字段含义】是否将音乐传到远端。 【推荐取值】YES：音乐在本地播放的同时，远端用户也能听到该音乐；NO：主播只能在本地听到该音乐，远端观众听不到。默认值：NO。
startTimeMS	【字段含义】音乐开始播放时间点，单位：毫秒。

TXBeautyManager

最近更新时间：2024-03-06 14:51:31

Copyright (c) 2021 Tencent. All rights reserved.

Module: 美颜与图像处理参数设置类

Function: 修改美颜、滤镜、绿幕等参数

TXBeautyManager

TXBeautyManager

函数列表	描述
setBeautyStyle:	设置美颜（磨皮）算法
setBeautyLevel:	设置美颜级别
setWhitenessLevel:	设置美白级别
enableSharpnessEnhancement:	开启清晰度增强
setRuddyLevel:	设置红润级别
setFilter:	设置色彩滤镜效果
setFilterStrength:	设置色彩滤镜的强度
setGreenScreenFile:	设置绿幕背景视频
setEyeScaleLevel:	设置大眼级别
setFaceSlimLevel:	设置瘦脸级别
setFaceVLevel:	设置 V 脸级别
setChinLevel:	设置下巴拉伸或收缩
setFaceShortLevel:	设置短脸级别
setFaceNarrowLevel:	设置窄脸级别

setNoseSlimLevel:	设置瘦鼻级别
setEyeLightenLevel:	设置亮眼级别
setToothWhitenLevel:	设置牙齿美白级别
setWrinkleRemoveLevel:	设置祛皱级别
setPounchRemoveLevel:	设置祛眼袋级别
setSmileLinesRemoveLevel:	设置法令纹去除级别
setForeheadLevel:	设置发际线调整级别
setEyeDistanceLevel:	设置眼距
setEyeAngleLevel:	设置眼角调整级别
setMouthShapeLevel:	设置嘴型调整级别
setNoseWingLevel:	设置鼻翼调整级别
setNosePositionLevel:	设置鼻子位置
setLipsThicknessLevel:	设置嘴唇厚度
setFaceBeautyLevel:	设置脸型
setMotionTmpl:inDir:	选择 AI 动效挂件
setMotionMute:	是否在动效素材播放时静音

枚举类型

枚举类型	描述
TXBeautyStyle	美颜（磨皮）算法

setBeautyStyle:

setBeautyStyle:

- (void)setBeautyStyle: (TXBeautyStyle)beautyStyle

设置美颜（磨皮）算法

TRTC 内置多种不同的磨皮算法，您可以选择最适合您产品定位的方案：

参数	描述
beautyStyle	美颜风格，TXBeautyStyleSmooth：光滑；TXBeautyStyleNature：自然；TXBeautyStylePitu：优图。

setBeautyLevel:

setBeautyLevel:

- (void)setBeautyLevel: (float)beautyLevel

设置美颜级别

参数	描述
beautyLevel	美颜级别，取值范围 0 - 9； 0 表示关闭，9 表示效果最明显。

setWhitenessLevel:

setWhitenessLevel:

- (void)setWhitenessLevel: (float)whitenessLevel

设置美白级别

参数	描述
whitenessLevel	美白级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

enableSharpnessEnhancement:

enableSharpnessEnhancement:

```
- (void)enableSharpnessEnhancement:    (BOOL)enable
```

开启清晰度增强

setRuddyLevel:

setRuddyLevel:

```
- (void)setRuddyLevel:    (float)ruddyLevel
```

设置红润级别

参数	描述
ruddyLevel	红润级别，取值范围0 - 9；0 表示关闭，9 表示效果最明显。

setFilter:

setFilter:

```
- (void)setFilter:    (nullable TXImage *)image
```

设置色彩滤镜效果

色彩滤镜，是一副包含色彩映射关系的颜色查找表图片，您可以在我们提供的官方 Demo 中找到预先准备好的几张滤镜图片。
SDK 会根据该查找表中的映射关系，对摄像头采集出的原始视频画面进行二次处理，以达到预期的滤镜效果。

参数	描述

image	包含色彩映射关系的颜色查找表图片，必须是 png 格式。

setFilterStrength:

setFilterStrength:

- (void)setFilterStrength: (float)strength

设置色彩滤镜的强度

该数值越高，色彩滤镜的作用强度越明显，经过滤镜处理后的视频画面跟原画面的颜色差异越大。
我默认的滤镜浓度是 0.5，如果您觉得默认的滤镜效果不明显，可以设置为 0.5 以上的数字，最大值为 1。

参数	描述
strength	从 0 到 1，数值越大滤镜效果越明显，默认值为 0.5。

setGreenScreenFile:

setGreenScreenFile:

- (int)setGreenScreenFile: (nullable NSString *)path

设置绿幕背景视频

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见 腾讯美颜特效SDK ）中生效。此接口所开启的绿幕功能不具备智能去除背景的能力，需要被拍摄者的背后有一块绿色的幕布来辅助产生特效。

参数	描述
path	MP4格式的视频文件路径; 设置空值表示关闭特效。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeScaleLevel:

setEyeScaleLevel:

- (int)setEyeScaleLevel: (float)eyeScaleLevel

设置大眼级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
eyeScaleLevel	大眼级别，取值范围 0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceSlimLevel:

setFaceSlimLevel:

- (int)setFaceSlimLevel: (float)faceSlimLevel

设置瘦脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
faceSlimLevel	瘦脸级别，取值范围0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceVLevel:

setFaceVLevel:

- (int)setFaceVLevel: (float)faceVLevel

设置 V 脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
faceVLevel	V 脸级别，取值范围 0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setChinLevel:

setChinLevel:

- (int)setChinLevel: (float)chinLevel

设置下巴拉伸或收缩

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
chinLevel	下巴拉伸或收缩级别，取值范围 -9 – 9；0 表示关闭，小于 0 表示收缩，大于 0 表示拉伸。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceShortLevel:

setFaceShortLevel:

```
- (int)setFaceShortLevel:      (float)faceShortLevel
```

设置短脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
faceShortLevel	短脸级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceNarrowLevel:

setFaceNarrowLevel:

```
- (int)setFaceNarrowLevel:      (float)faceNarrowLevel
```

设置窄脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
level	窄脸级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setNoseSlimLevel:

setNoseSlimLevel:

```
- (int)setNoseSlimLevel:      (float)noseSlimLevel
```

设置瘦鼻级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
noseSlimLevel	瘦鼻级别，取值范围0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeLightenLevel:

setEyeLightenLevel:

```
- (int)setEyeLightenLevel:      (float)eyeLightenLevel
```

设置亮眼级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
eyeLightenLevel	亮眼级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setToothWhitenLevel:

setToothWhitenLevel:

```
- (int)setToothWhitenLevel:      (float)toothWhitenLevel
```

设置牙齿美白级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
toothWhitenLevel	白牙级别，取值范围 0 - 9；0表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setWrinkleRemoveLevel:

setWrinkleRemoveLevel:

```
- (int)setWrinkleRemoveLevel:    (float)wrinkleRemoveLevel
```

设置祛皱级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
wrinkleRemoveLevel	祛皱级别，取值范围0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setPounchRemoveLevel:

setPounchRemoveLevel:

```
- (int)setPounchRemoveLevel:    (float)pounchRemoveLevel
```

设置祛眼袋级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
pounchRemoveLevel	祛眼袋级别，取值范围 0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setSmileLinesRemoveLevel:

setSmileLinesRemoveLevel:

```
- (int)setSmileLinesRemoveLevel:      (float)smileLinesRemoveLevel
```

设置法令纹去除级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
smileLinesRemoveLevel	法令纹级别，取值范围 0 – 9；0表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setForeheadLevel:

setForeheadLevel:

```
- (int)setForeheadLevel:      (float)foreheadLevel
```

设置发际线调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
foreheadLevel	发际线级别，取值范围-9 – 9；0表示关闭，9表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeDistanceLevel:

setEyeDistanceLevel:

```
- (int)setEyeDistanceLevel:      (float)eyeDistanceLevel
```

设置眼距

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
eyeDistanceLevel	眼距级别，取值范围 -9 – 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeAngleLevel:

setEyeAngleLevel:

```
- (int)setEyeAngleLevel:      (float)eyeAngleLevel
```

设置眼角调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
eyeAngleLeve	眼角调整级别，取值范围-9 – 9；0表示关闭，9表示效果最明显。

l	

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setMouthShapeLevel:

setMouthShapeLevel:

```
- (int)setMouthShapeLevel:    (float)mouthShapeLevel
```

设置嘴型调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见 腾讯美颜特效SDK ）中生效。

参数	描述
mouthShapeLevel	嘴型级别，取值范围 -9 – 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setNoseWingLevel:

setNoseWingLevel:

```
- (int)setNoseWingLevel:    (float)noseWingLevel
```

设置鼻翼调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见 腾讯美颜特效SDK ）中生效。

参数	描述
noseWingLevel	鼻翼调整级别，取值范围 -9 – 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setNosePositionLevel:

setNosePositionLevel:

- (int)setNosePositionLevel: (float)nosePositionLevel

设置鼻子位置

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见 腾讯美颜特效SDK ）中生效。

参数	描述
nosePositionLevel	鼻子位置级别，取值范围 -9 – 9；0 表示关闭，小于 0 表示抬高，大于 0 表示降低。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setLipsThicknessLevel:

setLipsThicknessLevel:

- (int)setLipsThicknessLevel: (float)lipsThicknessLevel

设置嘴唇厚度

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见 腾讯美颜特效SDK ）中生效。

参数	描述
lipsThicknessLevel	嘴唇厚度级别，取值范围 -9 – 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceBeautyLevel:

setFaceBeautyLevel:

```
- (int)setFaceBeautyLevel:    (float)faceBeautyLevel
```

设置脸型

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
faceBeautyLevel	美型级别，取值范围 0 - 9；0 表示关闭，1 - 9 值越大，效果越明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setMotionTmpl:inDir:

setMotionTmpl:inDir:

```
- (void)setMotionTmpl:    (nullable NSString *)tmplName
                 inDir:    (nullable NSString *)tmplDir
```

选择 AI 动效挂件

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
tmplDir	动效素材文件所在目录。
tmplName	动效挂件名称。

setMotionMute:

setMotionMute:

- (void)setMotionMute: (BOOL)motionMute

是否在动效素材播放时静音

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见 腾讯美颜特效SDK ）中生效。
有些挂件本身会有声音特效，通过此 API 可以关闭这些特效播放时所带的声音效果。

参数	描述
motionMute	YES：静音；NO：不静音。

TXBeautyStyle

TXBeautyStyle

美颜（磨皮）算法

TRTC 内置多种不同的磨皮算法，您可以选择最适合您产品定位的方案。

枚举	取值	描述
TXBeautyStyleSmooth	0	光滑，算法比较激进，磨皮效果比较明显，适用于秀场直播。
TXBeautyStyleNature	1	自然，算法更多地保留了面部细节，磨皮效果更加自然，适用于绝大多数直播场景。
TXBeautyStylePitu	2	优图，由优图实验室提供，磨皮效果介于光滑和自然之间，比光滑保留更多皮肤细节，比自然磨皮程度更高。

TXDeviceManager

最近更新时间：2024-03-06 14:51:31

Copyright (c) 2021 Tencent. All rights reserved.

Module: 音视频设备管理模块

Function: 用于管理摄像头、麦克风和扬声器等音视频相关的硬件设备

TXDeviceManager

TXDeviceObserver

函数列表	描述
onDeviceChanged:type:state:	本地设备的通断状态发生变化（仅适用于桌面系统）

TXDeviceManager

函数列表	描述
isFrontCamera	判断当前是否为前置摄像头（仅适用于移动端）
switchCamera:	切换前置或后置摄像头（仅适用于移动端）
isCameraZoomSupported	查询当前摄像头是否支持缩放（仅适用于移动端）
getCameraZoomMaxRatio	获取摄像头的最大缩放倍数（仅适用于移动端）
setCameraZoomRatio:	设置摄像头的缩放倍数（仅适用于移动端）
isAutoFocusEnabled	查询是否支持自动识别人脸位置（仅适用于移动端）
enableCameraAutoFocus:	开启自动对焦功能（仅适用于移动端）
setCameraFocusPosition:	设置摄像头的对焦位置（仅适用于移动端）
isCameraTorchSupported	查询是否支持开启闪光灯（仅适用于移动端）
enableCameraTorch:	开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）

setAudioRoute:	设置音频路由（仅适用于移动端）
setExposureCompensation:	设置摄像头的曝光参数，取值范围从-1到1
getDevicesList:	获取设备列表（仅适用于桌面端）
setCurrentDevice:deviceId:	设置当前要使用的设备（仅适用于桌面端）
getCurrentDevice:	获取当前正在使用的设备（仅适用于桌面端）
setCurrentDeviceVolume:deviceType:	设置当前设备的音量（仅适用于桌面端）
getCurrentDeviceVolume:	获取当前设备的音量（仅适用于桌面端）
setCurrentDeviceMute:deviceType:	设置当前设备的静音状态（仅适用于桌面端）
getCurrentDeviceMute:	获取当前设备的静音状态（仅适用于桌面端）
enableFollowingDefaultAudioDevice:enable:	设置 SDK 使用的音频设备根据跟随系统默认设备（仅适用于桌面端）
startCameraDeviceTest:	开始摄像头测试（仅适用于桌面端）
stopCameraDeviceTest	结束摄像头测试（仅适用于桌面端）
startMicDeviceTest:	开始麦克风测试（仅适用于桌面端）
startMicDeviceTest:playback:	开始麦克风测试（仅适用于桌面端）
stopMicDeviceTest	结束麦克风测试（仅适用于桌面端）
startSpeakerDeviceTest:	开始扬声器测试（仅适用于桌面端）
stopSpeakerDeviceTest	结束扬声器测试（仅适用于桌面端）
setObserver:	设备热插拔回调（仅适用于 Mac 系统）
setSystemVolumeType:	设置系统音量类型（仅适用于移动端）

结构体类型

函数列表	描述
TXMediaDeviceInfo	音视频设备的相关信息（仅适用于桌面平台）

枚举类型

枚举类型	描述
TXSystemVolumeType	系统音量类型（仅适用于移动设备）
TXAudioRoute	音频路由（即声音的播放模式）
TXMediaDeviceType	设备类型（仅适用于桌面平台）
TXMediaDeviceState	设备操作

onDeviceChanged:type:state:

onDeviceChanged:type:state:

```
- (void)onDeviceChanged:(NSString*)deviceId
                   type:(TXMediaDeviceType)mediaType
                  state:(TXMediaDeviceState)mediaState
```

本地设备的通断状态发生变化（仅适用于桌面系统）

当本地设备（包括摄像头、麦克风以及扬声器）被插入或者拔出时，SDK 便会抛出此事件回调。

参数	描述
deviceId	设备 ID
state	通断状态，0：设备已添加；1：设备已被移除；2：设备已启用。
type	设备类型

isFrontCamera

isFrontCamera

判断当前是否为前置摄像头（仅适用于移动端）

switchCamera:

switchCamera:

```
- (NSInteger)switchCamera:      (BOOL)frontCamera
```

切换前置或后置摄像头（仅适用于移动端）

isCameraZoomSupported

isCameraZoomSupported

查询当前摄像头是否支持缩放（仅适用于移动端）

getCameraZoomMaxRatio

getCameraZoomMaxRatio

获取摄像头的最大缩放倍数（仅适用于移动端）

setCameraZoomRatio:

setCameraZoomRatio:

```
- (NSInteger)setCameraZoomRatio:      (CGFloat)zoomRatio
```

设置摄像头的缩放倍数（仅适用于移动端）

参数	描述
zoomRat	取值范围1 - 5，取值为1表示最远视角（正常镜头），取值为5表示最近视角（放大镜

io	头）。最大值推荐为5，若超过5，视频数据会变得模糊不清。

isAutoFocusEnabled

isAutoFocusEnabled

查询是否支持自动识别人脸位置（仅适用于移动端）

enableCameraAutoFocus:

enableCameraAutoFocus:

```
- (NSInteger)enableCameraAutoFocus:    (BOOL)enabled
```

开启自动对焦功能（仅适用于移动端）

开启后，SDK 会自动检测画面中的人脸位置，并将摄像头的焦点始终对焦在人脸位置上。

setCameraFocusPosition:

setCameraFocusPosition:

```
- (NSInteger)setCameraFocusPosition:    (CGPoint)position
```

设置摄像头的对焦位置（仅适用于移动端）

您可以通过该接口实现如下交互：
1. 在本地摄像头的预览画面上，允许用户单击操作。
2. 在用户的单击位置显示一个矩形方框，以示摄像头会在此处对焦。
3. 随后将用户点击位置的坐标通过本接口传递给 SDK，之后 SDK 会操控摄像头按照用户期望的位置进行对焦。

参数	描述
position	对焦位置，请传入期望对焦点的坐标值

注意

使用该接口的前提是先通过 enableCameraAutoFocus 关闭自动对焦功能。

返回值说明：

0：操作成功；负数：操作失败。

isCameraTorchSupported

isCameraTorchSupported

查询是否支持开启闪光灯（仅适用于移动端）

enableCameraTorch:

enableCameraTorch:

```
- (NSInteger)enableCameraTorch:(BOOL)enabled
```

开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）

setAudioRoute:

setAudioRoute:

```
- (NSInteger)setAudioRoute:(TXAudioRoute)route
```

设置音频路由（仅适用于移动端）

手机有两个音频播放设备：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。
设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。

setExposureCompensation:

setExposureCompensation:

```
- (NSInteger)setExposureCompensation:(CGFloat)value
```

设置摄像头的曝光参数，取值范围从-1到1

getDevicesList:

getDevicesList:

```
- (NSArray<TXMediaDeviceInfo *> * _Nullable)getDevicesList:(TXMediaDeviceType
```

获取设备列表（仅适用于桌面端）

参数	描述
type	设备类型，指定需要获取哪种设备的列表。详见 TXMediaDeviceType 定义。

注意

- 使用完毕后请调用 release 方法释放资源，这样可以让 SDK 维护 ITXDeviceCollection 对象的生命周期。
- 不要使用 delete 释放返回的 Collection 对象，delete ITXDeviceCollection* 指针会导致异常崩溃。
- type 只支持 TXMediaDeviceTypeMic、TXMediaDeviceTypeSpeaker、TXMediaDeviceTypeCamera。
- 此接口只支持 Mac 和 Windows 平台。

setCurrentDevice:deviceId:

setCurrentDevice:deviceId:

```
- (NSInteger)setCurrentDevice:(TXMediaDeviceType)type
                     deviceId:(NSString *)deviceId
```

设置当前要使用的设备（仅适用于桌面端）

参数	描述
deviceId	设备ID，您可以通过接口 getDevicesList 获得设备 ID。
type	设备类型，详见 TXMediaDeviceType 定义。

返回值说明：

0：操作成功；负数：操作失败。

getCurrentDevice:

getCurrentDevice:

```
- (TXMediaDeviceInfo * _Nullable)getCurrentDevice:(TXMediaDeviceType)type
```

获取当前正在使用的设备（仅适用于桌面端）

setCurrentDeviceVolume:deviceType:

setCurrentDeviceVolume:deviceType:

```
- (NSInteger)setCurrentDeviceVolume:(NSInteger)volume
                         deviceType:(TXMediaDeviceType)type
```

设置当前设备的音量（仅适用于桌面端）

这里的音量指的是麦克风的采集音量或者扬声器的播放音量，摄像头是不支持设置音量的。

参数	描述
volume	音量大小，取值范围为0 - 100，默认值：100。

getCurrentDeviceVolume:

getCurrentDeviceVolume:

```
- (NSInteger)getCurrentDeviceVolume:    (TXMediaDeviceType)type
```

获取当前设备的音量（仅适用于桌面端）

这里的音量指的是麦克风的采集音量或者扬声器的播放音量，摄像头是不支持获取音量的。

setCurrentDeviceMute:deviceType:

setCurrentDeviceMute:deviceType:

```
- (NSInteger)setCurrentDeviceMute:    (BOOL)mute
                       deviceType:    (TXMediaDeviceType)type
```

设置当前设备的静音状态（仅适用于桌面端）

这里的音量指的是麦克风和扬声器，摄像头是不支持静音操作的。

getCurrentDeviceMute:

getCurrentDeviceMute:

```
- (BOOL)getCurrentDeviceMute:    (TXMediaDeviceType)type
```

获取当前设备的静音状态（仅适用于桌面端）

这里的音量指的是麦克风和扬声器，摄像头是不支持静音操作的。

enableFollowingDefaultAudioDevice:enable:

enableFollowingDefaultAudioDevice:enable:

```
- (NSInteger)enableFollowingDefaultAudioDevice:    (TXMediaDeviceType)type
                                        enable:    (BOOL)enable
```

设置 SDK 使用的音频设备根据跟随系统默认设备（仅适用于桌面端）

仅支持设置麦克风和扬声器类型，摄像头暂不支持跟随系统默认设备

参数	描述
enable	是否跟随系统默认的音频设备。 • true：跟随。当系统默认音频设备发生改变或者有新音频设备插入时，SDK 立即切换音频设备。 • false：不跟随。当系统默认音频设备发生改变或者有新音频设备插入时，SDK 不会切换音频设备。
type	设备类型，详见 TXMediaDeviceType 定义。

startCameraDeviceTest:

startCameraDeviceTest:

```
- (NSInteger)startCameraDeviceTest:      (NSView *)view
```

开始摄像头测试（仅适用于桌面端）

> **注意**
> 在测试过程中可以使用 setCurrentDevice 接口切换摄像头。

stopCameraDeviceTest

stopCameraDeviceTest

结束摄像头测试（仅适用于桌面端）

startMicDeviceTest:

startMicDeviceTest:

```
- (NSInteger)startMicDeviceTest:      (NSInteger)interval
```

开始麦克风测试（仅适用于桌面端）

该接口可以测试麦克风是否能正常工作，测试到的麦克风采集音量的大小，会以回调的形式通知给您，其中 volume 的取值范围为0 - 100。

参数	描述
interval	麦克风音量的回调间隔。

注意
该接口调用后默认会回播麦克风录制到的声音到扬声器中。

startMicDeviceTest:playback:

startMicDeviceTest:playback:

```
- (NSInteger)startMicDeviceTest:    (NSInteger)interval
                       playback:    (BOOL)playback
```

开始麦克风测试（仅适用于桌面端）

该接口可以测试麦克风是否能正常工作，测试到的麦克风采集音量的大小，会以回调的形式通知给您，其中 volume 的取值范围为0 - 100。

参数	描述
interval	麦克风音量的回调间隔。
playback	是否开启回播麦克风声音，开启后用户测试麦克风时会听到自己的声音。

stopMicDeviceTest

stopMicDeviceTest

结束麦克风测试（仅适用于桌面端）

startSpeakerDeviceTest:

startSpeakerDeviceTest:

- (NSInteger)startSpeakerDeviceTest: (NSString *)audioFilePath

开始扬声器测试（仅适用于桌面端）

该接口通过播放指定的音频文件，用于测试播放设备是否能正常工作。如果用户在测试时能听到声音，说明播放设备能正常工作。

参数	描述
filePath	声音文件的路径

stopSpeakerDeviceTest

stopSpeakerDeviceTest

结束扬声器测试（仅适用于桌面端）

setObserver:

setObserver:

- (void)setObserver: (nullable id<TXDeviceObserver>) observe

设备热插拔回调（仅适用于 Mac 系统）

setSystemVolumeType:

setSystemVolumeType:

- (NSInteger)setSystemVolumeType: (TXSystemVolumeType)type

设置系统音量类型（仅适用于移动端）

@deprecated v9.5 版本开始不推荐使用，建议使用 TRTCCloud 中的 startLocalAudio(quality) 接口替代之，通过 quality 参数来决策音质。

TXSystemVolumeType(Deprecated)

TXSystemVolumeType(Deprecated)

系统音量类型（仅适用于移动设备）

@deprecated v9.5 版本开始不推荐使用。

现代智能手机中一般都具备两套系统音量类型，即“通话音量”和“媒体音量”。

- 通话音量：手机专门为接打电话所设计的音量类型，自带回声抵消（AEC）功能，并且支持通过蓝牙耳机上的麦克风进行拾音，缺点是音质比较一般。

当您通过手机侧面的音量按键下调手机音量时，如果无法将其调至零（也就是无法彻底静音），说明您的手机当前处于通话音量。

- 媒体音量：手机专门为音乐场景所设计的音量类型，无法使用系统的 AEC 功能，并且不支持通过蓝牙耳机的麦克风进行拾音，但具备更好的音乐播放效果。

当您通过手机侧面的音量按键下调手机音量时，如果能够将手机音量调至彻底静音，说明您的手机当前处于媒体音量。

SDK 目前提供了三种系统音量类型的控制模式：自动切换模式、全程通话音量模式、全程媒体音量模式。

枚举	取值	描述
TXSystemVolumeTypeAuto	0	自动切换模式
TXSystemVolumeTypeMedia	1	全程媒体音量
TXSystemVolumeTypeVOIP	2	全程通话音量

TXAudioRoute

TXAudioRoute

音频路由（即声音的播放模式）

音频路由，即声音是从手机的扬声器还是从听筒中播放出来，因此该接口仅适用于手机等移动端设备。
手机有两个扬声器：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。

- 设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
- 设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。

枚举	取值	描述
TXAudioRouteSpeakerphone	0	Speakerphone：使用扬声器播放（即“免提”），扬声器位于手机底部，声音偏大，适合外放音乐。
TXAudioRouteEarpiece	1	Earpiece：使用听筒播放，听筒位于手机顶部，声音偏小，适合需要保护隐私的通话场景。

TXMediaDeviceType

TXMediaDeviceType

设备类型（仅适用于桌面平台）

该枚举值用于定义三种类型的音视频设备，即摄像头、麦克风和扬声器，以便让一套设备管理接口可以操控三种不同类型的设备。

枚举	取值	描述
TXMediaDeviceTypeUnknown	-1	未定义的设备类型
TXMediaDeviceTypeAudioInput	0	麦克风类型设备
TXMediaDeviceTypeAudioOutput	1	扬声器类型设备
TXMediaDeviceTypeVideoCamera	2	摄像头类型设备

TXMediaDeviceState

TXMediaDeviceState

设备操作

该枚举值用于本地设备的状态变化通知 onDeviceChanged 。

枚举	取值	描述
TXMediaDeviceStateAdd	0	设备已被插入
TXMediaDeviceStateRemove	1	设备已被移除
TXMediaDeviceStateActive	2	设备已启用
TXMediaDefaultDeviceChanged	3	系统默认设备变更

TXMediaDeviceInfo

TXMediaDeviceInfo

音视频设备的相关信息（仅适用于桌面平台）

该结构体用于描述一个音视频设备的关键信息，比如设备 ID、设备名称等等，以便用户能够在用户界面上选择自己期望使用的音视频设备。

枚举类型	描述
deviceId	设备 ID（UTF-8）
deviceName	设备名称（UTF-8）
deviceProperties	设备属性
type	设备类型

类型定义

最近更新时间：2024-03-06 14:51:31

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTC 关键类型定义

Function: 分辨率、质量等级等枚举和常量值的定义

类型定义

结构体类型

函数列表	描述
TRTCParams	进房参数
TRTCVideoEncParam	视频编码参数
TRTCNetworkQosParam	网络流控（Qos）参数集
TRTCRenderParams	视频画面的渲染参数
TRTCQualityInfo	网络质量
TRTCVolumeInfo	音量大小
TRTCSpeedTestParams	测速参数
TRTCSpeedTestResult	网络测速结果
TRTCVideoFrame	视频帧信息
TRTCAudioFrame	音频帧数据
TRTCMixUser	云端混流中各路画面的描述信息
TRTCTranscodingConfig	云端混流的排版布局和转码参数
TRTCPublishCDNParam	向非腾讯云 CDN 上发布音视频流时需设置的转推参数
TRTCAudioRecordingParams	本地音频文件的录制参数

TRTCLocalRecordingParams	本地媒体文件的录制参数
TRTCAudioEffectParam	音效参数（已废弃）
TRTCSwitchRoomConfig	房间切换参数
TRTCAudioFrameDelegateFormat	音频自定义回调的格式参数
TRTCScreenCaptureSourceInfo	屏幕分享的目标信息（仅适用于桌面系统）
TRTCAudioParallelParams	远端音频流智能并发播放策略的参数
TRTCUser	媒体流发布相关配置的用户信息
TRTCPublishCdnUrl	向腾讯或者第三方 CDN 上发布音视频流时需设置的 url 配置
TRTCPublishTarget	目标推流配置
TRTCVideoLayout	转码视频布局
TRTCWatermark	水印布局
TRTCStreamEncoderParam	媒体流编码输出参数
TRTCStreamMixingConfig	媒体流转码配置参数
TRTCPayloadPrivateEncryptionConfig	媒体流私有加密配置
TRTCAudioVolumeEvaluateParams	音量评估等相关参数设置

枚举类型

枚举类型	描述
TRTCVideoResolution	视频分辨率
TRTCVideoResolutionMode	视频宽高比模式
TRTCVideoStreamType	视频流类型
TRTCVideoFillMode	视频画面填充模式

TRTCVideoRotation	视频画面旋转方向
TRTCBeautyStyle	美颜（磨皮）算法
TRTCVideoPixelFormat	视频像素格式
TRTCVideoBufferType	视频数据传递方式
TRTCVideoMirrorType	视频的镜像类型
TRTCSnapshotSourceType	本地视频截图的数据源
TRTCAppScene	应用场景
TRTCRoleType	角色
TRTCQosControlMode	流控模式（已废弃）
TRTCVideoQosPreference	画质偏好
TRTCQuality	网络质量
TRTCAVStatusType	视频状态类型
TRTCAVStatusChangeReason	视频状态变化原因类型
TRTCAudioSampleRate	音频采样率
TRTCAudioQuality	声音音质
TRTCAudioRoute	音频路由（即声音的播放模式）
TRTCReverbType	声音混响模式
TRTCVoiceChangerType	变声类型
TRTCSystemVolumeType	系统音量类型（仅适用于移动设备）
TRTCAudioFrameOperationMode	音频回调数据读写模式
TRTCLogLevel	Log 级别
TRTCGSensorMode	重力感应开关（仅适用于移动端）

TRTCScreenCaptureSourceType	屏幕分享的目标类型（仅适用于桌面端）
TRTCTranscodingConfigMode	云端混流的排版模式
TRTCRecordType	媒体录制类型
TRTCMixInputType	混流输入类型
TRTCAudioRecordingContent	音频录制内容类型
TRTCPublishMode	媒体流发布模式
TRTCEncryptionAlgorithm	加密算法
TRTCSpeedTestScene	测速场景
TRTCGravitySensorAdaptiveMode	设置重力感应的适配模式（仅适用于移动端）

TRTCVideoResolution

TRTCVideoResolution

视频分辨率

此处仅定义横屏分辨率（如 640 × 360），如需使用竖屏分辨率（如360 × 640），需要同时指定 TRTCVideoResolutionMode 为 Portrait。

枚举	取值	描述
TRTCVideoResolution_120_120	1	宽高比 1:1；分辨率 120x120；建议码率（VideoCall）80kbps; 建议码率（LIVE）120kbps。
TRTCVideoResolution_160_160	3	宽高比 1:1 分辨率 160x160；建议码率（VideoCall）100kbps; 建议码率（LIVE）150kbps。
TRTCVideoResolution_270_270	5	宽高比 1:1；分辨率 270x270；建议码率（VideoCall）200kbps; 建议码率（LIVE）

		300kbps。
TRTCVideoResolution_480_480	7	宽高比 1:1；分辨率 480x480；建议码率（VideoCall）350kbps; 建议码率（LIVE）500kbps。
TRTCVideoResolution_160_120	50	宽高比4:3；分辨率 160x120；建议码率（VideoCall）100kbps; 建议码率（LIVE）150kbps。
TRTCVideoResolution_240_180	52	宽高比 4:3；分辨率 240x180；建议码率（VideoCall）150kbps; 建议码率（LIVE）250kbps。
TRTCVideoResolution_280_210	54	宽高比 4:3；分辨率 280x210；建议码率（VideoCall）200kbps; 建议码率（LIVE）300kbps。
TRTCVideoResolution_320_240	56	宽高比 4:3；分辨率 320x240；建议码率（VideoCall）250kbps; 建议码率（LIVE）375kbps。
TRTCVideoResolution_400_300	58	宽高比 4:3；分辨率 400x300；建议码率（VideoCall）300kbps; 建议码率（LIVE）450kbps。
TRTCVideoResolution_480_360	60	宽高比 4:3；分辨率 480x360；建议码率（VideoCall）400kbps; 建议码率（LIVE）600kbps。
TRTCVideoResolution_640_480	62	宽高比 4:3；分辨率 640x480；建议码率（VideoCall）600kbps; 建议码率（LIVE）900kbps。
TRTCVideoResolution_960_720	64	宽高比 4:3；分辨率 960x720；建议码率（VideoCall）1000kbps; 建议码率（LIVE）1500kbps。
TRTCVideoResolution_160_90	100	宽高比 16:9；分辨率 160x90；建议码率（VideoCall）150kbps; 建议码率（LIVE）250kbps。
TRTCVideoResolution_256_144	102	宽高比 16:9；分辨率 256x144；建议码率（VideoCall）200kbps; 建议码率（LIVE）300kbps。
TRTCVideoResolution_320_180	104	宽高比 16:9；分辨率 320x180；建议码率（VideoCall）250kbps; 建议码率（LIVE）

		400kbps。
TRTCVideoResolution_480_270	106	宽高比 16:9；分辨率 480x270；建议码率（VideoCall）350kbps; 建议码率（LIVE）550kbps。
TRTCVideoResolution_640_360	108	宽高比 16:9；分辨率 640x360；建议码率（VideoCall）500kbps; 建议码率（LIVE）900kbps。
TRTCVideoResolution_960_540	110	宽高比 16:9；分辨率 960x540；建议码率（VideoCall）850kbps; 建议码率（LIVE）1300kbps。
TRTCVideoResolution_1280_720	112	宽高比 16:9；分辨率 1280x720；建议码率（VideoCall）1200kbps; 建议码率（LIVE）1800kbps。
TRTCVideoResolution_1920_1080	114	宽高比 16:9；分辨率 1920x1080；建议码率（VideoCall）2000kbps; 建议码率（LIVE）3000kbps。

TRTCVideoResolutionMode

TRTCVideoResolutionMode

视频宽高比模式

TRTCVideoResolution 中仅定义了横屏分辨率（如 640 × 360），如需使用竖屏分辨率（如360 × 640），需要同时指定 TRTCVideoResolutionMode 为 Portrait。

枚举	取值	描述
TRTCVideoResolutionModeLandscape	0	横屏分辨率，例如：TRTCVideoResolution_640_360 + TRTCVideoResolutionModeLandscape = 640 × 360。
TRTCVideoResolutionModePortrait	1	竖屏分辨率，例如：TRTCVideoResolution_640_360 + TRTCVideoResolutionModePortrait = 360 × 640。

TRTCVideoStreamType

TRTCVideoStreamType

视频流类型

TRTC 内部有三种不同的视频流，分别是：

- 高清大画面：一般用来传输摄像头的视频数据。
- 低清小画面：小画面和大画面的内容相互，但是分辨率和码率都比大画面低，因此清晰度也更低。
- 辅流画面：一般用于屏幕分享，同一时间在同一个房间中只允许一个用户发布辅流视频，其他用户必须要等该用户关闭之后才能发布自己的辅流。

> **注意**
> 不支持单独开启低清小画面，小画面必须依附于大画面而存在，SDK 会自动设定低清小画面的分辨率和码率。

枚举	取值	描述
TRTCVideoStreamTypeBig	0	高清大画面，一般用来传输摄像头的视频数据。
TRTCVideoStreamTypeSmall	1	低清小画面：小画面和大画面的内容相互，但是分辨率和码率都比大画面低，因此清晰度也更低。
TRTCVideoStreamTypeSub	2	辅流画面：一般用于屏幕分享，同一时间在同一个房间中只允许一个用户发布辅流视频，其他用户必须要等该用户关闭之后才能发布自己的辅流。

TRTCVideoFillMode

TRTCVideoFillMode

视频画面填充模式

如果视频显示区域的宽高比不等于视频内容的宽高比时，需要指定画面的填充模式。

枚举	取值	描述
TRTCVideoFillMode	0	填充模式：即将画面内容居中等比缩放以充满整个显示区域，

_Fill		超出显示区域的部分将会被裁剪掉，此模式下画面可能不完整。
TRTCVideoFillMode_Fit	1	适应模式：即按画面长边进行缩放以适应显示区域，短边部分会被填充为黑色，此模式下图像完整但可能留有黑边。

TRTCVideoRotation

TRTCVideoRotation

视频画面旋转方向

TRTC 提供了对本地和远程画面的旋转角度设置 API，下列的旋转角度都是指顺时针方向的。

枚举	取值	描述
TRTCVideoRotation_0	0	不旋转。
TRTCVideoRotation_90	1	顺时针旋转90度。
TRTCVideoRotation_180	2	顺时针旋转180度。
TRTCVideoRotation_270	3	顺时针旋转270度。

TRTCBeautyStyle

TRTCBeautyStyle

美颜（磨皮）算法

TRTC 内置多种不同的磨皮算法，您可以选择最适合您产品定位的方案。

枚举	取值	描述
TRTCBeautyStyleSmooth	0	光滑，算法比较激进，磨皮效果比较明显，适用于秀场直播。

TRTCBeautyStyleNature	1	自然，算法更多地保留了面部细节，磨皮效果更加自然，适用于绝大多数直播场景。
TRTCBeautyStylePitu	2	优图，由优图实验室提供，磨皮效果介于光滑和自然之间，比光滑保留更多皮肤细节，比自然磨皮程度更高。

TRTCVideoPixelFormat

TRTCVideoPixelFormat

视频像素格式

TRTC 提供针对视频的自定义采集和自定义渲染功能：

- 在自定义采集功能中，您可以用下列枚举值描述您采集的视频像素格式。
- 在自定义渲染功能中，您可以指定您期望 SDK 回调出的视频像素格式。

枚举	取值	描述
TRTCVideoPixelFormat_Unknown	0	未定义的格式。
TRTCVideoPixelFormat_I420	1	YUV420P(I420) 格式。
TRTCVideoPixelFormat_Texture_2D	7	OpenGL 2D 纹理格式。
TRTCVideoPixelFormat_32BGRA	6	BGRA 格式
TRTCVideoPixelFormat_NV12	5	YUV420SP（NV12）格式。

TRTCVideoBufferType

TRTCVideoBufferType

视频数据传递方式

在自定义采集和自定义渲染功能，您需要用到下列枚举值来指定您希望以什么方式传递视频数据：

- 方案一：使用内存 Buffer 传递视频数据，该方案在 iOS 效率尚可，但在 Android 系统上效率较差，Windows 暂时仅支持内存 Buffer 的传递方式。
- 方案二：使用 Texture 纹理传递视频数据，该方案在 iOS 和 Android 系统下均有较高的效率，Windows 暂不支持，需要您有一定的 OpenGL 编程基础。

枚举	取值	描述
TRTCVideoBufferType_Unknown	0	未定义的传递方式。
TRTCVideoBufferType_PixelBuffer	1	使用内存 Buffer 传递视频数据，iOS：PixelBuffer；Android：用于 JNI 层的 DirectBuffer；Win：内存数据块。
TRTCVideoBufferType_NSData	2	使用内存 Buffer 传递视频数据，iOS：经过一次额外整理后更加紧凑的 NSData 类型的内存块；Android：用于 JAVA 层的 byte[]。 该传递的方式的性能是几种方案中效率较差的一种。
TRTCVideoBufferType_Texture	3	使用 Texture 纹理传递视频数据。

TRTCVideoMirrorType

TRTCVideoMirrorType

视频的镜像类型

视频的镜像是指对视频内容进行左右翻转，尤其是对本地的摄像头预览视频，开启镜像后能给主播带来熟悉的“照镜子”体验。

枚举	取值	描述
TRTCVideoMirrorTypeAuto	0	自动模式：如果正使用前置摄像头则开启镜像，如果是后置摄像头则不开启镜像（仅适用于移动设备）。
TRTCVideoMirrorTypeEnable	1	强制开启镜像，不论当前使用的是前置摄像头还是后置摄像头。
TRTCVideoMirrorTypeDisable	2	强制关闭镜像，不论当前使用的是前置摄像头还是后置摄像头。

TRTCSnapshotSourceType

TRTCSnapshotSourceType

本地视频截图的数据源

SDK 支持从如下两种数据源中截取图片并保存成本地文件：

- 视频流：从视频流中截取原生的视频内容，截取的内容不受渲染控件的显示控制。
- 渲染层：从渲染控件中截取显示的视频内容，可以做到用户所见即所得的效果，但如果显示区域过小，截取出的图片也会很小。

枚举	取值	描述
TRTCSnapshotSourceTypeStream	0	从视频流中截取原生的视频内容，截取的内容不受渲染控件的显示控制。
TRTCSnapshotSourceTypeView	1	从渲染控件中截取显示的视频内容，可以做到用户所见即所得的效果，但如果显示区域过小，截取出的图片也会很小。

TRTCAppScene

TRTCAppScene

应用场景

TRTC 针对常见的音视频应用场景都进行了定向优化，以满足各种垂直场景下的差异化要求，主要场景可以分为如下两类：

- 直播（LIVE）场景：包括 LIVE 和 VoiceChatRoom，前者是音频+视频，后者是纯音频。

直播场景下，用户被分成“主播”和“观众”两种角色，单个房间中同时最多支持10万人在线，适合于观众人数众多的直播场景。

- 实时（RTC）场景：包括 VideoCall 和 AudioCall，前者是音频+视频，后者是纯音频。

实时场景下，用户没有角色的差异，但单个房间中同时最多支持 300 人在线，适合于小范围实时通信的场景。

枚举	取值	描述
TRTCAppSceneVideoCall	0	视频通话场景，支持720P、1080P高清画质，单个房间最多支持300人同时在线，最高支持50人同时发言。

		适用于[1对1视频通话]、[300人视频会议]、[在线问诊]、[教育小班课]、[远程面试]等业务场景。
TRTCAppSceneLIVE	1	视频互动直播，支持平滑上下麦，切换过程无需等待，主播延时小于300ms；支持十万级别观众同时播放，播放延时低至1000ms。 适用于[低延时互动直播]、[大班课]、[主播PK]、[视频相亲]、[在线互动课堂]、[远程培训]、[超大型会议]等业务场景。 **注意** 此场景下，您必须通过 TRTCParams 中的 role 字段指定当前用户的角色。
TRTCAppSceneAudioCall	2	语音通话场景，默认采用 SPEECH 音质，单个房间最多支持300人同时在线，最高支持50人同时发言。 适用于 [1对1语音通话]、[300人语音会议]、[语音聊天]、[语音会议]、[在线狼人杀] 等业务场景。
TRTCAppSceneVoiceChatRoom	3	语音互动直播，支持平滑上下麦，切换过程无需等待，主播延时小于 300ms；支持十万级别观众同时播放，播放延时低至 1000ms。 适用于 [语音俱乐部]、[在线K歌房]、[音乐直播间]、[FM电台] 等业务场景。 **注意** 此场景下，您必须通过 TRTCParams 中的 role 字段指定当前用户的角色。

TRTCRoleType

TRTCRoleType

角色

仅适用于直播类场景（即 TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom），把用户区分成两种不同的身份：

- 主播：可以随时发布自己的音视频流，但人数有限制，同一个房间中最多只允许 50 个主播同时发布自己的音视频流。

- 观众：只能观看其他用户的音视频流，要发布音视频流，需要先通过 switchRole 切换成主播，同一个房间中最多能容纳10万观众。

枚举	取值	描述
TRTCRoleAnchor	20	主播：可以随时发布自己的音视频流，但人数有限制，同一个房间中最多只允许 50 个主播同时发布自己的音视频流。
TRTCRoleAudience	21	观众：只能观看其他用户的音视频流，要发布音视频流，需要先通过 switchRole 切换成主播，同一个房间中最多能容纳10万观众。

TRTCQosControlMode(Deprecated)

TRTCQosControlMode(Deprecated)

流控模式（已废弃）

枚举	取值	描述
TRTCQosControlModeClient	0	本地控制，用于 SDK 开发内部调试，客户请勿使用。
TRTCQosControlModeServer	1	云端控制，默认模式，推荐选择。

TRTCVideoQosPreference

TRTCVideoQosPreference

画质偏好

TRTC 在弱网络环境下有两种调控模式：“优先保证画面清晰”或“优先保证画面流畅”，两种模式均会优先保障声音数据的传输。

枚举	取值	描述
TRTCVideoQosPreferenceSmooth	1	流畅优先：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的流畅性，代价就是画面会比较模糊且伴随有较多的马赛克。

TRTCVideoQosPreferenceClear	2	清晰优先（默认值）：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的清晰度，代价就是画面会比较卡顿。

TRTCQuality

TRTCQuality

网络质量

TRTC 会每隔两秒对当前的网络质量进行评估，评估结果为六个等级：Excellent 表示最好，Down 表示最差。

枚举	取值	描述
TRTCQuality_Unknown	0	未定义。
TRTCQuality_Excellent	1	当前网络非常好。
TRTCQuality_Good	2	当前网络比较好。
TRTCQuality_Poor	3	当前网络一般。
TRTCQuality_Bad	4	当前网络较差。
TRTCQuality_Vbad	5	当前网络很差。
TRTCQuality_Down	6	当前网络不满足 TRTC 的最低要求。

TRTCAVStatusType

TRTCAVStatusType

视频状态类型

该枚举类型用于视频状态变化回调接口 onRemoteVideoStatusUpdated ，用于指定当前的视频状态。

枚举	取值	描述

TRTCAVStatusStopped	0	停止播放。
TRTCAVStatusPlaying	1	正在播放。
TRTCAVStatusLoading	2	正在加载。

TRTCAVStatusChangeReason

TRTCAVStatusChangeReason

视频状态变化原因类型

该枚举类型用于视频状态变化回调接口 onRemoteVideoStatusUpdated ，用于指定当前的视频状态原因。

枚举	取值	描述
TRTCAVStatusChangeReasonInternal	0	缺省值。
TRTCAVStatusChangeReasonBufferingBegin	1	网络缓冲。
TRTCAVStatusChangeReasonBufferingEnd	2	结束缓冲。
TRTCAVStatusChangeReasonLocalStarted	3	本地启动视频流播放。
TRTCAVStatusChangeReasonLocalStopped	4	本地停止视频流播放。
TRTCAVStatusChangeReasonRemoteStarted	5	远端视频流开始（或继续）。
TRTCAVStatusChangeReasonRemoteStopped	6	远端视频流停止（或中断）。

TRTCAudioSampleRate

版权所有：腾讯云计算（北京）有限责任公司

TRTCAudioSampleRate

音频采样率

音频采样率用来衡量声音的保真程度，采样率越高保真程度越好，如果您的应用场景有音乐的存在，推荐使用 TRTCAudioSampleRate48000。

枚举	取值	描述
TRTCAudioSampleRate16000	16000	16k 采样率。
TRTCAudioSampleRate32000	32000	32k 采样率。
TRTCAudioSampleRate44100	44100	44.1k 采样率。
TRTCAudioSampleRate48000	48000	48k采样率

TRTCAudioQuality

TRTCAudioQuality

声音音质

TRTC 提供了三种精心校调好的模式，用来满足各种垂直场景下对音质的差异化追求：

- 人声模式（Speech）：适用于以人声沟通为主的应用场景，该模式下音频传输的抗性较强，TRTC 会通过各种人声处理技术保障在弱网络环境下的流畅度最佳。
- 音乐模式（Music）：适用于对声乐要求很苛刻的场景，该模式下音频传输的数据量很大，TRTC 会通过各项技术确保音乐信号在各频段均能获得高保真的细节还原度。
- 默认模式（Default）：介于 Speech 和 Music 之间的档位，对音乐的还原度比人声模式要好，但传输数据量比音乐模式要低很多，对各种场景均有不错的适应性。

枚举	取值	描述
TRTCAudioQualitySpeech	1	人声模式：采样率：16k；单声道；编码码率：16kbps；具备几个模式中最强的网络抗性，适合语音通话为主的场景，比如在线会议，语音通话等。

TRTCAudioQualityDefault	2	默认模式：采样率：48k；单声道；编码码率：50kbps；介于 Speech 和 Music 之间的档位，SDK 默认档位，推荐选择。
TRTCAudioQualityMusic	3	音乐模式：采样率：48k；全频带立体声；编码码率：128kbps；适合需要高保真传输音乐的场景，比如在线K歌、音乐直播等。

TRTCAudioRoute

TRTCAudioRoute

音频路由（即声音的播放模式）

音频路由，即声音是从手机的扬声器还是从听筒中播放出来，因此该接口仅适用于手机等移动端设备。
手机有两个扬声器：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。

- 设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
- 设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。
- 音频路由为有线耳机。
- 音频路由为蓝牙耳机。
- 音频路由为USB专业声卡设备。

枚举	取值	描述
TRTCAudioModeSpeakerphone	0	Speakerphone：使用扬声器播放（即“免提”），扬声器位于手机底部，声音偏大，适合外放音乐。
TRTCAudioModeEarpiece	1	Earpiece：使用听筒播放，听筒位于手机顶部，声音偏小，适合需要保护隐私的通话场景。
TRTCAudioModeWiredHeadset	2	WiredHeadset：使用有线耳机播放。
TRTCAudioModeBluetoothHeadset	3	BluetoothHeadset：使用蓝牙耳机播放。
TRTCAudioModeSoundCard	4	SoundCard：使用 USB 声卡播放。

TRTCReverbType

TRTCReverbType

声音混响模式

该枚举值应用于设定直播场景中的混响模式，常用于秀场直播中。

枚举	取值	描述
TRTCReverbType_0	0	关闭混响。
TRTCReverbType_1	1	KTV。
TRTCReverbType_2	2	小房间。
TRTCReverbType_3	3	大会堂。
TRTCReverbType_4	4	低沉。
TRTCReverbType_5	5	洪亮。
TRTCReverbType_6	6	金属声。
TRTCReverbType_7	7	磁性。

TRTCVoiceChangerType

TRTCVoiceChangerType

变声类型

该枚举值应用于设定直播场景中的变声模式，常用于秀场直播中。

枚举	取值	描述

TRTCVoiceChangerType_0	0	关闭变声。
TRTCVoiceChangerType_1	1	熊孩子。
TRTCVoiceChangerType_2	2	萝莉。
TRTCVoiceChangerType_3	3	大叔。
TRTCVoiceChangerType_4	4	重金属。
TRTCVoiceChangerType_5	5	感冒。
TRTCVoiceChangerType_6	6	外国人。
TRTCVoiceChangerType_7	7	困兽。
TRTCVoiceChangerType_8	8	死肥仔。
TRTCVoiceChangerType_9	9	强电流。
TRTCVoiceChangerType_10	10	重机械。
TRTCVoiceChangerType_11	11	空灵。

TRTCSystemVolumeType

TRTCSystemVolumeType

系统音量类型（仅适用于移动设备）

现代智能手机中一般都具备两套系统音量类型，即“通话音量”和“媒体音量”。

- 通话音量：手机专门为接打电话所设计的音量类型，自带回声抵消（AEC）功能，并且支持通过蓝牙耳机

上的麦克风进行拾音，缺点是音质比较一般。当您通过手机侧面的音量按键下调手机音量时，如果无法将其调至零（也就是无法彻底静音），说明您的手机当前处于通话音量。

- 媒体音量：手机专门为音乐场景所设计的音量类型，无法使用系统的 AEC 功能，并且不支持通过蓝牙耳机的麦克风进行拾音，但具备更好的音乐播放效果。当您通过手机侧面的音量按键下调手机音量时，如果能够将手机音量调至彻底静音，说明您的手机当前处于媒体音量。

SDK 目前提供了三种系统音量类型的控制模式：自动切换模式、全程通话音量模式、全程媒体音量模式。

枚举	取值	描述
TRTCSystemVolumeTypeAuto	0	自动切换模式： 也被称为“麦上通话，麦下媒体”，即主播上麦时使用通话音量，观众不上麦则使用媒体音量，适合在线直播场景。 如果您在 enterRoom 时选择的场景为 TRTCAppSceneLIVE 或 TRTCAppSceneVoiceChatRoom，SDK 会自动使用该模式。
TRTCSystemVolumeTypeMedia	1	全程媒体音量： 通话全程使用媒体音量，并不是非常常用的音量类型，适用于对音质要求比较苛刻的音乐场景中。 如果您的用户大都使用外接设备（比如外接声卡）为主，可以使用该模式，否则请慎用。
TRTCSystemVolumeTypeVOIP	2	全程通话音量： 该方案的优势在于用户在上下麦时音频模块无需切换工作模式，可以做到无缝上下麦，适合于用户需要频繁上下麦的应用场景。 如果您在 enterRoom 时选择的场景为 TRTCAppSceneVideoCall 或 TRTCAppSceneAudioCall，SDK 会自动使用该模式。

TRTCAudioFrameOperationMode

TRTCAudioFrameOperationMode

音频回调数据读写模式

TRTC 提供了两种音频回调数据的操作模式。

- 读写模式（ReadWrite）：可以获取并修改回调的音频数据，默认模式。
- 只读模式（ReadOnly）：仅从回调中获取音频数据。

枚举	取值	描述
TRTCAudioFrameOperationModeReadWrite	0	读写模式：可以获取并修改回调的音频数据。
TRTCAudioFrameOperationModeReadOnly	1	只读模式：仅从回调中获取音频数据。

TRTCLogLevel

TRTCLogLevel

Log 级别

不同的日志等级定义了不同的详实程度和日志数量，推荐一般情况下将日志等级设置为：TRTCLogLevelInfo。

枚举	取值	描述
TRTCLogLevelVerbose	0	输出所有级别的 Log。
TRTCLogLevelDebug	1	输出 DEBUG，INFO，WARNING，ERROR 和 FATAL 级别的 Log。
TRTCLogLevelInfo	2	输出 INFO，WARNING，ERROR 和 FATAL 级别的 Log。
TRTCLogLevelWarn	3	输出WARNING，ERROR 和 FATAL 级别的 Log。
TRTCLogLevelError	4	输出ERROR 和 FATAL 级别的 Log。
TRTCLogLevelFatal	5	仅输出 FATAL 级别的 Log。
TRTCLogLevelNone	6	不输出任何 SDK Log。

TRTCGSensorMode

TRTCGSensorMode

重力感应开关（仅适用于移动端）

枚举	取值	描述
TRTCGSensorMode_Disable	0	不适配重力感应，该模式是桌面平台上的默认值，该模式下，当前用户发布出去的视频画面不受重力感应方向变化的影响。
TRTCGSensorMode_UIAutoLayout	1	适配重力感应，该模式是移动平台上的默认值，该模式下，当前用户发布出去的视频画面会跟随设备的重力感应方向进行相应的调整，同时本地预览画面保持方向不变。SDK 目前支持的一种适配模式是：当手机或 Pad 上下颠倒时，为了保证远端用户看到的画面方向正常，SDK 会自动将发布出去的画面上下旋转180度。如果您的 APP 的界面层开启了重力感应自适应，推荐使用 UIFixLayout 模式。
TRTCGSensorMode_UIFixLayout	2	适配重力感应 该模式下，当前用户发布出去的视频画面会跟随设备的重力感应方向进行相应的调整，同时本地预览画面也会进行相应的旋转适配。 目前支持的一种特性是：当手机或 Pad 上下颠倒时，为了保证远端用户看到的画面方向正常，SDK 会自动将发布出去的画面上下旋转180度。 如果您的 APP 的界面层不支持重力感应自适应，并且希望 SDK 的视频画面能够适配重力感应方向，推荐使用 UIFixLayout 模式。 @deprecated 从v11.5版本开始，不再支持 TRTCGSensorMode_UIFixLayout，只支持上面两种模式。

TRTCScreenCaptureSourceType

TRTCScreenCaptureSourceType

屏幕分享的目标类型（仅适用于桌面端）

枚举	取值	描述

TRTCScreenCaptureSourceTypeUnknown	-1	未定义。
TRTCScreenCaptureSourceTypeWindow	0	该分享目标是某一个应用的窗口。
TRTCScreenCaptureSourceTypeScreen	1	该分享目标是某一台显示器的屏幕。

TRTCTranscodingConfigMode

TRTCTranscodingConfigMode

云端混流的排版模式

TRTC 的云端混流服务能够将房间中的多路音视频流混合成一路，因此您需要指定画面的排版方案，我们提供了如下几种排版模式。

枚举	取值	描述
TRTCTranscodingConfigMode_Unknown	0	未定义。
TRTCTranscodingConfigMode_Manual	1	全手动排版模式。 该模式下，您需要指定每一路画面的精确排版位置。该模式的自由度最高，但易用性也最差： • 您需要填写 TRTCTranscodingConfig 中的所有参数，包括每一路画面（TRTCMixUser）的位置坐标。 • 您需要监听 TRTCCloudDelegate 中的 onUserVideoAvailable() 和 onUserAudioAvailable() 事件回调，并根据当前房间中各个麦上用户的音视频状态不断地调整 mixUsers 参数。
TRTCTranscodingConfigMode_Template_PureAudio	2	纯音频模式。 该模式适用于语音通话（AudioCall）和语音聊天室

		（VoiceChatRoom）等纯音频的应用场景。 • 您只需要在进入房间后，通过 setMixTranscodingConfig 接口设置一次，之后 SDK 就会自动把房间内所有上麦用户的声音混流到当前用户的直播流上。 • 您无需设置 TRTCTranscodingConfig 中的 mixUsers 参数，只需设置 audioSampleRate、audioBitrate 和 audioChannels 等参数即可。
TRTCTranscodingConfigMode_Template_PresetLayout	3	预排版模式。 最受欢迎的排版模式，因为该模式支持您通过占位符提前对各路画面的位置进行设定，之后 SDK 会自动根据房间中画面的路数动态进行适配调整。 此模式下，您依然需要设置 mixUsers 参数，但可以将 userId 设置为“占位符”，可选的占位符有： • "$PLACE_HOLDER_REMOTE$" ：指代远程用户的画面，可以设置多个。 • "$PLACE_HOLDER_LOCAL_MAIN$"： 指代本地摄像头画面，只允许设置一个。 • "$PLACE_HOLDER_LOCAL_SUB$"：指代本地屏幕分享画面，只允许设置一个。 此模式下，您不需要监听 TRTCCloudDelegate 中的 onUserVideoAvailable 和 onUserAudioAvailable 回调进行实时调整，只需要在进房成功后调用一次 setMixTranscodingConfig 即可，之后 SDK 会自动将真实的 userId 补位到您设置的占位符上。

TRTCTranscodingConfigMode_Template_ScreenSharing	4	屏幕分享模式。 适用于在线教育场景等以屏幕分享为主的应用场景，仅支持 Windows 和 Mac 两个平台的 SDK。 该模式下，SDK 会先根据您通过 videoWidth 和 videoHeight 参数设置的目标分辨率构建一张画布， • 当老师未开启屏幕分享时，SDK 会将老师的摄像头画面等比例拉伸绘制到该画布上； • 当老师开启屏幕分享之后，SDK 会将屏幕分享画面绘制到同样的画布上。 此种排版模式的目的是为了确保混流模块的输出分辨率一致，避免课程回放和网页观看的花屏问题（网页播放器不支持可变分辨率）。 同时，连麦学生的声音也会被默认混合到老师的音视频流中。 由于教学模式下的视频内容以屏幕分享为主，因此同时传输摄像头画面和屏幕分享画面是非常浪费带宽的。 推荐的做法是直接将摄像头画面通过 setLocalVideoRenderCallback 接口自定义绘制到当前屏幕上。 在该模式下，您无需设置 TRTCTranscodingConfig 中的 mixUsers 参数，SDK 不会混合学生的画面，以免干扰屏幕分享的效果。 您可以将 TRTCTranscodingConfig 中的 width x height 设为 0px × 0px，SDK 会自动根据用户当前屏幕的宽高比计算出一个合适的分辨率： • 如果老师当前屏幕宽度 <= 1920px，SDK 会使用老师当前屏幕的实际分辨率。 • 如果老师当前屏幕宽度 > 1920px，SDK 会根据当前屏幕宽高比，选择 1920x1080(16:9)、1920x1200(16:10)、

		1920x1440(4:3) 三种分辨率中的一种。

TRTCRecordType

TRTCRecordType

媒体录制类型

该枚举类型用于本地媒体录制接口 startLocalRecording，用于指定是录制音视频文件还是纯音频文件。

枚举	取值	描述
TRTCRecordTypeAudio	0	仅录制音频。
TRTCRecordTypeVideo	1	仅录制视频。
TRTCRecordTypeBoth	2	同时录制音频和视频。

TRTCMixInputType

TRTCMixInputType

混流输入类型

枚举	取值	描述
TRTCMixInputTypeUndefined	0	默认值，考虑到针对老版本的兼容性，如果您指定了 inputType 为 Undefined，SDK 会根据另一个参数 pureAudio 的数值决定混流输入类型。
TRTCMixInputTypeAudioVideo	1	混入音频和视频。
TRTCMixInputTypePureVideo	2	只混入视频。

TRTCMixInputTypePureAudio	3	只混入音频。
TRTCMixInputTypeWatermark	4	混入水印，此时您无需指定 userId 字段，但需要指定 image 字段，推荐使用 png 格式的图片。

TRTCAudioRecordingContent

TRTCAudioRecordingContent

音频录制内容类型

该枚举类型用于音频录制接口 startAudioRecording ，用于指定录制音频的内容。

枚举	取值	描述
TRTCAudioRecordingContentAll	0	录制本地和远端所有音频。
TRTCAudioRecordingContentLocal	1	仅录制本地音频。
TRTCAudioRecordingContentRemote	2	仅录制远端音频。

TRTCPublishMode

TRTCPublishMode

媒体流发布模式

该枚举类型用于媒体流发布接口 startPublishMediaStream TRTC 的媒体流发布服务能够将房间中的多路音视频流混合成一路发布至 CDN 或者回推到房间内，也可以将您当前的这路音视频发布到腾讯或者第三方 CDN 因此您需要指定对应媒体流的发布模式，我们提供了如下几种模式。

枚举	取值	描述
TRTCPublishModeUnknown	0	未定义。

TRTCPublishBigStreamToCdn	1	您可以通过设置该参数将您房间内的主路流（TRTCVideoStreamTypeBig）发布到腾讯或者第三方直播 CDN 服务商（仅支持标准 RTMP 协议）。
TRTCPublishSubStreamToCdn	2	您可以通过设置该参数将您房间内的辅路流（TRTCVideoStreamTypeSub）发布到腾讯或者第三方直播 CDN 服务商（仅支持标准 RTMP 协议）。
TRTCPublishMixStreamToCdn	3	您可以通过设置该参数，配合编码输出参数（TRTCStreamEncoderParam）和混流转码参数（TRTCStreamMixingConfig），将您指定的多路音视频流进行转码并发布到腾讯或者第三方直播 CDN 服务商（仅支持标准 RTMP 协议）。
TRTCPublishMixStreamToRoom	4	您可以通过设置该参数，配合媒体流编码输出参数（TRTCStreamEncoderParam）和混流转码参数（TRTCStreamMixingConfig），将您指定的多路音视频流进行转码并发布到您指定的房间中。 • 通过 TRTCPublishTarget 中的 TRTCUser 进行指定回推房间的机器人信息。

TRTCEncryptionAlgorithm

TRTCEncryptionAlgorithm

加密算法

该枚举类型用于媒体流私有加密算法选择。

枚举	取值	描述
TRTCEncryptionAlgorithmAes128Gcm	0	AES GCM 128。
TRTCEncryptionAlgorithmAes256Gcm	1	AES GCM 256。

TRTCSpeedTestScene

TRTCSpeedTestScene

测速场景

该枚举类型用于测速场景选择。

枚举	取值	描述
TRTCSpeedTestScene_DelayTesting	1	延迟测试。
TRTCSpeedTestScene_DelayAndBandwidthTesting	2	延迟与带宽测试。
TRTCSpeedTestScene_OnlineChorusTesting	3	在线合唱测试。

TRTCGravitySensorAdaptiveMode

TRTCGravitySensorAdaptiveMode

设置重力感应的适配模式（仅适用于移动端）

枚举	取值	描述
TRTCGravitySensorAdaptiveMode_Disable	0	关闭重力感应，根据当前采集分辨率与设置的编码分辨率决策，如果两者不一致，则通过旋转90度，保证最大画幅。
TRTCGravitySensorAdaptiveMode_FillByCenterCrop	1	开启重力感应，始终保证远端画面图像为正，中间过程需要处理分辨率不一致时，采用居中裁剪模式。
TRTCGravitySensorAdaptiveMode_FitWithBlackBorder	2	开启重力感应，始终保证远端画面图像为正，中间过程需要处理分辨率不一致时，采用叠加黑边模式。

TRTCParams

TRTCParams

进房参数

作为 TRTC SDK 的进房参数，只有该参数填写正确，才能顺利进入 roomId 或者 strRoomId 所指定的音视频房间。
由于历史原因，TRTC 支持数字和字符串两种类型的房间号，分别是 roomId 和 strRoomId。
请注意：不要混用 roomId 和 strRoomId，因为它们之间是不互通的，比如数字 123 和字符串 “123” 在 TRTC 看来是两个完全不同的房间。

枚举类型	描述
bussInfo	【字段含义】业务数据字段（选填），部分高级特性才需要用到此字段。 【推荐取值】请不要自行设置该字段。
privateMapKey	【字段含义】用于权限控制的权限票据（选填），当您希望某个房间只能让特定的 userId 进入时，需要使用 privateMapKey 进行权限保护。 【推荐取值】仅建议有高级别安全需求的客户使用，更多详情请参见 进房权限保护 。
role	【字段含义】直播场景下的角色，仅适用于直播场景（ TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom ），通话场景下指定该参数是无效的。 【推荐取值】默认值：主播（ TRTCRoleAnchor ）。
roomId	【字段含义】数字房间号，在同一个房间里的用户（userId）可以彼此看到对方并进行音视频通话。 【推荐取值】取值范围：1 - 4294967294。 【特别说明】roomId 与 strRoomId 是互斥的，若您选用 strRoomId，则 roomId 需要填写为 0。若两者都填，SDK 将优先选用 roomId。 【请您注意】不要混用 roomId 和 strRoomId，因为它们之间是不互通的，比如数字 123 和字符串 `123` 在 TRTC 看来是两个完全不同的房间。
sdkAppId	【字段含义】应用标识（必填），腾讯云基于 sdkAppId 完成计费统计。 【推荐取值】在 实时音视频控制台 创建应用后可以在账号信息页面中得到该 ID。
strRoomId	【字段含义】字符串房间号，在同一个房间里的用户（userId）可以彼此看到对方并进行音视频通话。 【特别说明】roomId 与 strRoomId 是互斥的，若您选用 strRoomId，则 roomId 需要填写为0。若两者都填，SDK 将优先选用 roomId。 【请您注意】不要混用 roomId 和 strRoomId，因为它们之间是不互通的，比如数字 123 和字符串 `123` 在 TRTC 看来是两个完全不同的房间。 【推荐取值】限制长度为 64 字节。以下为支持的字符集范围（共 89 个字符）： • 大小写英文字母（a-zA-Z）；

	• 数字（0-9）； • 空格、`!`、`#`、`$`、`%`、`&`、`(`、`)`、`+`、`-`、`:`、`;`、`<`、`=`、`.`、`>`、`?`、`@`、`[`、`]`、`^`、`_`、`{`、`}`、`|`、`~`、`,`。
streamId	【字段含义】用于指定在腾讯云直播平台上的 streamId（选填），设置之后，您可以在腾讯云直播 CDN 上通过标准拉流方案（FLV 或 HLS）播放该用户的音视频流。 【推荐取值】限制长度为64字节，可以不填写，一种推荐的方案是使用 `sdkappid_roomid_userid_main` 作为 streamid，这种命名方式容易辨认且不会在您的多个应用中发生冲突。 【特殊说明】要使用腾讯云直播 CDN，您需要先在 控制台 中的功能配置页开启“启动自动旁路直播”开关。 【参考文档】CDN 旁路直播 。
userDefineRecordId	【字段含义】云端录制开关（选填），用于指定是否要在云端将该用户的音视频流录制下来。 【参考文档】云端录制 。 【推荐取值】限制长度为64字节，只允许包含大小写英文字母（a-zA-Z）、数字（0-9）及下划线和连词符。 方案一：手动录制方案： 1. 在 控制台>应用管理>云端录制配置 中开启云端录制。 2. 设置 `录制形式` 为 `手动录制` 。 3. 设置手动录制后，在一个 TRTC 房间中只有设置了 userDefineRecordId 参数的用户才会在云端录制出视频文件，不指定该参数的用户不会产生录制行为。 4. 云端会以 “userDefineRecordId_起始时间_结束时间” 的格式命名录制下来的文件。 方案二：自动录制方案： 1. 需要在 控制台>应用管理>云端录制配置 中开启云端录制。 2. 设置 `录制形式` 为 `自动录制` 。 3. 设置自动录制后，在一个 TRTC 房间中的任何一个有音视频上行的用户，均会在云端录制出视频文件。 4. 文件会以 `userDefineRecordId_起始时间_结束时间` 的格式命名，如果不指定 userDefineRecordId，则文件会以 `streamId_起始时间_结束时间` 命名。
userId	【字段含义】用户标识（必填），当前用户的 userId，相当于用户名，使用 UTF-8 编码。 【推荐取值】如果一个用户在您的账号系统中的 ID 为“mike”，则 userId 即可设置为“mike”。
userSig	【字段含义】用户签名（必填），当前 userId 对应的验证签名，相当于使用云服务的登录密码。 【推荐取值】具体计算方法请参见 如何计算UserSig 。

TRTCVideoEncParam

TRTCVideoEncParam

视频编码参数

该设置决定远端用户看到的画面质量，同时也决定了云端录制出的视频文件的画面质量。

枚举类型	描述
enableAdjustRes	【字段含义】是否允许动态调整分辨率（开启后会对云端录制产生影响）。 【推荐取值】该功能适用于不需要云端录制的场景，开启后 SDK 会根据当前网络情况，智能选择出一个合适的分辨率，避免出现“大分辨率+小码率”的低效编码模式。 【特别说明】默认值：关闭。如有云端录制的需求，请不要开启此功能，因为如果视频分辨率发生变化后，云端录制出的 MP4 在普通的播放器上无法正常播放。
minVideoBitrate	【字段含义】最低视频码率，SDK 会在网络不佳的情况下主动降低视频码率以保持流畅度，最低会降至 minVideoBitrate 所设定的数值。 【特别说明】默认值：0，此时最低码率由 SDK 会根据您指定的分辨率，自动计算出合适的数值。 【推荐取值】您可以通过同时设置 videoBitrate 和 minVideoBitrate 两个参数，用于约束 SDK 对视频码率的调整范围： • 如果您追求 弱网络下允许卡顿但要保持清晰 的效果，可以设置 minVideoBitrate 为 videoBitrate 的 60%。 • 如果您追求 弱网络下允许模糊但要保持流畅 的效果，可以设置 minVideoBitrate 为一个较低的数值（比如 100kbps）。 • 如果您将 videoBitrate 和 minVideoBitrate 设置为同一个值，等价于关闭 SDK 对视频码率的自适应调节能力。
resMode	【字段含义】分辨率模式（横屏分辨率 or 竖屏分辨率）。 【推荐取值】手机平台（iOS、Android）建议选择 Portrait，桌面平台（Windows、Mac）建议选择 Landscape。 【特别说明】如需使用竖屏分辨率，请指定 resMode 为 Portrait，例如：640 × 360 + Portrait = 360 × 640。
videoBitrate	【字段含义】目标视频码率，SDK 会按照目标码率进行编码，只有在弱网络环境下才会主动降低视频码率。 【推荐取值】请参见本 TRTCVideoResolution 在各档位注释的最佳码率，也可以在此基础上适当调高。比如：TRTCVideoResolution_1280_720 对

	应 1200kbps 的目标码率，您也可以设置为 1500kbps 用来获得更好的观感清晰度。 【特别说明】您可以通过同时设置 videoBitrate 和 minVideoBitrate 两个参数，用于约束 SDK 对视频码率的调整范围： • 如果您追求“弱网络下允许卡顿但要保持清晰”的效果，可以设置 minVideoBitrate 为 videoBitrate 的 60%。 • 如果您追求“弱网络下允许模糊但要保持流畅”的效果，可以设置 minVideoBitrate 为一个较低的数值（比如 100kbps）。 • 如果您将 videoBitrate 和 minVideoBitrate 设置为同一个值，等价于关闭 SDK 对视频码率的自适应调节能力。
videoFps	【字段含义】视频采集帧率。 【推荐取值】15fps或20fps。5fps以下，卡顿感明显。10fps以下，会有轻微卡顿感。20fps以上，会浪费带宽（电影的帧率为24fps）。 【特别说明】部分 Android 手机的前置摄像头并不支持15fps以上的采集帧率，部分主打美颜功能的 Android 手机的前置摄像头的采集帧率可能低于10fps。
videoResolution	【字段含义】视频分辨率。 【特别说明】如需使用竖屏分辨率，请指定 resMode 为 Portrait，例如：640 × 360 + Portrait = 360 × 640。 【推荐取值】 • 手机视频通话：建议选择 360 × 640 及以下分辨率，resMode 选择 Portrait，即竖屏分辨率。 • 手机在线直播：建议选择 540 × 960，resMode 选择 Portrait，即竖屏分辨率。 • 桌面平台（Win + Mac）：建议选择 640 × 360 及以上分辨率，resMode 选择 Landscape，即横屏分辨率。

TRTCNetworkQosParam

TRTCNetworkQosParam

网络流控（Qos）参数集

网络流控相关参数，该设置决定 SDK 在弱网络环境下的调控策略（例如：“清晰优先”或“流畅优先”）。

枚举类型	描述
controlMode	【字段含义】流控模式（已废弃）。 【推荐取值】云端控制。 【特别说明】请设置为云端控制模式（TRTCQosControlModeServer）。

preference	【字段含义】清晰优先还是流畅优先。 【推荐取值】清晰优先。 【特别说明】该参数主要影响 TRTC 在较差网络环境下的音视频表现： • 流畅优先：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的流畅性，代价就是画面会比较模糊且伴随有较多的马赛克。参见 TRTCVideoQosPreferenceSmooth • 清晰优先（默认值）：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的清晰度，代价就是画面会比较卡顿。参见 TRTCVideoQosPreferenceClear

TRTCRenderParams

TRTCRenderParams

视频画面的渲染参数

您可以通过设置此参数来控制画面的旋转角度、填充模式和左右镜像模式。

枚举类型	描述
fillMode	【字段含义】画面填充模式。 【推荐取值】填充（画面可能会被拉伸裁剪）或适应（画面可能会有黑边），默认值：TRTCVideoFillMode_Fill。
mirrorType	【字段含义】画面镜像模式。 【推荐取值】默认值：TRTCVideoMirrorTypeAuto。
rotation	【字段含义】图像的顺时针旋转角度。 【推荐取值】支持 90、180 以及 270 旋转角度，默认值：TRTCVideoRotation_0。

TRTCQualityInfo

TRTCQualityInfo

网络质量

表征网络质量的好坏，您可以通过该数值在用户界面上展示每个用户的网络质量。

枚举类型	描述
quality	网络质量
userId	用户 ID

TRTCVolumeInfo

TRTCVolumeInfo

音量大小

表征语音音量的评估值，您可以通过该数值在用户界面上展示每个用户的音量大小。

枚举类型	描述
pitch	本地用户的人声频率（单位：Hz），取值范围[0 – 4000]，对于远端用户，该值始终为0。
spectrumData	音频频谱数据是将音频数据在频率域中的分布，划分为 256 个频率段，使用 spectrumData 记录各个频率段的能量值，每个能量值的取值范围为 [−300, 0]，单位为 dBFS。 **注意** 本地频谱使用编码前的音频数据计算，会受到本地采集音量、BGM等影响；远端频谱使用接收到的音频数据计算，本地调整远端播放音量等操作不会对其产生影响。
userId	说话者的 userId, 如果 userId 为空则代表是当前用户自己。
vad	是否检测到人声，0：非人声 1：人声。
volume	说话者的音量大小, 取值范围[0 – 100]。

TRTCSpeedTestParams

TRTCSpeedTestParams

测速参数

您可以在用户进入房间前通过 startSpeedTest 接口测试网速（注意：请不要在通话中调用）。

枚举类型	描述
expectedDownBandwidth	预期的下行带宽（kbps，取值范围： 10 ~ 5000，为 0 时不测试）。 **注意** 当参数 scene 设置为 TRTCSpeedTestScene_OnlineChorusTesting 时，为了获取更准确的 rtt / jitter 等信息，取值范围限制为 10 ~ 1000。
expectedUpBandwidth	预期的上行带宽（kbps，取值范围： 10 ~ 5000，为 0 时不测试）。 **注意** 当参数 scene 设置为 TRTCSpeedTestScene_OnlineChorusTesting 时，为了获取更准确的 rtt / jitter 等信息，取值范围限制为 10 ~ 1000。
scene	测速场景。
sdkAppId	应用标识，请参见 TRTCParams 中的相关说明。
userId	用户标识，请参见 TRTCParams 中的相关说明。
userSig	用户签名，请参见 TRTCParams 中的相关说明。

TRTCSpeedTestResult

TRTCSpeedTestResult

网络测速结果

您可以在用户进入房间前通过 startSpeedTest 接口进行测速（注意：请不要在通话中调用）。

枚举类型	描述
availableDownBand	下行带宽（kbps，-1：无效值）。

width	
availableUpBandwidth	上行带宽（kbps，-1：无效值）。
downJitter	下行数据包抖动（ms），指用户当前网络环境下数据通信的稳定性，该值越小越好，正常数值范围是0ms - 100ms，-1 代表此次测速没有成功测得有效值，一般情况下 WiFi 网络的 Jitter 会比 4G/5G 环境稍大。
downLostRate	下行丢包率，取值范围是 [0 - 1.0]，例如 0.2 表示每从服务器收取 10 个数据包可能会在中途丢失 2 个。
errMsg	带宽测试错误信息。
ip	服务器 IP 地址。
quality	内部通过评估算法测算出的网络质量，更多信息请参见 TRTCQuality 。
rtt	延迟（毫秒），指当前设备到 TRTC 服务器的一次网络往返时间，该值越小越好，正常数值范围是10ms - 100ms。
success	测试是否成功。
upJitter	上行数据包抖动（ms），指用户当前网络环境下数据通信的稳定性，该值越小越好，正常数值范围是0ms - 100ms，-1 代表此次测速没有成功测得有效值，一般情况下 WiFi 网络的 Jitter 会比 4G/5G 环境稍大。
upLostRate	上行丢包率，取值范围是 [0 - 1.0]，例如 0.3 表示每向服务器发送 10 个数据包可能会在中途丢失 3 个。

TRTCVideoFrame

TRTCVideoFrame

视频帧信息

TRTCVideoFrame 用来描述一帧视频画面的裸数据，也就是编码前或者解码后的视频画面数据。

枚举类型	描述
bufferType	【字段含义】视频数据结构类型。
data	【字段含义】bufferType 为 TRTCVideoBufferType_NSData 时的视频数据，承载 NSData 类型的内存数据块。

height	【字段含义】视频高度。
pixelBuffer	【字段含义】bufferType 为 TRTCVideoBufferType_PixelBuffer 时的视频数据，承载 iOS 平台专用的 PixelBuffer。
pixelFormat	【字段含义】视频的像素格式。
rotation	【字段含义】视频像素的顺时针旋转角度。
textureId	【字段含义】视频纹理 ID,bufferType 为 TRTCVideoBufferType_Texture 时的视频数据，承载用于 OpenGL 渲染的纹理数据。
timestamp	【字段含义】视频帧的时间戳，单位毫秒。 【推荐取值】自定义视频采集时可以设置为0。若该参数为0，SDK 会自定填充 timestamp 字段，但请“均匀”地控制 sendCustomVideoData 的调用间隔。
width	【字段含义】视频宽度。

TRTCAudioFrame

TRTCAudioFrame

音频帧数据

枚举类型	描述
channels	【字段含义】声道数。
data	【字段含义】音频数据。
extraData	【字段含义】音频额外数据，远端用户通过 `onLocalProcessedAudioFrame` 写入的数据会通过该字段回调。
sampleRate	【字段含义】采样率。
timestamp	【字段含义】时间戳，单位ms。

TRTCMixUser

TRTCMixUser

云端混流中各路画面的描述信息

TRTCMixUser 用于指定云端混流中每一路视频画面的位置、大小、图层以及流类型等信息。

枚举类型	描述
image	【字段含义】占位图或水印图 • 占位图是指当对应 userId 混流内容为纯音频时，混合后的画面中显示的是占位图片。 • 水印图是指一张贴在混合后画面中的半透明图片，这张图片会一直覆盖于混合后的画面上。 • 当指定 inputType 为 TRTCMixInputTypePureAudio 时，image 为占位图，此时需要您指定 userId。 • 当指定 inputType 为 TRTCMixInputTypeWatermark 时，image 为水印图，此时不需要您指定 userId。 【推荐取值】默认值：空值，即不设置占位图或者水印图。 【特别说明】 • 您可以将 image 设置为控制台中的某一个素材 ID，这需要您事先在 “控制台 => 应用管理 => 功能配置 => 素材管理” 中单击 [新增图片] 按钮进行上传。 • 上传成功后可以获得对应的“图片ID”，然后将“图片ID”转换成字符串类型并设置给 image 字段即可（比如假设“图片ID” 为 63，可以设置 image = @"63"）。 • 您也可以将 image 设置为图片的 URL 地址，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式，推荐使用 png 格式的半透明图片作为水印。 • image 仅在 inputType 为 TRTCMixInputTypePureAudio 或者 TRTCMixInputTypeWatermark 时才生效。
inputType	【字段含义】指定该路流的混合内容（只混音频、只混视频、混合音视频、混入水印）。 【默认取值】默认值：TRTCMixInputTypeUndefined。 【特别说明】 • 当指定 inputType 为 TRTCMixInputTypeUndefined 并设置 pureAudio 为 YES 时，等效于设置 inputType 为 TRTCMixInputTypePureAudio。 • 当指定 inputType 为 TRTCMixInputTypeUndefined 并设置 pureAudio 为 NO 时，等效于设置 inputType 为 TRTCMixInputTypeAudioVideo。 • 当指定 inputType 为 TRTCMixInputTypeWatermark 时，您可以不指定 userId 字段，但需要指定 image 字段。

pureAudio	【字段含义】指定该路流是不是只混合声音。 【推荐取值】默认值：NO。 【特别说明】已废弃，推荐使用8.5版本开始新引入的字段：inputType。
rect	【字段含义】指定该路画面的坐标区域（单位：像素）。
renderMode	【字段含义】该画面在输出时的显示模式。 【推荐取值】默认值：视频流默认为0。0为裁剪，1为缩放，2为缩放并显示黑底。 【特别说明】水印图和占位图暂时不支持设置 renderMode，默认强制拉伸处理。
roomID	【字段含义】该路音视频流所在的房间号（设置为空值代表当前用户所在的房间号）。
soundLevel	【字段含义】该路音频参与混音时的音量等级（取值范围：0 - 100）。 【默认取值】默认值：100。
streamType	【字段含义】指定该路画面是主路画面（TRTCVideoStreamTypeBig）还是辅路画面（TRTCVideoStreamTypeSub）。
userId	【字段含义】用户 ID
zOrder	【字段含义】指定该路画面的层级（取值范围：1 - 15，不可重复）。

TRTCTranscodingConfig

TRTCTranscodingConfig

云端混流的排版布局和转码参数

用于指定混流时各路画面的排版位置信息和云端转码的编码参数。

枚举类型	描述
appId	【字段含义】腾讯云直播服务的 AppID。 【推荐取值】请在 实时音视频控制台 依次单击【应用管理】=>【应用信息】，并在【旁路直播信息】中获取 appid。
audioBitrate	【字段含义】指定云端转码的目标音频码率。 【推荐取值】默认值：64kbps，取值范围是 [32，192]。
audioChannels	【字段含义】指定云端转码的音频声道数。 【推荐取值】默认值：1，代表单声道。可设定的数值只有两个数字：1-单声道，2-双声道。

audioCodec	【字段含义】指定云端转码的输出流音频编码类型。 【推荐取值】默认值：0，代表LC-AAC。可设定的数值只有三个数字：0 - LC-AAC，1 - HE-AAC，2 - HE-AACv2。 【特别说明】HE-AAC 和 HE-AACv2 支持的输出流音频采样率范围为 [48000, 44100, 32000, 24000, 16000]。 【特别说明】当音频编码设置为 HE-AACv2 时，只支持输出流音频声道数为双声道。 【特别说明】HE-AAC 和 HE-AACv2 取值仅在输出流为您额外设置的 streamId 上时才生效。
audioSampleRate	【字段含义】指定云端转码的目标音频采样率。 【推荐取值】默认值：48000Hz。支持12000HZ、16000HZ、22050HZ、24000HZ、32000HZ、44100HZ、48000HZ。
backgroundColor	【字段含义】指定混合画面的底色颜色。 【推荐取值】默认值：0x000000 代表黑色。格式为十六进制数字，比如：“0x61B9F1” 代表 RGB 分别为(97,158,241)。
backgroundImage	【字段含义】指定混合画面的背景图片。 【推荐取值】默认值：空值，即不设置背景图片。 【特别说明】 • 您可以将 image 设置为控制台中的某一个素材 ID，这需要您事先在 “控制台 => 应用管理 => 功能配置 => 素材管理” 中单击 [新增图片] 按钮进行上传。 • 上传成功后可以获得对应的“图片ID”，然后将“图片ID”转换成字符串类型并设置给 image 字段即可（比如假设“图片ID” 为 63，可以设置 image = @"63"）。 • 您也可以将 image 设置为图片的 URL 地址，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式。
bizId	【字段含义】腾讯云直播服务的 bizid。 【推荐取值】请在 实时音视频控制台 依次单击【应用管理】=>【应用信息】，并在【旁路直播信息】中获取 bizid。
mixUsers	【字段含义】指定云端混流中每一路视频画面的位置、大小、图层以及流类型等信息。 【推荐取值】该字段是一个 TRTCMixUser 类型的数组，数组中的每一个元素都用来代表每一路画面的信息。
mode	【字段含义】排版模式。 【推荐取值】请根据您的业务场景要求自行选择，预排版模式是适用性较好的一种模式。
streamId	【字段含义】输出到 CDN 上的直播流 ID。

	【推荐取值】默认值：空值，即房间里的多路音视频流最终会混合到接口调用者的那一路音视频流上。 • 如不设置该参数，SDK 会执行默认逻辑，即房间里的多路音视频流会混合到该接口调用者的那一路音视频流上，也就是 A + B => A。 • 如您设置该参数，SDK 会将房间里的多路音视频流混合到您指定的直播流上，也就是 A + B => C（C 代表您指定的 streamId）。
videoBitrate	【字段含义】指定云端转码的目标视频码率（kbps）。 【推荐取值】如果填0，TRTC 会根据 videoWidth 和 videoHeight 估算出一个合理的码率值，您也可以参考视频分辨率枚举定义中所推荐的码率值（见注释部分）。
videoFramerate	【字段含义】指定云端转码的目标视频帧率（FPS）。 【推荐取值】默认值：15fps，取值范围是 (0,30]。
videoGOP	【字段含义】指定云端转码的目标视频关键帧间隔（GOP）。 【推荐取值】默认值：2，单位为秒，取值范围是 [1,8]。
videoHeight	【字段含义】指定云端转码的目标分辨率（高度）。 【推荐取值】单位：像素值，推荐值：640，如果你只混合音频流，请将 width 和 height 均设置为 0，否则混流转码后的直播流中会有黑色背景。

videoSeiParams	【字段含义】混流 SEI 参数，默认不填写。
	【特别说明】参数以 json 字符串形式传入，示例如下：
	<pre>{
	"payLoadContent":"xxx",
	"payloadType":5,
	
	"payloadUuid":"1234567890abcdef1234567890abcdef",
	"interval":1000,
	"followIdr":false
	}</pre>
	当前支持的字段及含义：
	• payloadContent: 必填。透传 sei 的 payload 内容，不能为空；
	• payloadType: 必填。sei 消息的类型，取值范围: 5 或 [100, 254] 范围内的整数（244 除外，244 是内部自定义的时间戳 sei）；
	• payloadUuid: 当 payloadType 为 5 时必须填写，其他情况下该值会被忽略。该值必须是长度为 32 的十六进制数字；
	• interval: 选填，默认 1000。sei 的发送间隔，单位毫秒；
	• followIdr: 选填，默认 false。该值为 true 时，发送关键帧时会确保带 sei，否则不确保。

videoWidth	【字段含义】指定云端转码的目标分辨率（宽度）。

	【推荐取值】单位：像素值，推荐值：360，如果你只混合音频流，请将 width 和 height 均设置为 0，否则混流转码后的直播流中会有黑色背景。

TRTCPublishCDNParam

TRTCPublishCDNParam

向非腾讯云 CDN 上发布音视频流时需设置的转推参数

TRTC 的后台服务支持通过标准 RTMP 协议，将其中的音视频流发布到第三方直播 CDN 服务商。
如果您使用腾讯云直播 CDN 服务，可无需关注此参数，直接使用 startPublish 接口即可。

枚举类型	描述
appId	【字段含义】腾讯云直播服务的 AppID。 【推荐取值】请在 实时音视频控制台 依次单击【应用管理】=>【应用信息】，并在【旁路直播信息】中获取 appid。
bizId	【字段含义】腾讯云直播服务的 bizid。 【推荐取值】请在 实时音视频控制台 依次单击【应用管理】=>【应用信息】，并在【旁路直播信息】中获取 bizid。
streamId	【字段含义】需要转推的 streamId。 【推荐取值】默认值：空值。如果不填写，则默认转推调用者的旁路流。
url	【字段含义】指定该路音视频流在第三方直播服务商的推流地址（RTMP 格式）。 【推荐取值】各家服务商的推流地址规则差异较大，请根据目标服务商的要求填写合法的推流 URL，TRTC 的后台服务器会按照您填写的 URL 向第三方服务商推送标准格式音视频流。 【特别说明】推流 URL 必须为 RTMP 格式，必须符合您的目标直播服务商的规范要求，否则目标服务商会拒绝来自 TRTC 后台服务的推流请求。

TRTCAudioRecordingParams

TRTCAudioRecordingParams

本地音频文件的录制参数

该参数用于在音频录制接口 startAudioRecording 中指定录制参数。

枚举类型	描述
filePath	【字段含义】录音文件的保存路径（必填）。 【特别说明】该路径需精确到文件名及格式后缀，格式后缀用于决定录音文件的格式，目前支持的格式有 PCM、WAV 和 AAC。 例如：假如您指定路径为 "mypath/record/audio.aac"，代表您希望 SDK 生成一个 AAC 格式的音频录制文件。 请您指定一个有读写权限的合法路径，否则录音文件无法生成。
maxDurationPerFile	【字段含义】maxDurationPerFile 录制文件分片时长，单位毫秒，最小值10000。默认值为0，表示不分片。
recordingContent	【字段含义】音频录制内容类型。 【特别说明】默认录制所有本地和远端音频。

TRTCLocalRecordingParams

TRTCLocalRecordingParams

本地媒体文件的录制参数

该参数用于在本地媒体文件的录制接口 startLocalRecording 中指定录制相关参数。
接口 startLocalRecording 是接口 startAudioRecording 的能力加强版本，前者可以录制视频文件，后者只能录制音频文件。

枚举类型	描述
filePath	【字段含义】录制的文件地址（必填），请确保路径有读写权限且合法，否则录制文件无法生成。 【特别说明】该路径需精确到文件名及格式后缀，格式后缀用于决定录制出的文件格式，目前支持的格式暂时只有 MP4。 例如：假如您指定路径为 "mypath/record/test.mp4"，代表您希望 SDK 生成一个 MP4 格式的本地视频文件。 请您指定一个有读写权限的合法路径，否则录制文件无法生成。
interval	【字段含义】interval 录制信息更新频率，单位毫秒，有效范围：1000-10000。默认值为-1，表示不回调。
maxDurationPerFile	【字段含义】maxDurationPerFile 录制文件分片时长，单位毫秒，最小值10000。默认值为0，表示不分片。
recordType	【字段含义】媒体录制类型，默认值：TRTCRecordTypeBoth，即同时录制音频和视频。

TRTCSwitchRoomConfig

TRTCSwitchRoomConfig

房间切换参数

该参数用于切换房间接口 switchRoom ，可以让用户从一个房间快速切换到另一个房间。

枚举类型	描述
privateMapKey	【字段含义】用于权限控制的权限票据（选填），当您希望某个房间只能让特定的 userId 进入时，需要使用 privateMapKey 进行权限保护。 【推荐取值】仅建议有高级别安全需求的客户使用，更多详情请参见 进房权限保护 。
roomId	【字段含义】数字房间号码 [选填]，在同一个房间内的用户可以看到彼此并能够进行音视频通话。 【推荐取值】取值范围：1 – 4294967294。 【特别说明】roomId 和 strRoomId 必须并且只能填一个。若两者都填，则优先选择 roomId。
strRoomId	【字段含义】字符串房间号码 [选填]，在同一个房间内的用户可以看到彼此并能够进行音视频通话。 【特别说明】roomId 和 strRoomId 必须并且只能填一个。若两者都填，则优先选择 roomId。
userSig	【字段含义】用户签名 [选填]，当前 userId 对应的验证签名，相当于登录密码。 如果您在切换房间时不指定新计算出的 userSig，SDK 会继续使用您在进入房间时（enterRoom）时所指定的 userSig。 这就需要您必须保证旧的 userSig 在切换房间的那一刻仍在签名允许的效期内，否则会导致房间切换失败。 【推荐取值】具体计算方法请参见 如何计算UserSig 。

TRTCAudioFrameDelegateFormat

TRTCAudioFrameDelegateFormat

音频自定义回调的格式参数

该参数用于在音频自定义回调相关的接口中，设置 SDK 回调出来的音频数据的相关格式（包括采样率、声道数等）。

枚举类型	描述
channels	【字段含义】声道数。 【推荐取值】默认值：1，代表单声道。可设定的数值只有两个数字：1-单声道，2-双声道。
mode	【字段含义】回调数据读写模式 【推荐取值】TRTCAudioFrameOperationModeReadOnly：仅从回调中获取音频数据。可设定的模式有 TRTCAudioFrameOperationModeReadOnly，TRTCAudioFrameOperationModeReadWrite。
sampleRate	【字段含义】采样率。 【推荐取值】默认值：48000Hz。支持 16000, 32000, 44100, 48000。
samplesPerCall	【字段含义】采样点数。 【推荐取值】取值必须是 sampleRate/100 的整数倍。

TRTCScreenCaptureSourceInfo

TRTCScreenCaptureSourceInfo

屏幕分享的目标信息（仅适用于桌面系统）

在用户进行屏幕分享时，可以选择抓取整个桌面，也可以仅抓取某个程序的窗口。TRTCScreenCaptureSourceInfo 用于描述待分享目标的信息，包括 ID、名称、缩略图等，该结构体中的字段信息均是只读的。

枚举类型	描述
extInfo	【字段含义】窗口的扩展信息。
icon	【字段含义】分享窗口的图标。
sourceId	【字段含义】采集源的ID，对于窗口，该字段代表窗口的 ID；对于屏幕，该字段代表显示器的 ID。
sourceName	【字段含义】采集源名称（采用 UTF8 编码）。
thumbnail	【字段含义】分享窗口的缩略图。

type	【字段含义】采集源类型（是分享整个屏幕？还是分享某个窗口？）。

TRTCAudioParallelParams

TRTCAudioParallelParams

远端音频流智能并发播放策略的参数

该参数用于设置远端音频流智能并发播放策略。

枚举类型	描述
includeUsers	【字段含义】指定用户必定能并发播放。 【特殊说明】指定必定并发播放的用户 ID 列表。这些用户不参与智能选择。includeUsers 的数量必须小于 maxCount，否则本次并发播放设置失效。includeUsers 仅在 maxCount > 0 时有效。当 includeUsers 生效时，参与智能并发选择的最大播放数 = maxCount − 有效 includeUsers 的数量。
maxCount	【字段含义】最大并发播放数。默认值：0。 • 如果 maxCount > 0，且实际人数 > maxCount，会实时智能选出 maxCount 路数据进行播放，这会极大的降低性能消耗。 • 如果 maxCount = 0，SDK 不限制并发播放数，在上麦人数比较多的房间可能会引发性能问题。

TRTCUser

TRTCUser

媒体流发布相关配置的用户信息

您可以通过设置该参数，配合媒体流目标发布参数（TRTCPublishTarget）和混流转码参数（TRTCStreamMixingConfig），将您指定的多路音视频流进行转码并发布到您填写的目标发布地址中

枚举类型	描述
intRoomId	【字段含义】数字房间号，需要和您的进房参数（TRTCParams）中的房间号类型相匹配。 【推荐取值】取值范围：1 – 4294967294。

	【特别说明】intRoomId 与 strRoomId 是互斥的，若您进房参数中选用 strRoomId，则 intRoomId 需要填写为0。若两者都填，SDK 将优先选用 intRoomId。
strRoomId	【字段含义】字符串房间号，需要和您的进房参数（TRTCParams）中的房间号类型相匹配。 【特别说明】intRoomId 与 strRoomId 是互斥的，若您进房参数中选用 roomId，则 strRoomId 无需填写。若两者都填，SDK 将优先选用 intRoomId。 【推荐取值】限制长度为 64 字节。以下为支持的字符集范围（共 89 个字符）： • 大小写英文字母（a-zA-Z）； • 数字（0-9）； • 空格、!、#、$、%、&、(、)、+、-、:、;、<、=、.、>、?、@、[、]、^、_、{、}、|、~、,。
userId	【字段含义】用户标识，当前用户的 userId，相当于用户名，使用 UTF-8 编码。 【推荐取值】如果一个用户在您的账号系统中的 ID 为“mike”，则 userId 即可设置为“mike”。

TRTCPublishCdnUrl

TRTCPublishCdnUrl

向腾讯或者第三方 CDN 上发布音视频流时需设置的 url 配置

该配置用于媒体流发布接口 startPublishMediaStream 中的目标推流配置（TRTCPublishTarget）

枚举类型	描述
isInternalLine	【字段含义】指定该路音视频流是否发布至腾讯云。 【推荐取值】默认值：true。 【特别说明】若您的目标直播服务商为腾讯，请将此参数设置为 true，此时后台计费系统不会对此计算转推服务费。
rtmpUrl	【字段含义】指定该路音视频流在腾讯或者第三方直播服务商的推流地址（RTMP 格式）。 【推荐取值】各家服务商的推流地址规则差异较大，请根据目标服务商的要求填写合法的推流 URL，TRTC 的后台服务器会按照您填写的 URL 向第三方服务商推送标准格式音视频流。 【特别说明】推流 URL 必须为 RTMP 格式，必须符合您的目标直播服务商的规范要求，否则目标服务商会拒绝来自 TRTC 后台服务的推流请求。

TRTCPublishTarget

TRTCPublishTarget

目标推流配置

该配置用于媒体流发布接口 startPublishMediaStream 。

枚举类型	描述
cdnUrlList	【字段含义】发布至腾讯或者第三方直播服务商的推流地址（RTMP 格式）。 【特别说明】若您的 mode 选择为 TRTCPublishMixStreamToRoom，此时您不需要设置该参数。
mixStreamIdentity	【字段含义】回推房间机器人信息。 【特别说明】仅当您的 mode 选择为 TRTCPublishMixStreamToRoom 时，您需要设置该参数。 【特别说明】设置后，该路转码音视频数据将回推到您指定的房间中。建议设置为特殊的 userId，以避免难以区分回推机器人和您通过 TRTC SDK 进房的主播。 【特别说明】参与混流的用户不支持订阅该转码流。 【特别说明】当您进房前设置的订阅模式(setDefaultStreamRecvMode)均为手动时，您需要自行管理您想要拉取的音视频流（通常当您拉取回推房间的转码流时，您应该取消订阅参与转码的对应音视频单流）。 【特别说明】当您进房前设置的订阅模式(setDefaultStreamRecvMode)均为自动时，不参与转码的用户将自动收到后台下发的转码流并不再继续接收参与转码的音视频单流。除非您明确进行取消订阅（muteRemoteVideoStream 和 muteRemoteAudio），否则转码流数据将持续下发。
mode	【字段含义】媒体流发布模式。 【推荐取值】请根据您的业务场景要求自行选择，TRTC 支持转推、转码和回推到 RTC 房间的模式。 【特别说明】若您的业务场景需要多个发布模式，您可以通过多次调用媒体流发布接口（startPublishMediaStream）并分别设置不同的 TRTCPublishTarget。 【特别说明】同一个 mode 请对应一个媒体流发布接口（startPublishMediaStream），并在后续需要调整时，使用 updatePublishCDNStream 进行更新。

TRTCVideoLayout

TRTCVideoLayout

转码视频布局

该配置用于媒体流发布接口（startPublishMediaStream）中的转码配置（TRTCStreamMixingConfig）。
用于指定转码流中每一路视频画面的位置、大小、图层以及流类型等信息。

枚举类型	描述
backgroundColor	【字段含义】指定混合画面的底色颜色。 【推荐取值】默认值：0x000000 代表黑色。格式为十六进制数字，比如：“0x61B9F1” 代表 RGB 分别为 (97,158,241)。
fillMode	【字段含义】画面填充模式。 【推荐取值】填充（画面可能会被拉伸裁剪）或适应（画面可能会有黑边），默认值：TRTCVideoFillMode_Fill。
fixedVideoStreamType	【字段含义】指定该路画面是主路画面（TRTCVideoStreamTypeBig）还是辅路画面（TRTCVideoStreamTypeSub）。
fixedVideoUser	【字段含义】参与转码的用户信息。 【特别说明】用户信息（TRTCUser）支持不填写（即 userId、intRoomId 和 strRoomId 均不填写）。此时当发起混流的房间中有主播上行音视频数据时，TRTC 后台服务器将自动将对应主播音视频填充到您指定的布局中。
placeHolderImage	【字段含义】占位图 URL，即当指定用户暂时仅上行音频时，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 【推荐取值】默认值：空值，即不设置占位图。 【特别说明】此时需要您指定用户信息 fixedVideoUser 中的 userId。 【特别说明】 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式，推荐使用 png 格式的半透明图片作为占位图。
rect	【字段含义】指定该路画面的坐标区域（单位：像素）。
zOrder	【字段含义】指定该路画面的层级（取值范围：0 – 15，不可重复）。

TRTCWatermark

TRTCWatermark

水印布局

该配置用于媒体流发布接口（startPublishMediaStream）中的转码配置（TRTCStreamMixingConfig）

枚举类型	描述
rect	【字段含义】指定该路水印画面的坐标区域（单位：像素）。
watermarkUrl	【字段含义】水印 URL，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 【特别说明】 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式，推荐使用 png 格式的半透明图片作为水印。
zOrder	【字段含义】指定该路水印画面的层级（取值范围：0 - 15，不可重复）。

TRTCStreamEncoderParam

TRTCStreamEncoderParam

媒体流编码输出参数

【字段含义】该配置用于媒体流发布接口（startPublishMediaStream）。
【特别说明】当您的发布目标（TRTCPublishTarget）中的 mode 配置为 TRTCPublish_MixStream_ToCdn 或者 TRTCPublish_MixStream_ToRoom 时，该参数为必填。
【特别说明】当您使用转推服务（mode 为 TRTCPublish_BigStream_ToCdn 或者 TRTCPublish_SubStream_ToCdn）时，为了更好的转推稳定性以及更好的 CDN 播放兼容性，也建议您填写该配置的具体参数。

枚举类型	描述
audioEncodedChannelNum	【字段含义】指定媒体发布流的目标音频声道数。 【推荐取值】默认值：1，代表单声道。可设定的数值只有两个数字：1-单声道，2-双声道。

audioEncodedCodecType	【字段含义】指定媒体发布流的目标音频编码类型。 【推荐取值】默认值：0，代表 LC-AAC。可设定的数值只有三个数字：0 - LC-AAC，1 - HE-AAC，2 - HE-AACv2。 【特别说明】HE-AAC 和 HE-AACv2 支持的输出流音频采样率范围为 [48000, 44100, 32000, 24000, 16000]。 【特别说明】当音频编码设置为 HE-AACv2 时，只支持输出流音频声道数为双声道。
audioEncodedKbps	【字段含义】指定媒体发布流的目标音频码率（kbps）。 【推荐取值】默认值：50kbps，取值范围是 [32，192]。
audioEncodedSampleRate	【字段含义】指定媒体发布流的目标音频采样率。 【推荐取值】默认值：48000Hz。取值为 [48000, 44100, 32000, 24000, 16000, 8000]，单位是 Hz。
videoEncodedCodecType	【字段含义】指定媒体发布流的目标视频编码类型。 【推荐取值】默认值：0，代表 H264。可设定的数值只有两个数字：0 - H264，1 - H265。
videoEncodedFPS	【字段含义】指定媒体发布流的目标视频帧率（FPS）。 【推荐取值】推荐值：20fps，取值范围是 (0,30]。
videoEncodedGOP	【字段含义】指定媒体发布流的目标视频关键帧间隔（GOP）。 【推荐取值】推荐值：3，单位为秒，取值范围是 [1,5]。
videoEncodedHeight	【字段含义】指定媒体发布流的目标分辨率（高度）。 【推荐取值】单位：像素值，推荐值：640，如果你只混合音频流，请将 width 和 height 均设置为 0，否则混流转码后的直播流中会有黑色背景。
videoEncodedKbps	【字段含义】指定媒体发布流的目标视频码率（kbps）。 【推荐取值】如果填 0，TRTC 会根据 videoEncodedWidth 和 videoEncodedHeight 估算出一个合理的码率值，您也可以参考视频分辨率枚举定义中所推荐的码率值（见注释部分）。
videoEncodedWidth	【字段含义】指定媒体发布流的目标分辨率（宽度）。 【推荐取值】单位：像素值，推荐值：368，如果你只混合音频流，请将 width 和 height 均设置为 0，否则混流转码后的直播流中会有黑色背景。
videoSeiParams	【字段含义】混流 SEI 参数，默认不填写。 【特别说明】参数以 json 字符串形式传入，示例如下： { "payLoadContent":"xxx", "payloadType":5, "payloadUuid":"1234567890abcdef1234567890abcdef", "interval":1000,

```
  "followIdr":false
}
```

当前支持的字段及含义：

- payloadContent: 必填。透传 sei 的 payload 内容，不能为空；
- payloadType: 必填。sei 消息的类型，取值范围: 5 或 [100, 254] 范围内的整数（244 除外，244 是内部自定义的时间戳 sei）；
- payloadUuid: 当 payloadType 为 5 时必须填写，其他情况下该值会被忽略。该值必须是长度为 32 的十六进制数字；
- interval: 选填，默认 1000。sei 的发送间隔，单位毫秒；
- followIdr: 选填，默认 false。该值为 true 时，发送关键帧时会确保带 sei，否则不确保。

TRTCStreamMixingConfig

TRTCStreamMixingConfig

媒体流转码配置参数

该配置用于媒体流发布接口（ startPublishMediaStream ）。
用于指定转码时各路画面的排版位置信息和输入的音频信息。

枚举类型	描述
audioMixUserList	【字段含义】指定转码流中的每一路输入音频的信息。 【推荐取值】该字段是一个 TRTCUser 类型的数组，数组中的每一个元素都用来代表每一路输入音频的信息。 【特别说明】用户信息支持不填写（即 audioMixUserList 为空）。此时若设置了 TRTCStreamEncoderParam 中音频相关编码输出参数，TRTC 后台服务器将自动将所有主播的音频混合输出（当前仅支持最高 16 路音视频输入）。
backgroundColor	【字段含义】指定混合画面的底色颜色。 【推荐取值】默认值：0x000000 代表黑色。格式为十六进制数字，比如：“0x61B9F1” 代表 RGB 分别为（97、158、241）。
backgroundImage	【字段含义】指定混合画面的背景图 URL，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 【推荐取值】默认值：空值，即不设置背景图片。 【特别说明】 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。

	• 图片格式支持 png、jpg、jpeg、bmp 格式，推荐使用 png 格式的半透明图片作为背景图。
videoLayoutList	【字段含义】指定混合画面中的每一路视频画面的位置、大小、图层以及流类型等信息。 【推荐取值】该字段是一个 TRTCVideoLayout 类型的数组，数组中的每一个元素都用来代表每一路画面的信息。
watermarkList	【字段含义】指定混合画面中的每一路水印画面的位置、大小、图层等信息。 【推荐取值】该字段是一个 TRTCWatermark 类型的数组，数组中的每一个元素都用来代表每一路水印的信息。

TRTCPayloadPrivateEncryptionConfig

TRTCPayloadPrivateEncryptionConfig

媒体流私有加密配置

该配置用于设置媒体流私有加密的算法和密钥。

枚举类型	描述
encryptionAlgorithm	【字段含义】加密算法，默认为 TRTCEncryptionAlgorithmAes128Gcm。
encryptionKey	【字段含义】加密用密钥，字符串类型。 【推荐取值】若加密算法为 TRTCEncryptionAlgorithmAes128Gcm，密匙长度需为 16 字节，若加密算法为 TRTCEncryptionAlgorithmAes256Gcm，密匙长度需为 32 字节。
encryptionSalt	【字段含义】盐，加密用初始向量。 【推荐取值】需确保填入该参数的数组不为空、不全为 0 且数据长度为 32 字节。

TRTCAudioVolumeEvaluateParams

TRTCAudioVolumeEvaluateParams

音量评估等相关参数设置

该设置用于开启人声检测、声音频谱计算。

枚举类型	描述
enablePitchCalculation	【字段含义】是否开启本地人声频率计算
enableSpectrumCalculation	【字段含义】是否开启声音频谱计算。
enableVadDetection	【字段含义】是否开启本地人声检测。 【请您注意】在 startLocalAudio 之前调用才可以生效。
interval	【字段含义】设置 onUserVoiceVolume 回调的触发间隔，单位为毫秒，最小间隔为 100ms，如果小于等于 0 则会关闭回调。 【推荐取值】推荐值：300，单位为毫秒。 【特别说明】interval 大于 0 时，音量提示将默认开启，无需另外设置。

废弃接口

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2022 Tencent. All rights reserved.

Module: DeprecatedTRTCCloud @ TXLiteAVSDK

Function: TRTC 废弃接口

废弃接口

TRTCCloud

函数列表	描述
destroySharedIntance	销毁 TRTCCloud 实例（单例模式）
delegate	设置 TRTC 事件回调
setBeautyStyle:beautyLevel:whitenessLevel:ruddinessLevel:	设置美颜、美白以及红润效果级别
setEyeScaleLevel:	设置大眼级别
setFaceScaleLevel:	设置瘦脸级别
setFaceVLevel:	设置 V 脸级别
setChinLevel:	设置下巴拉伸或收缩幅度
setFaceShortLevel:	设置短脸级别
setNoseSlimLevel:	设置瘦鼻级别
selectMotionTmpl:	设置动效贴纸
setMotionMute:	设置动效静音
setFilter:	设置色彩滤镜效果
setFilterConcentration:	设置色彩滤镜浓度
setGreenScreenFile:	设置绿幕背景视频

setReverbType:	设置混响效果
setVoiceChangerType:	设置变声类型
enableAudioEarMonitoring:	开启（或关闭）耳返
enableAudioVolumeEvaluation:	启用音量大小提示
enableAudioVolumeEvaluation:enable_vad:	启用音量大小提示
switchCamera	切换摄像头
isCameraZoomSupported	查询当前摄像头是否支持缩放
setZoom:	设置摄像头缩放倍数（焦距）
isCameraTorchSupported	查询是否支持开关闪光灯
enbaleTorch:	开关/关闭闪光灯
isCameraFocusPositionInPreviewSupported	查询摄像头是否支持设置焦点
setFocusPosition:	设置摄像头焦点坐标位置
isCameraAutoFocusFaceModeSupported	查询是否支持自动识别人脸位置
enableAutoFaceFoucs:	开启/关闭人脸跟踪对焦
setSystemVolumeType:	设置系统音量类型
snapshotVideo:type:	视频截图
startScreenCaptureByReplaykit:appGroup:	开始全系统的屏幕分享（iOS）
startLocalAudio	设置音频质量
startRemoteView:view:	开始显示远端视频画面
stopRemoteView:	停止显示远端视频画面，同时不再拉取该远端用户的视频数据流
setLocalViewFillMode:	设置本地图像的渲染模式
setLocalViewRotation:	设置本地图像的顺时针旋转角度

setLocalViewMirror:	设置本地摄像头预览画面的镜像模式
setRemoteViewFillMode:mode:	设置远端图像的渲染模式
setRemoteViewRotation:rotation:	设置远端图像的顺时针旋转角度
startRemoteSubStreamView:view:	开始显示远端用户的辅路画面
stopRemoteSubStreamView:	停止显示远端用户的辅路画面
setRemoteSubStreamViewFillMode:mode:	设置辅路画面的填充模式
setRemoteSubStreamViewRotation:rotation:	设置辅路画面的顺时针旋转角度
setAudioQuality:	设置音频质量
setPriorRemoteVideoStreamType:	设定优先观看大画面还是小画面
setMicVolumeOnMixing:	设置麦克风音量大小
playBGM:	启动播放背景音乐
stopBGM	停止播放背景音乐
pauseBGM	停止播放背景音乐
resumeBGM	停止播放背景音乐
getBGMDuration:	获取背景音乐总时长（单位：毫秒）
setBGMPosition:	设置背景音乐的播放进度
setBGMVolume:	设置背景音乐的音量大小
setBGMPlayoutVolume:	设置背景音乐的本地播放音量
setBGMPublishVolume:	设置背景音乐的远端播放音量
playAudioEffect:	播放音效
setAudioEffectVolume:volume:	设置音效音量
stopAudioEffect:	停止播放音效
stopAllAudioEffects	停止所有音效
setAllAudioEffectsVolume:	设置所有音效音量

版权所有：腾讯云计算（北京）有限责任公司

pauseAudioEffect:	暂停音效
resumeAudioEffect:	暂停音效
enableCustomVideoCapture:	启用视频自定义采集模式
sendCustomVideoData:	投送自己采集的视频数据
muteLocalVideo:	暂停/恢复发布本地的视频流
muteRemoteVideoStream:mute:	暂停 / 恢复订阅远端用户的视频流
startSpeedTest:userId:userSig:	开始进行网络测速（进入房间前使用）
startScreenCapture:	启动屏幕分享
getCameraDevicesList	获取摄像头设备列表
setCurrentCameraDevice:	选定当前要使用的摄像头
getCurrentCameraDevice	获取当前使用的摄像头
getMicDevicesList	获取麦克风设备列表
getCurrentMicDevice	获取当前的麦克风设备
setCurrentMicDevice:	选定当前使用的麦克风
getCurrentMicDeviceVolume	获取当前麦克风的设备音量
setCurrentMicDeviceVolume:	设置当前麦克风的设备音量
setCurrentMicDeviceMute:	设置系统当前麦克风设备的静音状态
getCurrentMicDeviceMute	获取系统当前麦克风设备是否静音
getSpeakerDevicesList	获取扬声器设备列表
getCurrentSpeakerDevice	获取当前的扬声器设备
setCurrentSpeakerDevice:	设置要使用的扬声器
getCurrentSpeakerDeviceVolume	获取当前扬声器的设备音量
setCurrentSpeakerDeviceVolume:	设置当前扬声器的设备音量
getCurrentSpeakerDeviceMute	获取系统当前扬声器设备是否静音
setCurrentSpeakerDeviceMute:	设置系统当前扬声器设备的静音状态

版权所有：腾讯云计算（北京）有限责任公司

startCameraDeviceTestInView:	开始进行摄像头测试
stopCameraDeviceTest	停止进行摄像头测试
startMicDeviceTest:	开始进行麦克风测试
stopMicDeviceTest	开始进行麦克风测试
startSpeakerDeviceTest:	开始进行扬声器测试
stopSpeakerDeviceTest	停止进行扬声器测试
startScreenCaptureInApp:	开始应用内的屏幕分享（iOS）
setVideoEncoderRotation:	设置视频编码器输出的画面方向
setVideoEncoderMirror:	设置编码器输出的画面镜像模式
setGSensorMode:	设置重力感应的适配模式

destroySharedIntance

destroySharedIntance

销毁 TRTCCloud 实例（单例模式）

@deprecated 11.5 版本不推荐使用 建议使用 destroySharedInstance

delegate

delegate

设置 TRTC 事件回调

@deprecated v11.4 版本开始不推荐使用，建议使用 addDelegate 替代之。

setBeautyStyle:beautyLevel:whitenessLevel:ruddinessLevel:

setBeautyStyle:beautyLevel:whitenessLevel:ruddinessLevel:

```
- (void)setBeautyStyle:    (TRTCBeautyStyle)beautyStyle
```

```
beautyLevel:      (NSInteger)beautyLevel
whitenessLevel:   (NSInteger)whitenessLevel
ruddinessLevel:   (NSInteger)ruddinessLevel
```

设置美颜、美白以及红润效果级别

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setEyeScaleLevel:

setEyeScaleLevel:

```
- (void)setEyeScaleLevel:      (float)eyeScaleLevel
```

设置大眼级别

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setFaceScaleLevel:

setFaceScaleLevel:

```
- (void)setFaceScaleLevel:      (float)faceScaleLevel
```

设置瘦脸级别

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setFaceVLevel:

setFaceVLevel:

```
- (void)setFaceVLevel:      (float)faceVLevel
```

设置 V 脸级别

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setChinLevel:

setChinLevel:

```
- (void)setChinLevel:    (float)chinLevel
```

设置下巴拉伸或收缩幅度

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setFaceShortLevel:

setFaceShortLevel:

```
- (void)setFaceShortLevel:    (float)faceShortlevel
```

设置短脸级别

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setNoseSlimLevel:

setNoseSlimLevel:

```
- (void)setNoseSlimLevel:    (float)noseSlimLevel
```

设置瘦鼻级别

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

selectMotionTmpl:

selectMotionTmpl:

```
- (void)selectMotionTmpl:    (NSString *)tmplPath
```

设置动效贴纸

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setMotionMute:

setMotionMute:

```
- (void)setMotionMute:    (BOOL)motionMute
```

设置动效静音

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setFilter:

setFilter:

```
- (void)setFilter:    (TXImage *)image
```

设置色彩滤镜效果

@deprecated v7.2 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setFilterConcentration:

setFilterConcentration:

```
- (void)setFilterConcentration:    (float)concentration
```

设置色彩滤镜浓度

@deprecated v7.2 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setGreenScreenFile:

setGreenScreenFile:

```
- (void)setGreenScreenFile:    (NSURL *)file
```

设置绿幕背景视频

@deprecated v7.2 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setReverbType:

setReverbType:

```
- (void)setReverbType:(TRTCReverbType)reverbType
```

设置混响效果

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setVoiceReverbType 替代之。

setVoiceChangerType:

setVoiceChangerType:

```
- (void)setVoiceChangerType:(TRTCVoiceChangerType)voiceChanger
```

设置变声类型

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setVoiceChangerType 替代之。

enableAudioEarMonitoring:

enableAudioEarMonitoring:

```
- (void)enableAudioEarMonitoring:(BOOL)enable
```

开启（或关闭）耳返

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setVoiceEarMonitor 替代之。

enableAudioVolumeEvaluation:

enableAudioVolumeEvaluation:

```
- (void)enableAudioVolumeEvaluation:(NSUInteger)interval
```

启用音量大小提示

@deprecated v10.1 版本开始不推荐使用，建议使用 enableAudioVolumeEvaluation(enable, params) 替代之。

enableAudioVolumeEvaluation:enable_vad:

enableAudioVolumeEvaluation:enable_vad:

```
- (void)enableAudioVolumeEvaluation:(NSUInteger)interval
                         enable_vad:(BOOL)enable_vad
```

启用音量大小提示

@deprecated v11.2 版本开始不推荐使用，建议使用 enableAudioVolumeEvaluation(enable, params) 替代之。

switchCamera

switchCamera

切换摄像头

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 switchCamera 接口替代之。

isCameraZoomSupported

isCameraZoomSupported

查询当前摄像头是否支持缩放

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 isCameraZoomSupported 接口替代之。

setZoom:

setZoom:

```
- (void)setZoom:      (CGFloat)distance
```

设置摄像头缩放倍数（焦距）

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCameraZoomRatio 接口替代之。

isCameraTorchSupported

isCameraTorchSupported

查询是否支持开关闪光灯

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 isCameraTorchSupported 接口替代之。

enbaleTorch:

enbaleTorch:

```
- (BOOL)enbaleTorch:      (BOOL)enable
```

开关/关闭闪光灯

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 enableCameraTorch 接口替代之。

isCameraFocusPositionInPreviewSupported

isCameraFocusPositionInPreviewSupported

查询摄像头是否支持设置焦点

@deprecated v8.0 版本开始不推荐使用。

setFocusPosition:

setFocusPosition:

```
- (void)setFocusPosition:(CGPoint)touchPoint
```

设置摄像头焦点坐标位置

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCameraFocusPosition 接口替代之。

isCameraAutoFocusFaceModeSupported

isCameraAutoFocusFaceModeSupported

查询是否支持自动识别人脸位置

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 isAutoFocusEnabled 接口替代之。

enableAutoFaceFoucs:

enableAutoFaceFoucs:

```
- (void)enableAutoFaceFoucs:(BOOL)enable
```

开启/关闭人脸跟踪对焦

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 enableCameraAutoFocus 接口替代之。

setSystemVolumeType:

setSystemVolumeType:

```
- (void)setSystemVolumeType:(TRTCSystemVolumeType)type
```

设置系统音量类型

@deprecated v8.0 版本开始不推荐使用，建议使用 startLocalAudio(quality) 替代之，通过 quality 参数来决策音质。

snapshotVideo:type:

snapshotVideo:type:

```
- (void)snapshotVideo:(NSString *)userId
                 type:(TRTCVideoStreamType)streamType
```

视频截图

@deprecated v8.2 版本开始不推荐使用，建议使用 snapshotVideo 替代之。

startScreenCaptureByReplaykit:appGroup:

startScreenCaptureByReplaykit:appGroup:

```
- (void)startScreenCaptureByReplaykit:(TRTCVideoEncParam *)encParams
                             appGroup:(NSString *)appGroup
```

开始全系统的屏幕分享（iOS）

@deprecated v8.6 版本开始不推荐使用，建议使用 startScreenCaptureByReplaykit 接口替代之。

startLocalAudio

startLocalAudio

设置音频质量

@deprecated v8.0 版本开始不推荐使用，建议使用 startLocalAudio 替代之。

startRemoteView:view:

startRemoteView:view:

```
- (void)startRemoteView:    (NSString *)userId
                   view:    (TXView *)view
```

开始显示远端视频画面

@deprecated v8.0 版本开始不推荐使用，建议使用 startRemoteView 替代之。

stopRemoteView:

stopRemoteView:

```
- (void)stopRemoteView:    (NSString *)userId
```

停止显示远端视频画面，同时不再拉取该远端用户的视频数据流

@deprecated v8.0 版本开始不推荐使用，建议使用 stopRemoteView 替代之。

setLocalViewFillMode:

setLocalViewFillMode:

```
- (void)setLocalViewFillMode:    (TRTCVideoFillMode)mode
```

设置本地图像的渲染模式

@deprecated v8.0 版本开始不推荐使用，建议使用 setLocalRenderParams 替代之。

setLocalViewRotation:

setLocalViewRotation:

```
- (void)setLocalViewRotation:(TRTCVideoRotation)rotation
```

设置本地图像的顺时针旋转角度

@deprecated v8.0 版本开始不推荐使用，建议使用 setLocalRenderParams 替代之。

setLocalViewMirror:

setLocalViewMirror:

```
- (void)setLocalViewMirror:(TRTCLocalVideoMirrorType)mirror
```

设置本地摄像头预览画面的镜像模式

@deprecated v8.0 版本开始不推荐使用，建议使用 setLocalRenderParams 替代之。

setRemoteViewFillMode:mode:

setRemoteViewFillMode:mode:

```
- (void)setRemoteViewFillMode:(NSString*)userId
                         mode:(TRTCVideoFillMode)mode
```

设置远端图像的渲染模式

@deprecated v8.0 版本开始不推荐使用，建议使用 setRemoteRenderParams 替代之。

setRemoteViewRotation:rotation:

setRemoteViewRotation:rotation:

```
- (void)setRemoteViewRotation:(NSString*)userId
                     rotation:(TRTCVideoRotation)rotation
```

设置远端图像的顺时针旋转角度

@deprecated v8.0 版本开始不推荐使用，建议使用 setRemoteRenderParams 替代之。

startRemoteSubStreamView:view:

startRemoteSubStreamView:view:

```
- (void)startRemoteSubStreamView:    (NSString *)userId
                            view:    (TXView *)view
```

开始显示远端用户的辅路画面

@deprecated v8.0 版本开始不推荐使用，建议使用 startRemoteView 替代之。

stopRemoteSubStreamView:

stopRemoteSubStreamView:

```
- (void)stopRemoteSubStreamView:    (NSString *)userId
```

停止显示远端用户的辅路画面

@deprecated v8.0 版本开始不推荐使用，建议使用 stopRemoteView 替代之。

setRemoteSubStreamViewFillMode:mode:

setRemoteSubStreamViewFillMode:mode:

```
- (void)setRemoteSubStreamViewFillMode:    (NSString *)userId
                                   mode:    (TRTCVideoFillMode)mode
```

设置辅路画面的填充模式

@deprecated v8.0 版本开始不推荐使用，建议使用 setRemoteRenderParams 替代之。

setRemoteSubStreamViewRotation:rotation:

setRemoteSubStreamViewRotation:rotation:

```
- (void)setRemoteSubStreamViewRotation:(NSString*)userId
                              rotation:(TRTCVideoRotation)rotation
```

设置辅路画面的顺时针旋转角度

@deprecated v8.0 版本开始不推荐使用，建议使用 setRemoteRenderParams:streamType:params: 替代之。

setAudioQuality:

setAudioQuality:

```
- (void)setAudioQuality:(TRTCAudioQuality)quality
```

设置音频质量

@deprecated v8.0 版本开始不推荐使用，建议使用 startLocalAudio 替代之。

setPriorRemoteVideoStreamType:

setPriorRemoteVideoStreamType:

```
- (void)setPriorRemoteVideoStreamType:(TRTCVideoStreamType)streamType
```

设定优先观看大画面还是小画面

@deprecated v8.0 版本开始不推荐使用，建议使用 startRemoteView:streamType:view: 替代之。

setMicVolumeOnMixing:

setMicVolumeOnMixing:

```
- (void)setMicVolumeOnMixing:(NSInteger)volume
```

设置麦克风音量大小

playBGM:

playBGM:

```
- (void) playBGM:      (NSString *)path
```

启动播放背景音乐

@deprecated v7.3 版本开始不推荐使用，建议使用 getAudioEffectManager 替代之。

stopBGM

stopBGM

停止播放背景音乐

@deprecated v7.3 版本开始不推荐使用，建议使用 getAudioEffectManager 替代之。

pauseBGM

pauseBGM

停止播放背景音乐

@deprecated v7.3 版本开始不推荐使用，建议使用 getAudioEffectManager 替代之。

resumeBGM

resumeBGM

停止播放背景音乐

@deprecated v7.3 版本开始不推荐使用，建议使用 getAudioEffectManager 替代之。

getBGMDuration:

getBGMDuration:

```
- (NSInteger)getBGMDuration:    (NSString *)path
```

获取背景音乐总时长（单位：毫秒）

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 getMusicDurationInMS 替代之。

setBGMPosition:

setBGMPosition:

```
- (int)setBGMPosition:    (NSInteger)pos
```

设置背景音乐的播放进度

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 seekMusicToPosInMS 替代之。

setBGMVolume:

setBGMVolume:

```
- (void)setBGMVolume:    (NSInteger)volume
```

设置背景音乐的音量大小

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setMusicVolume 替代之。

setBGMPlayoutVolume:

setBGMPlayoutVolume:

```
- (void)setBGMPlayoutVolume:    (NSInteger)volume
```

设置背景音乐的本地播放音量

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setMusicPlayoutVolume 替代之。

setBGMPublishVolume:

setBGMPublishVolume:

```
- (void)setBGMPublishVolume:(NSInteger)volume
```

设置背景音乐的远端播放音量

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setBGMPublishVolume 替代之。

playAudioEffect:

playAudioEffect:

```
- (void)playAudioEffect:(TRTCAudioEffectParam *)effect
```

播放音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 startPlayMusic 替代之。

setAudioEffectVolume:volume:

setAudioEffectVolume:volume:

```
- (void)setAudioEffectVolume:(int)effectId
                      volume:(int) volume
```

设置音效音量

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setMusicPublishVolume 和 setMusicPlayoutVolume 替代之。

stopAudioEffect:

stopAudioEffect:

- (void)stopAudioEffect: (int)effectId

停止播放音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 stopPlayMusic 替代之。

stopAllAudioEffects

stopAllAudioEffects

停止所有音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 stopPlayMusic 替代之。

setAllAudioEffectsVolume:

setAllAudioEffectsVolume:

- (void)setAllAudioEffectsVolume: (int)volume

设置所有音效音量

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setMusicPublishVolume 和 setMusicPlayoutVolume 替代之。

pauseAudioEffect:

pauseAudioEffect:

```
- (void)pauseAudioEffect:    (int)effectId
```

暂停音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 pauseAudioEffect 替代之。

resumeAudioEffect:

resumeAudioEffect:

```
- (void)resumeAudioEffect:    (int)effectId
```

暂停音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 resumePlayMusic 替代之。

enableCustomVideoCapture:

enableCustomVideoCapture:

```
- (void)enableCustomVideoCapture:    (BOOL)enable
```

启用视频自定义采集模式

@deprecated v8.5 版本开始不推荐使用，建议使用 enableCustomVideoCapture(streamType, enable) 接口替代之。

sendCustomVideoData:

sendCustomVideoData:

```
- (void)sendCustomVideoData:    (TRTCVideoFrame *)frame
```

投送自己采集的视频数据

@deprecated v8.5 版本开始不推荐使用，建议使用 sendCustomVideoData(streamType, TRTCVideoFrame) 接口替代之。

muteLocalVideo:

muteLocalVideo:

```
- (void)muteLocalVideo:    (BOOL)mute
```

暂停/恢复发布本地的视频流

@deprecated v8.9 版本开始不推荐使用，建议使用 muteLocalVideo(streamType, mute) 接口替代之。

muteRemoteVideoStream:mute:

muteRemoteVideoStream:mute:

```
- (void)muteRemoteVideoStream:    (NSString*)userId
                          mute:    (BOOL)mute
```

暂停 / 恢复订阅远端用户的视频流

@deprecated v8.9 版本开始不推荐使用，建议使用 muteRemoteVideoStream(userId, streamType, mute) 接口替代之。

startSpeedTest:userId:userSig:

startSpeedTest:userId:userSig:

```
- (void)startSpeedTest:    (uint32_t)sdkAppId
                userId:    (NSString *)userId
               userSig:    (NSString *)userSig
```

开始进行网络测速（进入房间前使用）

@deprecated v9.2 版本开始不推荐使用，建议使用 startSpeedTest(params) 接口替代之。

startScreenCapture:

startScreenCapture:

```
- (void)startScreenCapture:      (nullable NSView *)view
```

启动屏幕分享

@deprecated v7.2 版本开始不推荐使用，建议使用 startScreenCapture 替代之。

getCameraDevicesList

getCameraDevicesList

获取摄像头设备列表

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getDevicesList 接口替代之。

setCurrentCameraDevice:

setCurrentCameraDevice:

```
- (int)setCurrentCameraDevice:      (NSString *)deviceId
```

选定当前要使用的摄像头

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDevice 接口替代之。

getCurrentCameraDevice

getCurrentCameraDevice

获取当前使用的摄像头

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDevice 接口替代之。

getMicDevicesList

getMicDevicesList

获取麦克风设备列表

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getDevicesList 接口替代之。

getCurrentMicDevice

getCurrentMicDevice

获取当前的麦克风设备

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDevice 接口替代之。

setCurrentMicDevice:

setCurrentMicDevice:

```
- (int)setCurrentMicDevice:      (NSString*)deviceId
```

选定当前使用的麦克风

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDevice 接口替代之。

getCurrentMicDeviceVolume

getCurrentMicDeviceVolume

获取当前麦克风的设备音量

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDeviceVolume 接口替代之。

setCurrentMicDeviceVolume:

setCurrentMicDeviceVolume:

```
- (void)setCurrentMicDeviceVolume:(NSInteger)volume
```

设置当前麦克风的设备音量

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDeviceVolume 接口替代之。

setCurrentMicDeviceMute:

setCurrentMicDeviceMute:

```
- (void)setCurrentMicDeviceMute:(BOOL)mute
```

设置系统当前麦克风设备的静音状态

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDeviceMute 接口替代之。

getCurrentMicDeviceMute

getCurrentMicDeviceMute

获取系统当前麦克风设备是否静音

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDeviceMute 接口替代之。

getSpeakerDevicesList

getSpeakerDevicesList

获取扬声器设备列表

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getDevicesList 接口替代之。

getCurrentSpeakerDevice

getCurrentSpeakerDevice

获取当前的扬声器设备

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDevice 接口替代之。

setCurrentSpeakerDevice:

setCurrentSpeakerDevice:

```
- (int)setCurrentSpeakerDevice:      (NSString*)deviceId
```

设置要使用的扬声器

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDevice 接口替代之。

getCurrentSpeakerDeviceVolume

getCurrentSpeakerDeviceVolume

获取当前扬声器的设备音量

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDeviceVolume 接口替代之。

setCurrentSpeakerDeviceVolume:

setCurrentSpeakerDeviceVolume:

```
- (int)setCurrentSpeakerDeviceVolume:    (NSInteger)volume
```

设置当前扬声器的设备音量

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDeviceVolume 接口替代之。

getCurrentSpeakerDeviceMute

getCurrentSpeakerDeviceMute

获取系统当前扬声器设备是否静音

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDeviceMute 接口替代之。

setCurrentSpeakerDeviceMute:

setCurrentSpeakerDeviceMute:

```
- (void)setCurrentSpeakerDeviceMute:    (BOOL)mute
```

设置系统当前扬声器设备的静音状态

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDeviceMute 接口替代之。

startCameraDeviceTestInView:

startCameraDeviceTestInView:

```
- (void)startCameraDeviceTestInView:    (NSView *)view
```

开始进行摄像头测试

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 startCameraDeviceTest 接口替代之。

stopCameraDeviceTest

stopCameraDeviceTest

停止进行摄像头测试

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 stopCameraDeviceTest 接口替代之。

startMicDeviceTest:

startMicDeviceTest:

```
- (void)startMicDeviceTest:      (NSInteger)interval
```

开始进行麦克风测试

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 startMicDeviceTest 接口替代之。

stopMicDeviceTest

stopMicDeviceTest

开始进行麦克风测试

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 stopMicDeviceTest 接口替代之。

startSpeakerDeviceTest:

startSpeakerDeviceTest:

```
- (void)startSpeakerDeviceTest:      (NSString*)audioFilePath
```

开始进行扬声器测试

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 startSpeakerDeviceTest 接口替代之。

stopSpeakerDeviceTest

stopSpeakerDeviceTest

停止进行扬声器测试

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 stopSpeakerDeviceTest 接口替代之。

startScreenCaptureInApp:

startScreenCaptureInApp:

```
- (void)startScreenCaptureInApp:      (TRTCVideoEncParam *)encParams
```

开始应用内的屏幕分享（iOS）

@deprecated v8.6 版本开始不推荐使用，建议使用 startScreenCaptureInApp 接口替代之。

setVideoEncoderRotation:

setVideoEncoderRotation:

```
- (void)setVideoEncoderRotation:      (TRTCVideoRotation)rotation
```

设置视频编码器输出的画面方向

@deprecated v11.7 版本开始不推荐使用。

setVideoEncoderMirror:

setVideoEncoderMirror:

- (void)setVideoEncoderMirror: (BOOL)mirror

设置编码器输出的画面镜像模式

@deprecated v11.7 版本开始不推荐使用。

setGSensorMode:

setGSensorMode:

- (void)setGSensorMode: (TRTCGSensorMode) mode

设置重力感应的适配模式

@deprecated v11.7 版本开始不推荐使用。建议使用 setGravitySensorAdaptiveMode 接口替代之。

错误码表

最近更新时间：2023-04-23 17:14:23

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTC 错误码表。

Function: 用于通知客户 TRTC 在使用过程中出现的警告和错误。

iOS&Mac 的错误码，请查阅：全平台 C++ 错误码表 。

Android API 概览

最近更新时间：2024-03-06 14:51:32

API OVERVIEW

创建实例和事件回调

函数列表	描述
sharedInstance	创建 TRTCCloud 实例（单例模式）
destroySharedInstance	销毁 TRTCCloud 实例（单例模式）
addListener	添加 TRTC 事件回调
removeListener	移除 TRTC 事件回调
setListenerHandler	设置驱动 TRTCCloudListener 事件回调的队列

房间相关接口函数

函数列表	描述
enterRoom	进入房间
exitRoom	离开房间
switchRole	切换角色
switchRoom	切换房间
ConnectOtherRoom	请求跨房通话
DisconnectOtherRoom	退出跨房通话
setDefaultStreamRecvMode	设置订阅模式（需要在进入房前设置才能生效）
createSubCloud	创建子房间实例（用于多房间并发观看）

destroySubCloud	销毁子房间实例
updateOtherRoomForwardMode	更改跨房主播在本房间的上行能力

CDN 相关接口函数

函数列表	描述
startPublishing	开始向腾讯云直播 CDN 上发布音视频流
stopPublishing	停止向腾讯云直播 CDN 上发布音视频流
startPublishCDNStream	开始向非腾讯云 CDN 上发布音视频流
stopPublishCDNStream	停止向非腾讯云 CDN 上发布音视频流
setMixTranscodingConfig	设置云端混流的排版布局和转码参数
startPublishMediaStream	开始发布媒体流
updatePublishMediaStream	更新发布媒体流
stopPublishMediaStream	停止发布媒体流

视频相关接口函数

函数列表	描述
startLocalPreview	开启本地摄像头的预览画面（移动端）
updateLocalView	更新本地摄像头的预览画面
stopLocalPreview	停止摄像头预览
muteLocalVideo	暂停/恢复发布本地的视频流
setVideoMuteImage	设置本地画面被暂停期间的替代图片
startRemoteView	订阅远端用户的视频流，并绑定视频渲染控件

updateRemoteView	更新远端用户的视频渲染控件
stopRemoteView	停止订阅远端用户的视频流，并释放渲染控件
stopAllRemoteView	停止订阅所有远端用户的视频流，并释放全部渲染资源
muteRemoteVideoStream	暂停/恢复订阅远端用户的视频流
muteAllRemoteVideoStreams	暂停/恢复订阅所有远端用户的视频流
setVideoEncoderParam	设置视频编码器的编码参数
setNetworkQosParam	设置网络质量控制的相关参数
setLocalRenderParams	设置本地画面的渲染参数
setRemoteRenderParams	设置远端画面的渲染模式
enableEncSmallVideoStream	开启大小画面双路编码模式
setRemoteVideoStreamType	切换指定远端用户的大小画面
snapshotVideo	视频画面截图
setPerspectiveCorrectionPoints	视频画面透视校正坐标设置
setGravitySensorAdaptiveMode	设置重力感应的适配模式（11.7 及以上版本）

音频相关接口函数

函数列表	描述
startLocalAudio	开启本地音频的采集和发布
stopLocalAudio	停止本地音频的采集和发布
muteLocalAudio	暂停/恢复发布本地的音频流
muteRemoteAudio	暂停/恢复播放远端的音频流
muteAllRemoteAudio	暂停/恢复播放所有远端用户的音频流

setAudioRoute	设置音频路由
setRemoteAudioVolume	设定某一个远端用户的声音播放音量
setAudioCaptureVolume	设定本地音频的采集音量
getAudioCaptureVolume	获取本地音频的采集音量
setAudioPlayoutVolume	设定远端音频的播放音量
getAudioPlayoutVolume	获取远端音频的播放音量
enableAudioVolumeEvaluation	启用音量大小提示
startAudioRecording	开始录音
stopAudioRecording	停止录音
startLocalRecording	开启本地媒体录制
stopLocalRecording	停止本地媒体录制
setRemoteAudioParallelParams	设置远端音频流智能并发播放策略
enable3DSpatialAudioEffect	启用 3D 音效
updateSelf3DSpatialPosition	设置 3D 音效中自身坐标及朝向信息
updateRemote3DSpatialPosition	设置 3D 音效中远端用户坐标信息
set3DSpatialReceivingRange	设置指定用户所发出声音的可被接收范围

设备管理相关接口

函数列表	描述
getDeviceMa	获取设备管理类（TXDeviceManager）

nager	

美颜特效和图像水印

函数列表	描述
getBeautyManager	获取美颜管理类（TXBeautyManager）
setWatermark	添加水印

背景音乐和声音特效

函数列表	描述
getAudioEffectManager	获取音效管理类（TXAudioEffectManager）
startSystemAudioLoopback	开启系统声音采集
stopSystemAudioLoopback	停止系统声音采集（iOS 端暂未支持）

屏幕分享相关接口

函数列表	描述
startScreenCapture	启动屏幕分享
stopScreenCapture	停止屏幕分享
pauseScreenCapture	暂停屏幕分享
resumeScreenCapture	恢复屏幕分享
setSubStreamEncoderParam	设置屏幕分享（即辅路）的视频编码参数（桌面系统和移动系统均已支持）

自定义采集和自定义渲染

函数列表	描述

enableCustomVideoCapture	启用/关闭视频自定义采集模式
sendCustomVideoData	向 SDK 投送自己采集的视频帧
enableCustomAudioCapture	启用音频自定义采集模式
sendCustomAudioData	向 SDK 投送自己采集的音频数据
enableMixExternalAudioFrame	启用/关闭自定义音轨
mixExternalAudioFrame	向 SDK 混入自定义音轨
setMixExternalAudioVolume	设置推流时混入外部音频的推流音量和播放音量
generateCustomPTS	生成自定义采集时的时间戳
setLocalVideoProcessListener	设置第三方美颜的视频数据回调
setLocalVideoRenderListener	设置本地视频自定义渲染回调
setRemoteVideoRenderListener	设置远端视频自定义渲染回调
setAudioFrameListener	设置音频数据自定义回调
setCapturedAudioFrameCallbackFormat	设置本地麦克风采集出的音频帧回调格式
setLocalProcessedAudioFrameCallbackFormat	设置经过前处理后的本地音频帧回调格式
setMixedPlayAudioFrameCallbackFormat	设置最终要由系统播放出的音频帧回调格式
enableCustomAudioRendering	开启音频自定义播放
getCustomAudioRenderingFrame	获取可播放的音频数据

自定义消息发送接口

函数列表	描述
sendCustomCmdMsg	使用 UDP 通道发送自定义消息给房间内所有用户
sendSEIMsg	使用 SEI 通道发送自定义消息给房间内所有用户

网络测试接口

函数列表	描述
startSpeedTest	开始进行网速测试（进入房间前使用）
stopSpeedTest	停止网络测速

调试相关接口

函数列表	描述
getSDKVersion	获取 SDK 版本信息
setLogLevel	设置 Log 输出级别
setConsoleEnabled	启用/禁用控制台日志打印
setLogCompressEnabled	启用/禁用日志的本地压缩
setLogDirPath	设置本地日志的保存路径
setLogListener	设置日志回调
showDebugView	显示仪表盘
TRTCViewMargin	设置仪表盘的边距
callExperimentalAPI	调用实验性接口

加密接口

函数列表	描述
enablePayloadPrivateEncryption	开启或关闭媒体流私有加密

错误和警告事件

函数列表	描述
onError	错误事件回调

onWarning	警告事件回调

房间相关事件回调

函数列表	描述
onEnterRoom	进入房间成功与否的事件回调
onExitRoom	离开房间的事件回调
onSwitchRole	切换角色的事件回调
onSwitchRoom	切换房间的结果回调
onConnectOtherRoom	请求跨房通话的结果回调
onDisConnectOtherRoom	结束跨房通话的结果回调
onUpdateOtherRoomForwardMode	更改跨房主播上行能力的结果回调

用户相关事件回调

函数列表	描述
onRemoteUserEnterRoom	有用户加入当前房间
onRemoteUserLeaveRoom	有用户离开当前房间
onUserVideoAvailable	某远端用户发布/取消了主路视频画面
onUserSubStreamAvailable	某远端用户发布/取消了辅路视频画面
onUserAudioAvailable	某远端用户发布/取消了自己的音频
onFirstVideoFrame	SDK 开始渲染自己本地或远端用户的首帧画面
onFirstAudioFrame	SDK 开始播放远端用户的首帧音频
onSendFirstLocalVid	自己本地的首个视频帧已被发布出去

eoFrame	
onSendFirstLocalAudioFrame	自己本地的首个音频帧已被发布出去
onRemoteVideoStatusUpdated	远端视频状态变化的事件回调
onRemoteAudioStatusUpdated	远端音频状态变化的事件回调
onUserVideoSizeChanged	用户视频大小发生改变回调

网络和技术指标统计回调

函数列表	描述
onNetworkQuality	网络质量的实时统计回调
onStatistics	音视频技术指标的实时统计回调
onSpeedTestResult	网速测试的结果回调

与云端连接情况的事件回调

函数列表	描述
onConnectionLost	SDK 与云端的连接已经断开
onTryToReconnect	SDK 正在尝试重新连接到云端
onConnectionRecovery	SDK 与云端的连接已经恢复

硬件设备相关事件回调

函数列表	描述
onCameraDidReady	摄像头准备就绪

onMicDidReady	麦克风准备就绪
onAudioRouteChanged	当前音频路由发生变化（仅适用于移动设备）
onUserVoiceVolume	音量大小的反馈回调

自定义消息的接收事件回调

函数列表	描述
onRecvCustomCmdMsg	收到自定义消息的事件回调
onMissCustomCmdMsg	自定义消息丢失的事件回调
onRecvSEIMsg	收到 SEI 消息的回调

CDN 相关事件回调

函数列表	描述
onStartPublishing	开始向腾讯云直播 CDN 上发布音视频流的事件回调
onStopPublishing	停止向腾讯云直播 CDN 上发布音视频流的事件回调
onStartPublishCDNStream	开始向非腾讯云 CDN 上发布音视频流的事件回调
onStopPublishCDNStream	停止向非腾讯云 CDN 上发布音视频流的事件回调
onSetMixTranscodingConfig	设置云端混流的排版布局和转码参数的事件回调
onStartPublishMediaStream	开始发布媒体流的事件回调
onUpdatePublishMediaStream	更新媒体流的事件回调
onStopPublishMediaStream	停止媒体流的事件回调

onCdnStreamStateChanged	RTMP/RTMPS 推流状态发生改变回调

屏幕分享相关事件回调

函数列表	描述
onScreenCaptureStarted	屏幕分享开启的事件回调
onScreenCapturePaused	屏幕分享暂停的事件回调
onScreenCaptureResumed	屏幕分享恢复的事件回调
onScreenCaptureStopped	屏幕分享停止的事件回调

本地录制和本地截图的事件回调

函数列表	描述
onLocalRecordBegin	本地录制任务已经开始的事件回调
onLocalRecording	本地录制任务正在进行中的进展事件回调
onLocalRecordFragment	本地录制分片的事件回调
onLocalRecordComplete	本地录制任务已经结束的事件回调
onSnapshotComplete	本地截图完成的事件回调

废弃的事件回调

函数列表	描述
onUserEnter	有主播加入当前房间（已废弃）
onUserExit	有主播离开当前房间（已废弃）

onAudioEffectFinished	音效播放已结束（已废弃）
onSpeedTest	服务器测速的结果回调（已废弃）

视频数据自定义回调

函数列表	描述
onRenderVideoFrame	自定义视频渲染回调
onGLContextCreated	SDK 内部 OpenGL 环境已经创建的通知
onProcessVideoFrame	用于对接第三方美颜组件的视频处理回调
onGLContextDestory	SDK 内部 OpenGL 环境被销毁的通知

音频数据自定义回调

函数列表	描述
onCapturedAudioFrame	本地采集并经过音频模块前处理后的音频数据回调
onLocalProcessedAudioFrame	本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调
onRemoteUserAudioFrame	混音前的每一路远程用户的音频数据
onMixedPlayAudioFrame	将各路待播放音频混合之后并在最终提交系统播放之前的数据回调
onMixedAllAudioFrame	SDK 所有音频混合后的音频数据（包括采集到的和待播放的）
onVoiceEarMonitorAudioFrame	耳返的音频数据

更多事件回调接口

函数	描述

列表	
onLog	本地 LOG 的打印回调

背景音乐预加载事件回调

函数列表	描述
onLoadProgress	背景音乐预加载进度
onLoadError	背景音乐预加载出错

背景音乐的播放事件回调

函数列表	描述
onStart	背景音乐开始播放
onPlayProgress	背景音乐的播放进度
onComplete	背景音乐已经播放完毕

人声相关的特效接口

函数列表	描述
enableVoiceEarMonitor	开启耳返
setVoiceEarMonitorVolume	设置耳返音量
setVoiceReverbType	设置人声的混响效果
setVoiceChangerType	设置人声的变声特效
setVoiceCaptureVolume	设置语音音量

setVoicePitch	设置语音音调

背景音乐的相关接口

函数列表	描述
setMusicObserver	设置背景音乐的事件回调接口
startPlayMusic	开始播放背景音乐
stopPlayMusic	停止播放背景音乐
pausePlayMusic	暂停播放背景音乐
resumePlayMusic	恢复播放背景音乐
setAllMusicVolume	设置所有背景音乐的本地音量和远端音量的大小
setMusicPublishVolume	设置某一首背景音乐的远端音量的大小
setMusicPlayoutVolume	设置某一首背景音乐的本地音量的大小
setMusicPitch	调整背景音乐的音调高低
setMusicSpeedRate	调整背景音乐的变速效果
getMusicCurrentPosInMS	获取背景音乐的播放进度（单位：毫秒）
getMusicDurationInMS	获取背景音乐的总时长（单位：毫秒）
seekMusicToPosInMS	设置背景音乐的播放进度（单位：毫秒）
setMusicScratchSpeedRate	调整搓碟的变速效果
setPreloadObserver	设置预加载事件回调
preloadMusic	预加载背景音乐
getMusicTrackCount	获取背景音乐的音轨数量
setMusicTrack	指定背景音乐的播放音轨

版权所有：腾讯云计算（北京）有限责任公司

美颜相关接口

函数列表	描述
setBeautyStyle	设置美颜（磨皮）算法
setBeautyLevel	设置美颜级别
setWhitenessLevel	设置美白级别
enableSharpnessEnhancement	开启清晰度增强
setRuddyLevel	设置红润级别
setFilter	设置色彩滤镜效果
setFilterStrength	设置色彩滤镜的强度
setGreenScreenFile	设置绿幕背景视频
setEyeScaleLevel	设置大眼级别
setFaceSlimLevel	设置瘦脸级别
setFaceVLevel	设置 V 脸级别
setChinLevel	设置下巴拉伸或收缩
setFaceShortLevel	设置短脸级别
setFaceNarrowLevel	设置窄脸级别
setNoseSlimLevel	设置瘦鼻级别
setEyeLightenLevel	设置亮眼级别
setToothWhitenLevel	设置牙齿美白级别
setWrinkleRemoveLevel	设置祛皱级别
setPounchRemoveLevel	设置祛眼袋级别
setSmileLinesRemoveLevel	设置法令纹去除级别

setForeheadLevel	设置发际线调整级别
setEyeDistanceLevel	设置眼距
setEyeAngleLevel	设置眼角调整级别
setMouthShapeLevel	设置嘴型调整级别
setNoseWingLevel	设置鼻翼调整级别
setNosePositionLevel	设置鼻子位置
setLipsThicknessLevel	设置嘴唇厚度
setFaceBeautyLevel	设置脸型
setMotionTmpl	选择 AI 动效挂件
setMotionMute	是否在动效素材播放时静音

设备操作接口

函数列表	描述
isFrontCamera	判断当前是否为前置摄像头（仅适用于移动端）
switchCamera	切换前置或后置摄像头（仅适用于移动端）
getCameraZoomMaxRatio	获取摄像头的最大缩放倍数（仅适用于移动端）
setCameraZoomRatio	设置摄像头的缩放倍数（仅适用于移动端）
isAutoFocusEnabled	查询是否支持自动识别人脸位置（仅适用于移动端）
enableCameraAutoFocus	开启自动对焦功能（仅适用于移动端）
setCameraFocusPosition	设置摄像头的对焦位置（仅适用于移动端）
enableCameraTorc	开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）

版权所有：腾讯云计算（北京）有限责任公司

h	
setAudioRoute	设置音频路由（仅适用于移动端）
setExposureCompensation	设置摄像头的曝光参数，取值范围从-1到1
setCameraCapturerParam	设置摄像头采集偏好

弃用接口

函数列表	描述
setSystemVolumeType	设置系统音量类型（仅适用于移动端）

废弃接口

函数列表	描述
setListener	设置 TRTC 事件回调
setBeautyStyle	设置美颜、美白以及红润效果级别
setEyeScaleLevel	设置大眼级别
setFaceSlimLevel	设置瘦脸级别
setFaceVLevel	设置 V 脸级别
setChinLevel	设置下巴拉伸或收缩幅度
setFaceShortLevel	设置短脸级别
setNoseSlimLevel	设置瘦鼻级别
selectMotionTmpl	设置动效贴纸
setMotionMute	设置动效静音
setFilter	设置色彩滤镜效果
setFilterConcentration	设置色彩滤镜浓度
setGreenScreenFile	设置绿幕背景视频
setReverbType	设置混响效果

版权所有：腾讯云计算（北京）有限责任公司

setVoiceChangerType	设置变声类型
enableAudioEarMonitoring	开启（或关闭）耳返
enableAudioVolumeEvaluation	启用音量大小提示
switchCamera	切换摄像头
isCameraZoomSupported	查询当前摄像头是否支持缩放
setZoom	设置摄像头缩放倍数（焦距）
isCameraTorchSupported	查询是否支持开关闪光灯
enableTorch	开关/关闭闪光灯
isCameraFocusPositionInPreviewSupported	查询摄像头是否支持设置焦点
setFocusPosition	设置摄像头焦点坐标位置
isCameraAutoFocusFaceModeSupported	查询是否支持自动识别人脸位置
setSystemVolumeType	设置系统音量类型
checkAudioCapabilitySupport	查询是否支持音频某种能力（仅适用于Android）
startLocalAudio	设置音频质量
startRemoteView	开始显示远端视频画面
stopRemoteView	停止显示远端视频画面，同时不再拉取该远端用户的视频数据流
setLocalViewFillMode	设置本地图像的渲染模式
setLocalViewRotation	设置本地图像的顺时针旋转角度
setLocalViewMirror	设置本地摄像头预览画面的镜像模式
setRemoteViewFillMode	设置远端图像的渲染模式
setRemoteViewRotation	设置远端图像的顺时针旋转角度
startRemoteSubStreamView	开始显示远端用户的辅路画面
stopRemoteSubStreamView	停止显示远端用户的辅路画面

版权所有：腾讯云计算（北京）有限责任公司

setRemoteSubStreamViewFillMode	设置辅路画面的填充模式
setRemoteSubStreamViewRotation	设置辅路画面的顺时针旋转角度
setAudioQuality	设置音频质量
setPriorRemoteVideoStreamType	设定优先观看大画面还是小画面
setMicVolumeOnMixing	设置麦克风音量大小
playBGM	启动播放背景音乐
stopBGM	停止播放背景音乐
pauseBGM	停止播放背景音乐
resumeBGM	停止播放背景音乐
getBGMDuration	获取背景音乐总时长（单位：毫秒）
setBGMPosition	设置背景音乐的播放进度
setBGMVolume	设置背景音乐的音量大小
setBGMPlayoutVolume	设置背景音乐的本地播放音量
setBGMPublishVolume	设置背景音乐的远端播放音量
playAudioEffect	播放音效
setAudioEffectVolume	设置音效音量
stopAudioEffect	停止播放音效
stopAllAudioEffects	停止所有音效
setAllAudioEffectsVolume	设置所有音效音量
pauseAudioEffect	暂停音效
resumeAudioEffect	暂停音效
enableCustomVideoCapture	启用视频自定义采集模式
sendCustomVideoData	投送自己采集的视频数据
muteLocalVideo	暂停/恢复发布本地的视频流

版权所有：腾讯云计算（北京）有限责任公司

muteRemoteVideoStream	暂停 / 恢复订阅远端用户的视频流
snapshotVideo	视频画面截图
startSpeedTest	开始进行网络测速（进入房间前使用）
startScreenCapture	启动屏幕分享
setVideoEncoderRotation	设置视频编码器输出的画面方向
setVideoEncoderMirror	设置编码器输出的画面镜像模式
setGSensorMode	设置重力感应的适配模式

版权所有：腾讯云计算（北京）有限责任公司

TRTCCloud

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTCCloud @ TXLiteAVSDK
Function: 腾讯云 TRTC 主功能接口
Version: 11.7

TRTCCloud

TRTCCloud

函数列表	描述
sharedInstance	创建 TRTCCloud 实例（单例模式）
destroySharedInstance	销毁 TRTCCloud 实例（单例模式）
addListener	添加 TRTC 事件回调
removeListener	移除 TRTC 事件回调
setListenerHandler	设置驱动 TRTCCloudListener 事件回调的队列
enterRoom	进入房间
exitRoom	离开房间
switchRole	切换角色
switchRole	切换角色（支持设置权限位）
switchRoom	切换房间
ConnectOtherRoom	请求跨房通话
DisconnectOtherRoom	退出跨房通话
setDefaultStreamRecvMode	设置订阅模式（需要在进入房前设置才能生效）
createSubCloud	创建子房间实例（用于多房间并发观看）

destroySubCloud	销毁子房间实例
updateOtherRoomForwardMode	更改跨房主播在本房间的上行能力
startPublishing	开始向腾讯云直播 CDN 上发布音视频流
stopPublishing	停止向腾讯云直播 CDN 上发布音视频流
startPublishCDNStream	开始向非腾讯云 CDN 上发布音视频流
stopPublishCDNStream	停止向非腾讯云 CDN 上发布音视频流
setMixTranscodingConfig	设置云端混流的排版布局和转码参数
startPublishMediaStream	开始发布媒体流
updatePublishMediaStream	更新发布媒体流
stopPublishMediaStream	停止发布媒体流
startLocalPreview	开启本地摄像头的预览画面（移动端）
updateLocalView	更新本地摄像头的预览画面
stopLocalPreview	停止摄像头预览
muteLocalVideo	暂停/恢复发布本地的视频流
setVideoMuteImage	设置本地画面被暂停期间的替代图片
startRemoteView	订阅远端用户的视频流，并绑定视频渲染控件
updateRemoteView	更新远端用户的视频渲染控件
stopRemoteView	停止订阅远端用户的视频流，并释放渲染控件
stopAllRemoteView	停止订阅所有远端用户的视频流，并释放全部渲染资源
muteRemoteVideoStream	暂停/恢复订阅远端用户的视频流
muteAllRemoteVideoStreams	暂停/恢复订阅所有远端用户的视频流
setVideoEncoderParam	设置视频编码器的编码参数
setNetworkQosParam	设置网络质量控制的相关参数
setLocalRenderParams	设置本地画面的渲染参数
setRemoteRenderParams	设置远端画面的渲染模式

版权所有：腾讯云计算（北京）有限责任公司

enableEncSmallVideoStream	开启大小画面双路编码模式
setRemoteVideoStreamType	切换指定远端用户的大小画面
snapshotVideo	视频画面截图
setPerspectiveCorrectionPoints	视频画面透视校正坐标设置
setGravitySensorAdaptiveMode	设置重力感应的适配模式（11.7 及以上版本）
startLocalAudio	开启本地音频的采集和发布
stopLocalAudio	停止本地音频的采集和发布
muteLocalAudio	暂停/恢复发布本地的音频流
muteRemoteAudio	暂停/恢复播放远端的音频流
muteAllRemoteAudio	暂停/恢复播放所有远端用户的音频流
setAudioRoute	设置音频路由
setRemoteAudioVolume	设定某一个远端用户的声音播放音量
setAudioCaptureVolume	设定本地音频的采集音量
getAudioCaptureVolume	获取本地音频的采集音量
setAudioPlayoutVolume	设定远端音频的播放音量
getAudioPlayoutVolume	获取远端音频的播放音量
enableAudioVolumeEvaluation	启用音量大小提示
startAudioRecording	开始录音
stopAudioRecording	停止录音
startLocalRecording	开启本地媒体录制
stopLocalRecording	停止本地媒体录制
setRemoteAudioParallelParams	设置远端音频流智能并发播放策略
enable3DSpatialAudioEffect	启用 3D 音效
updateSelf3DSpatialPosition	设置 3D 音效中自身坐标及朝向信息
updateRemote3DSpatialPosition	设置 3D 音效中远端用户坐标信息

版权所有：腾讯云计算（北京）有限责任公司

set3DSpatialReceivingRange	设置指定用户所发出声音的可被接收范围
getDeviceManager	获取设备管理类（TXDeviceManager）
getBeautyManager	获取美颜管理类（TXBeautyManager）
setWatermark	添加水印
getAudioEffectManager	获取音效管理类（TXAudioEffectManager）
startSystemAudioLoopback	开启系统声音采集
stopSystemAudioLoopback	停止系统声音采集（iOS 端暂未支持）
startScreenCapture	启动屏幕分享
stopScreenCapture	停止屏幕分享
pauseScreenCapture	暂停屏幕分享
resumeScreenCapture	恢复屏幕分享
setSubStreamEncoderParam	设置屏幕分享（即辅路）的视频编码参数（桌面系统和移动系统均已支持）
enableCustomVideoCapture	启用/关闭视频自定义采集模式
sendCustomVideoData	向 SDK 投送自己采集的视频帧
enableCustomAudioCapture	启用音频自定义采集模式
sendCustomAudioData	向 SDK 投送自己采集的音频数据
enableMixExternalAudioFrame	启用/关闭自定义音轨
mixExternalAudioFrame	向 SDK 混入自定义音轨
setMixExternalAudioVolume	设置推流时混入外部音频的推流音量和播放音量
generateCustomPTS	生成自定义采集时的时间戳
setLocalVideoProcessListener	设置第三方美颜的视频数据回调
setLocalVideoRenderListener	设置本地视频自定义渲染回调
setRemoteVideoRenderListener	设置远端视频自定义渲染回调
setAudioFrameListener	设置音频数据自定义回调
setCapturedAudioFrameCallback	设置本地麦克风采集出的音频帧回调格式

版权所有：腾讯云计算（北京）有限责任公司

Format	
setLocalProcessedAudioFrameCallbackFormat	设置经过前处理后的本地音频帧回调格式
setMixedPlayAudioFrameCallbackFormat	设置最终要由系统播放出的音频帧回调格式
enableCustomAudioRendering	开启音频自定义播放
getCustomAudioRenderingFrame	获取可播放的音频数据
sendCustomCmdMsg	使用 UDP 通道发送自定义消息给房间内所有用户
sendSEIMsg	使用 SEI 通道发送自定义消息给房间内所有用户
startSpeedTest	开始进行网速测试（进入房间前使用）
stopSpeedTest	停止网络测速
getSDKVersion	获取 SDK 版本信息
setLogLevel	设置 Log 输出级别
setConsoleEnabled	启用/禁用控制台日志打印
setLogCompressEnabled	启用/禁用日志的本地压缩
setLogDirPath	设置本地日志的保存路径
setLogListener	设置日志回调
showDebugView	显示仪表盘
TRTCViewMargin	设置仪表盘的边距
callExperimentalAPI	调用实验性接口
enablePayloadPrivateEncryption	开启或关闭媒体流私有加密

sharedInstance

sharedInstance

TRTCCloud sharedInstance (Context context)

创建 TRTCCloud 实例（单例模式）

参数	描述
context	仅适用于 Android 平台，SDK 内部会将其转化为 Android 平台的 ApplicationContext 用于调用 Android System API。 如果传入的 context 参数为空，SDK 内部会自动获取当前进程的 ApplicationContext。

注意

1. 如果您使用 delete ITRTCCloud* 会导致编译错误，请使用 destroyTRTCCloud 释放对象指针。
2. 在 Windows、Mac 和 iOS 平台上，请调用 getTRTCShareInstance() 接口。
3. 在 Android 平台上，请调用 getTRTCShareInstance(void *context) 接口。

destroySharedInstance

destroySharedInstance

销毁 TRTCCloud 实例（单例模式）

addListener

addListener

void addListener (TRTCCloudListener listener)

添加 TRTC 事件回调

您可以通过 TRTCCloudListener 获得来自 SDK 的各类事件通知（比如：错误码，警告码，音视频状态参数等）。

removeListener

removeListener

void removeListener (TRTCCloudListener listener)

移除 TRTC 事件回调

setListenerHandler

setListenerHandler

void setListenerHandler (Handler listenerHandler)

设置驱动 TRTCCloudListener 事件回调的队列

如果您不指定自己的 listenerHandler 默认会采用 MainQueue 作为驱动 TRTCCloudListener 事件回调的队列。
也就是当您不设置 listenerHandler 属性时，TRTCCloudListener 中的所有回调函数都是由 MainQueue 来驱动的。

参数	描述
listenerHandler	事件回调句柄

注意
如果您指定了自己的 listenerHandler TRTCCloudListener 回调函数中操作 UI，否则会引发线程安全问题。

enterRoom

enterRoom

void enterRoom (TRTCCloudDef.TRTCParams param, int scene)

进入房间

TRTC 的所有用户都需要进入房间才能“发布”或“订阅”音视频流，“发布”是指将自己的音频和视频推送到云端，“订阅”是指从云端拉取房间里其他用户的音视频流。
调用该接口时，您需要指定您的应用场景 TRTCAppScene 以获取最佳的音视频传输体验，这些场景可以分成两大类：

实时通话：
包括 TRTC_APP_SCENE_VIDEOCALL 和 TRTC_APP_SCENE_AUDIOCALL 两个可选项，分别是视频通话和语音通话，该模式适合 1对1 的音视频通话，或者参会人数在 300 人以内的在线会议。

在线直播：
包括 TRTC_APP_SCENE_LIVE 和 TRTC_APP_SCENE_VOICE_CHATROOM 两个可选项，分别是视频直播和语音直播，该模式适合十万人以内的在线直播场景，但需要您在接下来介绍的 TRTCParams 参数中指定 **角色(role)** 这个字段，也就是将房间中的用户区分为 **主播**（TRTCRoleAnchor）和 **观众**（TRTCRoleAudience）两种不同的角色。

调用该接口后，您会收到来自 TRTCCloudListener 中的 onEnterRoom(result) 回调：

- 如果进房成功，参数 result 会是一个正数（result > 0），表示从函数调用到进入房间所花费的时间，单位是毫秒（ms）。
- 如果进房失败，参数 result 会是一个负数（result < 0），表示进房失败的错误码。

参数	描述
param	进房参数，用于指定用户的身份、角色和安全票据等信息，详情请参见 TRTCParams 。
scene	应用场景，用于指定您的业务场景，同一个房间内的所有用户需要设定相同的 TRTCAppScene 。

注意
1. 同一个房间内的所有用户需要设定相同的 scene。不同的 scene 会导致偶现的异常问题。
2. 当您指定参数 scene 为 TRTC_APP_SCENE_LIVE 或 TRTC_APP_SCENE_VOICE_CHATROOM 时，您必须通过 TRTCParams 中的“role” 字段为当前用户设定他/她在房间中的角色。
3. 请您尽量保证 enterRoom 与 exitRoom 前后配对使用，即保证”先退出前一个房间再进入下一个房间”，否则会导致很多异常问题。

exitRoom

exitRoom

离开房间

调用该接口会让用户离开自己所在的音视频房间，并释放摄像头、麦克风、扬声器等设备资源。
等资源释放完毕之后，SDK 会通过 TRTCCloudListener 中的 onExitRoom 回调向您通知。

如果您要再次调用 enterRoom 或者切换到其他的供应商的 SDK，建议等待 onExitRoom 回调到来之后再执行之后的操作，以避免摄像头或麦克风被占用的问题。

switchRole

switchRole

void switchRole (int role)

切换角色

调用本接口可以实现用户在 主播 和 观众 两种角色之间来回切换。

由于视频直播和语音聊天室需要支持多达10万名观众同时观看，所以设定了 **只有主播才能发布自己的音视频** 的规则。因此，当有些观众希望发布自己的音视频流（以便能跟主播互动）时，就需要先把自己的角色切换成 **主播**。

您可以在进入房间时通过 TRTCParams 中的 role 字段事先确定用户的角色，也可以在进入房间后通过 switchRole 接口动态切换角色。

参数	描述
role	角色，默认为 **主播**。 • TRTCRoleAnchor ：主播，可以发布自己的音视频，同一个房间里最多支持50个主播同时发布音视频。 • TRTCRoleAudience ：观众，不能发布自己的音视频流，只能观看房间中其他主播的音视频。如果要发布自己的音视频，需要先通过 switchRole 切换成 **主播**，同一个房间内同时最多可以容纳 10 万名观众。

注意

1. 该接口仅适用于视频直播（TRTC_APP_SCENE_LIVE）和语音聊天室（TRTC_APP_SCENE_VOICE_CHATROOM）这两个场景。

2. 如果您在 enterRoom 时指定的 scene 为 TRTC_APP_SCENE_VIDEOCALL 或 TRTC_APP_SCENE_AUDIOCALL，请不要调用这个接口。

switchRole

switchRole

```
void switchRole (int role
                 final String privateMapKey)
```

切换角色（支持设置权限位）

调用本接口可以实现用户在 主播 和 观众 两种角色之间来回切换。

由于视频直播和语音聊天室需要支持多达10万名观众同时观看，所以设定了“只有主播才能发布自己的音视频”的规则。
因此，当有些观众希望发布自己的音视频流（以便能跟主播互动）时，就需要先把自己的角色切换成“主播”。
您可以在进入房间时通过 TRTCParams 中的 role 字段事先确定用户的角色，也可以在进入房间后通过 switchRole 接口动态切换角色。

参数	描述
privateMapKey	用于权限控制的权限票据，当您希望某个房间只能让特定的 userId 进入或者上行视频时，需要使用 privateMapKey 进行权限保护。 • 仅建议有高级别安全需求的客户使用，更多详情请参见 开启高级权限控制 。
role	角色，默认为“主播”： • TRTCRoleAnchor ：主播，可以发布自己的音视频，同一个房间里最多支持50个主播同时发布音视频。 • TRTCRoleAudience ：观众，不能发布自己的音视频流，只能观看房间中其他主播的音视频。如果要发布自己的音视频，需要先通过 switchRole 切换成“主播”，同一个房间内同时最多可以容纳 10 万名观众。

注意

1. 该接口仅适用于视频直播（TRTC_APP_SCENE_LIVE）和语音聊天室（TRTC_APP_SCENE_VOICE_CHATROOM）这两个场景。
2. 如果您在 enterRoom 时指定的 scene 为 TRTC_APP_SCENE_VIDEOCALL 或 TRTC_APP_SCENE_AUDIOCALL，请不要调用这个接口。

switchRoom

switchRoom

void switchRoom	(final TRTCCloudDef.TRTCSwitchRoomConfig config)

切换房间

使用该接口可以让用户可以快速从一个房间切换到另一个房间。

- 如果用户的身份是“观众”，该接口的调用效果等同于 exitRoom(当前房间) + enterRoom（新的房间）。
- 如果用户的身份是“主播”，该接口在切换房间的同时还会保持自己的音视频发布状态，因此在房间切换过程中，摄像头的预览和声音的采集都不会中断。

该接口适用于在线教育场景中，监课老师在多个房间中进行快速切换的场景。在该场景下使用 switchRoom 可以获得比 exitRoom+enterRoom 更好的流畅性和更少的代码量。
接口调用结果会通过 TRTCCloudListener 中的 onSwitchRoom(errCode, errMsg) 回调。

参数	描述
config	房间参数，详情请参见 TRTCSwitchRoomConfig 。

注意
由于对老版本 SDK 兼容的需求，参数 config 中同时包含 roomId 与 strRoomId 两个参数，这两个参数的填写格外讲究，请注意如下事项：
1. 若您选用 strRoomId，则 roomId 需要填写为0。若两者都填，将优先选用 roomId。
2. 所有房间需要同时使用 strRoomId 或同时使用 roomId，不可混用，否则将会出现很多预期之外的 bug。

ConnectOtherRoom

ConnectOtherRoom

void	(String param)

ConnectOtherRoom

请求跨房通话

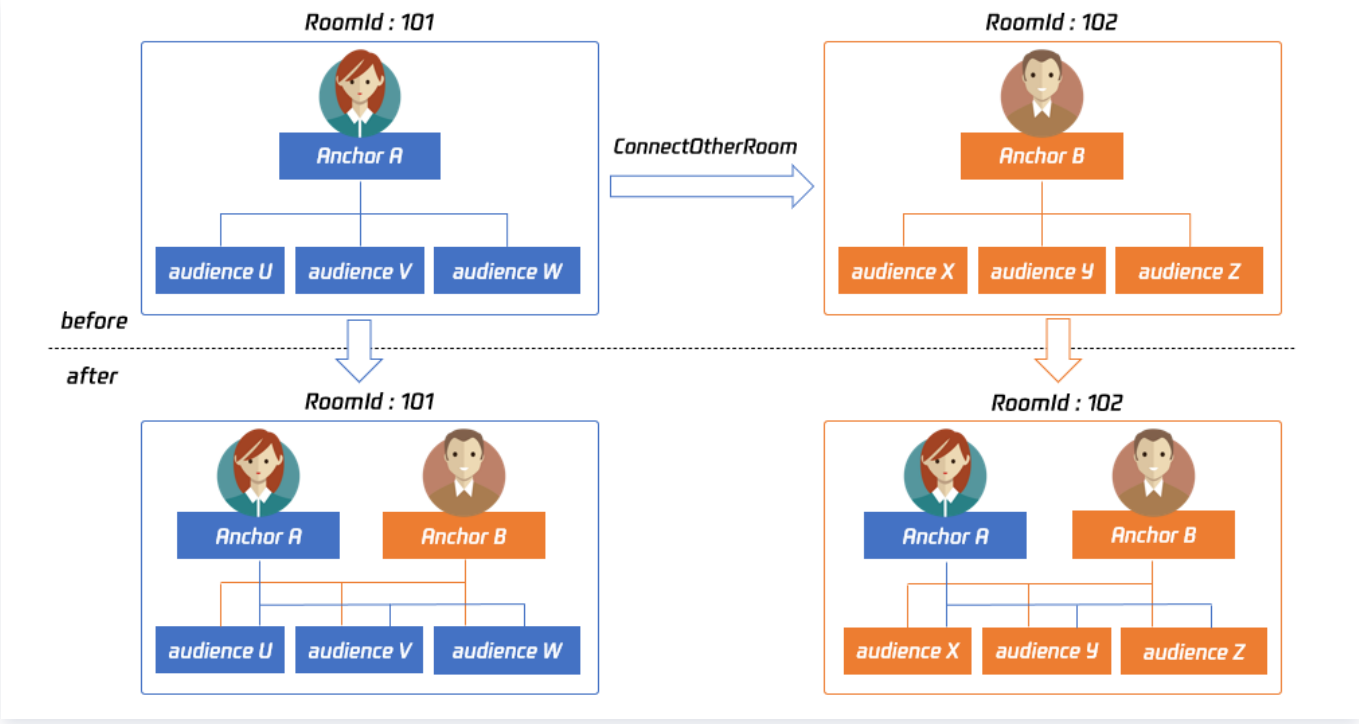
默认情况下，只有同一个房间中的用户之间可以进行音视频通话，不同的房间之间的音视频流是相互隔离的。但您可以通过调用该接口，将另一个房间中某个主播音视频流发布到自己所在的房间中，与此同时，该接口也会将自己的音视频流发布到目标主播的房间中。

也就是说，您可以使用该接口让身处两个不同房间中的主播进行跨房间的音视频流分享，从而让每个房间中的观众都能观看到这两个主播的音视频。该功能可以用来实现主播之间的 PK 功能。

跨房通话的请求结果会通过 TRTCCloudListener 中的 onConnectOtherRoom 回调通知给您。

例如：当房间“101”中的主播 A 通过 connectOtherRoom() 跟房间“102”中的主播 B 建立跨房通话后，

- 房间“101”中的用户都会收到主播 B 的 onRemoteUserEnterRoom(B) 和 onUserVideoAvailable(B,true) 这两个事件回调，即房间“101”中的用户都可以订阅主播 B 的音视频。
- 房间“102”中的用户都会收到主播 A 的 onRemoteUserEnterRoom(A) 和 onUserVideoAvailable(A,true) 这两个事件回调，即房间“102”中的用户都可以订阅主播 A 的音视频。



跨房通话的参数考虑到后续扩展字段的兼容性问题，暂时采用了 JSON 格式的参数：

情况一：数字房间号
如果房间“101”中的主播 A 要跟房间“102”中的主播 B 连麦，那么主播 A 调用该接口时需要传入：{"roomId": 102, "userId": "userB"}
示例代码如下：

```
JSONObject jsonObj = new JSONObject();
jsonObj.put("roomId", 102);
jsonObj.put("userId", "userB");
trtc.ConnectOtherRoom(jsonObj.toString());
```

情况二：字符串房间号
如果您使用的是字符串房间号，务必请将 json 中的 “roomId” 替换成 “strRoomId”：{"strRoomId": "102", "userId": "userB"}
示例代码如下：

```
JSONObject jsonObj = new JSONObject();
jsonObj.put("strRoomId", "102");
jsonObj.put("userId", "userB");
trtc.ConnectOtherRoom(jsonObj.toString());
```

参数	描述
param	需要你传入 JSON 格式的字符串参数，roomId 代表数字格式的房间号，strRoomId 代表字符串格式的房间号，userId 代表目标主播的用户 ID。

DisconnectOtherRoom

DisconnectOtherRoom

退出跨房通话

退出结果会通过 TRTCCloudListener 中的 onDisconnectOtherRoom 回调通知给您。

setDefaultStreamRecvMode

setDefaultStreamRecvMode

void setDefaultStreamRecvMode	(boolean autoRecvAudio boolean autoRecvVideo)

设置订阅模式（需要在进入房前设置才能生效）

您可以通过该接口在“自动订阅”和“手动订阅”两种模式下进行切换：

- 自动订阅：默认模式，用户在进入房间后会立刻接收到该房间中的音视频流，音频会自动播放，视频会自动开始解码（依然需要您通过 startRemoteView 接口绑定渲染控件）。
- 手动订阅：在用户进入房间后，需要手动调用 startRemoteView 接口才能启动视频流的订阅和解码，需要手动调用 muteRemoteAudio (false) 接口才能启动声音的播放。

在绝大多数场景下，用户进入房间后都会订阅房间中所有主播的音视频流，因此 TRTC 默认采用了自动订阅模式，以求得最佳的“秒开体验”。
如果您的应用场景中每个房间同时会有很多路音视频流在发布，而每个用户只想选择性地订阅其中的 1-2 路，则推荐使用“手动订阅”模式以节省流量费用。

参数	描述
autoRecvAudio	true：自动订阅音频；false：需手动调用 muteRemoteAudio(false) 订阅音频。默认值：true。
autoRecvVideo	true：自动订阅视频；false：需手动调用 startRemoteView 订阅视频。默认值：true。

注意

1. 需要在进入房间前调用该接口进行设置才能生效。
2. 在自动订阅模式下，如果用户在进入房间后没有调用 startRemoteView 订阅视频流，SDK 会自动停止订阅视频流，以便达到节省流量的目的。

createSubCloud

createSubCloud

创建子房间实例（用于多房间并发观看）

TRTCCloud 一开始被设计成单例模式，限制了多房间并发观看的能力。

通过调用该接口，您可以创建出多个 TRTCCloud 实例，以便同时进入多个不同的房间观看音视频流。

但需要注意的是，您在多个 TRTCCloud 实例中发布音视频流的能力会受到一定限制。

该功能主要用于在线教育场景中一种被称为“超级小班课”的业务场景中，用于解决“每个 TRTC 的房间中最多只能有 50 人同时发布自己音视频流”的限制。

示例代码如下：

```
    //In the small room that needs interaction, enter the room as an anchor and
push audio and video streams
    TRTCCloud mainCloud = TRTCCloud.sharedInstance(mContext);
    TRTCCloudDef.TRTCParams mainParams = new TRTCCloudDef.TRTCParams();
    //Fill your params
    mainParams.role = TRTCCloudDef.TRTCRoleAnchor;
    mainCloud.enterRoom(mainParams, TRTCCloudDef.TRTC_APP_SCENE_LIVE);
    //...
    mainCloud.startLocalPreview(true, videoView);
    mainCloud.startLocalAudio(TRTCCloudDef.TRTC_AUDIO_QUALITY_DEFAULT);

    //In the large room that only needs to watch, enter the room as an audience
and pull audio and video streams
    TRTCCloud subCloud = mainCloud.createSubCloud();
    TRTCCloudDef.TRTCParams subParams = new TRTCCloudDef.TRTCParams();
    //Fill your params
    subParams.role = TRTCCloudDef.TRTCRoleAudience;
    subCloud.enterRoom(subParams, TRTCCloudDef.TRTC_APP_SCENE_LIVE);
    //...
    subCloud.startRemoteView(userId,
TRTCCloudDef.TRTC_VIDEO_STREAM_TYPE_BIG, view);
    //...
    //Exit from new room and release it.
    subCloud.exitRoom();
    mainCloud.destroySubCloud(subCloud);
```

注意

- 同一个用户，可以使用同一个 userId 进入多个不同 roomId 的房间。
- 两台不同的终端设备不可以同时使用同一个 userId 进入同一个 roomId 的房间。
- 您可以分别为不同实例分别设置 TRTCCloudListener 获取各自的事件通知。
- 同一个用户可以在多个 TRTCCloud 实例中推流，也可以调用子实例中与本地音视频相关的接口。但需要注意：

- 多实例的音频需要同时为麦克风采集或自定义采集，而且与音频设备相关的接口调用会以最后一次为准；
- 与摄像头相关的调用会以最后一次为准：startLocalPreview 。

返回值说明：

子 TRTCCloud 实例

destroySubCloud

destroySubCloud

void destroySubCloud (final TRTCCloud subCloud)

销毁子房间实例

参数	描述
subCloud	子房间实例

updateOtherRoomForwardMode

updateOtherRoomForwardMode

void updateOtherRoomForwardMode (String param)

更改跨房主播在本房间的上行能力

通常情况下，在调用 connectOtherRoom 接口与另一个房间的主播进行跨房通话后，本房间内的所有观众都将收到该主播发布的音视频流。
您可以通过调用该接口，限制跨房主播在本房间内的上行能力，禁止或允许跨房主播发布音频/主路视频/辅路视频，该行为会影响房间内的所有用户。
在禁用跨房主播某种上行能力后，本房间内所有用户将无法收到对应音视频流，且无法再订阅对应的音视频。

需要注意的是，该接口只能由进行跨房通话的主播调用，且通过该接口设置的限制会因为跨房通话的中断或对应主播退房重置。

该接口的调用结果会通过 TRTCCloudListener 中的 onUpdateOtherRoomForwardMode 回调通知给您。

例如：
房间“101”中有主播 A 与观众 B，房间“102”中有主播 C，主播 C 正常发布音视频流。主播 A 通过 connectOtherRoom() 跟主播 C 建立跨房通话。

- 此时，主播 A 与观众 B 都会收到主播 C 的 onRemoteUserEnterRoom(C) 和 onUserVideoAvailable(C,true)、onUserAudioAvailable(C,true) 这三个事件回调，可以订阅主播 C 的音视频。

之后，主播 A 调用该接口禁用主播 C 在本房间发布音频的能力。

- 此后，主播 C 的音频流将无法发布至房间“101”，主播 A 与观众 B 都将收到 onUserAudioAvailable(C,false) 事件回调，且无法再通过 muteRemoteAudio(C,false) 订阅主播 C 的音频。
- 主播 C 的视频流将不受影响。房间“102”中的其他观众也不受影响，可以正常订阅主播 C 的音频。

该接口的参数考虑到后续扩展字段的兼容性问题，暂时采用了 JSON 格式的参数，示例如下：
情况一：数字房间号

```
{
 "roomId":102,
 "userId":"userC",
 "muteAudio":false,
 "muteVideo":true,
 "muteSubStream":false
}
```

情况二：字符串房间号

```
{
 "strRoomId":"102",
 "userId":"userC",
 "muteAudio":false,
 "muteVideo":true,
 "muteSubStream":false
}
```

参数	描述
param	需要你传入 JSON 格式的字符串参数，roomId 代表数字格式的房间号，strRoomId 代表字符串格式的房间号，userId 代表目标主播的用户 ID，muteAudio、muteVideo、muteSubStream均为可选项，分表代表禁止或允许跨房主播发布音频/主路视频/辅路视频的能力。

startPublishing

startPublishing

```
void startPublishing (final String streamId
                      final int streamType)
```

开始向腾讯云直播 CDN 上发布音视频流

该接口会向 TRTC 服务器发送指令，要求其将当前用户的音视频流旁路到直播 CDN 上。您可以通过参数 streamId 设定直播流的 StreamId，从而可以指定该用户的音视频流对应在直播 CDN 上的播放地址。

例如：您可以通过该接口将当前用户的直播流 ID 指定为 user_stream_001，那么该用户音视频流对应的 CDN 播放地址为：
“http://yourdomain/live/user_stream_001.flv”，其中 yourdomain 为您自己备案的播放域名，

您可以在直播控制台 配置您的播放域名，腾讯云不提供默认的播放域名。

您也可以在设置 enterRoom 的参数 TRTCParams 时指定 streamId, 而且我们更推荐您采用这种方案。

参数	描述
streamId	自定义流 ID。
streamType	仅支持 TRTC_VIDEO_STREAM_TYPE_BIG 和 TRTC_VIDEO_STREAM_TYPE_SUB 。

注意
您需要提前在 实时音视频控制台 中的功能配置页面上开启“启用旁路推流”才能生效。
- 若您选择“指定流旁路”，则您可以通过该接口将对应音视频流推送到腾讯云 CDN 且指定为填写的流 ID。

- 若您选择“全局自动旁路”，则您可以通过该接口调整默认的流 ID。

stopPublishing

stopPublishing

停止向腾讯云直播 CDN 上发布音视频流

startPublishCDNStream

startPublishCDNStream

void startPublishCDNStream (TRTCCloudDef.TRTCPublishCDNParam param)

开始向非腾讯云 CDN 上发布音视频流

该接口跟 startPublishing 功能类似，不同之处在于，startPublishing 仅支持向腾讯云的 CDN 发布，而本接口支持向非腾讯云的直播 CDN 上转推音视频流。

参数	描述
param	CDN 转推参数，详情请参见 TRTCPublishCDNParam

注意

- 使用 startPublishing 接口向腾讯云的直播 CDN 上发布音视频流不会收取额外费用。
- 使用 startPublishCDNStream 接口向非腾讯云的直播 CDN 上发布音视频流，需要收取额外的转推带宽费用。

stopPublishCDNStream

stopPublishCDNStream

停止向非腾讯云 CDN 上发布音视频流

setMixTranscodingConfig

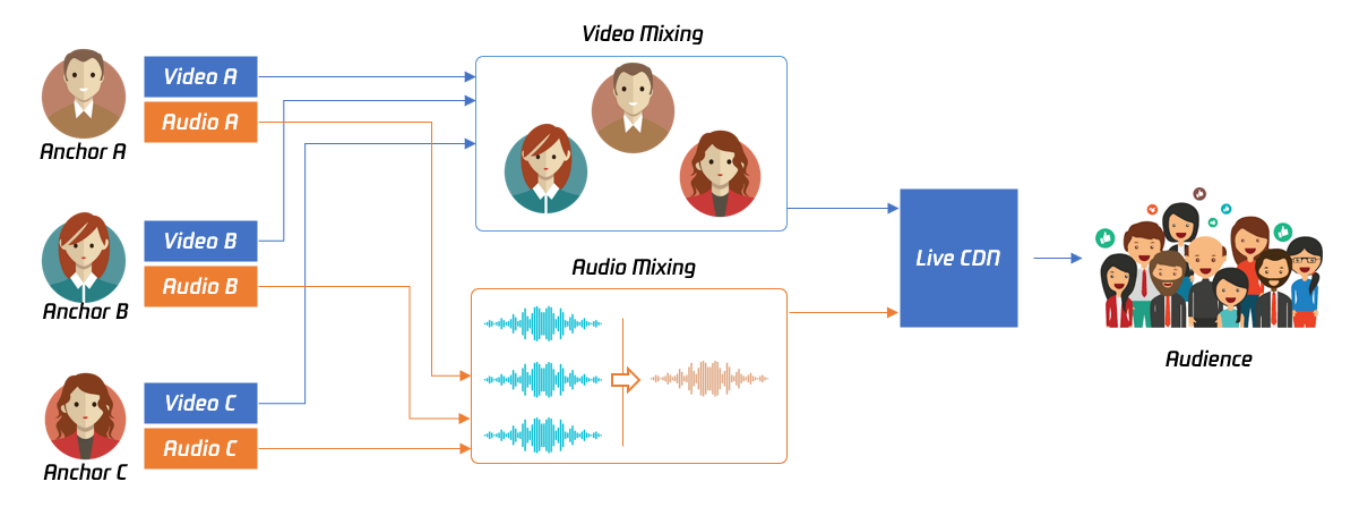
setMixTranscodingConfig

void setMixTranscodingConfig (TRTCCloudDef.TRTCTranscodingConfig config)

设置云端混流的排版布局和转码参数

在一个直播间中可能同时会有多个主播发布自己的音视频流，但对于直播 CDN 上的观众而言，只需要观看一条 HTTP-FLV 或 HLS 格式的视频流即可。
当您调用本接口函数时，SDK 会向腾讯云的 TRTC 混流服务器发送一条指令，混流服务器会将房间里的多路音视频流混合成一路。
您可以通过 TRTCTranscodingConfig 参数来调整每一路画面的排版布局，也可以设置混合后的音视频流的各项编码参数。

参考文档：云端混流转码 。



参数	描述
config	如果 config 不为空，则开启云端混流，如果 config 为空则停止云端混流。详情请参见 TRTCTranscodingConfig 。

注意
关于云端混流的注意事项：

- 混流转码为收费功能，调用接口将产生云端混流转码费用，详见 云端混流转码计费说明 。
- 调用该接口的用户，如果没设定 config 参数中的 streamId 字段，TRTC 会将房间中的多路画面混合到当前用户所对应的音视频流上，即 A + B => A。
- 调用该接口的用户，如果设定了 config 参数中的 streamId 字段，TRTC 会将房间中的多路画面混合到您指定的 streamId 上，即 A + B => streamId。
- 请注意，若您还在房间中且不再需要混流，请务必再次调用本接口并将 config 设置为空以进行取消，不及时取消混流可能会引起不必要的计费损失。
- 请放心，当您退房时 TRTC 会自动取消混流状态。

startPublishMediaStream

startPublishMediaStream

```
void startPublishMediaStream (TRTCCloudDef.TRTCPublishTarget target
                              TRTCCloudDef.TRTCStreamEncoderParam params
                              TRTCCloudDef.TRTCStreamMixingConfig config)
```

开始发布媒体流

该接口会向 TRTC 服务器发送指令，要求其将当前用户的音视频流转推/转码到直播 CDN 或者回推到 TRTC 房间中您可以通过 TRTCPublishTarget 配置中的 TRTCPublishMode 指定具体的发布模式

参数	描述
config	媒体流转码配置参数。具体配置参考 TRTCStreamMixingConfig 。转码和回推到 TRTC 房间中时为必填项，您需要指定您预期的转码配置参数。转推模式下则无效。
params	媒体流编码输出参数，具体配置参考 TRTCStreamEncoderParam 。转码和回推到 TRTC 房间中时为必填项，您需要指定您预期的转码输出参数。在转推时，为了更好的转推稳定性和 CDN 兼容性，也建议您进行配置。
target	媒体流发布的目标地址，具体配置参考 TRTCPublishTarget 。支持转推/转码到腾讯或者第三方 CDN，也支持转码回推到 TRTC 房间中。

注意

1. SDK 会通过回调 onStartPublishMediaStream 带给您后台启动的任务标识（即 taskId）。
2. 同一个任务（TRTCPublishMode 与 TRTCPublishCdnUrl 均相同）仅支持启动一次。若您后续需要更新或者停止该项任务，需要记录并使用返回的 taskId，通过

updatePublishMediaStream 或者 stopPublishMediaStream 来操作。
3. target 支持同时配置多个 CDN URL（最多同时 10 个）。若您的同一个转推/转码任务需要发布至多路 CDN，则仅需要在 target 中配置多个 CDN URL 即可。同一个转码任务即使有多个转推地址，对应的转码计费仍只收取一份。
4. 使用时需要注意不要多个任务同时往相同的 URL 地址推送，以免引起异常推流状态。一种推荐的方案是 URL 中使用 “sdkappid_roomid_userid_main” 作为区分标识，这种命名方式容易辨认且不会在您的多个应用中发生冲突。

updatePublishMediaStream

updatePublishMediaStream

```
void updatePublishMediaStream(final String taskId,
                              TRTCCloudDef.TRTCPublishTarget target,
                              TRTCCloudDef.TRTCStreamEncoderParam params,
                              TRTCCloudDef.TRTCStreamMixingConfig config)
```

更新发布媒体流

该接口会向 TRTC 服务器发送指令，更新通过 startPublishMediaStream 启动的媒体流

参数	描述
config	媒体流转码配置参数。具体配置参考 TRTCStreamMixingConfig 。转码和回推到 TRTC 房间中时为必填项，您需要指定您预期的转码配置参数。转推模式下则无效。
params	媒体流编码输出参数，具体配置参考 TRTCStreamEncoderParam 。转码和回推到 TRTC 房间中时为必填项，您需要指定您预期的转码输出参数。在转推时，为了更好的转推稳定性和 CDN 兼容性，也建议您进行配置。
target	媒体流发布的目标地址，具体配置参考 TRTCPublishTarget 。支持转推/转码到腾讯或者第三方 CDN，也支持转码回推到 TRTC 房间中。
taskId	通过回调 onStartPublishMediaStream 带给您后台启动的任务标识（即 taskId）

注意

1. 您可以通过本接口来更新发布的 CDN URL（支持增删，最多同时 10 个），但您使用时需要注意不要多个任务同时往相同的 URL 地址推送，以免引起异常推流状态。
2. 您可以通过 taskId 来更新调整转推/转码任务。例如在 pk 业务中，您可以先通过 startPublishMediaStream 发起转推，接着在主播发起 pk 时，通过 taskId 和本接口将转推更新为转码任务。此时，CDN 播放将连续并且不会发生断流（您需要保持媒体流编码输出参数 param 一致）。
3. 同一个任务不支持纯音频、音视频、纯视频之间的切换。

stopPublishMediaStream

stopPublishMediaStream

void stopPublishMediaStream	(final String taskId)

停止发布媒体流

该接口会向 TRTC 服务器发送指令，停止通过 startPublishMediaStream 启动的媒体流

参数	描述
taskId	通过回调 onStartPublishMediaStream 带给您后台启动的任务标识（即 taskId）

注意

1. 若您的业务后台并没有保存该 taskId，在您的主播异常退房重进后，如果您需要重新获取 taskId，您可以再次调用 startPublishMediaStream 启动任务。此时 TRTC 后台会返回任务启动失败，同时带给您上一次启动的 taskId
2. 若 taskId 填空字符串，将会停止该用户所有通过 startPublishMediaStream 启动的媒体流，如果您只启动了一个媒体流或者想停止所有通过您启动的媒体流，推荐使用这种方式。

startLocalPreview

startLocalPreview

void startLocalPreview	(boolean frontCamera TXCloudVideoView view)

开启本地摄像头的预览画面（移动端）

在 enterRoom 之前调用此函数，SDK 只会开启摄像头，并一直等到您调用 enterRoom 之后才开始推流。
在 enterRoom 之后调用此函数，SDK 会开启摄像头并自动开始视频推流。当开始渲染首帧摄像头画面时，您会收到 TRTCCloudListener 中的 onCameraDidReady 回调通知。

参数	描述
frontCamera	true：前置摄像头；false：后置摄像头。
view	承载视频画面的控件。

注意
如果希望开播前预览摄像头画面并通过 BeautyManager 调节美颜参数，您可以：
- 方案一：在调用 enterRoom 之前调用 startLocalPreview。
- 方案二：在调用 enterRoom 之后调用 startLocalPreview + muteLocalVideo(true)。

updateLocalView

updateLocalView

```
void updateLocalView (TXCloudVideoView view)
```

更新本地摄像头的预览画面

stopLocalPreview

stopLocalPreview

停止摄像头预览

muteLocalVideo

muteLocalVideo

```
void muteLocalVideo (int streamType
                     boolean mute)
```

暂停/恢复发布本地的视频流

该接口可以暂停（或恢复）发布本地的视频画面，暂停之后，同一房间中的其他用户将无法继续看到自己画面。
该接口在指定 TRTC_VIDEO_STREAM_TYPE_BIG 时等效于 start/stopLocalPreview 这两个接口，但具有更好的响应速度。
因为 start/stopLocalPreview 需要打开和关闭摄像头，而打开和关闭摄像头都是硬件设备相关的操作，非常耗时。
相比之下，muteLocalVideo 只需要在软件层面对数据流进行暂停或者放行即可，因此效率更高，也更适合需要频繁打开关闭的场景。

当暂停/恢复发布指定 TRTC_VIDEO_STREAM_TYPE_BIG 后，同一房间中的其他用户将会收到 onUserVideoAvailable 回调通知。
当暂停/恢复发布指定 TRTC_VIDEO_STREAM_TYPE_SUB 后，同一房间中的其他用户将会收到 onUserSubStreamAvailable 回调通知。

参数	描述
mute	true：暂停；false：恢复。
streamType	要暂停/恢复的视频流类型（仅支持 TRTC_VIDEO_STREAM_TYPE_BIG 和 TRTC_VIDEO_STREAM_TYPE_SUB ）。

setVideoMuteImage

setVideoMuteImage

```
void setVideoMuteImage (Bitmap image
                        int fps)
```

设置本地画面被暂停期间的替代图片

当您调用 muteLocalVideo(true) 暂停本地画面时，您可以通过调用本接口设置一张替代图片，设置后，房间中的其他用户会看到这张替代图片，而不是黑屏画面。

参数	描述
fps	设置替代图片帧率，最小值为5，最大值为10，默认5。
image	设置替代图片，空值代表在 muteLocalVideo 之后不再发送视频流数据，默认值为空。

startRemoteView

startRemoteView

```
void startRemoteView(String userId
                     int streamType
                     TXCloudVideoView view)
```

订阅远端用户的视频流，并绑定视频渲染控件

调用该接口可以让 SDK 拉取指定 userid 的视频流，并渲染到参数 view 指定的渲染控件上。您可以通过 setRemoteRenderParams 设置画面的显示模式。

- 如果您已经知道房间中有视频流的用户的 userid，可以直接调用 startRemoteView 订阅该用户的画面。
- 如果您不知道房间中有哪些用户在发布视频，您可以在 enterRoom 之后等待来自 onUserVideoAvailable 的通知。

调用本接口只是启动视频流的拉取，此时画面还需要加载和缓冲，当缓冲完毕后您会收到来自 onFirstVideoFrame 的通知。

参数	描述
streamType	指定要观看 userId 的视频流类型。 • 高清大画面：TRTC_VIDEO_STREAM_TYPE_BIG 。 • 低清小画面：TRTC_VIDEO_STREAM_TYPE_SMALL（需要远端用户通

	过 enableEncSmallVideoStream 开启双路编码后才有效果）。 • 辅流画面（常用于屏幕分享）：TRTC_VIDEO_STREAM_TYPE_SUB 。
userId	指定远端用户的 ID。
view	用于承载视频画面的渲染控件。

注意

注意几点规则需要您关注：

1. SDK 支持同时观看某 userid 的大画面和辅路画面，或者同时观看某 userid 的小画面和辅路画面，但不支持同时观看大画面和小画面。
2. 只有当指定的 userid 通过 enableEncSmallVideoStream 开启双路编码后，才能观看该用户的小画面。
3. 当指定的 userid 的小画面不存在时，SDK 默认切换到该用户的大画面。

updateRemoteView

updateRemoteView

```
void updateRemoteView (String userId
                       int streamType
                       TXCloudVideoView view)
```

更新远端用户的视频渲染控件

该接口可用于更新远端视频画面的渲染控件，常被用于切换显示区域的交互场景中。

参数	描述
streamType	要设置预览窗口的流类型（仅支持 TRTC_VIDEO_STREAM_TYPE_BIG 和 TRTC_VIDEO_STREAM_TYPE_SUB ）。
userId	指定远端用户的 ID。
view	承载视频画面的控件。

stopRemoteView

stopRemoteView

void stopRemoteView	(String userId int streamType)

停止订阅远端用户的视频流，并释放渲染控件

调用此接口会让 SDK 停止接收该用户的视频流，并释放该路视频流的解码和渲染资源。

参数	描述
streamType	指定要观看 userId 的视频流类型。 • 高清大画面：TRTC_VIDEO_STREAM_TYPE_BIG。 • 低清小画面：TRTC_VIDEO_STREAM_TYPE_SMALL。 • 辅流画面（常用于屏幕分享）：TRTC_VIDEO_STREAM_TYPE_SUB。
userId	指定远端用户的 ID。

stopAllRemoteView

stopAllRemoteView

停止订阅所有远端用户的视频流，并释放全部渲染资源

调用此接口会让 SDK 停止接收所有来自远端的视频流，并释放全部的解码和渲染资源。

> **注意**
> 如果当前有正在显示的辅路画面（屏幕分享）也会一并被停止。

muteRemoteVideoStream

muteRemoteVideoStream

void	(String userId

muteRemoteVideoStream

int streamType

boolean mute)

暂停/恢复订阅远端用户的视频流

该接口仅暂停/恢复接收指定用户的视频流，但并不释放显示资源，视频画面会被冻屏在接口调用时的最后一帧。

参数	描述
mute	是否暂停接收。
streamType	要暂停/恢复的视频流类型。 • 高清大画面：TRTC_VIDEO_STREAM_TYPE_BIG 。 • 低清小画面：TRTC_VIDEO_STREAM_TYPE_SMALL 。 • 辅流画面（常用于屏幕分享）：TRTC_VIDEO_STREAM_TYPE_SUB 。
userId	指定远端用户的 ID。

注意

该接口支持您在进入房间（ enterRoom ）前调用，暂停状态在退出房间（ exitRoom ）之后会被重置。

调用该接口暂停接收指定用户的视频流后，只调用 startRemoteView 接口无法播放指定用户的视频，需要调用 muteRemoteVideoStream (false) 或 muteAllRemoteVideoStreams (false) 才能恢复。

muteAllRemoteVideoStreams

muteAllRemoteVideoStreams

void muteAllRemoteVideoStreams (boolean mute)

暂停/恢复订阅所有远端用户的视频流

该接口仅暂停/恢复接收所有用户的视频流，但并不释放显示资源，视频画面会被冻屏在接口调用时的最后一帧。

参数	描述
mute	是否暂停接收。

注意
该接口支持您在进入房间（ enterRoom ）前调用，暂停状态在退出房间（ exitRoom ）之后会被重置。
调用该接口暂停接收所有用户的视频流后，只调用 startRemoteView 接口无法播放指定用户的视频，需要调用 muteRemoteVideoStream(false) 或 muteAllRemoteVideoStreams(false) 才能恢复。

setVideoEncoderParam

setVideoEncoderParam

void setVideoEncoderParam (TRTCCloudDef.TRTCVideoEncParam param)

设置视频编码器的编码参数

该设置能够决定远端用户看到的画面质量，同时也能决定云端录制出的视频文件的画面质量。

参数	描述
param	用于设置视频编码器的相关参数，详情请参见 TRTCVideoEncParam 。

注意
从v11.5版本开始，编码输出分辨率会按照宽8高2字节对齐，并且是向下调整，eg:输入分辨率540x960，实际编码输出分辨率536x960。

setNetworkQosParam

setNetworkQosParam

void (TRTCCloudDef.TRTCNetworkQosParam param)

setNetworkQosParam

设置网络质量控制的相关参数

该设置决定在差网络环境下的质量调控策略，如“画质优先”或“流畅优先”等策略。

参数	描述
param	用于设置网络质量控制的相关参数，详情请参见 TRTCNetworkQosParam 。

setLocalRenderParams

setLocalRenderParams

void setLocalRenderParams (TRTCCloudDef. TRTCRenderParams params)

设置本地画面的渲染参数

可设置的参数包括有：画面的旋转角度、填充模式以及左右镜像等。

参数	描述
params	画面渲染参数，详情请参见 TRTCRenderParams 。

setRemoteRenderParams

setRemoteRenderParams

void setRemoteRenderParams (String userId

int streamType

TRTCCloudDef. TRTCRenderParams params)

设置远端画面的渲染模式

可设置的参数包括有：画面的旋转角度、填充模式以及左右镜像等。

参数	描述
params	画面渲染参数，详情请参见 TRTCRenderParams 。
streamType	可以设置为主路画面（ TRTC_VIDEO_STREAM_TYPE_BIG ）或辅路画面（ TRTC_VIDEO_STREAM_TYPE_SUB ）。
userId	指定远端用户的 ID。

enableEncSmallVideoStream

enableEncSmallVideoStream

```
int enableEncSmallVideoStream(boolean enable
                              TRTCCloudDef.TRTCVideoEncParam smallVideoEncPa
```

开启大小画面双路编码模式

开启双路编码模式后，当前用户的编码器会同时输出【高清大画面】和【低清小画面】两路视频流（但只有一路音频流）。
如此以来，房间中的其他用户就可以根据自身的网络情况或屏幕大小选择订阅【高清大画面】或是【低清小画面】。

参数	描述
enable	是否开启小画面编码，默认值：false。
smallVideoEncParam	小流的视频参数。

注意
双路编码开启后，会消耗更多的 CPU 和 网络带宽，所以 Mac、Windows 或者高性能 Pad 可以考虑开启，不建议手机端开启。

返回值说明：

0：成功；-1：当前大画面已被设置为较低画质，开启双路编码已无必要。

setRemoteVideoStreamType

setRemoteVideoStreamType

int setRemoteVideoStreamType (String userId

int streamType)

切换指定远端用户的大小画面

当房间中某个主播开启了双路编码之后，房间中其他用户通过 startRemoteView 订阅到的画面默认会是【高清大画面】。
您可以通过此接口选定希望订阅的画面是大画面还是小画面，该接口在 startRemoteView 之前和之后调用均可生效。

参数	描述
streamType	视频流类型，即选择看大画面还是小画面，默认为大画面。
userId	指定远端用户的 ID。

注意
此功能需要目标用户已经通过 enableEncSmallVideoStream 提前开启了双路编码模式，否则此调用无实际效果。

snapshotVideo

snapshotVideo

void snapshotVideo (String userId

int streamType

int sourceType

TRTCCloudListener.TRTCSnapshotListener listener)

视频画面截图

您可以通过本接口截取本地的视频画面，远端用户的主路画面以及远端用户的辅路（屏幕分享）画面。

参数	描述
sourceType	画面来源，可选择截取视频流画面（TRTCSnapshotSourceTypeStream）或视频渲染画面（TRTCSnapshotSourceTypeView），前者一般更清晰。
streamType	视频流类型，可选择截取主路画面（TRTC_VIDEO_STREAM_TYPE_BIG，常用于摄像头）或辅路画面（TRTC_VIDEO_STREAM_TYPE_SUB，常用于屏幕分享）。
userId	用户 ID，如指定空置表示截取本地的视频画面。

注意
Windows 平台目前仅支持截取 TRTCSnapshotSourceTypeStream 来源的视频画面。

setPerspectiveCorrectionPoints

setPerspectiveCorrectionPoints

```
void setPerspectiveCorrectionPoints (String userId
                                     PointF[] srcPoints
                                     PointF[] dstPoints)
```

视频画面透视校正坐标设置

您可以通过本接口指定渲染画面透视校正的坐标区域。

参数	描述
dstPoints	期望校正到的目标坐标区域4个顶点坐标，需按照左上、左下、右上、右下的顺序传入，所有的坐标需要根据画面宽高做 [0,1] 区间的归一化；传 null 则停止透视校正。
srcPoints	原始画面坐标区域4个顶点坐标，需按照左上、左下、右上、右下的顺序传入，所有的坐标需要根据画面宽高做 [0,1] 区间的归一化; 传 null 则停止透视校正。
userId	用户 ID，如指定空值表示校正本地的视频画面。

setGravitySensorAdaptiveMode

setGravitySensorAdaptiveMode

void setGravitySensorAdaptiveMode (int mode)

设置重力感应的适配模式（11.7 及以上版本）

开启重力感应后，如果采集端的设备发生旋转，采集端和观众端的画面都会进行相应地渲染以确保视野中的画面始终朝上。
只在sdk内部摄像头采集场景生效，并且只在移动端生效。
1. 该接口仅对采集端起作用，如果只是观看房间中的画面，开启此接口是无效的
2. 当采集端设备发生 90 度或 270 度旋转时，采集端或者观众看到的画面可能会被裁剪以保持比例的协调

参数	描述
mode	重力感应模式，详情请参见 TRTC_GRAVITY_SENSOR_ADAPTIVE_MODE_DISABLE 、TRTC_GRAVITY_SENSOR_ADAPTIVE_MODE_FILL_BY_CENTER_CROP 和 TRTC_GRAVITY_SENSOR_ADAPTIVE_MODE_FIT_WITH_BLACK_BORDER 默认值：TRTC_GRAVITY_SENSOR_ADAPTIVE_MODE_DISABLE 。

startLocalAudio

startLocalAudio

void startLocalAudio (int quality)

开启本地音频的采集和发布

SDK 默认不开启麦克风，当用户需要发布本地音频时，需要调用该接口开启麦克风采集，并将音频编码并发布到当前的房间中。
开启本地音频的采集和发布后，房间中的其他用户会收到 onUserAudioAvailable(userId, true) 的通知。

参数	描述
quality	声音音质

- TRTC_AUDIO_QUALITY_SPEECH，流畅：采样率：16k；单声道；音频裸码率：16kbps；适合语音通话为主的场景，比如在线会议，语音通话。
- TRTC_AUDIO_QUALITY_DEFAULT，默认：采样率：48k；单声道；音频裸码率：50kbps；SDK 默认的音频质量，如无特殊需求推荐选择之。
- TRTC_AUDIO_QUALITY_MUSIC，高音质：采样率：48k；双声道 + 全频带；音频裸码率：128kbps；适合需要高保真传输音乐的场景，比如在线K歌、音乐直播等。

注意
该函数会检查麦克风的使用权限，如果当前 App 没有麦克风权限，SDK 会自动向用户申请麦克风使用权限。

stopLocalAudio

stopLocalAudio

停止本地音频的采集和发布

停止本地音频的采集和发布后，房间中的其他用户会收到 onUserAudioAvailable(userId, false) 的通知。

muteLocalAudio

muteLocalAudio

void muteLocalAudio	(boolean mute)

暂停/恢复发布本地的音频流

当您暂停发布本地音频流之后，房间中的其他用户会收到 onUserAudioAvailable(userId, false) 的通知。
当您恢复发布本地音频流之后，房间中的其他用户会收到 onUserAudioAvailable(userId, true) 的通知。

与 stopLocalAudio 的不同之处在于，muteLocalAudio(true) 并不会释放麦克风权限，而是继续发送码率极低的静音包。

这对于需要云端录制的场景非常适用，因为 MP4 等格式的视频文件，对于音频数据的连续性要求很高，使用 stopLocalAudio 会导致录制出的 MP4 文件不易播放。
因此在对录制文件的质量要求较高的场景中，建议选择 muteLocalAudio 而不建议使用 stopLocalAudio。

参数	描述
mute	true：静音；false：恢复。

muteRemoteAudio

muteRemoteAudio

```
void muteRemoteAudio (String userId
                      boolean mute)
```

暂停/恢复播放远端的音频流

当您静音某用户的远端音频时，SDK 会停止播放指定用户的声音，同时也会停止拉取该用户的音频数据。

参数	描述
mute	true：静音；false：取消静音。
userId	用于指定远端用户的 ID。

注意
在进入房间（enterRoom）之前或之后调用本接口均生效，静音状态在退出房间（exitRoom）之后会被重置为 false。

muteAllRemoteAudio

muteAllRemoteAudio

```
void (boolean mute)
```

muteAllRemoteAudio

暂停/恢复播放所有远端用户的音频流

当您静音所有用户的远端音频时，SDK 会停止播放所有来自远端的音频流，同时也会停止拉取所有用户的音频数据。

参数	描述
mute	true：静音；false：取消静音。

注意
在进入房间（enterRoom）之前或之后调用本接口均生效，静音状态在退出房间（exitRoom）之后会被重置为 false。

setAudioRoute

setAudioRoute

void setAudioRoute (int route)

设置音频路由

设置“音频路由”，即设置声音是从手机的扬声器还是从听筒中播放出来，因此该接口仅适用于手机等移动端设备。

手机有两个扬声器：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。
设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。

参数	描述
route	音频路由，即声音由哪里输出（扬声器、听筒），默认值：TRTC_AUDIO_ROUTE_SPEAKER。

setRemoteAudioVolume

setRemoteAudioVolume

void setRemoteAudioVolume	(String userId int volume)

设定某一个远端用户的声音播放音量

您可以通过 setRemoteAudioVolume(userId, 0) 将某一个远端用户的声音静音。

参数	描述
userId	用于指定远端用户的 ID。
volume	音量大小，取值范围为0 – 100，默认值：100。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setAudioCaptureVolume

setAudioCaptureVolume

void setAudioCaptureVolume	(int volume)

设定本地音频的采集音量

参数	描述
volume	音量大小，取值范围为0 – 100；默认值：100。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

getAudioCaptureVolume

getAudioCaptureVolume

获取本地音频的采集音量

setAudioPlayoutVolume

setAudioPlayoutVolume

void setAudioPlayoutVolume (int volume)

设定远端音频的播放音量

该接口会控制 SDK 最终交给系统播放的声音音量，调节效果会影响到本地音频录制文件的音量大小，但不会影响到耳返的音量大小。

参数	描述
volume	音量大小，取值范围为0 – 100，默认值：100。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

getAudioPlayoutVolume

getAudioPlayoutVolume

获取远端音频的播放音量

enableAudioVolumeEvaluation

enableAudioVolumeEvaluation

void enableAudioVolumeEvaluation	(boolean enable TRTCCloudDef. TRTCAudioVolumeEvaluateParams

启用音量大小提示

开启此功能后，SDK 会在 TRTCCloudListener 中的 onUserVoiceVolume 回调中反馈本地或远端用户的音频音量评估信息，包括音量大小、人声检测、音频频谱等。

参数	描述
enable	是否启用音量提示，默认为关闭状态。
params	音量评估等相关参数，请参见 TRTCAudioVolumeEvaluateParams 。

startAudioRecording

startAudioRecording

int startAudioRecording (TRTCCloudDef. TRTCAudioRecordingParams param)

开始录音

当您调用该接口后， SDK 会将本地和远端的所有音频（包括本地音频，远端音频，背景音乐和音效等）混合并录制到一个本地文件中。
该接口在进入房间前后调用均可生效，如果录制任务在退出房间前尚未通过 stopAudioRecording 停止，则退出房间后录制任务会自动被停止。
本次录制的启动、完成状态会通过本地录制相关回调进行通知。参见 TRTCCloud 相关回调。

参数	描述
par	录音参数，请参见 TRTCAudioRecordingParams 。

am	

注意

自 v11.5 版本，音频录制的状态结果由返回值统一调整为异步回调进行通知。参见 TRTCCloud 相关回调。

返回值说明：

0：成功；-1：录音已开始；-2：文件或目录创建失败；-3：后缀指定的音频格式不支持。

stopAudioRecording

stopAudioRecording

停止录音

如果录制任务在退出房间前尚未通过本接口停止，则退出房间后录音任务会自动被停止。

startLocalRecording

startLocalRecording

void startLocalRecording (TRTCCloudDef.TRTCLocalRecordingParams params)

开启本地媒体录制

开启后把直播过程中的音视频内容录制到本地的一个文件中。

参数	描述
params	录制参数，请参见 TRTCLocalRecordingParams 。

stopLocalRecording

stopLocalRecording

停止本地媒体录制

如果录制任务在退出房间前尚未通过本接口停止，则退出房间后录音任务会自动被停止。

setRemoteAudioParallelParams

setRemoteAudioParallelParams

void setRemoteAudioParallelParams (TRTCCloudDef.TRTCAudioParallelParams params

设置远端音频流智能并发播放策略

设置远端音频流智能并发播放策略，适用于上麦人数比较多的场景。

参数	描述
params	音频并发参数，请参见 TRTCAudioParallelParams 。

enable3DSpatialAudioEffect

enable3DSpatialAudioEffect

void enable3DSpatialAudioEffect (boolean enabled)

启用 3D 音效

启用 3D 音效。注意需使用流畅音质 TRTC_AUDIO_QUALITY_SPEECH 或默认音质 TRTC_AUDIO_QUALITY_DEFAULT 。

参数	描述
enabled	是否启用 3D 音效，默认为关闭状态。

updateSelf3DSpatialPosition

updateSelf3DSpatialPosition

```
void updateSelf3DSpatialPosition (int[] position
                                  float[] axisForward
                                  float[] axisRight
                                  float[] axisUp)
```

设置 3D 音效中自身坐标及朝向信息

更新自身在世界坐标系中的位置和朝向， SDK 会根据该方法参数计算自身和远端用户之间的相对位置，进而渲染出空间音效。注意各参数应分别传入长度为 3 的数组。

参数	描述
axisForward	自身坐标系前轴在世界坐标系中的单位向量，三个值依次表示前、右、上坐标值。
axisRight	自身坐标系右轴在世界坐标系中的单位向量，三个值依次表示前、右、上坐标值。
axisUp	自身坐标系上轴在世界坐标系中的单位向量，三个值依次表示前、右、上坐标值。
position	自身在世界坐标系中的坐标，三个值依次表示前、右、上坐标值。

注意
请适当限制调用频率，推荐两次坐标设置至少间隔 100ms。

updateRemote3DSpatialPosition

updateRemote3DSpatialPosition

```
void updateRemote3DSpatialPosition (String userId
                                    int[] position)
```

设置 3D 音效中远端用户坐标信息

更新远端用户在世界坐标系中的位置，SDK 会根据自身和远端用户之间的相对位置，进而渲染出空间音效。注意参数为长度等于 3 的数组。

参数	描述
position	该远端用户在世界坐标系中的坐标，三个值依次表示前、右、上坐标值。
userId	指定远端用户的 ID。

注意
请适当限制调用频率，推荐同一远端用户两次坐标设置至少间隔 100ms。

set3DSpatialReceivingRange

set3DSpatialReceivingRange

```
void set3DSpatialReceivingRange (String userId
                                 int range)
```

设置指定用户所发出声音的可被接收范围

设置该范围大小之后，该指定用户的声音将在该范围内可被听见，超出该范围将被衰减为 0。

参数	描述
range	声音最大可被接收范围。
userId	指定远端用户的 ID。

getDeviceManager

getDeviceManager

获取设备管理类（TXDeviceManager）

getBeautyManager

getBeautyManager

获取美颜管理类（TXBeautyManager）

通过美颜管理，您可以使用以下功能：

- 设置“磨皮”、“美白”、“红润”等美颜特效。
- 设置“大眼”、“瘦脸”、“V脸”、“下巴”、“短脸”、“小鼻”、“亮眼”、“白牙”、“祛眼袋”、“祛皱纹”、“祛法令纹”等修脸特效。
- 设置“发际线”、“眼间距”、“眼角”、“嘴形”、“鼻翼”、“鼻子位置”、“嘴唇厚度”、“脸型”等修脸特效。
- 设置“眼影”、“腮红”等美妆特效。
- 设置动态贴纸和人脸挂件等动画特效。

setWatermark

setWatermark

```
void
setWatermark
```

```
(Bitmap image

int streamType

float x

float y

float width)
```

添加水印

水印的位置是通过 rect 参数来指定的，rect 是一个四元组参数，其格式为 (x，y，width，height)

- x：水印的坐标，取值范围为0 – 1的浮点数。
- y：水印的坐标，取值范围为0 – 1的浮点数。
- width：水印的宽度，取值范围为0 – 1的浮点数。
- height：是不用设置的，SDK 内部会根据水印图片的宽高比自动计算一个合适的高度。

参数设置举例：
如果当前视频的编码分辨率是 540 × 960，且 rect 参数被您设置为（0.1，0.1，0.2，0.0），

那么水印的左上坐标点就是（540 × 0.1，960 × 0.1）即（54，96），水印的宽度是 540 × 0.2 = 108px，水印的高度会根据水印图片的宽高比由 SDK 自动算出。

参数	描述
image	水印图片，**必须使用透明底色的 png 格式**。
rect	水印相对于编码分辨率的归一化坐标，x，y，width，height 取值范围0 – 1。
streamType	指定给哪一路画面设置水印，详情请参见 TRTCVideoStreamType 。

注意
如果您要给主画面（一般为摄像头）和辅路画面（一般用作屏幕分享）同时设置水印，需要调用该接口两次，并设定不同的 streamType。

getAudioEffectManager

getAudioEffectManager

获取音效管理类（TXAudioEffectManager）

TXAudioEffectManager 是音效管理接口，您可以通过该接口实现如下功能：

- 背景音乐：支持在线音乐和本地音乐，支持变速、变调等特效、支持原生和伴奏并播放和循环播放。
- 耳机耳返：麦克风捕捉的声音实时通过耳机播放，常用于音乐直播。
- 混响效果：KTV、小房间、大会堂、低沉、洪亮...。
- 变声特效：萝莉、大叔、重金属...。
- 短音效：鼓掌声、欢笑声等简短的音效文件（对于小于10秒的文件，请将 isShortFile 参数设置为 true）。

startSystemAudioLoopback

startSystemAudioLoopback

开启系统声音采集

该接口会采集其他应用音频数据，并将其混入到 SDK 当前的音频数据流中，从而使房间中的其他用户也能听到主播的其他应用所播放出的声音。
在线教育场景中，老师可以使用此功能让 SDK 采集教学影片中的声音，并广播给同房间中的学生。
音乐直播场景中，主播可以使用此功能让 SDK 采集音乐播放器中的音乐，从而为自己的直播间增加背景音乐。

> **注意**
> 1. 该接口只在 Android API 29 及以上的版本生效。
> 2. 您需要先使用该接口来开启系统声音采集，当使用屏幕分享接口开启屏幕分享时才会真正生效。
> 3. 您需要添加一个前台服务来确保系统声音采集不被静默，并设置 android:foregroundServiceType="mediaProjection"。
> 4. SDK 只采集满足捕获策略和音频用法的应用音频，目前采集的音频用法包括USAGE_MEDIA, USAGE_GAME。

stopSystemAudioLoopback

stopSystemAudioLoopback

停止系统声音采集（iOS 端暂未支持）

startScreenCapture

startScreenCapture

```
void startScreenCapture (int streamType
                         TRTCCloudDef.TRTCVideoEncParam encParams
                         TRTCCloudDef.TRTCScreenShareParams shareParams)
```

启动屏幕分享

该接口支持抓取整个 Android 系统的屏幕，可以实现类似腾讯会议的全系统级的屏幕分享。

参考文档：实时屏幕分享(Android)

Android 推荐的屏幕分享视频编码参数（ TRTCVideoEncParam ）：

- 分辨率(videoResolution)：1280×720。
- 帧率(videoFps)：10 FPS。
- 码率(videoBitrate)：1200 kbps。
- 分辨率自适应(enableAdjustRes)：false。

参数	描述
encParams	编码参数，请参见 TRTCVideoEncParam 。如果您指定 encParams 为 null，SDK 会自动使用之前设定的编码参数。
shareParams	请参见 TRTCScreenShareParams ，您可以通过其中的 floatingView 参数弹出悬浮窗（也可以使用 Android 系统的 WindowManager 接口自行弹出）。
streamType	屏幕分享使用的线路，可以设置为主路（TRTC_VIDEO_STREAM_TYPE_BIG）或者辅路（TRTC_VIDEO_STREAM_TYPE_SUB）。

stopScreenCapture

stopScreenCapture

停止屏幕分享

pauseScreenCapture

pauseScreenCapture

暂停屏幕分享

> **注意**
> 从v11.5版本开始，暂停屏幕采集会使用最后一帧按照1fps帧率输出。

resumeScreenCapture

resumeScreenCapture

恢复屏幕分享

setSubStreamEncoderParam

setSubStreamEncoderParam

void setSubStreamEncoderParam	(TRTCCloudDef.TRTCVideoEncParam param)

设置屏幕分享（即辅路）的视频编码参数（桌面系统和移动系统均已支持）

该接口可以设定远端用户所看到的屏幕分享（即辅路）的画面质量，同时也能决定云端录制出的视频文件中屏幕分享的画面质量。
请注意如下两个接口的差异：

- setVideoEncoderParam 用于设置主路画面（TRTC_VIDEO_STREAM_TYPE_BIG，一般用于摄像头）的视频编码参数。
- setSubStreamEncoderParam 用于设置辅路画面（TRTC_VIDEO_STREAM_TYPE_SUB，一般用于屏幕分享）的视频编码参数。

参数	描述
param	辅流编码参数，详情请参见 TRTCVideoEncParam。

enableCustomVideoCapture

enableCustomVideoCapture

void enableCustomVideoCapture	(int streamType boolean enable)

启用/关闭视频自定义采集模式

开启该模式后，SDK 不在运行原有的视频采集流程，即不再继续从摄像头采集数据和美颜，而是只保留视频编码和发送能力。
您需要通过 sendCustomVideoData 不断地向 SDK 塞入自己采集的视频画面。

参数	描述
enable	是否启用，默认值：false。
streamType	用于指定视频流类型，TRTC_VIDEO_STREAM_TYPE_BIG：高清大画面；TRTC_VIDEO_STREAM_TYPE_SUB：辅路画面。

sendCustomVideoData

sendCustomVideoData

```
void sendCustomVideoData (int streamType
                          TRTCCloudDef.TRTCVideoFrame frame)
```

向 SDK 投送自己采集的视频帧

使用此接口可以向 SDK 投送自己采集的视频帧，SDK 会将视频帧进行编码并通过自身的网络模块传输出去。
Android 平台有两种投送方案：

- 内存传递方案：对接起来比较简单，但是性能较差，不适合分辨率较高的场景。
- 显存传递方案：对接需要一定的 OpenGL 基础，但是性能较好，640 × 360 以上的分辨率请采用该方案。

参考文档：自定义采集和渲染 。

参数	描述
frame	视频数据，如果是内存传递方案，请设置 data 字段；如果是显存传递方案，请设置 TRTCTexture 字段。详情请参见 TRTCVideoFrame 。
streamType	用于指定视频流类型，TRTC_VIDEO_STREAM_TYPE_BIG： 高清大画面；TRTC_VIDEO_STREAM_TYPE_SUB： 辅路画面。

注意

1. 推荐您在采集到的一帧视频帧后，即调用 generateCustomPTS 接口获取该帧的 timestamp 数值，这样可以获得最佳的音画同步效果。
2. SDK 最终编码出的视频帧率并不是由您调用本接口的频率决定的，而是由您在 setVideoEncoderParam 中所设置的 FPS 决定的。

3. 请尽量保持本接口的调用间隔是均匀的，否则会导致编码器输出帧率不稳或者音画不同步等问题。

enableCustomAudioCapture

enableCustomAudioCapture

void enableCustomAudioCapture	(boolean enable)

启用音频自定义采集模式

开启该模式后，SDK 不在运行原有的音频采集流程，即不再继续从麦克风采集音频数据，而是只保留音频编码和发送能力。
您需要通过 sendCustomAudioData 不断地向 SDK 塞入自己采集的音频数据。

参数	描述
enable	是否启用，默认值：false。

注意
由于回声抵消（AEC）需要严格的控制声音采集和播放的时间，所以开启自定义音频采集后，AEC 能力可能会失效。

sendCustomAudioData

sendCustomAudioData

void sendCustomAudioData	(TRTCCloudDef.TRTCAudioFrame frame)

向 SDK 投送自己采集的音频数据

参数 TRTCAudioFrame 推荐下列填写方式（其他字段不需要填写）：

- audioFormat：音频数据格式，仅支持 TRTCAudioFrameFormatPCM。
- data：音频帧 buffer。音频帧数据只支持 PCM 格式，支持[5ms ~ 100ms]帧长，推荐使用 20ms 帧长，长度计算方法：【48000采样率、单声道的帧长度：48000 × 0.02s × 1 × 16bit = 15360bit =

1920字节】。

- sampleRate：采样率，支持：16000、24000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- timestamp：时间戳，单位为毫秒（ms），请使用音频帧在采集时被记录下来的时间戳（可以在采集到一帧音频帧之后，通过调用 generateCustomPTS 获取时间戳）。

参考文档： 自定义采集和渲染 。

参数	描述
frame	音频数据

注意
请您精准地按每帧时长的间隔调用本接口，数据投送间隔不均匀时极易触发声音卡顿。

enableMixExternalAudioFrame

enableMixExternalAudioFrame

```
void enableMixExternalAudioFrame (boolean enablePublish
                                  boolean enablePlayout)
```

启用/关闭自定义音轨

开启后，您可以通过本接口向 SDK 混入一条自定义的音轨。通过两个布尔型参数，您可以控制该音轨是否要在远端和本地播放。

参数	描述
enablePlayout	控制混入的音轨是否要在本地播放，默认值：false。
enablePublish	控制混入的音轨是否要在远端播放，默认值：false。

注意
如果您指定参数 enablePublish 和 enablePlayout 均为 false，代表完全关闭您的自定义音轨。

mixExternalAudioFrame

mixExternalAudioFrame

int mixExternalAudioFrame (TRTCCloudDef.TRTCAudioFrame frame)

向 SDK 混入自定义音轨

调用该接口之前，您需要先通过 enableMixExternalAudioFrame 开启自定义音轨，之后就可以通过本接口将自己的音轨以 PCM 格式混入到 SDK 中。
理想情况下，我们期望您的代码能够以非常均匀的速度向 SDK 提供音轨数据。但我们也非常清楚，完美的调用间隔是一个巨大的挑战。
所以 SDK 内部会开启一个音轨数据的缓冲区，该缓冲区的作用类似一个“蓄水池”，它能够暂存您传入的音轨数据，平抑由于接口调用间隔的抖动问题。
本接口的返回值代表这个音轨缓冲区的大小，单位是毫秒（ms），比如：如果该接口返回 50，则代表当前的音轨缓冲区有 50ms 的音轨数据。因此只要您在 50ms 内再次调用本接口，SDK 就能保证您混入的音轨数据是连续的。
当您调用该接口后，如果发现返回值 > 100ms，则可以等待一帧音频帧的播放时间之后再次调用；如果返回值 < 100ms，则代表缓冲区比较小，您可以再次混入一些音轨数据以确保音轨缓冲区的大小维持在“安全水位”以上。

参数 TRTCAudioFrame 推荐下列填写方式（其他字段不需要填写）：

- data：音频帧 buffer。音频帧数据只支持 PCM 格式，支持[5ms ~ 100ms]帧长，推荐使用 20ms 帧长，长度计算方法：【48000采样率、单声道的帧长度：48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节】。
- sampleRate：采样率，支持：16000、24000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- timestamp：时间戳，单位为毫秒（ms），请使用音频帧在采集时被记录下来的时间戳（可以在获得一帧音频帧之后，通过调用 generateCustomPTS 获得时间戳）。

参数	描述
frame	音频数据

注意
请您精准地按每帧时长的间隔调用本接口，数据投送间隔不均匀时极易触发声音卡顿。

返回值说明：

>= 0 缓冲的长度，单位：ms。< 0 错误（-1 未启用 mixExternalAudioFrame）

setMixExternalAudioVolume

setMixExternalAudioVolume

```
void setMixExternalAudioVolume(int publishVolume, int playoutVolume)
```

设置推流时混入外部音频的推流音量和播放音量

参数	描述
playoutVolume	设置的播放音量大小，范围0 - 100，-1表示不改变。
publishVolume	设置的推流音量大小，范围0 - 100，-1表示不改变。

generateCustomPTS

generateCustomPTS

生成自定义采集时的时间戳

本接口仅适用于自定义采集模式，用于解决音视频帧的采集时间（capture time）和投送时间（send time）不一致所导致的音画不同步问题。

当您通过 sendCustomVideoData 或 sendCustomAudioData 等接口进行自定义视频或音频采集时，请按照如下操作使用该接口：
1. 首先，在采集到一帧视频或音频帧时，通过调用本接口获得当时的 PTS 时间戳。
2. 之后可以将该视频或音频帧送入您使用的前处理模块（如第三方美颜组件，或第三方音效组件）。
3. 在真正调用 sendCustomVideoData 或 sendCustomAudioData 进行投送时，请将该帧在采集时记录的 PTS 时间戳赋值给 TRTCVideoFrame 或 TRTCAudioFrame 中的 timestamp 字段。

返回值说明：

时间戳（单位：ms）

setLocalVideoProcessListener

setLocalVideoProcessListener

```
int setLocalVideoProcessListener    (int pixelFormat
                                     int bufferType
                                     TRTCCloudListener.TRTCVideoFrameListener liste
```

设置第三方美颜的视频数据回调

设置该回调之后，SDK 会把采集到的视频帧通过您设置的 listener 回调出来，用于第三方美颜组件进行二次处理，之后 SDK 会将处理后的视频帧进行编码和发送。

参数	描述
bufferType	指定回调的数据格式，目前支持： • TRTC_VIDEO_BUFFER_TYPE_TEXTURE：适用于 pixelFormat 被设置为 TRTC_VIDEO_PIXEL_FORMAT_Texture_2D 时。 • TRTC_VIDEO_BUFFER_TYPE_BYTE_BUFFER：适用于 pixelFormat 被设置为 TRTC_VIDEO_PIXEL_FORMAT_I420 时。 • TRTC_VIDEO_BUFFER_TYPE_BYTE_ARRAY：适用于 pixelFormat 被设置为 TRTC_VIDEO_PIXEL_FORMAT_I420 时。
listener	自定义预处理回调，详见 TRTCVideoFrameListener 。
pixelFormat	指定回调的像素格式，目前支持： • TRTC_VIDEO_PIXEL_FORMAT_Texture_2D：显存纹理方案。 • TRTC_VIDEO_PIXEL_FORMAT_I420：内存数据方案。

返回值说明：

0：成功；< 0：错误

setLocalVideoRenderListener

setLocalVideoRenderListener

```
int setLocalVideoRenderListener    (int pixelFormat
```

int bufferType

TRTCCloudListener.TRTCVideoRenderListener lister

设置本地视频自定义渲染回调

设置该回调之后，SDK 内部会跳过原来的渲染流程，并把采集到的数据回调出来，您需要自己完成画面渲染。

- pixelFormat 指定回调的数据格式，目前支持 Texture2D、I420 和 RGBA 三种格式。
- bufferType 指定 buffer 的类型，BYTE_BUFFER 适合在 jni 层使用，BYTE_ARRAY 则可用于 Java 层的直接操作。

参考文档：自定义采集和渲染 。

参数	描述
bufferType	指定视频帧的数据结构： • TRTC_VIDEO_BUFFER_TYPE_TEXTURE：适用于 pixelFormat 被设置为 TRTC_VIDEO_PIXEL_FORMAT_Texture_2D 时。 • TRTC_VIDEO_BUFFER_TYPE_BYTE_BUFFER：适用于 pixelFormat 被设置为 TRTC_VIDEO_PIXEL_FORMAT_I420 和 TRTC_VIDEO_PIXEL_FORMAT_RGBA 时。 • TRTC_VIDEO_BUFFER_TYPE_BYTE_ARRAY：适用于 pixelFormat 被设置为 TRTC_VIDEO_PIXEL_FORMAT_I420 和 TRTC_VIDEO_PIXEL_FORMAT_RGBA 时。
listener	自定义视频渲染回调，每一帧视频数据回调一次。
pixelFormat	指定视频帧的像素格式，如： • TRTC_VIDEO_PIXEL_FORMAT_Texture_2D：OpenGL 纹理格式，比较适合 GPU 进行处理，处理效率高。 • TRTC_VIDEO_PIXEL_FORMAT_I420：标准 i420 格式，比较适合 CPU 进行处理，处理效率较差。 • TRTC_VIDEO_PIXEL_FORMAT_RGBA：RGBA 格式，比较适合 CPU 进行处理，处理效率较差。

返回值说明：

0：成功；< 0：错误。

setRemoteVideoRenderListener

setRemoteVideoRenderListener

int setRemoteVideoRenderListener	(String userId int pixelFormat int bufferType TRTCCloudListener.TRTCVideoRenderListener liste

设置远端视频自定义渲染回调

设置该回调之后，SDK 内部会跳过原来的渲染流程，并把采集到的数据回调出来，您需要自己完成画面渲染。

- pixelFormat 指定回调的数据格式，例如 NV12、i420 以及 32BGRA。
- bufferType 指定 buffer 的类型，直接使用 PixelBuffer 效率最高；使用 NSData 相当于让 SDK 在内部做了一次内存转换，因此会有额外的性能损耗。

参考文档：自定义采集和渲染 。

参数	描述
bufferType	指定视频帧的数据结构，如 TRTC_VIDEO_BUFFER_TYPE_BYTE_BUFFER，TRTC_VIDEO_BUFFER_TYPE_BYTE_ARRAY 。
listener	自定义视频渲染回调，每一帧视频数据回调一次。
pixelFormat	指定视频帧的像素格式，目前支持 TRTC_VIDEO_PIXEL_FORMAT_I420，TRTC_VIDEO_PIXEL_FORMAT_RGBA 。
userId	指定远端用户的 ID。

注意
实际使用时，需要先调用 startRemoteView(userid, null) 启动远程视频流的拉取，并将 view 设置为 null，否则不会有数据回调出来。

返回值说明：

0：成功；< 0：错误。

setAudioFrameListener

setAudioFrameListener

void setAudioFrameListener (TRTCCloudListener.TRTCAudioFrameListener listener)

设置音频数据自定义回调

设置该回调之后，SDK 内部会把音频数据（PCM 格式）回调出来，包括：

- onCapturedAudioFrame：本地麦克风采集到的音频数据回调
- onLocalProcessedAudioFrame：本地采集并经过音频模块前处理后的音频数据回调
- onRemoteUserAudioFrame：混音前的每一路远程用户的音频数据
- onMixedPlayAudioFrame：将各路音频混合之后并最终要由系统播放出的音频数据回调

> **注意**
> 设置回调为空即代表停止自定义音频回调，反之，设置回调不为空则代表启动自定义音频回调。

setCapturedAudioFrameCallbackFormat

setCapturedAudioFrameCallbackFormat

int setCapturedAudioFrameCallbackFormat (TRTCCloudDef.TRTCAudioFrameCallbackF

设置本地麦克风采集出的音频帧回调格式

本接口用于设置 onCapturedAudioFrame 回调出来的 AudioFrame 的格式：

- sampleRate：采样率，支持：16000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- samplesPerCall：采样点数，定义回调数据帧长。帧长必须为 10ms 的整数倍。

如果希望用毫秒数计算回调帧长，则将毫秒数转换成采样点数的公式为：采样点数 = 毫秒数 * 采样率 / 1000。
举例：48000 采样率希望回调 20ms 帧长的数据，则采样点数应该填：960 = 20 * 48000 / 1000。

参数	描述
form	音频数据回调格式。

at	

注意

最终回调的帧长度是以字节为单位，采样点数转换成字节数的计算公式为：字节数 = 采样点数 * channel * 2（位宽）举例：48000 采样率，双声道，20ms 帧长，采样点数为 960，字节数为 3840 = 960 * 2 * 2

返回值说明：

0：成功；<0：错误

setLocalProcessedAudioFrameCallbackFormat

setLocalProcessedAudioFrameCallbackFormat

int setLocalProcessedAudioFrameCallbackFormat	(TRTCCloudDef.TRTCAudioFrameCa format)

设置经过前处理后的本地音频帧回调格式

本接口用于设置 onLocalProcessedAudioFrame 回调出来的 AudioFrame 的格式：

- sampleRate：采样率，支持：16000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- samplesPerCall：采样点数，定义回调数据帧长。帧长必须为 10ms 的整数倍。

如果希望用毫秒数计算回调帧长，则将毫秒数转换成采样点数的公式为：采样点数 = 毫秒数 * 采样率 / 1000。
举例：48000 采样率希望回调20ms帧长的数据，则采样点数应该填: 960 = 20 * 48000 / 1000。

参数	描述
format	音频数据回调格式。

注意

最终回调的帧长度是以字节为单位，采样点数转换成字节数的计算公式为：字节数 = 采样点数 * channel * 2（位宽） 举例：48000 采样率，双声道，20ms 帧长，采样点数为 960，字节数为 3840 = 960 * 2 * 2

返回值说明：

0：成功；<0：错误

setMixedPlayAudioFrameCallbackFormat

setMixedPlayAudioFrameCallbackFormat

int setMixedPlayAudioFrameCallbackFormat (TRTCCloudDef.TRTCAudioFrameCallback

设置最终要由系统播放出的音频帧回调格式

本接口用于设置 onMixedPlayAudioFrame 回调出来的 AudioFrame 的格式：

- sampleRate：采样率，支持：16000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- samplesPerCall：采样点数，定义回调数据帧长。帧长必须为 10ms 的整数倍。

如果希望用毫秒数计算回调帧长，则将毫秒数转换成采样点数的公式为：采样点数 = 毫秒数 * 采样率 / 1000。
举例：48000 采样率希望回调20ms帧长的数据，则采样点数应该填: 960 = 20 * 48000 / 1000。

参数	描述
format	音频数据回调格式。

注意
最终回调的帧长度是以字节为单位，采样点数转换成字节数的计算公式为：字节数 = 采样点数 * channel * 2（位宽）举例：48000 采样率，双声道，20ms 帧长，采样点数为 960，字节数为 3840 = 960 * 2 * 2

返回值说明：

0：成功；<0：错误

enableCustomAudioRendering

enableCustomAudioRendering

void enableCustomAudioRendering	(boolean enable)

开启音频自定义播放

如果您需要外接一些特定的音频设备，或者希望自己掌控音频的播放逻辑，您可以通过该接口启用音频自定义播放。
启用音频自定义播放后，SDK 将不再调用系统的音频接口播放数据，您需要通过 getCustomAudioRenderingFrame 获取 SDK 要播放的音频帧并自行播放。

参数	描述
enable	是否启用音频自定义播放，默认为关闭状态。

注意
需要您在进入房间前设置才能生效，暂不支持进入房间后再设置。

getCustomAudioRenderingFrame

getCustomAudioRenderingFrame

void getCustomAudioRenderingFrame	(final TRTCCloudDef.TRTCAudioFrame audioFram

获取可播放的音频数据

调用该接口之前，您需要先通过 enableCustomAudioRendering 开启音频自定义播放。

参数 TRTCAudioFrame 推荐下列填写方式（其他字段不需要填写）：

- sampleRate：采样率，必填，支持 16000、24000、32000、44100、48000。
- channel：声道数，必填，单声道请填1，双声道请填2，双声道时数据是交叉的。
- data：用于获取音频数据的 buffer。需要您根据一帧音频帧的帧长度分配好 data 的内存大小。获取的 PCM 数据支持 10ms 或 20ms 两种帧长，推荐使用 20ms 的帧长。

计算公式为：采样率 x 播放时长 x 声道数量 x 2（每个样点的字节数），例如：48000采样率、单声道、且播放时长为 20ms 的一帧音频帧的 buffer 大小为 48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节。

参数	描述
audioFrame	音频数据帧。

注意

1. 参数 audioFrame 中的 sampleRate、channel 需提前设置好，同时分配好所需读取帧长的 data 空间。
2. SDK 内部会根据 sampleRate 和 channel 自动填充 data 数据。
3. 建议由系统的音频播放线程直接驱动该函数的调用，在播放完一帧音频之后，即调用该接口获取下一帧可播放的音频数据。

sendCustomCmdMsg

sendCustomCmdMsg

```
boolean sendCustomCmdMsg(int cmdID
                         byte[] data
                         boolean reliable
                         boolean ordered)
```

使用 UDP 通道发送自定义消息给房间内所有用户

该接口可以让您借助 TRTC 的 UDP 通道，向当前房间里的其他用户广播自定义数据，以达到传输信令的目的。
房间中的其他用户可以通过 TRTCCloudListener 中的 onRecvCustomCmdMsg 回调接收消息。

参数	描述
cmdID	消息 ID，取值范围为1 – 10。
data	待发送的消息，单个消息的最大长度被限制为 1KB。
ordered	是否要求有序，即是否要求接收端的数据包顺序和发送端的数据包顺序一致（这会带来一定的接收延时）。
reliable	是否可靠发送，可靠发送可以获得更高的发送成功率，但可靠发送比不可靠发送会带来更

	大的接收延迟。

注意

1. 发送消息到房间内所有用户（暂时不支持 Web/小程序端），每秒最多能发送30条消息。
2. 每个包最大为 1KB，超过则很有可能会被中间路由器或者服务器丢弃。
3. 每个客户端每秒最多能发送总计 8KB 数据。
4. 请将 reliable 和 ordered 同时设置为 true 或同时设置为 false，暂不支持交叉设置。
5. 强烈建议您将不同类型的消息设定为不同的 cmdID，这样可以在要求有序的情况下减小消息时延。
6. 目前仅支持主播身份。

返回值说明：

true：消息已经发出；false：消息发送失败。

sendSEIMsg

sendSEIMsg

```
boolean
sendSEIMsg          (byte[] data

                    int repeatCount)
```

使用 SEI 通道发送自定义消息给房间内所有用户

该接口可以让您借助 TRTC 的 SEI 通道，向当前房间里的其他用户广播自定义数据，已达到传输信令的目的。

视频帧的头部有一个叫做 SEI 的头部数据块，该接口的原理就是利用这个被称为 SEI 的头部数据块，将您要发送的自定义信令嵌入其中，使其同视频帧一并发送出去。
因此，与 sendCustomCmdMsg 相比，SEI 通道传输的信令具有更好的兼容性：信令可以伴随着视频帧一直传输到直播 CDN 上。
不过，由于视频帧头部的数据块不能太大，建议您使用该接口时，尽量将信令控制在几个字节的大小。

最常见的用法是把自定义的时间戳（timestamp）用本接口嵌入视频帧中，实现消息和画面的完美对齐（比如：教育场景下的课件和视频信号的对齐）。
房间中的其他用户可以通过 TRTCCloudListener 中的 onRecvSEIMsg 回调接收消息。

参数	描述

data	待发送的数据，最大支持 1KB（1000字节）的数据大小
repeatCount	发送数据次数

注意

本接口有以下限制：

1. 数据在接口调用完后不会被即时发送出去，而是从下一帧视频帧开始与视频帧一起发送。
2. 发送消息到房间内所有用户，每秒最多能发送 30 条消息（与 sendCustomCmdMsg 共享限制）。
3. 每个包最大为 1KB，若发送大量数据，会导致视频码率增大，可能导致视频画质下降甚至卡顿（与 sendCustomCmdMsg 共享限制）。
4. 每个客户端每秒最多能发送总计8KB数据（与 sendCustomCmdMsg 共享限制）。
5. 若指定多次发送（repeatCount > 1），则数据会被带在后续的连续 repeatCount 个视频帧中发送出去，同样会导致视频码率增大。
6. 如果 repeatCount > 1，多次发送，接收消息 onRecvSEIMsg 回调也可能会收到多次相同的消息，需要去重。

返回值说明：

true：消息已通过限制，等待后续视频帧发送；false：消息被限制发送

startSpeedTest

startSpeedTest

int startSpeedTest (TRTCCloudDef.TRTCSpeedTestParams params)

开始进行网速测试（进入房间前使用）

参数	描述
params	测速选项

注意

1. 测速过程将产生少量的基础服务费用，详见 计费概述 > 基础服务 文档说明。
2. 请在进入房间前进行网速测试，在房间中网速测试会影响正常的音视频传输效果，而且由于干扰过多，网速测试结果也不准确。

3. 同一时间只允许一项网速测试任务运行。

返回值说明：

接口调用结果，< 0：失败

stopSpeedTest

stopSpeedTest

停止网络测速

getSDKVersion

getSDKVersion

获取 SDK 版本信息

setLogLevel

setLogLevel

void setLogLevel (int level)

设置 Log 输出级别

参数	描述
level	参见 TRTCLogLevel，默认值：TRTCLogLevelNone

setConsoleEnabled

setConsoleEnabled

void setConsoleEnabled	(boolean enabled)

启用/禁用控制台日志打印

参数	描述
enabled	指定是否启用，默认：禁止状态。

setLogCompressEnabled

setLogCompressEnabled

void setLogCompressEnabled	(boolean enabled)

启用/禁用日志的本地压缩

开启压缩后，Log 存储体积明显减小，但需要腾讯云提供的 Python 脚本解压后才能阅读。
禁用压缩后，Log 采用明文存储，可以直接用记事本打开阅读，但占用空间较大。

参数	描述
enabled	指定是否启用，默认为启动状态

setLogDirPath

setLogDirPath

void setLogDirPath	(String path)

设置本地日志的保存路径

通过该接口您可以更改 SDK 本地日志的默认存储路径，SDK 默认的本地日志的存储位置：

- Windows 平台：在 C:/Users/[系统用户名]/AppData/Roaming/liteav/log，即 %appdata%/liteav/log 下。

- iOS 或 Mac 平台：在 sandbox Documents/log 下。
- Android 平台：在 /app私有目录/files/log/liteav/ 下。

参数	描述
path	存储日志的路径

注意
请务必在所有其他接口之前调用，并且保证您指定的目录是存在的，并且您的应用程序拥有对该目录的读写权限。

setLogListener

setLogListener

void setLogListener (final TRTCCloudListener.TRTCLogListener logListener)

设置日志回调

showDebugView

showDebugView

void showDebugView (int showType)

显示仪表盘

“仪表盘”是位于视频渲染控件之上的一个半透明的调试信息浮层，用于展示音视频信息和事件信息，便于对接和调试。

参数	描述
showType	0：不显示；1：显示精简版（仅显示音视频信息）；2：显示完整版（包含音视频信息和事件信息）。

TRTCViewMargin

TRTCViewMargin

public TRTCViewMargin (float leftMargin
float rightMargin
float topMargin
float bottomMargin)

设置仪表盘的边距

用于调整仪表盘在视频渲染控件中的位置，必须在 showDebugView 之前调用才能生效。

参数	描述
margin	仪表盘内边距，注意这里是基于 parentView 的百分比，margin 的取值范围是0 – 1。
userId	用户 ID。

callExperimentalAPI

callExperimentalAPI

String callExperimentalAPI (String jsonStr)

调用实验性接口

enablePayloadPrivateEncryption

enablePayloadPrivateEncryption

int enablePayloadPrivateEncryption (boolean enabled
TRTCCloudDef. TRTCPayloadPrivateEncryptionCo

开启或关闭媒体流私有加密

在安全要求较高的场景下，TRTC 建议您在加入房间前，调用 enablePayloadPrivateEncryption 方法开启媒体流私有加密。
用户退出房间后，SDK 会自动关闭私有加密。如需重新开启私有加密，您需要在用户再次加入房间前调用该方法。

参数	描述
config	配置媒体流私有加密的算法和密钥，参见 TRTCPayloadPrivateEncryptionConfig 。
enabled	是否开启媒体流私有加密。

注意
TRTC 已经内置对媒体流进行加密后再传输，启用媒体流私有加密后将使用您传入的密匙与初始向量进行再次加密。

返回值说明：

接口调用结果，0: 方法调用成功， -1: 传入参数无效， -2: 功能已过期。若需解锁：请前往中国大陆站点开通 TRTC 旗舰版套餐 ，并填写 申请 ，审核通过后方可使用。

TRTCCloudListener

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTCCloudListener @ TXLiteAVSDK

Function: 腾讯云实时音视频的事件回调接口

TRTCCloudListener

TRTCVideoRenderListener

函数列表	描述
onRenderVideoFrame	自定义视频渲染回调

TRTCVideoFrameListener

函数列表	描述
onGLContextCreated	SDK 内部 OpenGL 环境已经创建的通知
onProcessVideoFrame	用于对接第三方美颜组件的视频处理回调
onGLContextDestory	SDK 内部 OpenGL 环境被销毁的通知

TRTCAudioFrameListener

函数列表	描述
onCapturedAudioFrame	本地采集并经过音频模块前处理后的音频数据回调
onLocalProcessedAudioFrame	本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调

onRemoteUserAudioFrame	混音前的每一路远程用户的音频数据
onMixedPlayAudioFrame	将各路待播放音频混合之后并在最终提交系统播放之前的数据回调
onMixedAllAudioFrame	SDK 所有音频混合后的音频数据（包括采集到的和待播放的）
onVoiceEarMonitorAudioFrame	耳返的音频数据

TRTCLogListener

函数列表	描述
onLog	本地 LOG 的打印回调

TRTCCloudListener

函数列表	描述
onError	错误事件回调
onWarning	警告事件回调
onEnterRoom	进入房间成功与否的事件回调
onExitRoom	离开房间的事件回调
onSwitchRole	切换角色的事件回调
onSwitchRoom	切换房间的结果回调
onConnectOtherRoom	请求跨房通话的结果回调
onDisConnectOtherRoom	结束跨房通话的结果回调
onUpdateOtherRoomForwardMode	更改跨房主播上行能力的结果回调
onRemoteUserEnterRoom	有用户加入当前房间

onRemoteUserLeaveRoom	有用户离开当前房间
onUserVideoAvailable	某远端用户发布/取消了主路视频画面
onUserSubStreamAvailable	某远端用户发布/取消了辅路视频画面
onUserAudioAvailable	某远端用户发布/取消了自己的音频
onFirstVideoFrame	SDK 开始渲染自己本地或远端用户的首帧画面
onFirstAudioFrame	SDK 开始播放远端用户的首帧音频
onSendFirstLocalVideoFrame	自己本地的首个视频帧已被发布出去
onSendFirstLocalAudioFrame	自己本地的首个音频帧已被发布出去
onRemoteVideoStatusUpdated	远端视频状态变化的事件回调
onRemoteAudioStatusUpdated	远端音频状态变化的事件回调
onUserVideoSizeChanged	用户视频大小发生改变回调
onNetworkQuality	网络质量的实时统计回调
onStatistics	音视频技术指标的实时统计回调
onSpeedTestResult	网速测试的结果回调
onConnectionLost	SDK 与云端的连接已经断开
onTryToReconnect	SDK 正在尝试重新连接到云端
onConnectionRecovery	SDK 与云端的连接已经恢复
onCameraDidReady	摄像头准备就绪
onMicDidReady	麦克风准备就绪
onAudioRouteChanged	当前音频路由发生变化（仅适用于移动设备）

版权所有：腾讯云计算（北京）有限责任公司

onUserVoiceVolume	音量大小的反馈回调
onRecvCustomCmdMsg	收到自定义消息的事件回调
onMissCustomCmdMsg	自定义消息丢失的事件回调
onRecvSEIMsg	收到 SEI 消息的回调
onStartPublishing	开始向腾讯云直播 CDN 上发布音视频流的事件回调
onStopPublishing	停止向腾讯云直播 CDN 上发布音视频流的事件回调
onStartPublishCDNStream	开始向非腾讯云 CDN 上发布音视频流的事件回调
onStopPublishCDNStream	停止向非腾讯云 CDN 上发布音视频流的事件回调
onSetMixTranscodingConfig	设置云端混流的排版布局和转码参数的事件回调
onStartPublishMediaStream	开始发布媒体流的事件回调
onUpdatePublishMediaStream	更新媒体流的事件回调
onStopPublishMediaStream	停止媒体流的事件回调
onCdnStreamStateChanged	RTMP/RTMPS 推流状态发生改变回调
onScreenCaptureStarted	屏幕分享开启的事件回调
onScreenCapturePaused	屏幕分享暂停的事件回调
onScreenCaptureResumed	屏幕分享恢复的事件回调
onScreenCaptureStopped	屏幕分享停止的事件回调
onLocalRecordBegin	本地录制任务已经开始的事件回调

版权所有：腾讯云计算（北京）有限责任公司

onLocalRecording	本地录制任务正在进行中的进展事件回调
onLocalRecordFragment	本地录制分片的事件回调
onLocalRecordComplete	本地录制任务已经结束的事件回调
onSnapshotComplete	本地截图完成的事件回调
onUserEnter	有主播加入当前房间（已废弃）
onUserExit	有主播离开当前房间（已废弃）
onAudioEffectFinished	音效播放已结束（已废弃）
onSpeedTest	服务器测速的结果回调（已废弃）

onRenderVideoFrame

onRenderVideoFrame

```
void onRenderVideoFrame(String userId,
                        int streamType,
                        TRTCCloudDef.TRTCVideoFrame frame)
```

自定义视频渲染回调

当您设置了本地或者远端的视频自定义渲染回调之后，SDK 就会将原本要交给渲染控件进行渲染的视频帧通过此回调接口抛送给您，以便于您进行自定义渲染。

参数	描述
frame	待渲染的视频帧信息。
streamType	频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。
userId	视频源的 userId，如果是本地视频回调（setLocalVideoRenderDelegate），该参数可以忽略。

onGLContextCreated

onGLContextCreated

SDK 内部 OpenGL 环境已经创建的通知

onProcessVideoFrame

onProcessVideoFrame

int onProcessVideoFrame (TRTCCloudDef.TRTCVideoFrame srcFrame, TRTCCloudDef.TRTCVideoFrame dstFrame)

用于对接第三方美颜组件的视频处理回调

如果您选购了第三方美颜组件，就需要在 TRTCCloud 中设置第三方美颜回调，之后 TRTC 就会将原本要进行预处理的视频帧通过此回调接口抛送给您。
之后您就可以将 TRTC 抛出的视频帧交给第三方美颜组件进行图像处理，由于抛出的数据是可读且可写的，因此第三方美颜的处理结果也可以同步给 TRTC 进行后续的编码和发送。

情况一：美颜组件自身会产生新的纹理。
如果您使用的美颜组件会在处理图像的过程中产生一帧全新的纹理（用于承载处理后的图像），那请您在回调函数中将 dstFrame.textureId 设置为新纹理的 ID：

```
private final TRTCVideoFrameListener mVideoFrameListener = new
TRTCVideoFrameListener() {
    @Override
    public void onGLContextCreated() {
        mFURenderer.onSurfaceCreated();
        mFURenderer.setUseTexAsync(true);
    }
    @Override
    public int onProcessVideoFrame(TRTCVideoFrame srcFrame, TRTCVideoFrame
dstFrame) {
        dstFrame.texture.textureId =
mFURenderer.onDrawFrameSingleInput(srcFrame.texture.textureId,
srcFrame.width, srcFrame.height);
```

```
        return 0;
    }
    @Override
    public void onGLContextDestory() {
        mFURenderer.onSurfaceDestroyed();
    }
};
```

情况二：美颜组件需要您提供目标纹理。
如果您使用的第三方美颜模块并不生成新的纹理，而是需要您设置给该模块一个输入纹理和一个输出纹理，则可以考虑如下方案：

```
int onProcessVideoFrame(TRTCCloudDef.TRTCVideoFrame srcFrame,
TRTCCloudDef.TRTCVideoFrame dstFrame) {
    thirdparty_process(srcFrame.texture.textureId, srcFrame.width,
srcFrame.height, dstFrame.texture.textureId);
    return 0;
}
```

参数	描述
dstFrame	用于接收第三方美颜处理过的视频画面。
srcFrame	用于承载 TRTC 采集到的摄像头画面。

注意
目前仅支持 OpenGL 纹理方案（ PC 仅支持 TRTCVideoBufferType_Buffer 格式）。

onGLContextDestory

onGLContextDestory

SDK 内部 OpenGL 环境被销毁的通知

onCapturedAudioFrame

onCapturedAudioFrame

```
void
onCapturedAudioFrame
(TRTCCloudDef.TRTCAudioFrame frame)
```

本地采集并经过音频模块前处理后的音频数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把刚采集到并经过前处理(ANS、AEC、AGC）之后的数据，以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。
3. 此接口回调出的音频数据已经经过了前处理(ANS、AEC、AGC），但**不包含**背景音、音效、混响等前处理效果，延迟较低。

onLocalProcessedAudioFrame

onLocalProcessedAudioFrame

```
void
onLocalProcessedAudioFrame
(TRTCCloudDef.TRTCAudioFrame frame)
```

本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把刚采集到并经过前处理、音效处理和混 BGM 之后的数据，在最终进行网络编码之前，以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

特殊说明：
您可以通过设置接口中的 `TRTCAudioFrame.extraData` 字段，达到传输信令的目的。由于音频帧头部的数据块不能太大，建议您写入 `extraData` 时，尽量将信令控制在几个字节的大小，如果超过 100 个字节，写入的数据不会被发送。

房间内其他用户可以通过 TRTCAudioFrameListener 中的 `onRemoteUserAudioFrame` 中的 `TRTCAudioFrame.extraData` 字段回调接收数据。

参数	描述
frame	PCM 格式的音频数据帧。

> **注意**
> 1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
> 2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。
> 3. 此接口回调出的数据已经经过了前处理(ANS、AEC、AGC）、音效和混 BGM 处理，声音的延迟相比于 onCapturedAudioFrame 要高一些。

onRemoteUserAudioFrame

onRemoteUserAudioFrame

```
void onRemoteUserAudioFrame(TRTCCloudDef.TRTCAudioFrame frame,
                            String userId)
```

混音前的每一路远程用户的音频数据

当您设置完音频数据自定义回调之后，SDK 内部会把远端的每一路原始数据，在最终混音之前，以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。
userId	用户标识。

注意
此接口回调出的音频数据是只读的，不支持修改。

onMixedPlayAudioFrame

onMixedPlayAudioFrame

void onMixedPlayAudioFrame (TRTCCloudDef.TRTCAudioFrame frame)

将各路待播放音频混合之后并在最终提交系统播放之前的数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把各路待播放的音频混合之后的音频数据，在提交系统播放之前，以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。
3. 此接口回调出的是对各路待播放音频数据的混合，但其中并不包含耳返的音频数据。

onMixedAllAudioFrame

onMixedAllAudioFrame

```
void onMixedAllAudioFrame (TRTCCloudDef.TRTCAudioFrame frame)
```

SDK 所有音频混合后的音频数据（包括采集到的和待播放的）

当您设置完音频数据自定义回调之后，SDK 内部会把所有采集到的和待播放的音频数据混合起来，以 PCM 格式的形式通过本接口回调给您，便于您进行自定义录制。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 此接口回调出的是SDK所有音频数据的混合数据，包括：经过 3A 前处理、特效叠加以及背景音乐混音后的本地音频，所有远端音频，但不包括耳返音频。
2. 此接口回调出的音频数据不支持修改。

onVoiceEarMonitorAudioFrame

onVoiceEarMonitorAudioFrame

void onVoiceEarMonitorAudioFrame	(TRTCCloudDef.TRTCAudioFrame frame)

耳返的音频数据

当您设置完音频数据自定义回调之后，SDK 内部会把耳返的音频数据在播放之前以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长不固定，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式 48000 采样率、单声道、16采样点位宽为例，0.02s 的音频数据字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，否则会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。

onLog

onLog

void onLog	(String log int level String module)

本地 LOG 的打印回调

如果您希望捕获 SDK 的本地日志打印行为，可以通过设置日志回调，让 SDK 将要打印的日志都通过本回调接口抛送给您。

参数	描述

level	日志等级请参见 TRTC_LOG_LEVEL。
log	日志内容。
module	保留字段，暂无具体意义，目前为固定值 TXLiteAVSDK。

onError

onError

```
void onError (int errCode
              String errMsg
              Bundle extraInfo)
```

错误事件回调

错误事件，表示 SDK 抛出的不可恢复的错误，比如进入房间失败或设备开启失败等。
参考文档：错误码表

参数	描述
errCode	错误码
errMsg	错误信息
extInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题

onWarning

onWarning

```
void onWarning (int warningCode
                String warningMsg
                Bundle extraInfo)
```

警告事件回调

警告事件，表示 SDK 抛出的提示性问题，比如视频出现卡顿或 CPU 使用率太高等。
参考文档：错误码表

参数	描述
extInfo	扩展信息字段，个别警告码可能会带额外的信息帮助定位问题
warningCode	警告码
warningMsg	警告信息

onEnterRoom

onEnterRoom

void onEnterRoom	(long result)

进入房间成功与否的事件回调

调用 TRTCCloud 中的 enterRoom() 接口执行进房操作后，会收到来自 TRTCCloudListener 的 onEnterRoom(result) 回调：

- 如果加入成功，回调 result 会是一个正数（result > 0），代表进入房间所消耗的时间，单位是毫秒（ms）。
- 如果加入失败，回调 result 会是一个负数（result < 0），代表失败原因的错误码。

进房失败的错误码含义请参见错误码表 。

参数	描述
result	result > 0 时为进房耗时（ms），result < 0 时为进房错误码。

注意
1. 在 Ver6.6 之前的版本，只有进房成功会抛出 onEnterRoom(result) 回调，进房失败由 onError() 回调抛出。

> 2. 在 Ver6.6 之后的版本：无论进房成功或失败，均会抛出 onEnterRoom(result) 回调，同时进房失败也会有 onError() 回调抛出。

onExitRoom

onExitRoom

```
void
onExitRoom          (int reason)
```

离开房间的事件回调

调用 TRTCCloud 中的 exitRoom 接口会执行退出房间的相关逻辑，例如释放音视频设备资源和编解码器资源等。
待 SDK 占用的所有资源释放完毕后，SDK 会抛出 onExitRoom() 回调通知到您。

如果您要再次调用 enterRoom 或者切换到其他的音视频 SDK，请等待 onExitRoom 回调到来后再执行相关操作。否则可能会遇到例如摄像头、麦克风设备被强占等各种异常问题。

参数	描述
reason	离开房间原因，0：主动调用 exitRoom 退出房间；1：被服务器踢出当前房间；2：当前房间整个被解散。

onSwitchRole

onSwitchRole

```
void
onSwitchRole        (final int errCode

                     final String errMsg)
```

切换角色的事件回调

调用 TRTCCloud 中的 switchRole() 接口会切换主播和观众的角色，该操作会伴随一个线路切换的过程，待 SDK 切换完成后，会抛出 onSwitchRole() 事件回调。

参数	描述
errCode	错误码，ERR_NULL 代表切换成功，其他请参见 错误码 。
errMsg	错误信息。

onSwitchRoom

onSwitchRoom

```
void onSwitchRoom (final int errCode
                   final String errMsg)
```

切换房间的结果回调

调用 TRTCCloud 中的 switchRoom() 接口可以让用户快速地从一个房间切换到另一个房间，待 SDK 切换完成后，会抛出 onSwitchRoom() 事件回调。

参数	描述
errCode	错误码，ERR_NULL 代表切换成功，其他请参见 错误码 。
errMsg	错误信息。

onConnectOtherRoom

onConnectOtherRoom

```
void onConnectOtherRoom (final String userId
                         final int errCode
                         final String errMsg)
```

请求跨房通话的结果回调

调用 TRTCCloud 中的 connectOtherRoom() 接口会将两个不同房间中的主播拉通视频通话，也就是所谓的“主播PK”功能。
调用者会收到 onConnectOtherRoom() 回调来获知跨房通话是否成功，
如果成功，两个房间中的所有用户都会收到来自另一个房间中的 PK 主播的 onUserVideoAvailable() 回调。

参数	描述
errCode	错误码，ERR_NULL 代表切换成功，其他请参见 错误码 。
errMsg	错误信息。
userId	要跨房通话的另一个房间中的主播的用户 ID。

onDisConnectOtherRoom

onDisConnectOtherRoom

```
void onDisConnectOtherRoom (final int errCode
                            final String errMsg)
```

结束跨房通话的结果回调

onUpdateOtherRoomForwardMode

onUpdateOtherRoomForwardMode

```
void onUpdateOtherRoomForwardMode (final int errCode
                                   final String errMsg)
```

更改跨房主播上行能力的结果回调

onRemoteUserEnterRoom

onRemoteUserEnterRoom

void onRemoteUserEnterRoom	(String userId)

有用户加入当前房间

出于性能方面的考虑，在 TRTC 两种不同的应用场景（即 AppScene，在 enterRoom 时通过第二个参数指定）下，该通知的行为会有差别：

- 直播类场景（TRTC_APP_SCENE_LIVE 和 TRTC_APP_SCENE_VOICE_CHATROOM）：该场景下的用户区分主播和观众两种角色，只有主播进入房间时才会触发该通知，观众进入房间时不会触发该通知。
- 通话类场景（TRTC_APP_SCENE_VIDEOCALL 和 TRTC_APP_SCENE_AUDIOCALL）：该场景下的用户没有角色的区分（可认为都是主播），任何用户进入房间都会触发该通知。

参数	描述
userId	远端用户的用户标识。

注意

1. 事件回调 onRemoteUserEnterRoom 和 onRemoteUserLeaveRoom 只适用于维护当前房间里的“用户列表”，有此事件回调不代表一定有视频画面。
2. 如果需要显示远程画面，请监听代表某个用户是否有视频画面的 onUserVideoAvailable() 事件回调。

onRemoteUserLeaveRoom

onRemoteUserLeaveRoom

void onRemoteUserLeaveRoom	(String userId int reason)

有用户离开当前房间

与 onRemoteUserEnterRoom 相对应，在两种不同的应用场景（即 AppScene，在 enterRoom 时通过第二个参数指定）下，该通知的行为会有差别：

- 直播类场景（TRTC_APP_SCENE_LIVE 和 TRTC_APP_SCENE_VOICE_CHATROOM）：只有主播离开房间时才会触发该通知，观众离开房间不会触发该通知。
- 通话类场景（TRTC_APP_SCENE_VIDEOCALL 和 TRTC_APP_SCENE_AUDIOCALL）：该场景下用户没有角色的区别，任何用户的离开都会触发该通知。

参数	描述
reason	离开原因，0表示用户主动退出房间，1表示用户超时退出，2表示被踢出房间，3表示主播因切换到观众退出房间。
userId	远端用户的用户标识。

onUserVideoAvailable

onUserVideoAvailable

```
void onUserVideoAvailable (String userId
                           boolean available)
```

某远端用户发布/取消了主路视频画面

主路画面 一般被用于承载摄像头画面。当您收到 onUserVideoAvailable(userId, true) 通知时，表示该路画面已经有可播放的视频帧到达。
此时，您需要调用 startRemoteView 接口订阅该用户的远程画面，订阅成功后，您会继续收到该用户的首帧画面渲染回调 onFirstVideoFrame(userid)。

当您收到 onUserVideoAvailable(userId, false) 通知时，表示该路远程画面已经被关闭，关闭的原因可能是该用户调用了 muteLocalVideo 或 stopLocalPreview 。

参数	描述
available	该用户是否发布（或取消发布）了主路视频画面，true：发布；false：取消发布。
userId	远端用户的用户标识。

onUserSubStreamAvailable

onUserSubStreamAvailable

void onUserSubStreamAvailable	(String userId boolean available)

某远端用户发布/取消了辅路视频画面

“辅路画面”一般被用于承载屏幕分享的画面。当您收到 onUserSubStreamAvailable(userId, true) 通知时，表示该路画面已经有可播放的视频帧到达。
此时，您需要调用 startRemoteView 接口订阅该用户的远程画面，订阅成功后，您会继续收到该用户的首帧画面渲染回调 onFirstVideoFrame(userid)。

参数	描述
available	该用户是否发布（或取消发布）了辅路视频画面，true: 发布；false：取消发布。
userId	远端用户的用户标识。

注意
显示辅路画面使用的函数是 startRemoteView 而非 startRemoteSubStreamView，startRemoteSubStreamView 已经被废弃。

onUserAudioAvailable

onUserAudioAvailable

void onUserAudioAvailable	(String userId boolean available)

某远端用户发布/取消了自己的音频

当您收到 onUserAudioAvailable(userId, true) 通知时，表示该用户发布了自己的声音，此时 SDK 的表现为：

- 在自动订阅模式下，您无需做任何操作，SDK 会自动播放该用户的声音。
- 在手动订阅模式下，您可以通过 muteRemoteAudio(userid, false) 来播放该用户的声音。

参数	描述
available	该用户是否发布（或取消发布）了自己的音频，true: 发布；false：取消发布。
userId	远端用户的用户标识。

注意
SDK 默认使用自动订阅模式，您可以通过 setDefaultStreamRecvMode 设置为手动订阅，但需要在您进入房间之前调用才生效。

onFirstVideoFrame

onFirstVideoFrame

```
void onFirstVideoFrame    (String userId
                           int streamType
                           int width
                           int height)
```

SDK 开始渲染自己本地或远端用户的首帧画面

SDK 会在渲染自己本地或远端用户的首帧画面时抛出该事件，您可以通过回调事件中的 userId 参数来判断事件来自于“本地”还是来自于“远端”。

- 如果 userId 为空值，代表 SDK 已经开始渲染自己本地的视频画面，不过前提是您已经调用了 startLocalPreview 或 startScreenCapture。
- 如果 userId 不为空，代表 SDK 已经开始渲染远端用户的视频画面，不过前提是您已经调用了 startRemoteView 订阅了该用户的视频画面。

参数	描述
height	画面的高度。
streamType	视频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。
userId	本地或远端的用户标识，如果 userId 为空值代表自己本地的首帧画面已到来，userId 不为空则代表远端用户的首帧画面已到来。
width	画面的宽度。

注意

1. 只有当您调用了 startLocalPreview 或 startScreenCapture 之后，才会触发自己本地的首帧画面事件回调。
2. 只有当您调用了 startRemoteView 或 startRemoteSubStreamView 之后，才会触发远端用户的首帧画面事件回调。

onFirstAudioFrame

onFirstAudioFrame

void onFirstAudioFrame (String userId)

SDK 开始播放远端用户的首帧音频

SDK 会在播放远端用户的首帧音频时抛出该事件，本地音频的首帧事件暂不抛出。

参数	描述
userId	远端用户的用户标识。

onSendFirstLocalVideoFrame

onSendFirstLocalVideoFrame

void onSendFirstLocalVideoFrame (int streamType)

自己本地的首个视频帧已被发布出去

当您成功进入房间并通过 startLocalPreview 或 startScreenCapture 开启本地视频采集之后（开启采集和进入房间的先后顺序无影响），SDK 就会开始进行视频编码，并通过自身的网络模块向云端发布自己本地的视频数据。当 SDK 成功地向云端送出自己的第一帧视频数据帧以后，就会抛出 onSendFirstLocalVideoFrame 事件回调。

参数	描述

streamType	视频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。

onSendFirstLocalAudioFrame

onSendFirstLocalAudioFrame

自己本地的首个音频帧已被发布出去

当您成功进入房间并通过 startLocalAudio 开启本地音频采集之后（开启采集和进入房间的先后顺序无影响），SDK 就会开始进行音频编码，并通过自身的网络模块向云端发布自己本地的音频数据。当 SDK 成功地向云端送出自己的第一帧音频数据帧以后，就会抛出 onSendFirstLocalAudioFrame 事件回调。

onRemoteVideoStatusUpdated

onRemoteVideoStatusUpdated

```
void onRemoteVideoStatusUpdated(String userId
                                int streamType
                                int status
                                int reason
                                Bundle extraInfo)
```

远端视频状态变化的事件回调

您可以通过此事件回调获取远端每一路画面的播放状态（包括 Playing、Loading 和 Stopped 三种状态），从而进行相应的 UI 展示。

参数	描述
extrainfo	额外信息。
reason	视频状态改变的原因。
status	视频状态：包括 Playing、Loading 和 Stopped 三种状态。

streamType	视频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。
userId	用户标识。

onRemoteAudioStatusUpdated

onRemoteAudioStatusUpdated

```
void onRemoteAudioStatusUpdated (String userId
                                 int status
                                 int reason
                                 Bundle extraInfo)
```

远端音频状态变化的事件回调

您可以通过此事件回调获取远端每一路音频的播放状态（包括 Playing、Loading 和 Stopped 三种状态），从而进行相应的 UI 展示。

参数	描述
extrainfo	额外信息。
reason	音频状态改变的原因。
status	音频状态：包括 Playing、Loading 和 Stopped 三种状态。
userId	用户标识。

onUserVideoSizeChanged

onUserVideoSizeChanged

```
void onUserVideoSizeChanged (String userId
                             int streamType
```

int newWidth

int newHeight)

用户视频大小发生改变回调

当您收到 onUserVideoSizeChanged(userId, streamtype, newWidth, newHeight) 通知时，表示该路画面大小发生了调整，调整的原因可能是该用户调用了 setVideoEncoderParam 或者 setSubStreamEncoderParam 重新设置了画面尺寸。

参数	描述
newHeight	视频流的高度（像素）。
newWidth	视频流的宽度（像素）。
streamType	视频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。
userId	用户标识。

onNetworkQuality

onNetworkQuality

void onNetworkQuality (TRTCCloudDef. TRTCQuality localQuality

ArrayList<TRTCCloudDef. TRTCQuality > remoteQuality)

网络质量的实时统计回调

该统计回调每间隔2秒抛出一次，用于通知 SDK 感知到的当前网络的上行和下行质量。
SDK 会使用一组内嵌的自研算法对当前网络的延迟高低、带宽大小以及稳定情况进行评估，并计算出一个评估结果：如果评估结果为 1（Excellent）代表当前的网络情况非常好，如果评估结果为 6（Down）代表当前网络无法支撑 TRTC 的正常通话。

参数	描述
localQuality	上行网络质量。
remoteQuality	下行网络质量，代表数据流历经“远端->云端->本端”这样一条完整的传输链路后，最终在本端统计到的数据质量。因此，这里的下行网络质量代表的是远端上

	行链路与本端下行链路共同影响的结果。

注意
暂时无法通过该接口单独判定远端用户的上行质量。

onStatistics

onStatistics

void onStatistics （TRTCStatistics statistics）

音视频技术指标的实时统计回调

该统计回调每间隔2秒抛出一次，用于通知 SDK 内部音频、视频以及网络相关的专业技术指标，这些信息在 TRTCStatistics 均有罗列：

- 视频统计信息：视频的分辨率（resolution）、帧率（FPS）和比特率（bitrate）等信息。
- 音频统计信息：音频的采样率（samplerate）、声道（channel）和比特率（bitrate）等信息。
- 网络统计信息：SDK 和云端一次往返（SDK > Cloud > SDK）的网络耗时（rtt）、丢包率（loss）、上行流量（sentBytes）和下行流量（receivedBytes）等信息。

参数	描述
statistics	统计数据，包括自己本地的统计信息和远端用户的统计信息，详情请参考 TRTCStatistics 。

注意
如果您只需要获知当前网络质量的好坏，并不需要花太多时间研究本统计回调，更推荐您使用 onNetworkQuality 来解决问题。

onSpeedTestResult

onSpeedTestResult

void onSpeedTestResult (TRTCCloudDef. TRTCSpeedTestResult result)

网速测试的结果回调

该统计回调由 startSpeedTest: 触发。

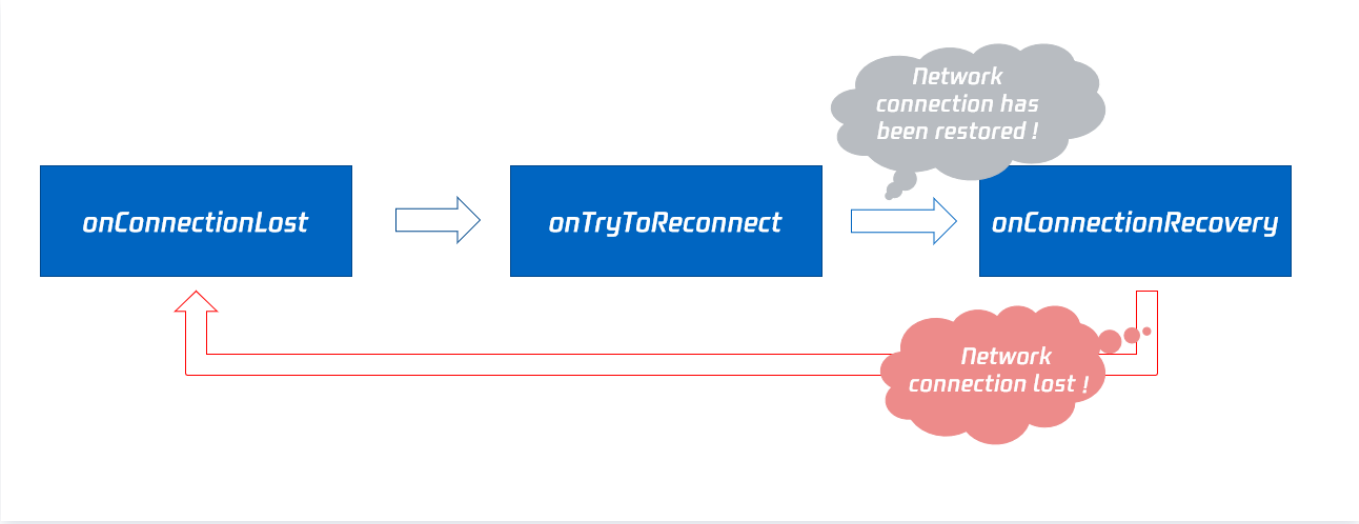
参数	描述
result	网速测试结果，包括丢包、往返延迟、上下行的带宽速率，详情请参见 TRTCSpeedTestResult 。

onConnectionLost

onConnectionLost

SDK 与云端的连接已经断开

SDK 会在跟云端的连接断开时抛出此事件回调，导致断开的原因大多是网络不可用或者网络切换所致，比如用户在通话中走进电梯时就可能会遇到此事件。
在抛出此事件之后，SDK 会努力跟云端重新建立连接，重连过程中会抛出 onTryToReconnect，连接恢复后会抛出 onConnectionRecovery 。
所以，SDK 会在如下三个连接相关的事件中按如下规律切换：



onTryToReconnect

onTryToReconnect

SDK 正在尝试重新连接到云端

SDK 会在跟云端的连接断开时抛出 onConnectionLost，之后会努力跟云端重新建立连接并抛出本事件，连接恢复后会抛出 onConnectionRecovery 。

onConnectionRecovery

onConnectionRecovery

SDK 与云端的连接已经恢复

SDK 会在跟云端的连接断开时抛出 onConnectionLost，之后会努力跟云端重新建立连接并抛出 onTryToReconnect，连接恢复后会抛出本事件回调。

onCameraDidReady

onCameraDidReady

摄像头准备就绪

当您调用 startLocalPreivew 之后，SDK 会尝试启动摄像头，如果摄像头能够启动成功就会抛出本事件。如果启动失败，大概率是因为当前应用没有获得访问摄像头的权限，或者摄像头当前正在被其他程序独占使用中。
您可以通过捕获 onError 事件回调获知这些异常情况并通过 UI 界面提示用户。

onMicDidReady

onMicDidReady

麦克风准备就绪

当您调用 startLocalAudio 之后，SDK 会尝试启动麦克风，如果麦克风能够启动成功就会抛出本事件。如果启动失败，大概率是因为当前应用没有获得访问麦克风的权限，或者麦克风当前正在被其他程序独占使用中。
您可以通过捕获 onError 事件回调获知这些异常情况并通过 UI 界面提示用户。

版权所有：腾讯云计算（北京）有限责任公司

onAudioRouteChanged

onAudioRouteChanged

```
void
onAudioRouteChanged
(int newRoute
int oldRoute)
```

当前音频路由发生变化（仅适用于移动设备）

所谓“音频路由”，是指声音是从手机的扬声器还是从听筒中播放出来，音频路由变化也就是声音的播放位置发生了变化。

- 当音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
- 当音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。
- 当音频路由为有线耳机。
- 当音频路由为蓝牙耳机。
- 当音频路由为USB专业声卡设备。

参数	描述
fromRoute	变更前的音频路由。
route	音频路由，即声音由哪里输出（扬声器、听筒）。

onUserVoiceVolume

onUserVoiceVolume

```
void
onUserVoiceVolume
(ArrayList<TRTCCloudDef.TRTCVolumeInfo> userVolumes
int totalVolume)
```

音量大小的反馈回调

SDK 可以评估每一路音频的音量大小，并每隔一段时间抛出该事件回调，您可以根据音量大小在 UI 上做出相应的提示，比如 `波形图` 或 `音量槽` 。

要完成这个功能， 您需要先调用 enableAudioVolumeEvaluation 开启这个能力并设定事件抛出的时间间隔。
需要补充说明的是，无论当前房间中是否有人说话，SDK 都会按照您设定的时间间隔定时抛出此事件回调。

参数	描述
totalVolume	所有远端用户的总音量大小，取值范围 0 – 100。
userVolumes	是一个数组，用于承载所有正在说话的用户的音量大小，取值范围 0 – 100。

注意
userVolumes 为一个数组，对于数组中的每一个元素，当 userId 为空时表示本地麦克风采集的音量大小，当 userId 不为空时代表远端用户的音量大小。

onRecvCustomCmdMsg

onRecvCustomCmdMsg

```
void onRecvCustomCmdMsg(String userId
                        int cmdID
                        int seq
                        byte[] message)
```

收到自定义消息的事件回调

当房间中的某个用户使用 sendCustomCmdMsg 发送自定义 UDP 消息时，房间中的其他用户可以通过 onRecvCustomCmdMsg 事件回调接收到该条消息。

参数	描述
cmdID	命令 ID。
message	消息数据。
seq	消息序号。

userId	用户标识。

onMissCustomCmdMsg

onMissCustomCmdMsg

```
void
onMissCustomCmdMsg
(String userId
int cmdID
int errCode
int missed)
```

自定义消息丢失的事件回调

当您使用 sendCustomCmdMsg 发送自定义 UDP 消息时，即使设置了可靠传输（reliable），也无法确 100% 不丢失，只是丢消息概率极低，能满足常规可靠性要求。
在发送端设置了可靠运输（reliable）后，SDK 都会通过此回调通知过去时间段内（通常为5s）传输途中丢失的自定义消息数量统计信息。

参数	描述
cmdID	命令 ID。
errCode	错误码。
missed	丢失的消息数量。
userId	用户标识。

注意
只有在发送端设置了可靠传输（reliable），接收方才能收到消息的丢失回调。

onRecvSEIMsg

onRecvSEIMsg

void onRecvSEIMsg	(String userId byte[] data)

收到 SEI 消息的回调

当房间中的某个用户使用 sendSEIMsg 借助视频数据帧发送 SEI 消息时，房间中的其他用户可以通过 onRecvSEIMsg 事件回调接收到该条消息。

参数	描述
message	数据。
userId	用户标识。

onStartPublishing

onStartPublishing

void onStartPublishing	(int err String errMsg)

开始向腾讯云直播 CDN 上发布音视频流的事件回调

当您调用 startPublishing 开始向腾讯云直播 CDN 上发布音视频流时，SDK 会立刻将这一指令同步给云端服务器。
随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	0表示成功，其余值表示失败。
errMsg	具体错误原因。

onStopPublishing

onStopPublishing

```
void onStopPublishing (int err
                       String errMsg)
```

停止向腾讯云直播 CDN 上发布音视频流的事件回调

当您调用 stopPublishing 停止向腾讯云直播 CDN 上发布音视频流时，SDK 会立刻将这一指令同步给云端服务器。
随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	0表示成功，其余值表示失败。
errMsg	具体错误原因。

onStartPublishCDNStream

onStartPublishCDNStream

```
void onStartPublishCDNStream (int err
                              String errMsg)
```

开始向非腾讯云 CDN 上发布音视频流的事件回调

当您调用 startPublishCDNStream 开始向非腾讯云直播 CDN 上发布音视频流时，SDK 会立刻将这一指令同步给云端服务器。随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	0表示成功，其余值表示失败。
errMsg	具体错误原因。

> **注意**
> 当您收到成功的事件回调时，只是说明您的发布指令已经同步到腾讯云后台服务器，但如果目标 CDN 厂商的服务器不接收该条视频流，依然可能导致发布失败。

onStopPublishCDNStream

onStopPublishCDNStream

```
void onStopPublishCDNStream(int err, String errMsg)
```

停止向非腾讯云 CDN 上发布音视频流的事件回调

当您调用 stopPublishCDNStream 开始向非腾讯云直播 CDN 上发布音视频流时，SDK 会立刻将这一指令同步给云端服务器。随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	0表示成功，其余值表示失败。
errMsg	具体错误原因。

onSetMixTranscodingConfig

onSetMixTranscodingConfig

```
void onSetMixTranscodingConfig(int err, String errMsg)
```

设置云端混流的排版布局和转码参数的事件回调

当您调用 setMixTranscodingConfig 调整云端混流的排版布局和转码参数时，SDK 会立刻将这一调整指令同步给云端服务器。随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	错误码：0表示成功，其余值表示失败。
errMsg	具体的错误原因。

onStartPublishMediaStream

onStartPublishMediaStream

```
void onStartPublishMediaStream (String taskId
                                int code
                                String message
                                Bundle extraInfo)
```

开始发布媒体流的事件回调

当您调用 startPublishMediaStream 开始向 TRTC 后台服务发布媒体流时，SDK 会立刻将这一指令同步给云端服务器，随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
code	回调结果，0 表示成功，其余值表示失败。
extraInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。
message	具体回调信息。
taskId	当请求成功时，TRTC 后台会在回调中提供给您这项任务的 taskId，后续您可以通过该 taskId 结合 updatePublishMediaStream 和 stopPublishMediaStream 进行更新和停止。

onUpdatePublishMediaStream

onUpdatePublishMediaStream

```
void onUpdatePublishMediaStream (String taskId
                                 int code
                                 String message
                                 Bundle extraInfo)
```

更新媒体流的事件回调

当您调用媒体流发布接口（updatePublishMediaStream）开始向 TRTC 后台服务更新媒体流时，SDK 会立刻将这一指令同步给云端服务器，随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
code	回调结果，0 表示成功，其余值表示失败。
extraInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。
message	具体回调信息。
taskId	您调用媒体流发布接口（updatePublishMediaStream）时传入的 taskId，会通过此回调再带回给您，用于标识该回调属于哪一次更新请求。

onStopPublishMediaStream

onStopPublishMediaStream

```
void onStopPublishMediaStream (String taskId
                               int code
                               String message
```

Bundle extraInfo)

停止媒体流的事件回调

当您调用停止发布媒体流（stopPublishMediaStream）开始向 TRTC 后台服务停止媒体流时，SDK 会立刻将这一指令同步给云端服务器，随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
code	回调结果，0 表示成功，其余值表示失败。
extraInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。
message	具体回调信息。
taskId	您调用停止发布媒体流（stopPublishMediaStream）时传入的 taskId，会通过此回调再带回给您，用于标识该回调属于哪一次停止请求。

onCdnStreamStateChanged

onCdnStreamStateChanged

```
void onCdnStreamStateChanged(String cdnUrl
                             int status
                             int code
                             String msg
                             Bundle extraInfo)
```

RTMP/RTMPS 推流状态发生改变回调

当您调用 startPublishMediaStream 开始向 TRTC 后台服务发布媒体流时，SDK 会立刻将这一指令同步给云端服务。
若您在目标推流配置（TRTCPublishTarget）设置了向腾讯或者第三方 CDN 上发布音视频流的 URL 配置，则具体 RTMP 或者 RTMPS 推流状态将通过此回调同步给您。

参数	描述
cdnUrl	您调用 startPublishMediaStream 时通过目标推流配置（TRTCPublishTarget）传入的 url，在推流状态变更时，会通过此回调同步给您。
code	推流结果，0 表示成功，其余值表示出错。
extraInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。
message	具体推流信息。
status	推流状态。 • 0：推流未开始或者已结束。在您调用 stopPublishMediaStream 时会返回该状态。 • 1：正在连接 TRTC 服务器和 CDN 服务器。若无法立刻成功，TRTC 后台服务会多次重试尝试推流，并返回该状态（5s回调一次）。如成功进行推流，则进入状态 2；如服务器出错或 60 秒内未成功推流，则进入状态 4。 • 2：CDN 推流正在进行。在成功推流后，会返回该状态。 • 3：TRTC 服务器和 CDN 服务器推流中断，正在恢复。当 CDN 出现异常，或推流短暂中断时，TRTC 后台服务会自动尝试恢复推流，并返回该状态（5s回调一次）。如成功恢复推流，则进入状态 2；如服务器出错或 60 秒内未成功恢复，则进入状态 4。 • 4：TRTC 服务器和 CDN 服务器推流中断，且恢复或连接超时。即此时推流失败，你可以再次调用 updatePublishMediaStream 尝试推流。 • 5：正在断开 TRTC 服务器和 CDN 服务器。在您调用 stopPublishMediaStream 时，TRTC 后台服务会依次同步状态 5 和状态 0。

onScreenCaptureStarted

onScreenCaptureStarted

屏幕分享开启的事件回调

当您通过 startScreenCapture 等相关接口启动屏幕分享时，SDK 便会抛出此事件回调。

onScreenCapturePaused

onScreenCapturePaused

屏幕分享暂停的事件回调

当您通过 pauseScreenCapture 暂停屏幕分享时，SDK 便会抛出此事件回调。

onScreenCaptureResumed

onScreenCaptureResumed

屏幕分享恢复的事件回调

当您通过 resumeScreenCapture 恢复屏幕分享时，SDK 便会抛出此事件回调。

onScreenCaptureStopped

onScreenCaptureStopped

```
void onScreenCaptureStopped (int reason)
```

屏幕分享停止的事件回调

当您通过 stopScreenCapture 停止屏幕分享时，SDK 便会抛出此事件回调。

参数	描述
reason	停止原因，0：用户主动停止；1：屏幕窗口关闭导致停止；2：表示屏幕分享的显示屏状态变更（如接口被拔出、投影模式变更等）。

onLocalRecordBegin

onLocalRecordBegin

```
void onLocalRecordBegin (int errCode
                         String storagePath)
```

本地录制任务已经开始的事件回调

当您调用 startLocalRecording 启动本地媒体录制任务时，SDK 会抛出该事件回调，用于通知您录制任务是否已经顺利启动。

参数	描述
errCode	状态码。 • 0：录制任务启动成功。 • -1：内部错误导致录制任务启动失败。 • -2：文件后缀名有误（比如不支持的录制格式）。 • -6：录制已经启动，需要先停止录制。 • -7：录制文件已存在，需要先删除文件。 • -8：录制目录无写入权限，请检查目录权限问题。
storagePath	录制文件存储路径。

onLocalRecording

onLocalRecording

```
void onLocalRecording    (long duration
                          String storagePath)
```

本地录制任务正在进行中的进展事件回调

当您调用 startLocalRecording 成功启动本地媒体录制任务后，SDK 会定时地抛出本事件回调。
您可通过捕获该事件回调来获知录制任务的健康状况。
您可以在 startLocalRecording 时设定本事件回调的抛出间隔。

参数	描述
duration	已经录制的累计时长，单位毫秒。
storagePath	录制文件存储路径。

onLocalRecordFragment

onLocalRecordFragment

void onLocalRecordFragment	(String storagePath)

本地录制分片的事件回调

当您开启分片录制时，每完成一个分片，都会通过此接口回调。

参数	描述
storagePath	分片文件存储路径。

onLocalRecordComplete

onLocalRecordComplete

void onLocalRecordComplete	(int errCode String storagePath)

本地录制任务已经结束的事件回调

当您调用 stopLocalRecording 停止本地媒体录制任务时，SDK 会抛出该事件回调，用于通知您录制任务的最终结果。

参数	描述
errCode	状态码。 • 0：结束录制任务成功。 • -1：录制失败。 • -2：切换分辨率或横竖屏导致录制结束。 • -3：录制时间太短，或未采集到任何视频或音频数据，请检查录制时长，或是否已开启音、视频采集。
storagePat	录制文件存储路径。

h	

onSnapshotComplete

onSnapshotComplete

void onSnapshotComplete (Bitmap bmp)

本地截图完成的事件回调

参数	描述
bmp	截图结果，如果 bmp 为 null 代表本次截图操作失败。
data	截图数据，为 nullptr 表示截图失败。
format	截图数据格式，目前只支持 TRTCVideoPixelFormat_BGRA32。
height	截图画面的高度。
length	截图数据长度，对于BGRA32而言，length = width * height * 4。
type	视频流类型。
userId	用户标识，如果 userId 为空字符串，则代表截取的是本地画面。
width	截图画面的宽度。

注意

全平台 C++ 接口和 Java 接口在参数上是不一样的，C++ 接口用 7 个参数描述一个截图画面，Java 接口只用一个 Bitmap 描述一个截图画面。

onUserEnter

onUserEnter

void onUserEnter	(String userId)

有主播加入当前房间（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 onRemoteUserEnterRoom 替代之。

onUserExit

onUserExit

void onUserExit	(String userId int reason)

有主播离开当前房间（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 onRemoteUserLeaveRoom 替代之。

onAudioEffectFinished

onAudioEffectFinished

void onAudioEffectFinished	(int effectId int code)

音效播放已结束（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 ITXAudioEffectManager 接口替代之。
新的接口中不再区分背景音乐和音效，而是统一用 startPlayMusic 取代之。

onSpeedTest

onSpeedTest

void	(TRTCCloudDef.TRTCSpeedTestResult currentResult

```
onSpeedTest
                    int finishedCount
                    int totalCount)
```

服务器测速的结果回调（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 onSpeedTestResult 接口替代之。

TRTCStatistics

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTC 音视频统计指标（只读）

Function: TRTC SDK 会以两秒钟一次的频率向您汇报当前实时的音视频指标（帧率、码率、卡顿情况等）

TRTCStatistics

结构体类型

函数列表	描述
TRTCLocalStatistics	本地的音视频统计指标
TRTCRemoteStatistics	远端的音视频统计指标
TRTCStatistics	网络和性能的汇总统计指标

TRTCLocalStatistics

TRTCLocalStatistics

本地的音视频统计指标

枚举类型	描述
audioBitrate	【字段含义】本地音频的码率，即每秒钟新产生音频数据的多少，单位 Kbps。
audioCaptureState	【字段含义】音频设备采集状态（用于检测音频外设的健康度）。 0：采集设备状态正常；1：检测到长时间静音；2：检测到破音；3：检测到声音异常间断。

audioSampleRate	【字段含义】本地音频的采样率，单位 Hz。
frameRate	【字段含义】本地视频的帧率，即每秒钟会有多少视频帧，单位：FPS。
height	【字段含义】本地视频的高度，单位 px。
streamType	【字段含义】视频流类型（高清大画面|低清小画面|辅流画面）。
videoBitrate	【字段含义】本地视频的码率，即每秒钟新产生视频数据的多少，单位 Kbps。
width	【字段含义】本地视频的宽度，单位 px。

TRTCRemoteStatistics

TRTCRemoteStatistics

远端的音视频统计指标

枚举类型	描述
audioBitrate	【字段含义】本地音频的码率，单位 Kbps
audioBlockRate	【字段含义】音频播放卡顿率，单位 (%) 音频播放卡顿率（audioBlockRate）= 音频播放的累计卡顿时长（audioTotalBlockTime）/ 音频播放的总时长
audioPacketLoss	【字段含义】音频流的总丢包率（%）。 audioPacketLoss 代表音频流历经 `主播>云端>观众` 这样一条完整的传输链路后，最终在观众端统计到的丢包率。 audioPacketLoss 越小越好，丢包率为0即表示该路音频流的所有数据均已经完整地到达了观众端。 如果出现了 downLoss == 0 但 audioPacketLoss != 0 的情况，说明该路音频流在“云端=>观众”这一段链路上没有出现丢包，但是在 `主播>云端` 这一段链路上出现了不可恢复的丢包。
audioSampleRate	【字段含义】本地音频的采样率，单位 Hz。
audioTotalBlockTime	【字段含义】音频播放的累计卡顿时长，单位 ms
finalLoss	【字段含义】该路音视频流的总丢包率（%）。已废弃，不推荐使用；建议使用 audioPacketLoss、videoPacketLoss 替代。

frameRate	【字段含义】远端视频的帧率，单位：FPS。
height	【字段含义】远端视频的高度，单位 px。
jitterBufferDelay	【字段含义】播放延迟，单位 ms 为了避免网络抖动和网络包乱序导致的声音和画面卡顿，TRTC 会在播放端管理一个播放缓冲区，用于对接收到的网络数据包进行整理，该缓冲区的大小会根据当前的网络质量进行自适应调整，该缓冲区的大小折算成以毫秒为单位的时间长度，也就是 jitterBufferDelay。
point2PointDelay	【字段含义】端到端延迟，单位 ms point2PointDelay 代表 “主播=>云端=>观众” 的延迟，更准确地说，它代表了“采集=>编码=>网络传输=>接收=>缓冲=>解码=>播放” 全链路的延迟。 point2PointDelay 需要本地和远端的 SDK 均为 8.5 及以上的版本才生效，若远端用户为 8.5 以前的版本，此数值会一直为0，代表无意义。
remoteNetworkRTT	【字段含义】该用户从 SDK 到云端的往返延时，单位 ms 该数值代表从 SDK 发送一个网络包到云端，再从云端回送一个网络包到 SDK 的总计耗时，也就是一个网络包经历 “SDK=>云端=>SDK” 的总耗时。 该数值越小越好：如果 remoteNetworkRTT < 50ms，意味着较低的音视频通话延迟；如果 remoteNetworkRTT > 200ms，则意味着较高的音视频通话延迟。 需要特别解释的是，remoteNetworkRTT 代表 “SDK=>云端=>SDK” 的总耗时，所以不需要区分 remoteNetworkUpRTT 和 remoteNetworkDownRTT
remoteNetworkUplinkLoss	【字段含义】该用户从 SDK 到云端的上行丢包率，单位 (%) 该数值越小越好，如果 remoteNetworkUplinkLoss 为 0%，则意味着上行链路的网络质量很好，上传到云端的数据包基本不发生丢失。 如果 remoteNetworkUplinkLoss 为 30%，则意味着 SDK 向云端发送的音视频数据包中，会有 30% 丢失在传输链路中。
streamType	【字段含义】视频流类型（高清大画面|低清小画面|辅流画面）。
userId	【字段含义】用户 ID
videoBitrate	【字段含义】远端视频的码率，单位 Kbps。
videoBlockRate	【字段含义】视频播放卡顿率，单位 (%) 视频播放卡顿率（videoBlockRate）= 视频播放的累计卡顿时长（videoTotalBlockTime）/ 视频播放的总时长。
videoPacketLoss	【字段含义】该路视频流的总丢包率（%）。 videoPacketLoss 代表该路视频流历经 `主播>云端>观众` 这样一条完整的传输链路后，最终在观众端统计到的丢包率。

	videoPacketLoss 越小越好，丢包率为0即表示该路视频流的所有数据均已经完整地到达了观众端。 如果出现了 downLoss == 0 但 videoPacketLoss != 0 的情况，说明该路视频流在 `云端>观众` 这一段链路上没有出现丢包，但是在 `主播>云端` 这一段链路上出现了不可恢复的丢包。
videoTotalBlockTime	【字段含义】视频播放的累计卡顿时长，单位 ms
width	【字段含义】远端视频的宽度，单位 px。

TRTCStatistics

TRTCStatistics

网络和性能的汇总统计指标

枚举类型	描述
appCpu	【字段含义】当前应用的 CPU 使用率，单位 (%)，Android 8.0 以上不支持。
downLoss	【字段含义】从云端到 SDK 的下行丢包率，单位 (%) 该数值越小越好，如果 downLoss 为 0%，则意味着下行链路的网络质量很好，从云端接收的数据包基本不发生丢失。 如果 downLoss 为 30%，则意味着云端向 SDK 传输的音视频数据包中，会有 30% 丢失在传输链路中。
gatewayRtt	【字段含义】从 SDK 到本地路由器的往返时延，单位 ms 该数值代表从 SDK 发送一个网络包到本地路由器网关，再从网关回送一个网络包到 SDK 的总计耗时，也就是一个网络包经历【SDK>网关>SDK【 的总耗时。 该数值越小越好：如果 gatewayRtt < 50ms，意味着较低的音视频通话延迟；如果 gatewayRtt > 200ms，则意味着较高的音视频通话延迟。 当网络类型为蜂窝网时，该值无效。
localArray	【字段含义】本地的音视频统计信息 由于本地可能有三路音视频流（即高清大画面，低清小画面，以及辅流画面），因此本地的音视频统计信息是一个数组。
receiveBytes	【字段含义】总接收字节数（包含信令数据和音视频数据），单位：字节数（Bytes）。
remoteArray	【字段含义】远端的音视频统计信息 因为同时可能有多个远端用户，而且每个远端用户同时可能有多路音视频流（即高清大画面，低清小画面，以及辅流画面），因此远端的音视频统计信息是一个数

	组。
rtt	【字段含义】从 SDK 到云端的往返延时，单位 ms 该数值代表从 SDK 发送一个网络包到云端，再从云端回送一个网络包到 SDK 的总计耗时，也就是一个网络包经历 “SDK=>云端=>SDK” 的总耗时。 该数值越小越好：如果 rtt < 50ms，意味着较低的音视频通话延迟；如果 rtt > 200ms，则意味着较高的音视频通话延迟。 需要特别解释的是，rtt 代表 “SDK=>云端=>SDK” 的总耗时，所以不需要区分 upRtt 和 downRtt。
sendBytes	【字段含义】总发送字节数（包含信令数据和音视频数据），单位：字节数（Bytes）。
systemCpu	【字段含义】当前系统的 CPU 使用率，单位 (%)，Android 8.0 以上不支持。
upLoss	【字段含义】从 SDK 到云端的上行丢包率，单位 (%) 该数值越小越好，如果 upLoss 为 0%，则意味着上行链路的网络质量很好，上传到云端的数据包基本不发生丢失。 如果 upLoss 为 30%，则意味着 SDK 向云端发送的音视频数据包中，会有 30% 丢失在传输链路中。

TXAudioEffectManager

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: 背景音乐、短音效和人声特效的管理类

Function: 用于对背景音乐、短音效和人声特效进行设置的管理类

TXAudioEffectManager

TXMusicPreloadObserver

函数列表	描述
onLoadProgress	背景音乐预加载进度
onLoadError	背景音乐预加载出错

TXMusicPlayObserver

函数列表	描述
onStart	背景音乐开始播放
onPlayProgress	背景音乐的播放进度
onComplete	背景音乐已经播放完毕

TXAudioEffectManager

函数列表	描述
enableVoiceEarMonitor	开启耳返
setVoiceEarMonitor	设置耳返音量

Volume	
setVoiceReverbType	设置人声的混响效果
setVoiceChangerType	设置人声的变声特效
setVoiceCaptureVolume	设置语音音量
setVoicePitch	设置语音音调
setMusicObserver	设置背景音乐的事件回调接口
startPlayMusic	开始播放背景音乐
stopPlayMusic	停止播放背景音乐
pausePlayMusic	暂停播放背景音乐
resumePlayMusic	恢复播放背景音乐
setAllMusicVolume	设置所有背景音乐的本地音量和远端音量的大小
setMusicPublishVolume	设置某一首背景音乐的远端音量的大小
setMusicPlayoutVolume	设置某一首背景音乐的本地音量的大小
setMusicPitch	调整背景音乐的音调高低
setMusicSpeedRate	调整背景音乐的变速效果
getMusicCurrentPosInMS	获取背景音乐的播放进度（单位：毫秒）
getMusicDurationInMS	获取背景音乐的总时长（单位：毫秒）
seekMusicToPosInMS	设置背景音乐的播放进度（单位：毫秒）
setMusicScratchSpeedRate	调整搓碟的变速效果
setPreloadObserver	设置预加载事件回调

preloadMusic	预加载背景音乐
getMusicTrackCount	获取背景音乐的音轨数量
setMusicTrack	指定背景音乐的播放音轨

结构体类型

函数列表	描述
AudioMusicParam	背景音乐的播放控制信息

枚举类型

枚举类型	描述
TXVoiceReverbType	混响特效
TXVoiceChangerType	变声特效

onLoadProgress

onLoadProgress

```
void onLoadProgress (int id
                     int progress)
```

背景音乐预加载进度

onLoadError

onLoadError

```
void (int id
```

```
onLoadError
                    int errorCode)
```

背景音乐预加载出错

参数	描述
errorCode	-4001:打开文件失败，如音频文件格式不支持，本地音频文件不存在，网络音频文件无法访问等；-4002:解码失败，如音频文件损坏，网络音频文件服务器无法访问等；-4003:预加载数量超上限，请先调用 stopPlayMusic 释放无用的预加载。

onStart

onStart

```
void
onStart         (int id

                 int errCode)
```

背景音乐开始播放

在背景音乐开始播放成功或者失败时调用。

参数	描述
errCode	错误码。0: 开始播放成功；-4001: 打开文件失败，如音频文件格式不支持，本地音频文件不存在，网络音频文件无法访问等。
id	音乐 ID。

onPlayProgress

onPlayProgress

```
void
onPlayProgress  (int id

                 long curPtsMS
```

long durationMS)

背景音乐的播放进度

onComplete

onComplete

void onComplete (int id
int errCode)

背景音乐已经播放完毕

在背景音乐播放完毕或播放错误时调用。

参数	描述
errCode	错误码。0: 播放结束；-4002: 解码失败，如音频文件损坏，网络音频文件服务器无法访问等。
id	音乐 ID。

enableVoiceEarMonitor

enableVoiceEarMonitor

void enableVoiceEarMonitor (boolean enable)

开启耳返

主播开启耳返后，可以在耳机里听到麦克风采集到的自己发出的声音，该特效适用于主播唱歌的应用场景中。

需要您注意的是，由于蓝牙耳机的硬件延迟非常高，所以在主播佩戴蓝牙耳机时无法开启此特效，请尽量在用户界面上提示主播佩戴有线耳机。

同时也需要注意，并非所有的手机开启此特效后都能达到优秀的耳返效果，我们已经对部分耳返效果不佳的手机屏蔽了该特效。

参数	描述
enable	true：开启；false：关闭。

注意
仅在主播佩戴耳机时才能开启此特效，同时请您提示主播佩戴有线耳机。

setVoiceEarMonitorVolume

setVoiceEarMonitorVolume

void setVoiceEarMonitorVolume (int volume)

设置耳返音量

通过该接口您可以设置耳返特效中声音的音量大小。

参数	描述
volume	音量大小，取值范围为 0 – 100，默认值：100。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setVoiceReverbType

setVoiceReverbType

void setVoiceReverbType (TXVoiceReverbType type)

设置人声的混响效果

通过该接口您可以设置人声的混响效果，具体特效请参见枚举定义 TXVoiceReverbType 。

> **注意**
> 设置的效果在退出房间后会自动失效，如果下次进房还需要对应特效，需要调用此接口再次进行设置。

setVoiceChangerType

setVoiceChangerType

void setVoiceChangerType (TXVoiceChangerType type)

设置人声的变声特效

通过该接口您可以设置人声的变声特效，具体特效请参见枚举定义 TXVoiceChangeType 。

> **注意**
> 设置的效果在退出房间后会自动失效，如果下次进房还需要对应特效，需要调用此接口再次进行设置。

setVoiceCaptureVolume

setVoiceCaptureVolume

void setVoiceCaptureVolume (int volume)

设置语音音量

该接口可以设置语音音量的大小，一般配合音乐音量的设置接口 setAllMusicVolume 协同使用，用于调谐语音和音乐在混音前各自的音量占比。

参数	描述
volu	音量大小，取值范围为0 - 100，默认值：100。

me	

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setVoicePitch

setVoicePitch

```
void setVoicePitch (double pitch)
```

设置语音音调

该接口可以设置语音音调，用于实现变调不变速的目的。

参数	描述
pitch	音调，取值范围为-1.0f~1.0f，默认值：0.0f。

setMusicObserver

setMusicObserver

```
void setMusicObserver (int id
                       TXMusicPlayObserver observer)
```

设置背景音乐的事件回调接口

请在播放背景音乐之前使用该接口设置播放事件回调，以便感知背景音乐的播放进度。

参数	描述
musicId	音乐 ID。

observer	具体参考 ITXMusicPlayObserver 中定义接口。

注意

1. 如果该 ID 不需要使用，observer 可设置为 NULL 彻底释放。

startPlayMusic

startPlayMusic

boolean startPlayMusic (final AudioMusicParam musicParam)

开始播放背景音乐

每个音乐都需要您指定具体的 ID，您可以通过该 ID 对音乐的开始、停止、音量等进行设置。

参数	描述
musicParam	音乐参数。

注意

1. 如果要多次播放同一首背景音乐，请不要每次播放都分配一个新的 ID，我们推荐使用相同的 ID。
2. 若您希望同时播放多首不同的音乐，请为不同的音乐分配不同的 ID 进行播放。
3. 如果使用同一个 ID 播放不同音乐，SDK 会先停止播放旧的音乐，再播放新的音乐。

stopPlayMusic

stopPlayMusic

void stopPlayMusic (int id)

停止播放背景音乐

参数	描述

id	音乐 ID。

pausePlayMusic

pausePlayMusic

void pausePlayMusic (int id)

暂停播放背景音乐

参数	描述
id	音乐 ID。

resumePlayMusic

resumePlayMusic

void resumePlayMusic (int id)

恢复播放背景音乐

参数	描述
id	音乐 ID。

setAllMusicVolume

setAllMusicVolume

void setAllMusicVolume (int volume)

设置所有背景音乐的本地音量和远端音量的大小

该接口可以设置所有背景音乐的本地音量和远端音量。

- 本地音量：即主播本地可以听到的背景音乐的音量大小。
- 远端音量：即观众端可以听到的背景音乐的音量大小。

参数	描述
volume	音量大小，取值范围为0 – 100，默认值：60。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPublishVolume

setMusicPublishVolume

```
void setMusicPublishVolume(int id, int volume)
```

设置某一首背景音乐的远端音量的大小

该接口可以细粒度地控制每一首背景音乐的远端音量，也就是观众端可听到的背景音乐的音量大小。

参数	描述
id	音乐 ID。
volume	音量大小，取值范围为0 – 100；默认值：60。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPlayoutVolume

setMusicPlayoutVolume

```
void setMusicPlayoutVolume(int id, int volume)
```

设置某一首背景音乐的本地音量的大小

该接口可以细粒度地控制每一首背景音乐的本地音量，也就是主播本地可以听到的背景音乐的音量大小。

参数	描述
id	音乐 ID。
volume	音量大小，取值范围为0 – 100，默认值：60。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPitch

setMusicPitch

```
void setMusicPitch(int id, float pitch)
```

调整背景音乐的音调高低

参数	描述
id	音乐 ID。

pitch	音调，默认值是0.0f，范围是：[-1 ~ 1] 之间的浮点数。

setMusicSpeedRate

setMusicSpeedRate

```
void setMusicSpeedRate(int id, float speedRate)
```

调整背景音乐的变速效果

参数	描述
id	音乐 ID。
speedRate	速度，默认值是1.0f，范围是：[0.5 ~ 2] 之间的浮点数。

getMusicCurrentPosInMS

getMusicCurrentPosInMS

```
long getMusicCurrentPosInMS(int id)
```

获取背景音乐的播放进度（单位：毫秒）

参数	描述
id	音乐 ID。

返回值说明：

成功返回当前播放时间，单位：毫秒，失败返回 -1。

getMusicDurationInMS

getMusicDurationInMS

long getMusicDurationInMS (String path)

获取背景音乐的总时长（单位：毫秒）

参数	描述
path	音乐文件路径。

返回值说明：

成功返回时长，失败返回 -1。

seekMusicToPosInMS

seekMusicToPosInMS

void seekMusicToPosInMS (int id

int pts)

设置背景音乐的播放进度（单位：毫秒）

参数	描述
id	音乐 ID。
pts	单位: 毫秒。

注意

请尽量避免过度频繁地调用该接口，因为该接口可能会再次读写音乐文件，耗时稍高。
因此，当用户拖拽音乐的播放进度条时，请在用户完成拖拽操作后再调用本接口。
因为 UI 上的进度条控件往往会以很高的频率反馈用户的拖拽进度，如不做频率限制，会导致较差的用户体验。

setMusicScratchSpeedRate

setMusicScratchSpeedRate

```
void
setMusicScratchSpeedRate
(int id
float scratchSpeedRate)
```

调整搓碟的变速效果

参数	描述
id	音乐 ID。
scratchSpeedRate	搓碟速度，默认值是1.0f，范围是：[-12.0 ~ 12.0] 之间的浮点数，速度值正/负表示方向正/反，绝对值大小表示速度快慢。

注意
前置条件 preloadMusic 成功。

setPreloadObserver

setPreloadObserver

```
void
setPreloadObserver
(TXMusicPreloadObserver observer)
```

设置预加载事件回调

请在预加载背景音乐之前使用该接口设置回调，以便感知背景音乐的预加载进度。

参数	描述
observer	具体参考 ITXMusicPreloadObserver 中定义接口。

preloadMusic

preloadMusic

boolean preloadMusic (final AudioMusicParam preloadParam)

预加载背景音乐

每个音乐都需要您指定具体的 ID，您可以通过该 ID 对音乐的开始、停止、音量等进行设置。

参数	描述
preloadParam	预加载音乐参数。

注意

1. 预先加载最多同时支持2个不同 ID 的预加载，且预加载时长不超过10分钟，使用完需 stopPlayMusic，否则内存不释放。
2. 若该ID对应的音乐正在播放中，预加载会失败，需先调用 stopPlayMusic。
3. 当 musicParam 和传入 startPlayMusic 的 musicParam 完全相同时，预加载有效。

getMusicTrackCount

getMusicTrackCount

int getMusicTrackCount (int id)

获取背景音乐的音轨数量

参数	描述
id	音乐 ID。

setMusicTrack

setMusicTrack

void setMusicTrack	(int id int trackIndex)

指定背景音乐的播放音轨

参数	描述
id	音乐 ID。
index	默认播放第一个音轨。取值范围[0, 音轨总数)。

注意
音轨总数量可通过 getMusicTrackCount 接口获取。

TXVoiceReverbType

TXVoiceReverbType

混响特效

混响特效可以作用于人声之上，通过声学算法对声音进行叠加处理，模拟出各种不同环境下的临场感受，目前支持如下几种混响效果：
0：关闭；1：KTV；2：小房间；3：大会堂；4：低沉；5：洪亮；6：金属声；7：磁性；8：空灵；9：录音棚；10：悠扬 11：录音棚2。

枚举	取值	描述
TXLiveVoiceReverbType_0	0	关闭特效
TXLiveVoiceReverbType_1	1	KTV
TXLiveVoiceReverbType_2	2	小房间
TXLiveVoiceReverbType_3	3	大会堂

TXLiveVoiceReverbType_4	4	低沉
TXLiveVoiceReverbType_5	5	洪亮
TXLiveVoiceReverbType_6	6	金属声
TXLiveVoiceReverbType_7	7	磁性
TXLiveVoiceReverbType_8	8	空灵
TXLiveVoiceReverbType_9	9	录音棚
TXLiveVoiceReverbType_10	10	悠扬
TXLiveVoiceReverbType_11	11	录音棚2

TXVoiceChangeType

TXVoiceChangeType

变声特效

变声特效可以作用于人声之上，通过声学算法对人声进行二次处理，以获得与原始声音所不同的音色，目前支持如下几种变声特效：
0：关闭；1：熊孩子；2：萝莉；3：大叔；4：重金属；5：感冒；6：外语腔；7：困兽；8：肥宅；9：强电流；10：重机械；11：空灵。

枚举	取值	描述
TXLiveVoiceChangerType_0	0	关闭
TXLiveVoiceChangerType_1	1	熊孩子

TXLiveVoiceChangerType_2	2	萝莉
TXLiveVoiceChangerType_3	3	大叔
TXLiveVoiceChangerType_4	4	重金属
TXLiveVoiceChangerType_5	5	感冒
TXLiveVoiceChangerType_6	6	外语腔
TXLiveVoiceChangerType_7	7	困兽
TXLiveVoiceChangerType_8	8	肥宅
TXLiveVoiceChangerType_9	9	强电流
TXLiveVoiceChangerType_10	10	重机械
TXLiveVoiceChangerType_11	11	空灵

TXAudioMusicParam

TXAudioMusicParam

背景音乐的播放控制信息

该信息用于在接口 startPlayMusic 中指定背景音乐的相关信息，包括播放 ID、文件路径和循环次数等：
1. 如果要多次播放同一首背景音乐，请不要每次播放都分配一个新的 ID，我们推荐使用相同的 ID。
2. 若您希望同时播放多首不同的音乐，请为不同的音乐分配不同的 ID 进行播放。
3. 如果使用同一个 ID 播放不同音乐，SDK 会先停止播放旧的音乐，再播放新的音乐。

枚举类型	描述
endTimeM	【字段含义】音乐结束播放时间点，单位毫秒，0表示播放至文件结尾。

S	
id	【字段含义】音乐 ID。 【特殊说明】SDK 允许播放多路音乐，因此需要使用 ID 进行标记，用于控制音乐的开始、停止、音量等。
isShortFile	【字段含义】播放的是否为短音乐文件。 【推荐取值】true：需要重复播放的短音乐文件；false：正常的音乐文件。默认值：false。
loopCount	【字段含义】音乐循环播放的次数。 【推荐取值】取值范围为0 - 任意正整数，默认值：0。0 表示播放音乐一次；1 表示播放音乐两次；以此类推。
path	【字段含义】音效文件的完整路径或 URL 地址。支持的音频格式包括 MP3、AAC、M4A、WAV。
publish	【字段含义】是否将音乐传到远端。 【推荐取值】true：音乐在本地播放的同时，远端用户也能听到该音乐；false：主播只能在本地听到该音乐，远端观众听不到。默认值：false。
startTimeMS	【字段含义】音乐开始播放时间点，单位：毫秒。

TXBeautyManager

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: 美颜与图像处理参数设置类

Function: 修改美颜、滤镜、绿幕等参数

TXBeautyManager

TXBeautyManager

函数列表	描述
setBeautyStyle	设置美颜（磨皮）算法
setBeautyLevel	设置美颜级别
setWhitenessLevel	设置美白级别
enableSharpnessEnhancement	开启清晰度增强
setRuddyLevel	设置红润级别
setFilter	设置色彩滤镜效果
setFilterStrength	设置色彩滤镜的强度
setGreenScreenFile	设置绿幕背景视频
setEyeScaleLevel	设置大眼级别
setFaceSlimLevel	设置瘦脸级别
setFaceVLevel	设置 V 脸级别
setChinLevel	设置下巴拉伸或收缩
setFaceShortLevel	设置短脸级别
setFaceNarrowLevel	设置窄脸级别

setNoseSlimLevel	设置瘦鼻级别
setEyeLightenLevel	设置亮眼级别
setToothWhitenLevel	设置牙齿美白级别
setWrinkleRemoveLevel	设置祛皱级别
setPounchRemoveLevel	设置祛眼袋级别
setSmileLinesRemoveLevel	设置法令纹去除级别
setForeheadLevel	设置发际线调整级别
setEyeDistanceLevel	设置眼距
setEyeAngleLevel	设置眼角调整级别
setMouthShapeLevel	设置嘴型调整级别
setNoseWingLevel	设置鼻翼调整级别
setNosePositionLevel	设置鼻子位置
setLipsThicknessLevel	设置嘴唇厚度
setFaceBeautyLevel	设置脸型
setMotionTmpl	选择 AI 动效挂件
setMotionMute	是否在动效素材播放时静音

枚举类型

枚举类型	描述
TXBeautyStyle	美颜（磨皮）算法

setBeautyStyle

setBeautyStyle

void setBeautyStyle	(int beautyStyle)

设置美颜（磨皮）算法

TRTC 内置多种不同的磨皮算法，您可以选择最适合您产品定位的方案：

参数	描述
beautyStyle	美颜风格，TXBeautyStyleSmooth：光滑；TXBeautyStyleNature：自然；TXBeautyStylePitu：优图。

setBeautyLevel

setBeautyLevel

void setBeautyLevel	(float beautyLevel)

设置美颜级别

参数	描述
beautyLevel	美颜级别，取值范围 0 - 9； 0 表示关闭，9 表示效果最明显。

setWhitenessLevel

setWhitenessLevel

void setWhitenessLevel	(float whitenessLevel)

设置美白级别

参数	描述
whitenessLevel	美白级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

enableSharpnessEnhancement

enableSharpnessEnhancement

void enableSharpnessEnhancement	(boolean enable)

开启清晰度增强

setRuddyLevel

setRuddyLevel

void setRuddyLevel	(float ruddyLevel)

设置红润级别

参数	描述
ruddyLevel	红润级别，取值范围0 - 9；0 表示关闭，9 表示效果最明显。

setFilter

setFilter

void setFilter	(Bitmap image)

版权所有：腾讯云计算（北京）有限责任公司

设置色彩滤镜效果

色彩滤镜，是一副包含色彩映射关系的颜色查找表图片，您可以在我们提供的官方 Demo 中找到预先准备好的几张滤镜图片。
SDK 会根据该查找表中的映射关系，对摄像头采集出的原始视频画面进行二次处理，以达到预期的滤镜效果。

参数	描述
image	包含色彩映射关系的颜色查找表图片，必须是 png 格式。

setFilterStrength

setFilterStrength

void setFilterStrength (float strength)

设置色彩滤镜的强度

该数值越高，色彩滤镜的作用强度越明显，经过滤镜处理后的视频画面跟原画面的颜色差异越大。
我默认的滤镜浓度是 0.5，如果您觉得默认的滤镜效果不明显，可以设置为 0.5 以上的数字，最大值为 1。

参数	描述
strength	从 0 到 1，数值越大滤镜效果越明显，默认值为 0.5。

setGreenScreenFile

setGreenScreenFile

int setGreenScreenFile (String path)

设置绿幕背景视频

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见 腾讯美颜特效SDK ）中生效。此接口所开启的绿幕功能不具备智能去除背景的能力，需要被拍摄者的背后有一块绿色的幕布来辅助产生

特效。

参数	描述
path	MP4格式的视频文件路径; 设置空值表示关闭特效。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeScaleLevel

setEyeScaleLevel

int setEyeScaleLevel (float eyeScaleLevel)

设置大眼级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见 腾讯美颜特效SDK ）中生效。

参数	描述
eyeScaleLevel	大眼级别，取值范围 0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceSlimLevel

setFaceSlimLevel

int setFaceSlimLevel (float faceSlimLevel)

设置瘦脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
faceSlimLevel	瘦脸级别，取值范围0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceVLevel

setFaceVLevel

int setFaceVLevel (float faceVLevel)

设置 V 脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
faceVLevel	V 脸级别，取值范围 0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setChinLevel

setChinLevel

int setChinLevel (float chinLevel)

设置下巴拉伸或收缩

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
chinLevel	下巴拉伸或收缩级别，取值范围 -9 – 9；0 表示关闭，小于 0 表示收缩，大于 0 表示拉伸。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceShortLevel

setFaceShortLevel

int setFaceShortLevel (float faceShortLevel)

设置短脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
faceShortLevel	短脸级别，取值范围 0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceNarrowLevel

setFaceNarrowLevel

int setFaceNarrowLevel (float faceNarrowLevel)

设置窄脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
level	窄脸级别，取值范围 0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setNoseSlimLevel

setNoseSlimLevel

int setNoseSlimLevel (float noseSlimLevel)

设置瘦鼻级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
noseSlimLevel	瘦鼻级别，取值范围0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeLightenLevel

setEyeLightenLevel

int setEyeLightenLevel (float eyeLightenLevel)

设置亮眼级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
eyeLightenLevel	亮眼级别，取值范围 0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setToothWhitenLevel

setToothWhitenLevel

int setToothWhitenLevel (float toothWhitenLevel)

设置牙齿美白级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
toothWhitenLevel	白牙级别，取值范围 0 – 9；0表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setWrinkleRemoveLevel

setWrinkleRemoveLevel

int setWrinkleRemoveLevel (float wrinkleRemoveLevel)

设置祛皱级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
wrinkleRemoveLevel	祛皱级别，取值范围0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setPounchRemoveLevel

setPounchRemoveLevel

int setPounchRemoveLevel (float pounchRemoveLevel)

设置祛眼袋级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
pounchRemoveLevel	祛眼袋级别，取值范围 0 – 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setSmileLinesRemoveLevel

setSmileLinesRemoveLevel

int setSmileLinesRemoveLevel (float smileLinesRemoveLevel)

设置法令纹去除级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
smileLinesRemoveLevel	法令纹级别，取值范围 0 – 9；0表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setForeheadLevel

setForeheadLevel

int setForeheadLevel (float foreheadLevel)

设置发际线调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
foreheadLevel	发际线级别，取值范围-9 – 9；0表示关闭，9表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeDistanceLevel

setEyeDistanceLevel

int setEyeDistanceLevel (float eyeDistanceLevel)

设置眼距

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
eyeDistanceLevel	眼距级别，取值范围 -9 – 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeAngleLevel

setEyeAngleLevel

int setEyeAngleLevel (float eyeAngleLevel)

设置眼角调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
eyeAngleLevel	眼角调整级别，取值范围-9 – 9；0表示关闭，9表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setMouthShapeLevel

setMouthShapeLevel

int setMouthShapeLevel (float mouthShapeLevel)

设置嘴型调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
mouthShapeLevel	嘴型级别，取值范围 -9 – 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setNoseWingLevel

setNoseWingLevel

int setNoseWingLevel (float noseWingLevel)

设置鼻翼调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
noseWingLevel	鼻翼调整级别，取值范围 -9 – 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setNosePositionLevel

setNosePositionLevel

int setNosePositionLevel (float nosePositionLevel)

设置鼻子位置

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
nosePositionLevel	鼻子位置级别，取值范围 -9 – 9；0 表示关闭，小于 0 表示抬高，大于 0 表示降低。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setLipsThicknessLevel

setLipsThicknessLevel

int setLipsThicknessLevel (float lipsThicknessLevel)

设置嘴唇厚度

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
lipsThicknessLevel	嘴唇厚度级别，取值范围 -9 – 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceBeautyLevel

setFaceBeautyLevel

int setFaceBeautyLevel (float faceBeautyLevel)

设置脸型

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
faceBeautyLevel	美型级别，取值范围 0 – 9；0 表示关闭，1 – 9 值越大，效果越明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setMotionTmpl

setMotionTmpl

void setMotionTmpl	(String tmplPath)

选择 AI 动效挂件

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。

参数	描述
tmplPath	动效素材文件所在目录。

setMotionMute

setMotionMute

void setMotionMute	(boolean motionMute)

是否在动效素材播放时静音

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见腾讯美颜特效SDK）中生效。
有些挂件本身会有声音特效，通过此 API 可以关闭这些特效播放时所带的声音效果。

参数	描述
motionMute	true：静音；false：不静音。

TXBeautyStyle

TXBeautyStyle

美颜（磨皮）算法

TRTC 内置多种不同的磨皮算法，您可以选择最适合您产品定位的方案。

枚举	取值	描述
TXBeautyStyleSmooth	0	光滑，算法比较激进，磨皮效果比较明显，适用于秀场直播。
TXBeautyStyleNature	1	自然，算法更多地保留了面部细节，磨皮效果更加自然，适用于绝大多数直播场景。
TXBeautyStylePitu	2	优图，由优图实验室提供，磨皮效果介于光滑和自然之间，比光滑保留更多皮肤细节，比自然磨皮程度更高。

TXDeviceManager

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: 音视频设备管理模块

Function: 用于管理摄像头、麦克风和扬声器等音视频相关的硬件设备

TXDeviceManager

TXDeviceManager

函数列表	描述
isFrontCamera	判断当前是否为前置摄像头（仅适用于移动端）
switchCamera	切换前置或后置摄像头（仅适用于移动端）
getCameraZoomMaxRatio	获取摄像头的最大缩放倍数（仅适用于移动端）
setCameraZoomRatio	设置摄像头的缩放倍数（仅适用于移动端）
isAutoFocusEnabled	查询是否支持自动识别人脸位置（仅适用于移动端）
enableCameraAutoFocus	开启自动对焦功能（仅适用于移动端）
setCameraFocusPosition	设置摄像头的对焦位置（仅适用于移动端）
enableCameraTorch	开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）
setAudioRoute	设置音频路由（仅适用于移动端）
setExposureCompensation	设置摄像头的曝光参数，取值范围从-1到1

setCameraCaptureParam	设置摄像头采集偏好
setSystemVolumeType	设置系统音量类型（仅适用于移动端）

结构体类型

函数列表	描述
TXCameraCaptureParam	摄像头采集参数

枚举类型

枚举类型	描述
TXSystemVolumeType	系统音量类型（仅适用于移动设备）
TXAudioRoute	音频路由（即声音的播放模式）
TXCameraCaptureMode	摄像头采集偏好

isFrontCamera

isFrontCamera

判断当前是否为前置摄像头（仅适用于移动端）

switchCamera

switchCamera

int switchCamera (boolean frontCamera)

切换前置或后置摄像头（仅适用于移动端）

getCameraZoomMaxRatio

getCameraZoomMaxRatio

获取摄像头的最大缩放倍数（仅适用于移动端）

setCameraZoomRatio

setCameraZoomRatio

int setCameraZoomRatio (float zoomRatio)

设置摄像头的缩放倍数（仅适用于移动端）

参数	描述
zoomRatio	取值范围1 – 5，取值为1表示最远视角（正常镜头），取值为5表示最近视角（放大镜头）。最大值推荐为5，若超过5，视频数据会变得模糊不清。

isAutoFocusEnabled

isAutoFocusEnabled

查询是否支持自动识别人脸位置（仅适用于移动端）

enableCameraAutoFocus

enableCameraAutoFocus

int enableCameraAutoFocus (boolean enabled)

开启自动对焦功能（仅适用于移动端）

开启后，SDK 会自动检测画面中的人脸位置，并将摄像头的焦点始终对焦在人脸位置上。

setCameraFocusPosition

setCameraFocusPosition

```
int setCameraFocusPosition(int x, int y)
```

设置摄像头的对焦位置（仅适用于移动端）

您可以通过该接口实现如下交互：
1. 在本地摄像头的预览画面上，允许用户单击操作。
2. 在用户的单击位置显示一个矩形方框，以示摄像头会在此处对焦。
3. 随后将用户点击位置的坐标通过本接口传递给 SDK，之后 SDK 会操控摄像头按照用户期望的位置进行对焦。

参数	描述
position	对焦位置，请传入期望对焦点的坐标值

> **注意**
> 使用该接口的前提是先通过 enableCameraAutoFocus 关闭自动对焦功能。

返回值说明：

0：操作成功；负数：操作失败。

enableCameraTorch

enableCameraTorch

```
boolean enableCameraTorch(boolean enable)
```

开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）

setAudioRoute

setAudioRoute

int setAudioRoute (TXAudioRoute route)

设置音频路由（仅适用于移动端）

手机有两个音频播放设备：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。
设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。

setExposureCompensation

setExposureCompensation

int setExposureCompensation (float value)

设置摄像头的曝光参数，取值范围从-1到1

setCameraCapturerParam

setCameraCapturerParam

void setCameraCapturerParam (TXCameraCaptureParam params)

设置摄像头采集偏好

setSystemVolumeType

setSystemVolumeType

int setSystemVolumeType (TXSystemVolumeType type)

设置系统音量类型（仅适用于移动端）

@deprecated v9.5 版本开始不推荐使用，建议使用 TRTCCloud 中的 startLocalAudio(quality) 接口替代之，通过 quality 参数来决策音质。

TXSystemVolumeType(Deprecated)

TXSystemVolumeType(Deprecated)

系统音量类型（仅适用于移动设备）

@deprecated v9.5 版本开始不推荐使用。

现代智能手机中一般都具备两套系统音量类型，即“通话音量”和“媒体音量”。

- 通话音量：手机专门为接打电话所设计的音量类型，自带回声抵消（AEC）功能，并且支持通过蓝牙耳机上的麦克风进行拾音，缺点是音质比较一般。

当您通过手机侧面的音量按键下调手机音量时，如果无法将其调至零（也就是无法彻底静音），说明您的手机当前处于通话音量。

- 媒体音量：手机专门为音乐场景所设计的音量类型，无法使用系统的 AEC 功能，并且不支持通过蓝牙耳机的麦克风进行拾音，但具备更好的音乐播放效果。

当您通过手机侧面的音量按键下调手机音量时，如果能够将手机音量调至彻底静音，说明您的手机当前处于媒体音量。

SDK 目前提供了三种系统音量类型的控制模式：自动切换模式、全程通话音量模式、全程媒体音量模式。

枚举	取值	描述
TXSystemVolumeTypeAuto	Not Defined	自动切换模式
TXSystemVolumeTypeMedia	Not Defined	全程媒体音量

TXSystemVolumeTypeVOIP	Not Defined	全程通话音量

TXAudioRoute

TXAudioRoute

音频路由（即声音的播放模式）

音频路由，即声音是从手机的扬声器还是从听筒中播放出来，因此该接口仅适用于手机等移动端设备。
手机有两个扬声器：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。

- 设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
- 设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。

枚举	取值	描述
TXAudioRouteSpeakerphone	Not Defined	Speakerphone：使用扬声器播放（即“免提”），扬声器位于手机底部，声音偏大，适合外放音乐。
TXAudioRouteEarpiece	Not Defined	Earpiece：使用听筒播放，听筒位于手机顶部，声音偏小，适合需要保护隐私的通话场景。

TXCameraCaptureMode

TXCameraCaptureMode

摄像头采集偏好

该枚举类型用于摄像头采集参数设置。

枚举	取值	描述
TXCameraResolutionStrategyAuto	Not Defined	自动调整采集参数。 SDK 根据实际的采集设备性能及网络情况，选择合适的摄像头输出参数，在设备性能及视频预览质量之间，维持平衡。

TXCameraResolutionStrategyPerformance	Not Defined	优先保证设备性能。 SDK 根据用户设置编码器的分辨率和帧率，选择最接近的摄像头输出参数，从而保证设备性能。
TXCameraResolutionStrategyHighQuality	Not Defined	优先保证视频预览质量。 SDK选择较高的摄像头输出参数，从而提高预览视频的质量。在这种情况下，会消耗更多的 CPU 及内存做视频前处理。
TXCameraCaptureManual	Not Defined	允许用户设置本地摄像头采集的视频宽高。

TXCameraCaptureParam

TXCameraCaptureParam

摄像头采集参数

该设置能决定本地预览图像画质。

枚举类型	描述
height	**字段含义：** 采集图像宽度
mode	**字段含义：** 摄像头采集偏好，请参见 TXCameraCaptureMode
width	**字段含义：** 采集图像长度

类型定义

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTC 关键类型定义

Function: 分辨率、质量等级等枚举和常量值的定义

类型定义

结构体类型

函数列表	描述
TRTCParams	进房参数
TRTCVideoEncParam	视频编码参数
TRTCNetworkQosParam	网络流控（Qos）参数集
TRTCRenderParams	视频画面的渲染参数
TRTCQuality	网络质量
TRTCVolumeInfo	音量大小
TRTCSpeedTestParams	测速参数
TRTCSpeedTestResult	网络测速结果
TRTCTexture	视频纹理数据
TRTCVideoFrame	视频帧信息
TRTCAudioFrame	音频帧数据
TRTCMixUser	云端混流中各路画面的描述信息
TRTCTranscodingConfig	云端混流的排版布局和转码参数
TRTCPublishCDNParam	向非腾讯云 CDN 上发布音视频流时需设置的转推参数

TRTCAudioRecordingParams	本地音频文件的录制参数
TRTCLocalRecordingParams	本地媒体文件的录制参数
TRTCAudioEffectParam	音效参数（已废弃）
TRTCSwitchRoomConfig	房间切换参数
TRTCAudioFrameCallbackFormat	音频自定义回调的格式参数
TRTCScreenShareParams	屏幕分享参数（仅适用于 Android 平台）
TRTCAudioParallelParams	远端音频流智能并发播放策略的参数
TRTCUser	媒体流发布相关配置的用户信息
TRTCPublishCdnUrl	向腾讯或者第三方 CDN 上发布音视频流时需设置的 url 配置
TRTCPublishTarget	目标推流配置
TRTCVideoLayout	转码视频布局
TRTCWatermark	水印布局
TRTCStreamEncoderParam	媒体流编码输出参数
TRTCStreamMixingConfig	媒体流转码配置参数
TRTCPayloadPrivateEncryptionConfig	媒体流私有加密配置
TRTCAudioVolumeEvaluateParams	音量评估等相关参数设置

枚举类型

枚举类型	描述
TRTCVideoResolution	视频分辨率
TRTCVideoResolutionMode	视频宽高比模式
TRTCVideoStreamType	视频流类型
TRTCVideoFillMode	视频画面填充模式

TRTCVideoRotation	视频画面旋转方向
TRTCBeautyStyle	美颜（磨皮）算法
TRTCVideoPixelFormat	视频像素格式
TRTCVideoBufferType	视频数据传递方式
TRTCVideoMirrorType	视频的镜像类型
TRTCSnapshotSourceType	本地视频截图的数据源
TRTCAppScene	应用场景
TRTCRoleType	角色
TRTCQosControlMode(Deprecated)	流控模式（已废弃）
TRTCVideoQosPreference	画质偏好
TRTCQuality	网络质量
TRTCAVStatusType	视频状态类型
TRTCAVStatusChangeReason	视频状态变化原因类型
TRTCAudioSampleRate	音频采样率
TRTCAudioQuality	声音音质
TRTCAudioRoute	音频路由（即声音的播放模式）
TRTCReverbType	声音混响模式
TRTCVoiceChangerType	变声类型
TRTCSystemVolumeType	系统音量类型（仅适用于移动设备）
TRTCAudioFrameFormat	音频帧的内容格式
TRTCAudioCapabilityTy	系统支持的音频能力类型（仅适用于Android设备）

版权所有：腾讯云计算（北京）有限责任公司

pe	
TRTCAudioFrameOperationMode	音频回调数据读写模式
TRTCLogLevel	Log 级别
TRTCGSensorMode	重力感应开关（仅适用于移动端）
TRTCTranscodingConfigMode	云端混流的排版模式
TRTCRecordType	媒体录制类型
TRTCMixInputType	混流输入类型
TRTCDebugViewLevel	用于渲染控件上展示的调试信息
TRTCAudioRecordingContent	音频录制内容类型
TRTCPublishMode	媒体流发布模式
TRTCEncryptionAlgorithm	加密算法
TRTCSpeedTestScene	测速场景
TRTCGravitySensorAdaptiveMode	设置重力感应的适配模式（仅适用于移动端）

TRTCVideoResolution

TRTCVideoResolution

视频分辨率

此处仅定义横屏分辨率（如 640 × 360），如需使用竖屏分辨率（如360 × 640），需要同时指定 TRTCVideoResolutionMode 为 Portrait。

枚举	取值	描述
TRTC_VIDEO_RESOLUTION_120_120	1	宽高比 1:1；分辨率 120x120；建议码率（VideoCall）80kbps; 建议码率（LIVE）120kbps。

TRTC_VIDEO_RESOLUTION_160_160	3	宽高比 1:1 分辨率 160x160；建议码率（VideoCall）100kbps; 建议码率（LIVE）150kbps。
TRTC_VIDEO_RESOLUTION_270_270	5	宽高比 1:1；分辨率 270x270；建议码率（VideoCall）200kbps; 建议码率（LIVE）300kbps。
TRTC_VIDEO_RESOLUTION_480_480	7	宽高比 1:1；分辨率 480x480；建议码率（VideoCall）350kbps; 建议码率（LIVE）500kbps。
TRTC_VIDEO_RESOLUTION_160_120	50	宽高比4:3；分辨率 160x120；建议码率（VideoCall）100kbps; 建议码率（LIVE）150kbps。
TRTC_VIDEO_RESOLUTION_240_180	52	宽高比 4:3；分辨率 240x180；建议码率（VideoCall）150kbps; 建议码率（LIVE）250kbps。
TRTC_VIDEO_RESOLUTION_280_210	54	宽高比 4:3；分辨率 280x210；建议码率（VideoCall）200kbps; 建议码率（LIVE）300kbps。
TRTC_VIDEO_RESOLUTION_320_240	56	宽高比 4:3；分辨率 320x240；建议码率（VideoCall）250kbps; 建议码率（LIVE）375kbps。
TRTC_VIDEO_RESOLUTION_400_300	58	宽高比 4:3；分辨率 400x300；建议码率（VideoCall）300kbps; 建议码率（LIVE）450kbps。
TRTC_VIDEO_RESOLUTION_480_360	60	宽高比 4:3；分辨率 480x360；建议码率（VideoCall）400kbps; 建议码率（LIVE）600kbps。
TRTC_VIDEO_RESOLUTION_640_480	62	宽高比 4:3；分辨率 640x480；建议码率（VideoCall）600kbps; 建议码率（LIVE）900kbps。
TRTC_VIDEO_RESOLUTION_960_720	64	宽高比 4:3；分辨率 960x720；建议码率（VideoCall）1000kbps; 建议码率（LIVE）1500kbps。
TRTC_VIDEO_RESOLUTION_160_90	100	宽高比 16:9；分辨率 160x90；建议码率（VideoCall）150kbps; 建议码率（LIVE）250kbps。

版权所有：腾讯云计算（北京）有限责任公司

TRTC_VIDEO_RESOLUTION_256_144	102	宽高比 16:9；分辨率 256x144；建议码率（VideoCall）200kbps; 建议码率（LIVE）300kbps。
TRTC_VIDEO_RESOLUTION_320_180	104	宽高比 16:9；分辨率 320x180；建议码率（VideoCall）250kbps; 建议码率（LIVE）400kbps。
TRTC_VIDEO_RESOLUTION_480_270	106	宽高比 16:9；分辨率 480x270；建议码率（VideoCall）350kbps; 建议码率（LIVE）550kbps。
TRTC_VIDEO_RESOLUTION_640_360	108	宽高比 16:9；分辨率 640x360；建议码率（VideoCall）500kbps; 建议码率（LIVE）900kbps。
TRTC_VIDEO_RESOLUTION_960_540	110	宽高比 16:9；分辨率 960x540；建议码率（VideoCall）850kbps; 建议码率（LIVE）1300kbps。
TRTC_VIDEO_RESOLUTION_1280_720	112	宽高比 16:9；分辨率 1280x720；建议码率（VideoCall）1200kbps; 建议码率（LIVE）1800kbps。
TRTC_VIDEO_RESOLUTION_1920_1080	114	宽高比 16:9；分辨率 1920x1080；建议码率（VideoCall）2000kbps; 建议码率（LIVE）3000kbps。

TRTCVideoResolutionMode

TRTCVideoResolutionMode

视频宽高比模式

TRTCVideoResolution 中仅定义了横屏分辨率（如 640 × 360），如需使用竖屏分辨率（如360 × 640），需要同时指定 TRTCVideoResolutionMode 为 Portrait。

枚举	取值	描述
TRTC_VIDEO_RESOLUTION_MODE_LANDSCAPE	0	横屏分辨率，例如：TRTCVideoResolution_640_360 + TRTCVideoResolutionModeLandscape = 640 × 360。

TRTC_VIDEO_RESOLUTION_MODE_PORTRAIT	1	竖屏分辨率，例如：TRTCVideoResolution_640_360 + TRTCVideoResolutionModePortrait = 360 × 640。

TRTCVideoStreamType

TRTCVideoStreamType

视频流类型

TRTC 内部有三种不同的视频流，分别是：

- 高清大画面：一般用来传输摄像头的视频数据。
- 低清小画面：小画面和大画面的内容相互，但是分辨率和码率都比大画面低，因此清晰度也更低。
- 辅流画面：一般用于屏幕分享，同一时间在同一个房间中只允许一个用户发布辅流视频，其他用户必须要等该用户关闭之后才能发布自己的辅流。

> **注意**
> 不支持单独开启低清小画面，小画面必须依附于大画面而存在，SDK 会自动设定低清小画面的分辨率和码率。

枚举	取值	描述
TRTC_VIDEO_STREAM_TYPE_BIG	0	高清大画面，一般用来传输摄像头的视频数据。
TRTC_VIDEO_STREAM_TYPE_SMALL	1	低清小画面：小画面和大画面的内容相互，但是分辨率和码率都比大画面低，因此清晰度也更低。
TRTC_VIDEO_STREAM_TYPE_SUB	2	辅流画面：一般用于屏幕分享，同一时间在同一个房间中只允许一个用户发布辅流视频，其他用户必须要等该用户关闭之后才能发布自己的辅流。

TRTCVideoFillMode

TRTCVideoFillMode

视频画面填充模式

如果视频显示区域的宽高比不等于视频内容的宽高比时，需要指定画面的填充模式。

枚举	取值	描述
TRTC_VIDEO_RENDER_MODE_FILL	0	填充模式：即将画面内容居中等比缩放以充满整个显示区域，超出显示区域的部分将会被裁剪掉，此模式下画面可能不完整。
TRTC_VIDEO_RENDER_MODE_FIT	1	适应模式：即按画面长边进行缩放以适应显示区域，短边部分会被填充为黑色，此模式下图像完整但可能留有黑边。

TRTCVideoRotation

TRTCVideoRotation

视频画面旋转方向

TRTC 提供了对本地和远程画面的旋转角度设置 API，下列的旋转角度都是指顺时针方向的。

枚举	取值	描述
TRTC_VIDEO_ROTATION_0	0	不旋转。
TRTC_VIDEO_ROTATION_90	1	顺时针旋转90度。
TRTC_VIDEO_ROTATION_180	2	顺时针旋转180度。
TRTC_VIDEO_ROTATION_270	3	顺时针旋转270度。

TRTCBeautyStyle

TRTCBeautyStyle

美颜（磨皮）算法

TRTC 内置多种不同的磨皮算法，您可以选择最适合您产品定位的方案。

枚举	取值	描述
TRTC_BEAUTY_STYLE_SMOOTH	0	光滑，算法比较激进，磨皮效果比较明显，适用于秀场直播。
TRTC_BEAUTY_STYLE_NATURE	1	自然，算法更多地保留了面部细节，磨皮效果更加自然，适用于绝大多数直播场景。
TRTC_BEAUTY_STYLE_PITU	2	优图，由优图实验室提供，磨皮效果介于光滑和自然之间，比光滑保留更多皮肤细节，比自然磨皮程度更高。

TRTCVideoPixelFormat

TRTCVideoPixelFormat

视频像素格式

TRTC 提供针对视频的自定义采集和自定义渲染功能：

- 在自定义采集功能中，您可以用下列枚举值描述您采集的视频像素格式。
- 在自定义渲染功能中，您可以指定您期望 SDK 回调出的视频像素格式。

枚举	取值	描述
TRTC_VIDEO_PIXEL_FORMAT_UNKNOWN	0	未定义的格式。
TRTC_VIDEO_PIXEL_FORMAT_I420	1	YUV420P(I420) 格式。
TRTC_VIDEO_PIXEL_FORMAT_Texture_2D	2	OpenGL 2D 纹理格式。
TRTC_VIDEO_PIXEL_FORMAT_TEXTURE_EXTERNAL_OES	3	OES 外部纹理格式（适用于 Android 平台）。
TRTC_VIDEO_PIXEL_FORMAT_NV21	4	NV21 格式。
TRTC_VIDEO_PIXEL_FORMAT_RGBA	5	RGBA 格式。

TRTCVideoBufferType

TRTCVideoBufferType

视频数据传递方式

在自定义采集和自定义渲染功能，您需要用到下列枚举值来指定您希望以什么方式传递视频数据：

- 方案一：使用内存 Buffer 传递视频数据，该方案在 iOS 效率尚可，但在 Android 系统上效率较差，Windows 暂时仅支持内存 Buffer 的传递方式。
- 方案二：使用 Texture 纹理传递视频数据，该方案在 iOS 和 Android 系统下均有较高的效率，Windows 暂不支持，需要您有一定的 OpenGL 编程基础。

枚举	取值	描述
TRTC_VIDEO_BUFFER_TYPE_UNKNOWN	0	未定义的传递方式。
TRTC_VIDEO_BUFFER_TYPE_BYTE_BUFFER	1	使用内存 Buffer 传递视频数据，iOS：PixelBuffer；Android：用于 JNI 层的 DirectBuffer；Win：内存数据块。
TRTC_VIDEO_BUFFER_TYPE_BYTE_ARRAY	2	使用内存 Buffer 传递视频数据，iOS：经过一次额外整理后更加紧凑的 NSData 类型的内存块；Android：用于 JAVA 层的 byte[]。 该传递的方式的性能是几种方案中效率较差的一种。
TRTC_VIDEO_BUFFER_TYPE_TEXTURE	3	使用 Texture 纹理传递视频数据。

TRTCVideoMirrorType

TRTCVideoMirrorType

视频的镜像类型

视频的镜像是指对视频内容进行左右翻转，尤其是对本地的摄像头预览视频，开启镜像后能给主播带来熟悉的“照镜子”体验。

枚举	取值	描述

TRTC_VIDEO_MIRROR_TYPE_AUTO	0	自动模式：如果正使用前置摄像头则开启镜像，如果是后置摄像头则不开启镜像（仅适用于移动设备）。
TRTC_VIDEO_MIRROR_TYPE_ENABLE	1	强制开启镜像，不论当前使用的是前置摄像头还是后置摄像头。
TRTC_VIDEO_MIRROR_TYPE_DISABLE	2	强制关闭镜像，不论当前使用的是前置摄像头还是后置摄像头。

TRTCSnapshotSourceType

TRTCSnapshotSourceType

本地视频截图的数据源

SDK 支持从如下两种数据源中截取图片并保存成本地文件：

- 视频流：从视频流中截取原生的视频内容，截取的内容不受渲染控件的显示控制。
- 渲染层：从渲染控件中截取显示的视频内容，可以做到用户所见即所得的效果，但如果显示区域过小，截取出的图片也会很小。

枚举	取值	描述
TRTC_SNAPSHOT_SOURCE_TYPE_STREAM	0	从视频流中截取原生的视频内容，截取的内容不受渲染控件的显示控制。
TRTC_SNAPSHOT_SOURCE_TYPE_VIEW	1	从渲染控件中截取显示的视频内容，可以做到用户所见即所得的效果，但如果显示区域过小，截取出的图片也会很小。

TRTCAppScene

TRTCAppScene

应用场景

TRTC 针对常见的音视频应用场景都进行了定向优化，以满足各种垂直场景下的差异化要求，主要场景可以分为如下两类：

- 直播（LIVE）场景：包括 LIVE 和 VoiceChatRoom，前者是音频+视频，后者是纯音频。

直播场景下，用户被分成“主播”和“观众”两种角色，单个房间中同时最多支持10万人在线，适合于观众人数众多的直播场景。

- 实时（RTC）场景：包括 VideoCall 和 AudioCall，前者是音频+视频，后者是纯音频。

实时场景下，用户没有角色的差异，但单个房间中同时最多支持 300 人在线，适合于小范围实时通信的场景。

枚举	取值	描述
TRTC_APP_SCENE_VIDEOCALL	0	视频通话场景，支持720P、1080P高清画质，单个房间最多支持300人同时在线，最高支持50人同时发言。 适用于[1对1视频通话]、[300人视频会议]、[在线问诊]、[教育小班课]、[远程面试]等业务场景。
TRTC_APP_SCENE_LIVE	1	视频互动直播，支持平滑上下麦，切换过程无需等待，主播延时小于300ms；支持十万级别观众同时播放，播放延时低至1000ms。 适用于[低延时互动直播]、[大班课]、[主播PK]、[视频相亲]、[在线互动课堂]、[远程培训]、[超大型会议]等业务场景。 **注意** 此场景下，您必须通过 TRTCParams 中的 role 字段指定当前用户的角色。
TRTC_APP_SCENE_AUDIOCALL	2	语音通话场景，默认采用 SPEECH 音质，单个房间最多支持300人同时在线，最高支持50人同时发言。 适用于 [1对1语音通话]、[300人语音会议]、[语音聊天]、[语音会议]、[在线狼人杀] 等业务场景。
TRTC_APP_SCENE_VOICE_CHATROOM	3	语音互动直播，支持平滑上下麦，切换过程无需等待，主播延时小于 300ms；支持十万级别观众同时播放，播放延时低至 1000ms。 适用于 [语音俱乐部]、[在线K歌房]、[音乐直播间]、[FM电台] 等业务场景。 **注意** 此场景下，您必须通过 TRTCParams 中的 role 字段指定当前用户的角色。

TRTCRoleType

TRTCRoleType

角色

仅适用于直播类场景（即 TRTC_APP_SCENE_LIVE 和 TRTC_APP_SCENE_VOICE_CHATROOM），把用户区分成两种不同的身份：

- 主播：可以随时发布自己的音视频流，但人数有限制，同一个房间中最多只允许 50 个主播同时发布自己的音视频流。
- 观众：只能观看其他用户的音视频流，要发布音视频流，需要先通过 switchRole 切换成主播，同一个房间中最多能容纳10万观众。

枚举	取值	描述
TRTCRoleAnchor	20	主播：可以随时发布自己的音视频流，但人数有限制，同一个房间中最多只允许 50 个主播同时发布自己的音视频流。
TRTCRoleAudience	21	观众：只能观看其他用户的音视频流，要发布音视频流，需要先通过 switchRole 切换成主播，同一个房间中最多能容纳10万观众。

TRTCQosControlMode(Deprecated)

TRTCQosControlMode(Deprecated)

流控模式（已废弃）

枚举	取值	描述
VIDEO_QOS_CONTROL_CLIENT	0	本地控制，用于 SDK 开发内部调试，客户请勿使用。
VIDEO_QOS_CONTROL_SERVER	1	云端控制，默认模式，推荐选择。

TRTCVideoQosPreference

TRTCVideoQosPreference

画质偏好

TRTC 在弱网络环境下有两种调控模式：“优先保证画面清晰”或“优先保证画面流畅”，两种模式均会优先保障声音数据的传输。

枚举	取值	描述
TRTC_VIDEO_QOS_PREFERENCE_SMOOTH	1	流畅优先：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的流畅性，代价就是画面会比较模糊且伴随有较多的马赛克。
TRTC_VIDEO_QOS_PREFERENCE_CLEAR	2	清晰优先（默认值）：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的清晰度，代价就是画面会比较卡顿。

TRTCQuality

TRTCQuality

网络质量

TRTC 会每隔两秒对当前的网络质量进行评估，评估结果为六个等级：Excellent 表示最好，Down 表示最差。

枚举	取值	描述
TRTC_QUALITY_UNKNOWN	0	未定义。
TRTC_QUALITY_Excellent	1	当前网络非常好。
TRTC_QUALITY_Good	2	当前网络比较好。
TRTC_QUALITY_Poor	3	当前网络一般。
TRTC_QUALITY_Bad	4	当前网络较差。
TRTC_QUALITY_Vb	5	当前网络很差。

ad		
TRTC_QUALITY_Down	6	当前网络不满足 TRTC 的最低要求。

TRTCAVStatusType

TRTCAVStatusType

视频状态类型

该枚举类型用于视频状态变化回调接口 onRemoteVideoStatusUpdated ，用于指定当前的视频状态。

枚举	取值	描述
TRTCAVStatusStopped	0	停止播放。
TRTCAVStatusPlaying	1	正在播放。
TRTCAVStatusLoading	2	正在加载。

TRTCAVStatusChangeReason

TRTCAVStatusChangeReason

视频状态变化原因类型

该枚举类型用于视频状态变化回调接口 onRemoteVideoStatusUpdated ，用于指定当前的视频状态原因。

枚举	取值	描述
TRTCAVStatusChangeReasonInternal	0	缺省值。
TRTCAVStatusChangeReasonBufferingBegin	1	网络缓冲。

TRTCAVStatusChangeReasonBufferingEnd	2	结束缓冲。
TRTCAVStatusChangeReasonLocalStarted	3	本地启动视频流播放。
TRTCAVStatusChangeReasonLocalStopped	4	本地停止视频流播放。
TRTCAVStatusChangeReasonRemoteStarted	5	远端视频流开始（或继续）。
TRTCAVStatusChangeReasonRemoteStopped	6	远端视频流停止（或中断）。

TRTCAudioSampleRate

TRTCAudioSampleRate

音频采样率

音频采样率用来衡量声音的保真程度，采样率越高保真程度越好，如果您的应用场景有音乐的存在，推荐使用 TRTCAudioSampleRate48000。

枚举	取值	描述
TRTCAudioSampleRate16000	16000	16k 采样率。
TRTCAudioSampleRate32000	32000	32k 采样率。
TRTCAudioSampleRate44100	44100	44.1k 采样率。
TRTCAudioSampleRate48000	48000	48k采样率

TRTCAudioQuality

TRTCAudioQuality

声音音质

TRTC 提供了三种精心校调好的模式，用来满足各种垂直场景下对音质的差异化追求：

- 人声模式（Speech）：适用于以人声沟通为主的应用场景，该模式下音频传输的抗性较强，TRTC 会通过各种人声处理技术保障在弱网络环境下的流畅度最佳。
- 音乐模式（Music）：适用于对声乐要求很苛刻的场景，该模式下音频传输的数据量很大，TRTC 会通过各项技术确保音乐信号在各频段均能获得高保真的细节还原度。
- 默认模式（Default）：介于 Speech 和 Music 之间的档位，对音乐的还原度比人声模式要好，但传输数据量比音乐模式要低很多，对各种场景均有不错的适应性。

枚举	取值	描述
TRTC_AUDIO_QUALITY_SPEECH	1	人声模式：采样率：16k；单声道；编码码率：16kbps；具备几个模式中最强的网络抗性，适合语音通话为主的场景，比如在线会议，语音通话等。
TRTC_AUDIO_QUALITY_DEFAULT	2	默认模式：采样率：48k；单声道；编码码率：50kbps；介于 Speech 和 Music 之间的档位，SDK 默认档位，推荐选择。
TRTC_AUDIO_QUALITY_MUSIC	3	音乐模式：采样率：48k；全频带立体声；编码码率：128kbps；适合需要高保真传输音乐的场景，比如在线K歌、音乐直播等。

TRTCAudioRoute

TRTCAudioRoute

音频路由（即声音的播放模式）

音频路由，即声音是从手机的扬声器还是从听筒中播放出来，因此该接口仅适用于手机等移动端设备。
手机有两个扬声器：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。

- 设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
- 设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。
- 音频路由为有线耳机。
- 音频路由为蓝牙耳机。
- 音频路由为USB专业声卡设备。

枚举	取值	描述

TRTC_AUDIO_ROUTE_SPEAKER	0	Speakerphone：使用扬声器播放（即“免提”），扬声器位于手机底部，声音偏大，适合外放音乐。
TRTC_AUDIO_ROUTE_EARPIECE	1	Earpiece：使用听筒播放，听筒位于手机顶部，声音偏小，适合需要保护隐私的通话场景。
TRTC_AUDIO_ROUTE_WIRED_HEADSET	2	WiredHeadset：使用有线耳机播放。
TRTC_AUDIO_ROUTE_BLUETOOTH_HEADSET	3	BluetoothHeadset：使用蓝牙耳机播放。
TRTC_AUDIO_ROUTE_SOUND_CARD	4	SoundCard：使用 USB 声卡播放。

TRTCReverbType

TRTCReverbType

声音混响模式

该枚举值应用于设定直播场景中的混响模式，常用于秀场直播中。

枚举	取值	描述
TRTC_REVERB_TYPE_0	0	关闭混响。
TRTC_REVERB_TYPE_1	1	KTV。
TRTC_REVERB_TYPE_2	2	小房间。
TRTC_REVERB_TYPE_3	3	大会堂。
TRTC_REVERB_TYPE_4	4	低沉。
TRTC_REVERB_TYPE_5	5	洪亮。

TRTC_REVERB_TYPE_6	6	金属声。
TRTC_REVERB_TYPE_7	7	磁性。

TRTCVoiceChangerType

TRTCVoiceChangerType

变声类型

该枚举值应用于设定直播场景中的变声模式，常用于秀场直播中。

枚举	取值	描述
TRTC_VOICE_CHANGER_TYPE_0	0	关闭变声。
TRTC_VOICE_CHANGER_TYPE_1	1	熊孩子。
TRTC_VOICE_CHANGER_TYPE_2	2	萝莉。
TRTC_VOICE_CHANGER_TYPE_3	3	大叔。
TRTC_VOICE_CHANGER_TYPE_4	4	重金属。
TRTC_VOICE_CHANGER_TYPE_5	5	感冒。
TRTC_VOICE_CHANGER_TYPE_6	6	外国人。
TRTC_VOICE_CHANGER_TYPE_7	7	困兽。
TRTC_VOICE_CHANGER_TYPE_8	8	死肥仔。
TRTC_VOICE_CHANGE	9	强电流。

R_TYPE_9		
TRTC_VOICE_CHANGER_TYPE_10	10	重机械。
TRTC_VOICE_CHANGER_TYPE_11	11	空灵。

TRTCSystemVolumeType

TRTCSystemVolumeType

系统音量类型（仅适用于移动设备）

现代智能手机中一般都具备两套系统音量类型，即“通话音量”和“媒体音量”。

- 通话音量：手机专门为接打电话所设计的音量类型，自带回声抵消（AEC）功能，并且支持通过蓝牙耳机上的麦克风进行拾音，缺点是音质比较一般。当您通过手机侧面的音量按键下调手机音量时，如果无法将其调至零（也就是无法彻底静音），说明您的手机当前处于通话音量。
- 媒体音量：手机专门为音乐场景所设计的音量类型，无法使用系统的 AEC 功能，并且不支持通过蓝牙耳机的麦克风进行拾音，但具备更好的音乐播放效果。当您通过手机侧面的音量按键下调手机音量时，如果能够将手机音量调至彻底静音，说明您的手机当前处于媒体音量。

SDK 目前提供了三种系统音量类型的控制模式：自动切换模式、全程通话音量模式、全程媒体音量模式。

枚举	取值	描述
TRTCSystemVolumeTypeAuto	0	自动切换模式： 也被称为“麦上通话，麦下媒体”，即主播上麦时使用通话音量，观众不上麦则使用媒体音量，适合在线直播场景。 如果您在 enterRoom 时选择的场景为 TRTC_APP_SCENE_LIVE 或 TRTC_APP_SCENE_VOICE_CHATROOM，SDK 会自动使用该模式。
TRTCSystemVolumeTypeMedia	1	全程媒体音量： 通话全程使用媒体音量，并不是非常常用的音量类型，适用于对音质要求比较苛刻的音乐场景中。 如果您的用户大都使用外接设备（比如外接声卡）为主，可以使用该模式，否则请慎用。
TRTCSystemVolumeTypeVOIP	2	全程通话音量： 该方案的优势在于用户在上下麦时音频模块无需切换工作模式，可以做到无缝上下麦，适合于用户需要频繁上下麦的应

		用场景。 如果您在 enterRoom 时选择的场景为 TRTCAppSceneVideoCall 或 TRTCAppSceneAudioCall，SDK 会自动使用该模式。

TRTCAudioFrameFormat

TRTCAudioFrameFormat

音频帧的内容格式

枚举	取值	描述
TRTC_AUDIO_FRAME_FORMAT_PCM	1	PCM 格式的音频数据。

TRTCAudioCapabilityType

TRTCAudioCapabilityType

系统支持的音频能力类型（仅适用于Android设备）

SDK 目前提供了两种系统音频能力类型是否支持的查询：低延时合唱能力、低延时耳返能力。

枚举	取值	描述
TRTCAudioCapabilityLowLatencyChorus	1	低延时合唱能力。
TRTCAudioCapabilityLowLatencyEarMonitor	2	低延时耳返能力。

TRTCAudioFrameOperationMode

TRTCAudioFrameOperationMode

音频回调数据读写模式

TRTC 提供了两种音频回调数据的操作模式。

- 读写模式（ReadWrite）：可以获取并修改回调的音频数据，默认模式。
- 只读模式（ReadOnly）：仅从回调中获取音频数据。

枚举	取值	描述
TRTC_AUDIO_FRAME_OPERATION_MODE_READWRITE	0	读写模式：可以获取并修改回调的音频数据。
TRTC_AUDIO_FRAME_OPERATION_MODE_READONLY	1	只读模式：仅从回调中获取音频数据。

TRTCLogLevel

TRTCLogLevel

Log 级别

不同的日志等级定义了不同的详实程度和日志数量，推荐一般情况下将日志等级设置为：TRTCLogLevelInfo。

枚举	取值	描述
TRTC_LOG_LEVEL_VERBOSE	0	输出所有级别的 Log。
TRTC_LOG_LEVEL_DEBUG	1	输出 DEBUG，INFO，WARNING，ERROR 和 FATAL 级别的 Log。
TRTC_LOG_LEVEL_INFO	2	输出 INFO，WARNING，ERROR 和 FATAL 级别的 Log。
TRTC_LOG_LEVEL_WARN	3	输出WARNING，ERROR 和 FATAL 级别的 Log。
TRTC_LOG_LEVEL_ERROR	4	输出ERROR 和 FATAL 级别的 Log。
TRTC_LOG_LEVEL_FATAL	5	仅输出 FATAL 级别的 Log。

TRTC_LOG_LEVEL_NULL	6	不输出任何 SDK Log。

TRTCGSensorMode

TRTCGSensorMode

重力感应开关（仅适用于移动端）

枚举	取值	描述
TRTC_GSENSOR_MODE_DISABLE	0	不适配重力感应，该模式是桌面平台上的默认值，该模式下，当前用户发布出去的视频画面不受重力感应方向变化的影响。
TRTC_GSENSOR_MODE_UIAUTOLAYOUT	1	适配重力感应，该模式是移动平台上的默认值，该模式下，当前用户发布出去的视频画面会跟随设备的重力感应方向进行相应的调整，同时本地预览画面保持方向不变。 SDK 目前支持的一种适配模式是：当手机或 Pad 上下颠倒时，为了保证远端用户看到的画面方向正常，SDK 会自动将发布出去的画面上下旋转180度。如果您的 APP 的界面层开启了重力感应自适应，推荐使用 UIFixLayout 模式。
TRTC_GSENSOR_MODE_UIFIXLAYOUT	2	适配重力感应 该模式下，当前用户发布出去的视频画面会跟随设备的重力感应方向进行相应的调整，同时本地预览画面也会进行相应的旋转适配。 目前支持的一种特性是：当手机或 Pad 上下颠倒时，为了保证远端用户看到的画面方向正常，SDK 会自动将发布出去的画面上下旋转180度。 如果您的 APP 的界面层不支持重力感应自适应，并且希望 SDK 的视频画面能够适配重力感应方向，推荐使用 UIFixLayout 模式。 @deprecated 从v11.5版本开始，不再支持 TRTCGSensorMode_UIFixLayout，只支持上面两种模式。

TRTCTranscodingConfigMode

TRTCTranscodingConfigMode

云端混流的排版模式

TRTC 的云端混流服务能够将房间中的多路音视频流混合成一路，因此您需要指定画面的排版方案，我们提供了如下几种排版模式。

枚举	取值	描述
TRTC_TranscodingConfigMode_Unknown	0	未定义。
TRTC_TranscodingConfigMode_Manual	1	全手动排版模式。 该模式下，您需要指定每一路画面的精确排版位置。该模式的自由度最高，但易用性也最差： • 您需要填写 TRTCTranscodingConfig 中的所有参数，包括每一路画面（TRTCMixUser）的位置坐标。 • 您需要监听 TRTCCloudListener 中的 onUserVideoAvailable() 和 onUserAudioAvailable() 事件回调，并根据当前房间中各个麦上用户的音视频状态不断地调整 mixUsers 参数。
TRTC_TranscodingConfigMode_Template_PureAudio	2	纯音频模式。 该模式适用于语音通话（AudioCall）和语音聊天室（VoiceChatRoom）等纯音频的应用场景。 • 您只需要在进入房间后，通过 setMixTranscodingConfig 接口设置一次，之后 SDK 就会自动把房间内所有上麦用户的声音混流到当前用户的直播流上。 • 您无需设置 TRTCTranscodingConfig 中的 mixUsers 参数，只需设置 audioSampleRate、

		audioBitrate 和 audioChannels 等参数即可。
TRTC_TranscodingConfigMode_Template_PresetLayout	3	预排版模式。 最受欢迎的排版模式，因为该模式支持您通过占位符提前对各路画面的位置进行设定，之后 SDK 会自动根据房间中画面的路数动态进行适配调整。 此模式下，您依然需要设置 mixUsers 参数，但可以将 userId 设置为“占位符”，可选的占位符有： • "$PLACE_HOLDER_REMOTE$" ：指代远程用户的画面，可以设置多个。 • "$PLACE_HOLDER_LOCAL_MAIN$"： 指代本地摄像头画面，只允许设置一个。 • "$PLACE_HOLDER_LOCAL_SUB$" ：指代本地屏幕分享画面，只允许设置一个。 此模式下，您不需要监听 TRTCCloudListener 中的 onUserVideoAvailable 和 onUserAudioAvailable 回调进行实时调整，只需要在进房成功后调用一次 setMixTranscodingConfig 即可，之后 SDK 会自动将真实的 userId 补位到您设置的占位符上。
TRTC_TranscodingConfigMode_Template_ScreenSharing	4	屏幕分享模式。 适用于在线教育场景等以屏幕分享为主的应用场景，仅支持 Windows 和 Mac 两个平台的 SDK。 该模式下，SDK 会先根据您通过 videoWidth 和 videoHeight 参

		数设置的目标分辨率构建一张画布， • 当老师未开启屏幕分享时，SDK 会将老师的摄像头画面等比例拉伸绘制到该画布上； • 当老师开启屏幕分享之后，SDK 会将屏幕分享画面绘制到同样的画布上。 此种排版模式的目的是为了确保混流模块的输出分辨率一致，避免课程回放和网页观看的花屏问题（网页播放器不支持可变分辨率）。 同时，连麦学生的声音也会被默认混合到老师的音视频流中。 由于教学模式下的视频内容以屏幕分享为主，因此同时传输摄像头画面和屏幕分享画面是非常浪费带宽的。 推荐的做法是直接将摄像头画面通过 setLocalVideoRenderCallback 接口自定义绘制到当前屏幕上。 在该模式下，您无需设置 TRTCTranscodingConfig 中的 mixUsers 参数，SDK 不会混合学生的画面，以免干扰屏幕分享的效果。 您可以将 TRTCTranscodingConfig 中的 width x height 设为 0px × 0px，SDK 会自动根据用户当前屏幕的宽高比计算出一个合适的分辨率： • 如果老师当前屏幕宽度 <= 1920px，SDK 会使用老师当前屏幕的实际分辨率。 • 如果老师当前屏幕宽度 > 1920px，SDK 会根据当前屏幕宽高比，选择 1920x1080(16:9)、1920x1200(16:10)、1920x1440(4:3) 三种分辨率中的一种。

TRTCRecordType

TRTCRecordType

媒体录制类型

该枚举类型用于本地媒体录制接口 startLocalRecording，用于指定是录制音视频文件还是纯音频文件。

枚举	取值	描述
TRTC_RECORD_TYPE_AUDIO	0	仅录制音频。
TRTC_RECORD_TYPE_VIDEO	1	仅录制视频。
TRTC_RECORD_TYPE_BOTH	2	同时录制音频和视频。

TRTCMixInputType

TRTCMixInputType

混流输入类型

枚举	取值	描述
TRTC_MixInputType_Undefined	0	默认值，考虑到针对老版本的兼容性，如果您指定了 inputType 为 Undefined，SDK 会根据另一个参数 pureAudio 的数值决定混流输入类型。
TRTC_MixInputType_AudioVideo	1	混入音频和视频。
TRTC_MixInputType_PureVideo	2	只混入视频。
TRTC_MixInputType_PureAudio	3	只混入音频。
TRTC_MixInputType_Wat	4	混入水印，此时您无需指定 userId 字段，但需要指定

ermark		image 字段，推荐使用 png 格式的图片。

TRTCDebugViewLevel

TRTCDebugViewLevel

用于渲染控件上展示的调试信息

枚举	取值	描述
TRTC_DEBUG_VIEW_LEVEL_GONE	0	不在渲染控件上展示调试信息。
TRTC_DEBUG_VIEW_LEVEL_STATUS	1	在渲染控件上展示音视频统计信息。
TRTC_DEBUG_VIEW_LEVEL_ALL	2	在渲染控件上展示音视频统计信息和关键历史事件。

TRTCAudioRecordingContent

TRTCAudioRecordingContent

音频录制内容类型

该枚举类型用于音频录制接口 startAudioRecording，用于指定录制音频的内容。

枚举	取值	描述
TRTC_AudioRecordingContent_All	0	录制本地和远端所有音频。
TRTC_AudioRecordingContent_Local	1	仅录制本地音频。
TRTC_AudioRecordingContent_Remote	2	仅录制远端音频。

TRTCPublishMode

TRTCPublishMode

媒体流发布模式

该枚举类型用于媒体流发布接口 startPublishMediaStream TRTC 的媒体流发布服务能够将房间中的多路音视频流混合成一路发布至 CDN 或者回推到房间内，也可以将您当前的这路音视频发布到腾讯或者第三方 CDN 因此您需要指定对应媒体流的发布模式，我们提供了如下几种模式。

枚举	取值	描述
TRTC_PublishMode_Unknown	0	未定义。
TRTC_PublishBigStream_ToCdn	1	您可以通过设置该参数将您房间内的主路流（TRTC_VIDEO_STREAM_TYPE_BIG）发布到腾讯或者第三方直播 CDN 服务商（仅支持标准 RTMP 协议）。
TRTC_PublishSubStream_ToCdn	2	您可以通过设置该参数将您房间内的辅路流（TRTC_VIDEO_STREAM_TYPE_SUB）发布到腾讯或者第三方直播 CDN 服务商（仅支持标准 RTMP 协议）。
TRTC_PublishMixStream_ToCdn	3	您可以通过设置该参数，配合编码输出参数（TRTCStreamEncoderParam）和混流转码参数（TRTCStreamMixingConfig），将您指定的多路音视频流进行转码并发布到腾讯或者第三方直播 CDN 服务商（仅支持标准 RTMP 协议）。
TRTC_PublishMixStream_ToRoom	4	您可以通过设置该参数，配合媒体流编码输出参数（TRTCStreamEncoderParam）和混流转码参数（TRTCStreamMixingConfig），将您指定的多路音视频流进行转码并发布到您指定的房间中。 • 通过 TRTCPublishTarget 中的 TRTCUser 进行指定回推房间的机器人信息。

TRTCEncryptionAlgorithm

TRTCEncryptionAlgorithm

加密算法

该枚举类型用于媒体流私有加密算法选择。

枚举	取值	描述
TRTC_EncryptionAlgorithm_Aes_128_Gcm	0	AES GCM 128。
TRTC_EncryptionAlgorithm_Aes_256_Gcm	1	AES GCM 256。

TRTCSpeedTestScene

TRTCSpeedTestScene

测速场景

该枚举类型用于测速场景选择。

枚举	取值	描述
TRTC_SpeedTestScene_Delay_Testing	1	延迟测试。
TRTC_SpeedTestScene_Delay_Bandwidth_Testing	2	延迟与带宽测试。
TRTC_SpeedTestScene_Online_Chorus_Testing	3	在线合唱测试。

TRTCGravitySensorAdaptiveMode

TRTCGravitySensorAdaptiveMode

设置重力感应的适配模式（仅适用于移动端）

枚举	取值	描述
TRTC_GRAVITY_SENSOR_ADAPTIVE_MODE_	0	关闭重力感应，根据当前采

DISABLE		集分辨率与设置的编码分辨率决策，如果两者不一致，则通过旋转90度，保证最大画幅。
TRTC_GRAVITY_SENSOR_ADAPTIVE_MODE_FILL_BY_CENTER_CROP	1	开启重力感应，始终保证远端画面图像为正，中间过程需要处理分辨率不一致时，采用居中裁剪模式。
TRTC_GRAVITY_SENSOR_ADAPTIVE_MODE_FIT_WITH_BLACK_BORDER	2	开启重力感应，始终保证远端画面图像为正，中间过程需要处理分辨率不一致时，采用叠加黑边模式。

TRTCParams

TRTCParams

进房参数

作为 TRTC SDK 的进房参数，只有该参数填写正确，才能顺利进入 roomId 或者 strRoomId 所指定的音视频房间。
由于历史原因，TRTC 支持数字和字符串两种类型的房间号，分别是 roomId 和 strRoomId。
请注意：不要混用 roomId 和 strRoomId，因为它们之间是不互通的，比如数字 123 和字符串 “123” 在 TRTC 看来是两个完全不同的房间。

枚举类型	描述
businessInfo	【字段含义】业务数据字段（选填），部分高级特性才需要用到此字段。 【推荐取值】请不要自行设置该字段。
privateMapKey	【字段含义】用于权限控制的权限票据（选填），当您希望某个房间只能让特定的 userId 进入时，需要使用 privateMapKey 进行权限保护。 【推荐取值】仅建议有高级别安全需求的客户使用，更多详情请参见 进房权限保护 。
role	【字段含义】直播场景下的角色，仅适用于直播场景（TRTC_APP_SCENE_LIVE 和 TRTC_APP_SCENE_VOICE_CHATROOM），通话场景下指定该参数是无效的。 【推荐取值】默认值：主播（TRTCRoleAnchor）。

roomId	【字段含义】数字房间号，在同一个房间里的用户（userId）可以彼此看到对方并进行音视频通话。 【推荐取值】取值范围：1 - 4294967294。 【特别说明】roomId 与 strRoomId 是互斥的，若您选用 strRoomId，则 roomId 需要填写为 0。若两者都填，SDK 将优先选用 roomId。 【请您注意】不要混用 roomId 和 strRoomId，因为它们之间是不互通的，比如数字 123 和字符串 `123` 在 TRTC 看来是两个完全不同的房间。
sdkAppId	【字段含义】应用标识（必填），腾讯云基于 sdkAppId 完成计费统计。 【推荐取值】在 实时音视频控制台 创建应用后可以在账号信息页面中得到该 ID。
strRoomId	【字段含义】字符串房间号，在同一个房间里的用户（userId）可以彼此看到对方并进行音视频通话。 【特别说明】roomId 与 strRoomId 是互斥的，若您选用 strRoomId，则 roomId 需要填写为0。若两者都填，SDK 将优先选用 roomId。 【请您注意】不要混用 roomId 和 strRoomId，因为它们之间是不互通的，比如数字 123 和字符串 `123` 在 TRTC 看来是两个完全不同的房间。 【推荐取值】限制长度为 64 字节。以下为支持的字符集范围（共 89 个字符）： • 大小写英文字母（a-zA-Z）； • 数字（0-9）； • 空格、`!`、`#`、`$`、`%`、`&`、`(`、`)`、`+`、`-`、`:`、`;`、`<`、`=`、`.`、`>`、`?`、`@`、`[`、`]`、`^`、`_`、`{`、`}`、`|`、`~`、`,`。
streamId	【字段含义】用于指定在腾讯云直播平台上的 streamId（选填），设置之后，您可以在腾讯云直播 CDN 上通过标准拉流方案（FLV 或 HLS）播放该用户的音视频流。 【推荐取值】限制长度为64字节，可以不填写，一种推荐的方案是使用 `sdkappid_roomid_userid_main` 作为 streamid，这种命名方式容易辨认且不会在您的多个应用中发生冲突。 【特殊说明】要使用腾讯云直播 CDN，您需要先在 控制台 中的功能配置页开启“启动自动旁路直播”开关。 【参考文档】CDN 旁路直播 。
userDefineRecordId	【字段含义】云端录制开关（选填），用于指定是否要在云端将该用户的音视频流录制下来。 【参考文档】云端录制 。 【推荐取值】限制长度为64字节，只允许包含大小写英文字母（a-zA-Z）、数字（0-9）及下划线和连词符。 方案一：手动录制方案： 1. 在 控制台>应用管理>云端录制配置 中开启云端录制。

	2. 设置 录制形式 为 手动录制 。 3. 设置手动录制后，在一个 TRTC 房间中只有设置了 userDefineRecordId 参数的用户才会在云端录制出视频文件，不指定该参数的用户不会产生录制行为。 4. 云端会以 “userDefineRecordId_起始时间_结束时间” 的格式命名录制下来的文件。 方案二：自动录制方案： 1. 需要在 控制台>应用管理>云端录制配置 中开启云端录制。 2. 设置 录制形式 为 自动录制 。 3. 设置自动录制后，在一个 TRTC 房间中的任何一个有音视频上行的用户，均会在云端录制出视频文件。 4. 文件会以 userDefineRecordId_起始时间_结束时间 的格式命名，如果不指定 userDefineRecordId，则文件会以 streamId_起始时间_结束时间 命名。
userId	【字段含义】用户标识（必填），当前用户的 userId，相当于用户名，使用 UTF-8 编码。 【推荐取值】如果一个用户在您的账号系统中的 ID 为“mike”，则 userId 即可设置为“mike”。
userSig	【字段含义】用户签名（必填），当前 userId 对应的验证签名，相当于使用云服务的登录密码。 【推荐取值】具体计算方法请参见 如何计算UserSig 。

TRTCVideoEncParam

TRTCVideoEncParam

视频编码参数

该设置决定远端用户看到的画面质量，同时也决定了云端录制出的视频文件的画面质量。

枚举类型	描述
enableAdjustRes	【字段含义】是否允许动态调整分辨率（开启后会对云端录制产生影响）。 【推荐取值】该功能适用于不需要云端录制的场景，开启后 SDK 会根据当前网络情况，智能选择出一个合适的分辨率，避免出现“大分辨率+小码率”的低效编码模式。 【特别说明】默认值：关闭。如有云端录制的需求，请不要开启此功能，因为如果视频分辨率发生变化后，云端录制出的 MP4 在普通的播放器上无法正常播放。
minVideoBitrate	【字段含义】最低视频码率，SDK 会在网络不佳的情况下主动降低视频码率

	以保持流畅度，最低会降至 minVideoBitrate 所设定的数值。 【特别说明】默认值：0，此时最低码率由 SDK 会根据您指定的分辨率，自动计算出合适的数值。 【推荐取值】您可以通过同时设置 videoBitrate 和 minVideoBitrate 两个参数，用于约束 SDK 对视频码率的调整范围： • 如果您追求 弱网络下允许卡顿但要保持清晰 的效果，可以设置 minVideoBitrate 为 videoBitrate 的 60%。 • 如果您追求 弱网络下允许模糊但要保持流畅 的效果，可以设置 minVideoBitrate 为一个较低的数值（比如 100kbps）。 • 如果您将 videoBitrate 和 minVideoBitrate 设置为同一个值，等价于关闭 SDK 对视频码率的自适应调节能力。
videoBitrate	【字段含义】目标视频码率，SDK 会按照目标码率进行编码，只有在弱网络环境下才会主动降低视频码率。 【推荐取值】请参见本 TRTCVideoResolution 在各档位注释的最佳码率，也可以在此基础上适当调高。比如：TRTCVideoResolution_1280_720 对应 1200kbps 的目标码率，您也可以设置为 1500kbps 用来获得更好的观感清晰度。 【特别说明】您可以通过同时设置 videoBitrate 和 minVideoBitrate 两个参数，用于约束 SDK 对视频码率的调整范围： • 如果您追求“弱网络下允许卡顿但要保持清晰”的效果，可以设置 minVideoBitrate 为 videoBitrate 的 60%。 • 如果您追求“弱网络下允许模糊但要保持流畅”的效果，可以设置 minVideoBitrate 为一个较低的数值（比如 100kbps）。 • 如果您将 videoBitrate 和 minVideoBitrate 设置为同一个值，等价于关闭 SDK 对视频码率的自适应调节能力。
videoFps	【字段含义】视频采集帧率。 【推荐取值】15fps或20fps。5fps以下，卡顿感明显。10fps以下，会有轻微卡顿感。20fps以上，会浪费带宽（电影的帧率为24fps）。 【特别说明】部分 Android 手机的前置摄像头并不支持15fps以上的采集帧率，部分主打美颜功能的 Android 手机的前置摄像头的采集帧率可能低于10fps。
videoResolution	【字段含义】视频分辨率。 【特别说明】如需使用竖屏分辨率，请指定 resMode 为 Portrait，例如：640 × 360 + Portrait = 360 × 640。 【推荐取值】 • 手机视频通话：建议选择 360 × 640 及以下分辨率，resMode 选择 Portrait，即竖屏分辨率。 • 手机在线直播：建议选择 540 × 960，resMode 选择 Portrait，即竖屏分辨率。 • 桌面平台（Win + Mac）：建议选择 640 × 360 及以上分辨率，resMode 选择 Landscape，即横屏分辨率。

videoResolutionMode	【字段含义】分辨率模式（横屏分辨率 or 竖屏分辨率）。 【推荐取值】手机平台（iOS、Android）建议选择 Portrait，桌面平台（Windows、Mac）建议选择 Landscape。 【特别说明】如需使用竖屏分辨率，请指定 resMode 为 Portrait，例如：640 × 360 + Portrait = 360 × 640。

TRTCNetworkQosParam

TRTCNetworkQosParam

网络流控（Qos）参数集

网络流控相关参数，该设置决定 SDK 在弱网络环境下的调控策略（例如：“清晰优先”或“流畅优先”）。

枚举类型	描述
controlMode	【字段含义】流控模式（已废弃）。 【推荐取值】云端控制。 【特别说明】请设置为云端控制模式（TRTCQosControlModeServer）。
preference	【字段含义】清晰优先还是流畅优先。 【推荐取值】清晰优先。 【特别说明】该参数主要影响 TRTC 在较差网络环境下的音视频表现： • 流畅优先：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的流畅性，代价就是画面会比较模糊且伴随有较多的马赛克。参见 TRTC_VIDEO_QOS_PREFERENCE_SMOOTH • 清晰优先（默认值）：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的清晰度，代价就是画面会比较卡顿。参见 TRTC_VIDEO_QOS_PREFERENCE_CLEAR

TRTCRenderParams

TRTCRenderParams

视频画面的渲染参数

您可以通过设置此参数来控制画面的旋转角度、填充模式和左右镜像模式。

枚举类型	描述

fillMode	【字段含义】画面填充模式。 【推荐取值】填充（画面可能会被拉伸裁剪）或适应（画面可能会有黑边），默认值：TRTC_VIDEO_RENDER_MODE_FILL。
mirrorType	【字段含义】画面镜像模式。 【推荐取值】默认值：TRTC_VIDEO_MIRROR_TYPE_AUTO。
rotation	【字段含义】图像的顺时针旋转角度。 【推荐取值】支持 90、180 以及 270 旋转角度，默认值：TRTC_VIDEO_ROTATION_0。

TRTCQualityInfo

TRTCQualityInfo

网络质量

表征网络质量的好坏，您可以通过该数值在用户界面上展示每个用户的网络质量。

枚举类型	描述
quality	网络质量
userId	用户 ID

TRTCVolumeInfo

TRTCVolumeInfo

音量大小

表征语音音量的评估值，您可以通过该数值在用户界面上展示每个用户的音量大小。

枚举类型	描述
pitch	本地用户的人声频率（单位：Hz），取值范围[0 – 4000]，对于远端用户，该值始终为0。
spectrumData	音频频谱数据是将音频数据在频率域中的分布，划分为 256 个频率段，使用 spectrumData 记录各个频率段的能量值，每个能量值的取值范围为 [-300,

	0]，单位为 dBFS。 **注意** 本地频谱使用编码前的音频数据计算，会受到本地采集音量、BGM等影响；远端频谱使用接收到的音频数据计算，本地调整远端播放音量等操作不会对其产生影响。
userId	说话者的 userId, 如果 userId 为空则代表是当前用户自己。
vad	是否检测到人声，0：非人声 1：人声。
volume	说话者的音量大小, 取值范围[0 – 100]。

TRTCSpeedTestParams

TRTCSpeedTestParams

测速参数

您可以在用户进入房间前通过 startSpeedTest 接口测试网速（注意：请不要在通话中调用）。

枚举类型	描述
expectedDownBandwidth	预期的下行带宽（kbps，取值范围： 10 ~ 5000，为 0 时不测试）。 **注意** 当参数 scene 设置为 TRTCSpeedTestScene_OnlineChorusTesting 时，为了获取更准确的 rtt / jitter 等信息，取值范围限制为 10 ~ 1000。
expectedUpBandwidth	预期的上行带宽（kbps，取值范围： 10 ~ 5000，为 0 时不测试）。 **注意** 当参数 scene 设置为 TRTCSpeedTestScene_OnlineChorusTesting 时，为了获取更准确的 rtt / jitter 等信息，取值范围限制为 10 ~ 1000。
scene	测速场景。

sdkAppId	应用标识，请参见 TRTCParams 中的相关说明。
userId	用户标识，请参见 TRTCParams 中的相关说明。
userSig	用户签名，请参见 TRTCParams 中的相关说明。

TRTCSpeedTestResult

TRTCSpeedTestResult

网络测速结果

您可以在用户进入房间前通过 startSpeedTest 接口进行测速（注意：请不要在通话中调用）。

枚举类型	描述
availableDownBandwidth	下行带宽（kbps，-1：无效值）。
availableUpBandwidth	上行带宽（kbps，-1：无效值）。
downJitter	下行数据包抖动（ms），指用户当前网络环境下数据通信的稳定性，该值越小越好，正常数值范围是0ms - 100ms，-1 代表此次测速没有成功测得有效值，一般情况下 WiFi 网络的 Jitter 会比 4G/5G 环境稍大。
downLostRate	下行丢包率，取值范围是 [0 - 1.0]，例如 0.2 表示每从服务器收取 10 个数据包可能会在中途丢失 2 个。
errMsg	带宽测试错误信息。
ip	服务器 IP 地址。
quality	内部通过评估算法测算出的网络质量，更多信息请参见 TRTCQuality 。
rtt	延迟（毫秒），指当前设备到 TRTC 服务器的一次网络往返时间，该值越小越好，正常数值范围是10ms - 100ms。
success	测试是否成功。
upJitter	上行数据包抖动（ms），指用户当前网络环境下数据通信的稳定性，该值越小越好，正常数值范围是0ms - 100ms，-1 代表此次测速没有成功测得有效值，一般情况下 WiFi 网络的 Jitter 会比 4G/5G 环境稍大。

upLostRate	上行丢包率，取值范围是 [0 – 1.0]，例如 0.3 表示每向服务器发送 10 个数据包可能会在中途丢失 3 个。

TRTCTexture

TRTCTexture

视频纹理数据

枚举类型	描述
eglContext10	【字段含义】使用 (javax.microedition.khronos.egl.*) 定义的 OpenGL 上下文环境。
eglContext14	【字段含义】使用 (android.opengl.*) 定义的 OpenGL 上下文环境。
textureId	【字段含义】视频纹理 ID。

TRTCVideoFrame

TRTCVideoFrame

视频帧信息

TRTCVideoFrame 用来描述一帧视频画面的裸数据，也就是编码前或者解码后的视频画面数据。

枚举类型	描述
buffer	【字段含义】bufferType 为 TRTCCloudDef#TRTC_VIDEO_BUFFER_TYPE_BYTE_BUFFER 时的视频数据，承载用于 JNI 层的 Direct Buffer。
bufferType	【字段含义】视频数据结构类型。
data	【字段含义】bufferType 为 TRTCCloudDef#TRTC_VIDEO_BUFFER_TYPE_BYTE_ARRAY 时的视频数据，承载用于 java 层的字节数组。
height	【字段含义】视频高度。

pixelFormat	【字段含义】视频的像素格式。
rotation	【字段含义】视频像素的顺时针旋转角度。
texture	【字段含义】bufferType 为 TRTCCloudDef#TRTC_VIDEO_PIXEL_FORMAT_Texture_2D 时的视频数据，承载用于 OpenGL 渲染的纹理数据。
timestamp	【字段含义】视频帧的时间戳，单位毫秒。 【推荐取值】自定义视频采集时可以设置为0。若该参数为0，SDK 会自定填充 timestamp 字段，但请“均匀”地控制 sendCustomVideoData 的调用间隔。
width	【字段含义】视频宽度。

TRTCAudioFrame

TRTCAudioFrame

音频帧数据

枚举类型	描述
channel	【字段含义】声道数。
data	【字段含义】音频数据。
extraData	【字段含义】音频额外数据，远端用户通过 `onLocalProcessedAudioFrame` 写入的数据会通过该字段回调。
sampleRate	【字段含义】采样率。
timestamp	【字段含义】时间戳，单位ms。

TRTCMixUser

TRTCMixUser

云端混流中各路画面的描述信息

TRTCMixUser 用于指定云端混流中每一路视频画面的位置、大小、图层以及流类型等信息。

枚举类型	描述
height	【字段含义】指定该路画面的高度（单位：像素）。
image	【字段含义】占位图或水印图 • 占位图是指当对应 userId 混流内容为纯音频时，混合后的画面中显示的是占位图片。 • 水印图是指一张贴在混合后画面中的半透明图片，这张图片会一直覆盖于混合后的画面上。 • 当指定 inputType 为 TRTCMixInputTypePureAudio 时，image 为占位图，此时需要您指定 userId。 • 当指定 inputType 为 TRTCMixInputTypeWatermark 时，image 为水印图，此时不需要您指定 userId。 【推荐取值】默认值：空值，即不设置占位图或者水印图。 【特别说明】 • 您可以将 image 设置为控制台中的某一个素材 ID，这需要您事先在 “控制台 => 应用管理 => 功能配置 => 素材管理” 中单击 [新增图片] 按钮进行上传。 • 上传成功后可以获得对应的“图片ID”，然后将“图片ID”转换成字符串类型并设置给 image 字段即可（比如假设“图片ID” 为 63，可以设置 image = @"63"）。 • 您也可以将 image 设置为图片的 URL 地址，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式，推荐使用 png 格式的半透明图片作为水印。 • image 仅在 inputType 为 TRTCMixInputTypePureAudio 或者 TRTCMixInputTypeWatermark 时才生效。
inputType	【字段含义】指定该路流的混合内容（只混音频、只混视频、混合音视频、混入水印）。 【默认取值】默认值：TRTCMixInputTypeUndefined。 【特别说明】 • 当指定 inputType 为 TRTCMixInputTypeUndefined 并设置 pureAudio 为 YES 时，等效于设置 inputType 为 TRTCMixInputTypePureAudio。 • 当指定 inputType 为 TRTCMixInputTypeUndefined 并设置 pureAudio 为 NO 时，等效于设置 inputType 为 TRTCMixInputTypeAudioVideo。 • 当指定 inputType 为 TRTCMixInputTypeWatermark 时，您可以不指定 userId 字段，但需要指定 image 字段。
pureAudio	【字段含义】指定该路流是不是只混合声音。 【推荐取值】默认值：false。 【特别说明】已废弃，推荐使用8.5版本开始新引入的字段：inputType。

renderMode	【字段含义】该画面在输出时的显示模式。 【推荐取值】默认值：视频流默认为0。0为裁剪，1为缩放，2为缩放并显示黑底。 【特别说明】水印图和占位图暂时不支持设置 renderMode，默认强制拉伸处理。
roomId	【字段含义】该路音视频流所在的房间号（设置为空值代表当前用户所在的房间号）。
soundLevel	【字段含义】该路音频参与混音时的音量等级（取值范围：0 - 100）。 【默认取值】默认值：100。
streamType	【字段含义】指定该路画面是主路画面（TRTC_VIDEO_STREAM_TYPE_BIG）还是辅路画面（TRTC_VIDEO_STREAM_TYPE_SUB）。
userId	【字段含义】用户 ID
width	【字段含义】指定该路画面的宽度（单位：像素）。
x	【字段含义】指定该路画面的 x 坐标（单位：像素）。
y	【字段含义】指定该路画面的 y 坐标（单位：像素）。
zOrder	【字段含义】指定该路画面的层级（取值范围：1 - 15，不可重复）。

TRTCTranscodingConfig

TRTCTranscodingConfig

云端混流的排版布局和转码参数

用于指定混流时各路画面的排版位置信息和云端转码的编码参数。

枚举类型	描述
appId	【字段含义】腾讯云直播服务的 AppID。 【推荐取值】请在 实时音视频控制台 依次单击【应用管理】=>【应用信息】，并在【旁路直播信息】中获取 appid。
audioBitrate	【字段含义】指定云端转码的目标音频码率。 【推荐取值】默认值：64kbps，取值范围是 [32，192]。
audioChannels	【字段含义】指定云端转码的音频声道数。 【推荐取值】默认值：1，代表单声道。可设定的数值只有两个数字：1-单声道，2-双声道。

audioCodec	【字段含义】指定云端转码的输出流音频编码类型。 【推荐取值】默认值：0，代表LC-AAC。可设定的数值只有三个数字：0 - LC-AAC，1 - HE-AAC，2 - HE-AACv2。 【特别说明】HE-AAC 和 HE-AACv2 支持的输出流音频采样率范围为 [48000, 44100, 32000, 24000, 16000]。 【特别说明】当音频编码设置为 HE-AACv2 时，只支持输出流音频声道数为双声道。 【特别说明】HE-AAC 和 HE-AACv2 取值仅在输出流为您额外设置的 streamId 上时才生效。
audioSampleRate	【字段含义】指定云端转码的目标音频采样率。 【推荐取值】默认值：48000Hz。支持12000HZ、16000HZ、22050HZ、24000HZ、32000HZ、44100HZ、48000HZ。
backgroundColor	【字段含义】指定混合画面的底色颜色。 【推荐取值】默认值：0x000000 代表黑色。格式为十六进制数字，比如：“0x61B9F1” 代表 RGB 分别为(97,158,241)。
backgroundImage	【字段含义】指定混合画面的背景图片。 【推荐取值】默认值：空值，即不设置背景图片。 【特别说明】 • 您可以将 image 设置为控制台中的某一个素材 ID，这需要您事先在 “控制台 => 应用管理 => 功能配置 => 素材管理” 中单击 [新增图片] 按钮进行上传。 • 上传成功后可以获得对应的“图片ID”，然后将“图片ID”转换成字符串类型并设置给 image 字段即可（比如假设“图片ID” 为 63，可以设置 image = @"63"）。 • 您也可以将 image 设置为图片的 URL 地址，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式。
bizId	【字段含义】腾讯云直播服务的 bizid。 【推荐取值】请在 实时音视频控制台 依次单击【应用管理】=>【应用信息】，并在【旁路直播信息】中获取 bizid。
mixUsers	【字段含义】指定云端混流中每一路视频画面的位置、大小、图层以及流类型等信息。 【推荐取值】该字段是一个 TRTCMixUser 类型的数组，数组中的每一个元素都用来代表每一路画面的信息。
mode	【字段含义】排版模式。 【推荐取值】请根据您的业务场景要求自行选择，预排版模式是适用性较好的一种模式。

streamId	【字段含义】输出到 CDN 上的直播流 ID。 【推荐取值】默认值：空值，即房间里的多路音视频流最终会混合到接口调用者的那一路音视频流上。 • 如不设置该参数，SDK 会执行默认逻辑，即房间里的多路音视频流会混合到该接口调用者的那一路音视频流上，也就是 A + B => A。 • 如您设置该参数，SDK 会将房间里的多路音视频流混合到您指定的直播流上，也就是 A + B => C（C 代表您指定的 streamId）。
videoBitrate	【字段含义】指定云端转码的目标视频码率（kbps）。 【推荐取值】如果填0，TRTC 会根据 videoWidth 和 videoHeight 估算出一个合理的码率值，您也可以参考视频分辨率枚举定义中所推荐的码率值（见注释部分）。
videoFramerate	【字段含义】指定云端转码的目标视频帧率（FPS）。 【推荐取值】默认值：15fps，取值范围是 (0,30]。
videoGOP	【字段含义】指定云端转码的目标视频关键帧间隔（GOP）。 【推荐取值】默认值：2，单位为秒，取值范围是 [1,8]。
videoHeight	【字段含义】指定云端转码的目标分辨率（高度）。 【推荐取值】单位：像素值，推荐值：640，如果你只混合音频流，请将 width 和 height 均设置为 0，否则混流转码后的直播流中会有黑色背景。
videoSeiParams	【字段含义】混流 SEI 参数，默认不填写。 【特别说明】参数以 json 字符串形式传入，示例如下： `{ "payLoadContent":"xxx", "payloadType":5, "payloadUuid":"1234567890abcdef1234567890abcdef", "interval":1000, "followIdr":false }` 当前支持的字段及含义： • payloadContent: 必填。透传 sei 的 payload 内容，不能为空； • payloadType: 必填。sei 消息的类型，取值范围: 5 或 [100, 254] 范围内的整数（244 除外，244 是内部自定义的时间戳 sei）； • payloadUuid: 当 payloadType 为 5 时必须填写，其他情况下该值会被忽略。该值必须是长度为 32 的十六进制数字； • interval: 选填，默认 1000。sei 的发送间隔，单位毫秒； • followIdr: 选填，默认 false。该值为 true 时，发送关键帧时会确保带 sei，否则不确保。

videoWidth	【字段含义】指定云端转码的目标分辨率（宽度）。 【推荐取值】单位：像素值，推荐值：360，如果你只混合音频流，请将 width 和 height 均设置为 0，否则混流转码后的直播流中会有黑色背景。

TRTCPublishCDNParam

TRTCPublishCDNParam

向非腾讯云 CDN 上发布音视频流时需设置的转推参数

TRTC 的后台服务支持通过标准 RTMP 协议，将其中的音视频流发布到第三方直播 CDN 服务商。
如果您使用腾讯云直播 CDN 服务，可无需关注此参数，直接使用 startPublish 接口即可。

枚举类型	描述
appId	【字段含义】腾讯云直播服务的 AppID。 【推荐取值】请在 实时音视频控制台 依次单击【应用管理】=>【应用信息】，并在【旁路直播信息】中获取 appid。
bizId	【字段含义】腾讯云直播服务的 bizid。 【推荐取值】请在 实时音视频控制台 依次单击【应用管理】=>【应用信息】，并在【旁路直播信息】中获取 bizid。
streamId	【字段含义】需要转推的 streamId。 【推荐取值】默认值：空值。如果不填写，则默认转推调用者的旁路流。
url	【字段含义】指定该路音视频流在第三方直播服务商的推流地址（RTMP 格式）。 【推荐取值】各家服务商的推流地址规则差异较大，请根据目标服务商的要求填写合法的推流 URL，TRTC 的后台服务器会按照您填写的 URL 向第三方服务商推送标准格式音视频流。 【特别说明】推流 URL 必须为 RTMP 格式，必须符合您的目标直播服务商的规范要求，否则目标服务商会拒绝来自 TRTC 后台服务的推流请求。

TRTCAudioRecordingParams

TRTCAudioRecordingParams

本地音频文件的录制参数

该参数用于在音频录制接口 startAudioRecording 中指定录制参数。

枚举类型	描述
filePath	【字段含义】录音文件的保存路径（必填）。 【特别说明】该路径需精确到文件名及格式后缀，格式后缀用于决定录音文件的格式，目前支持的格式有 PCM、WAV 和 AAC。 例如：假如您指定路径为 "mypath/record/audio.aac"，代表您希望 SDK 生成一个 AAC 格式的音频录制文件。 请您指定一个有读写权限的合法路径，否则录音文件无法生成。
maxDurationPerFile	【字段含义】maxDurationPerFile 录制文件分片时长，单位毫秒，最小值10000。默认值为0，表示不分片。
recordingContent	【字段含义】音频录制内容类型。 【特别说明】默认录制所有本地和远端音频。

TRTCLocalRecordingParams

TRTCLocalRecordingParams

本地媒体文件的录制参数

该参数用于在本地媒体文件的录制接口 startLocalRecording 中指定录制相关参数。
接口 startLocalRecording 是接口 startAudioRecording 的能力加强版本，前者可以录制视频文件，后者只能录制音频文件。

枚举类型	描述
filePath	【字段含义】录制的文件地址（必填），请确保路径有读写权限且合法，否则录制文件无法生成。 【特别说明】该路径需精确到文件名及格式后缀，格式后缀用于决定录制出的文件格式，目前支持的格式暂时只有 MP4。 例如：假如您指定路径为 "mypath/record/test.mp4"，代表您希望 SDK 生成一个 MP4 格式的本地视频文件。 请您指定一个有读写权限的合法路径，否则录制文件无法生成。
interval	【字段含义】interval 录制信息更新频率，单位毫秒，有效范围：1000-10000。默认值为-1，表示不回调。
maxDurationPerFile	【字段含义】maxDurationPerFile 录制文件分片时长，单位毫秒，最小值10000。默认值为0，表示不分片。
recordType	【字段含义】媒体录制类型，默认值：TRTCRecordTypeBoth，即同时录制音频和视频。

TRTCSwitchRoomConfig

TRTCSwitchRoomConfig

房间切换参数

该参数用于切换房间接口 switchRoom ，可以让用户从一个房间快速切换到另一个房间。

枚举类型	描述
privateMapKey	【字段含义】用于权限控制的权限票据（选填），当您希望某个房间只能让特定的 userId 进入时，需要使用 privateMapKey 进行权限保护。 【推荐取值】仅建议有高级别安全需求的客户使用，更多详情请参见 进房权限保护 。
roomId	【字段含义】数字房间号码 [选填]，在同一个房间内的用户可以看到彼此并能够进行音视频通话。 【推荐取值】取值范围：1 - 4294967294。 【特别说明】roomId 和 strRoomId 必须并且只能填一个。若两者都填，则优先选择 roomId。
strRoomId	【字段含义】字符串房间号码 [选填]，在同一个房间内的用户可以看到彼此并能够进行音视频通话。 【特别说明】roomId 和 strRoomId 必须并且只能填一个。若两者都填，则优先选择 roomId。
userSig	【字段含义】用户签名 [选填]，当前 userId 对应的验证签名，相当于登录密码。 如果您在切换房间时不指定新计算出的 userSig，SDK 会继续使用您在进入房间时（enterRoom）时所指定的 userSig。 这就需要您必须保证旧的 userSig 在切换房间的那一刻仍在签名允许的效期内，否则会导致房间切换失败。 【推荐取值】具体计算方法请参见 如何计算UserSig 。

TRTCAudioFrameDelegateFormat

TRTCAudioFrameDelegateFormat

音频自定义回调的格式参数

该参数用于在音频自定义回调相关的接口中，设置 SDK 回调出来的音频数据的相关格式（包括采样率、声道数等）。

枚举类型	描述
channel	【字段含义】声道数。 【推荐取值】默认值：1，代表单声道。可设定的数值只有两个数字：1-单声道，2-双声道。
mode	【字段含义】回调数据读写模式 【推荐取值】TRTCAudioFrameOperationModeReadOnly：仅从回调中获取音频数据。可设定的模式有 TRTCAudioFrameOperationModeReadOnly，TRTCAudioFrameOperationModeReadWrite。
sampleRate	【字段含义】采样率。 【推荐取值】默认值：48000Hz。支持 16000, 32000, 44100, 48000。
samplesPerCall	【字段含义】采样点数。 【推荐取值】取值必须是 sampleRate/100 的整数倍。

TRTCScreenShareParams

TRTCScreenShareParams

屏幕分享参数（仅适用于 Android 平台）

该参数用于在屏幕分享接口 startScreenCapture 中指定屏幕分享时的悬浮窗等相关信息。

枚举类型	描述
enableForegroundService	【字段含义】是否同时启用 SDK 内置的前台服务，停止屏幕分享时会停止该服务。 从 Android 10 开始，如果 APP 的 targetSdkVersion 大于等于 29 时， 需要有 mediaProjection 类型的前台服务才能成功开始，否则会报 SecurityException。 您可以自己启动一个前台服务，也可以使用 SDK 内置的前台服务。当使用内置的前台服务时， 您需要在 AndroidManifest.xml 中添加以下内容： <uses-permission android:name="android.permission.FOREGROUND_SERVICE" />

	<service android:name="com.tencent.rtmp.video.ScreenCaptureService" android:exported="false" android:foregroundServiceType="mediaProjection"> </service> 【推荐取值】推荐您自己定制前台服务，保持该值为 false 即可。
floatingView	【字段含义】可以通过此参数设置一个悬浮view。 【推荐取值】从 Android 7.0 系统开始，对于没有做过任何保活措施的后台应用，Android 系统会很快将其杀掉以减少系统能耗。 但当您的 App 处于屏幕分享状态时，不可避免地要被切到系统后台运行，此时如果能弹出悬浮窗可以避免被系统强杀掉。 同时，在手机屏幕上显示悬浮窗，也有利于告知用户当前正在做屏幕分享，以便提醒用户避免泄漏个人隐私。 【特殊说明】您也可以使用 Android 系统的 WindowsManager 接口达到同样的效果。
mediaProjection	【字段含义】可以通过此参数将你自己创建的 MediaProjection 传入给 SDK 使用。 【推荐取值】正常情况下，该参数保持为 null 即可。

TRTCAudioParallelParams

TRTCAudioParallelParams

远端音频流智能并发播放策略的参数

该参数用于设置远端音频流智能并发播放策略。

枚举类型	描述
includeUsers	【字段含义】指定用户必定能并发播放。 【特殊说明】指定必定并发播放的用户 ID 列表。这些用户不参与智能选择。includeUsers 的数量必须小于 maxCount，否则本次并发播放设置失效。includeUsers 仅在 maxCount > 0 时有效。当 includeUsers 生效时，参与智能并发选择的最大播放数 = maxCount - 有效 includeUsers 的数量。
maxCount	【字段含义】最大并发播放数。默认值：0。 • 如果 maxCount > 0，且实际人数 > maxCount，会实时智能选出 maxCount 路数据进行播放，这会极大的降低性能消耗。 • 如果 maxCount = 0，SDK 不限制并发播放数，在上麦人数比较多的房间可能会引发性能问题。

TRTCUser

TRTCUser

媒体流发布相关配置的用户信息

您可以通过设置该参数，配合媒体流目标发布参数（TRTCPublishTarget）和混流转码参数（TRTCStreamMixingConfig），将您指定的多路音视频流进行转码并发布到您填写的目标发布地址中

枚举类型	描述
intRoomId	【字段含义】数字房间号，需要和您的进房参数（TRTCParams）中的房间号类型相匹配。 【推荐取值】取值范围：1 - 4294967294。 【特别说明】intRoomId 与 strRoomId 是互斥的，若您进房参数中选用 strRoomId，则 intRoomId 需要填写为0。若两者都填，SDK 将优先选用 intRoomId。
strRoomId	【字段含义】字符串房间号，需要和您的进房参数（TRTCParams）中的房间号类型相匹配。 【特别说明】intRoomId 与 strRoomId 是互斥的，若您进房参数中选用 roomId，则 strRoomId 无需填写。若两者都填，SDK 将优先选用 intRoomId。 【推荐取值】限制长度为 64 字节。以下为支持的字符集范围（共 89 个字符）： • 大小写英文字母（a-zA-Z）； • 数字（0-9）； • 空格、`!`、`#`、`$`、`%`、`&`、`(`、`)`、`+`、`-`、`:`、`;`、`<`、`=`、`.`、`>`、`?`、`@`、`[`、`]`、`^`、`_`、`{`、`}`、`|`、`~`、`,`。
userId	【字段含义】用户标识，当前用户的 userId，相当于用户名，使用 UTF-8 编码。 【推荐取值】如果一个用户在您的账号系统中的 ID 为“mike”，则 userId 即可设置为“mike”。

TRTCPublishCdnUrl

TRTCPublishCdnUrl

向腾讯或者第三方 CDN 上发布音视频流时需设置的 url 配置

该配置用于媒体流发布接口 startPublishMediaStream 中的目标推流配置（TRTCPublishTarget）

枚举类型	描述
isInternalLine	【字段含义】指定该路音视频流是否发布至腾讯云。 【推荐取值】默认值：true。 【特别说明】若您的目标直播服务商为腾讯，请将此参数设置为 true，此时后台计费系统不会对此计算转推服务费。
rtmpUrl	【字段含义】指定该路音视频流在腾讯或者第三方直播服务商的推流地址（RTMP 格式）。 【推荐取值】各家服务商的推流地址规则差异较大，请根据目标服务商的要求填写合法的推流 URL，TRTC 的后台服务器会按照您填写的 URL 向第三方服务商推送标准格式音视频流。 【特别说明】推流 URL 必须为 RTMP 格式，必须符合您的目标直播服务商的规范要求，否则目标服务商会拒绝来自 TRTC 后台服务的推流请求。

TRTCPublishTarget

TRTCPublishTarget

目标推流配置

该配置用于媒体流发布接口 startPublishMediaStream 。

枚举类型	描述
cdnUrlList	【字段含义】发布至腾讯或者第三方直播服务商的推流地址（RTMP 格式）。 【特别说明】若您的 mode 选择为 TRTCPublishMixStreamToRoom，此时您不需要设置该参数。
mixStreamIdentity	【字段含义】回推房间机器人信息。 【特别说明】仅当您的 mode 选择为 TRTCPublishMixStreamToRoom 时，您需要设置该参数。 【特别说明】设置后，该路转码音视频数据将回推到您指定的房间中。建议设置为特殊的 userId，以避免难以区分回推机器人和您通过 TRTC SDK 进房的主播。 【特别说明】参与混流的用户不支持订阅该转码流。 【特别说明】当您进房前设置的订阅模式(setDefaultStreamRecvMode)均为手动时，您需要自行管理您想要拉取的音视频流（通常当您拉取回推房间的转码流时，您应该取消订阅参与转码的对应音视频单流）。

	【特别说明】当您进房前设置的订阅模式(setDefaultStreamRecvMode)均为自动时，不参与转码的用户将自动收到后台下发的转码流并不再继续接收参与转码的音视频单流。除非您明确进行取消订阅（muteRemoteVideoStream 和 muteRemoteAudio），否则转码流数据将持续下发。
mode	【字段含义】媒体流发布模式。 【推荐取值】请根据您的业务场景要求自行选择，TRTC 支持转推、转码和回推到 RTC 房间的模式。 【特别说明】若您的业务场景需要多个发布模式，您可以通过多次调用媒体流发布接口（startPublishMediaStream）并分别设置不同的 TRTCPublishTarget。 【特别说明】同一个 mode 请对应一个媒体流发布接口（startPublishMediaStream），并在后续需要调整时，使用 updatePublishCDNStream 进行更新。

TRTCVideoLayout

TRTCVideoLayout

转码视频布局

该配置用于媒体流发布接口（startPublishMediaStream）中的转码配置（TRTCStreamMixingConfig）。
用于指定转码流中每一路视频画面的位置、大小、图层以及流类型等信息。

枚举类型	描述
backgroundColor	【字段含义】指定混合画面的底色颜色。 【推荐取值】默认值：0x000000 代表黑色。格式为十六进制数字，比如：“0x61B9F1” 代表 RGB 分别为 (97,158,241)。
fillMode	【字段含义】画面填充模式。 【推荐取值】填充（画面可能会被拉伸裁剪）或适应（画面可能会有黑边），默认值：TRTCVideoFillMode_Fill。
fixedVideoStreamType	【字段含义】指定该路画面是主路画面（TRTC_VIDEO_STREAM_TYPE_BIG）还是辅路画面（TRTC_VIDEO_STREAM_TYPE_SUB）。
fixedVideoUser	【字段含义】参与转码的用户信息。 【特别说明】用户信息（TRTCUser）支持不填写（即 userId、intRoomId 和 strRoomId 均不填写）。此时当发起混流的房间中有主播

	上行音视频数据时，TRTC 后台服务器将自动将对应主播音视频填充到您指定的布局中。
height	【字段含义】指定该路画面的高度（单位：像素）。
placeHolderImage	【字段含义】占位图 URL，即当指定用户暂时仅上行音频时，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 【推荐取值】默认值：空值，即不设置占位图。 【特别说明】此时需要您指定用户信息 fixedVideoUser 中的 userId。 【特别说明】 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式，推荐使用 png 格式的半透明图片作为占位图。
width	【字段含义】指定该路画面的宽度（单位：像素）。
x	【字段含义】指定该路画面的 x 坐标（单位：像素）。
y	【字段含义】指定该路画面的 y 坐标（单位：像素）。
zOrder	【字段含义】指定该路画面的层级（取值范围：0 – 15，不可重复）。

TRTCWatermark

TRTCWatermark

水印布局

该配置用于媒体流发布接口（startPublishMediaStream）中的转码配置（TRTCStreamMixingConfig）

枚举类型	描述
height	【字段含义】指定该路水印画面的高度（单位：像素）。
watermarkUrl	【字段含义】水印 URL，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 【特别说明】 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式，推荐使用 png 格式的半透明图片作为水印。
width	【字段含义】指定该路水印画面的宽度（单位：像素）。

x	【字段含义】指定该路水印画面的 x 坐标（单位：像素）。
y	【字段含义】指定该路水印画面的 y 坐标（单位：像素）。
zOrder	【字段含义】指定该路水印画面的层级（取值范围：0 – 15，不可重复）。

TRTCStreamEncoderParam

TRTCStreamEncoderParam

媒体流编码输出参数

【字段含义】该配置用于媒体流发布接口（startPublishMediaStream）。
【特别说明】当您的发布目标（TRTCPublishTarget）中的 mode 配置为 TRTCPublish_MixStream_ToCdn 或者 TRTCPublish_MixStream_ToRoom 时，该参数为必填。
【特别说明】当您使用转推服务（mode 为 TRTCPublish_BigStream_ToCdn 或者 TRTCPublish_SubStream_ToCdn）时，为了更好的转推稳定性以及更好的 CDN 播放兼容性，也建议您填写该配置的具体参数。

枚举类型	描述
audioEncodedChannelNum	【字段含义】指定媒体发布流的目标音频声道数。 【推荐取值】默认值：1，代表单声道。可设定的数值只有两个数字：1–单声道，2–双声道。
audioEncodedCodecType	【字段含义】指定媒体发布流的目标音频编码类型。 【推荐取值】默认值：0，代表 LC–AAC。可设定的数值只有三个数字：0 – LC–AAC，1 – HE–AAC，2 – HE–AACv2。 【特别说明】HE–AAC 和 HE–AACv2 支持的输出流音频采样率范围为 [48000, 44100, 32000, 24000, 16000]。 【特别说明】当音频编码设置为 HE–AACv2 时，只支持输出流音频声道数为双声道。
audioEncodedKbps	【字段含义】指定媒体发布流的目标音频码率（kbps）。 【推荐取值】默认值：50kbps，取值范围是 [32，192]。
audioEncodedSampleRate	【字段含义】指定媒体发布流的目标音频采样率。 【推荐取值】默认值：48000Hz。取值为 [48000, 44100, 32000, 24000, 16000, 8000]，单位是 Hz。
videoEncodedCodecType	【字段含义】指定媒体发布流的目标视频编码类型。

	【推荐取值】默认值：0，代表 H264。可设定的数值只有两个数字：0 - H264，1 - H265。
videoEncodedFPS	【字段含义】指定媒体发布流的目标视频帧率（FPS）。 【推荐取值】推荐值：20fps，取值范围是 (0,30]。
videoEncodedGOP	【字段含义】指定媒体发布流的目标视频关键帧间隔（GOP）。 【推荐取值】推荐值：3，单位为秒，取值范围是 [1,5]。
videoEncodedHeight	【字段含义】指定媒体发布流的目标分辨率（高度）。 【推荐取值】单位：像素值，推荐值：640，如果你只混合音频流，请将 width 和 height 均设置为 0，否则混流转码后的直播流中会有黑色背景。
videoEncodedKbps	【字段含义】指定媒体发布流的目标视频码率（kbps）。 【推荐取值】如果填 0，TRTC 会根据 videoEncodedWidth 和 videoEncodedHeight 估算出一个合理的码率值，您也可以参考视频分辨率枚举定义中所推荐的码率值（见注释部分）。
videoEncodedWidth	【字段含义】指定媒体发布流的目标分辨率（宽度）。 【推荐取值】单位：像素值，推荐值：368，如果你只混合音频流，请将 width 和 height 均设置为 0，否则混流转码后的直播流中会有黑色背景。

videoSeiParams	【字段含义】混流 SEI 参数，默认不填写。
	【特别说明】参数以 json 字符串形式传入，示例如下：
	<pre>{
	"payLoadContent":"xxx",
	"payloadType":5,
	
	"payloadUuid":"1234567890abcdef1234567890abcdef",
	"interval":1000,
	"followIdr":false
	}</pre>
	当前支持的字段及含义：
	• payloadContent: 必填。透传 sei 的 payload 内容，不能为空；
	• payloadType: 必填。sei 消息的类型，取值范围: 5 或 [100, 254] 范围内的整数（244 除外，244 是内部自定义的时间戳 sei）；
	• payloadUuid: 当 payloadType 为 5 时必须填写，其他情况下该值会被忽略。该值必须是长度为 32 的十六进制数字；
	• interval: 选填，默认 1000。sei 的发送间隔，单位毫秒；
	• followIdr: 选填，默认 false。该值为 true 时，发送关键帧时会确保带 sei，否则不确保。

TRTCStreamMixingConfig

TRTCStreamMixingConfig

媒体流转码配置参数

该配置用于媒体流发布接口（ startPublishMediaStream ）。
用于指定转码时各路画面的排版位置信息和输入的音频信息。

枚举类型	描述
audioMixUserList	【字段含义】指定转码流中的每一路输入音频的信息。 【推荐取值】该字段是一个 TRTCUser 类型的数组，数组中的每一个元素都用来代表每一路输入音频的信息。 【特别说明】用户信息支持不填写（即 audioMixUserList 为空）。此时若设置了 TRTCStreamEncoderParam 中音频相关编码输出参数，TRTC 后台服务器将自动将所有主播的音频混合输出（当前仅支持最高 16 路音视频输入）。
backgroundColor	【字段含义】指定混合画面的底色颜色。 【推荐取值】默认值：0x000000 代表黑色。格式为十六进制数字，比如：“0x61B9F1” 代表 RGB 分别为（97、158、241）。
backgroundImage	【字段含义】指定混合画面的背景图 URL，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 【推荐取值】默认值：空值，即不设置背景图片。 【特别说明】 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式，推荐使用 png 格式的半透明图片作为背景图。
videoLayoutList	【字段含义】指定混合画面中的每一路视频画面的位置、大小、图层以及流类型等信息。 【推荐取值】该字段是一个 TRTCVideoLayout 类型的数组，数组中的每一个元素都用来代表每一路画面的信息。
watermarkList	【字段含义】指定混合画面中的每一路水印画面的位置、大小、图层等信息。 【推荐取值】该字段是一个 TRTCWatermark 类型的数组，数组中的每一个元素都用来代表每一路水印的信息。

TRTCPayloadPrivateEncryptionConfig

TRTCPayloadPrivateEncryptionConfig

媒体流私有加密配置

该配置用于设置媒体流私有加密的算法和密钥。

枚举类型	描述
encryptionAlgorithm	【字段含义】加密算法，默认为 TRTCEncryptionAlgorithmAes128Gcm。
encryptionKey	【字段含义】加密用密钥，字符串类型。 【推荐取值】若加密算法为 TRTCEncryptionAlgorithmAes128Gcm，密匙长度需为 16 字节，若加密算法为 TRTCEncryptionAlgorithmAes256Gcm，密匙长度需为 32 字节。
encryptionSalt	【字段含义】盐，加密用初始向量。 【推荐取值】需确保填入该参数的数组不为空、不全为 0 且数据长度为 32 字节。

TRTCAudioVolumeEvaluateParams

TRTCAudioVolumeEvaluateParams

音量评估等相关参数设置

该设置用于开启人声检测、声音频谱计算。

枚举类型	描述
enablePitchCalculation	【字段含义】是否开启本地人声频率计算
enableSpectrumCalculation	【字段含义】是否开启声音频谱计算。
enableVadDetection	【字段含义】是否开启本地人声检测。 【请您注意】在 startLocalAudio 之前调用才可以生效。
interval	【字段含义】设置 onUserVoiceVolume 回调的触发间隔，单位为毫秒，最小间隔为 100ms，如果小于等于 0 则会关闭回调。 【推荐取值】推荐值：300，单位为毫秒。

	【特别说明】interval 大于 0 时，音量提示将默认开启，无需另外设置。

废弃接口

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2022 Tencent. All rights reserved.

Module: DeprecatedTRTCCloud @ TXLiteAVSDK

Function: TRTC 废弃接口

废弃接口

DeprecatedTRTCCloud

函数列表	描述
setListener	设置 TRTC 事件回调
setBeautyStyle	设置美颜、美白以及红润效果级别
setEyeScaleLevel	设置大眼级别
setFaceSlimLevel	设置瘦脸级别
setFaceVLevel	设置 V 脸级别
setChinLevel	设置下巴拉伸或收缩幅度
setFaceShortLevel	设置短脸级别
setNoseSlimLevel	设置瘦鼻级别
selectMotionTmpl	设置动效贴纸
setMotionMute	设置动效静音
setFilter	设置色彩滤镜效果
setFilterConcentration	设置色彩滤镜浓度
setGreenScreenFile	设置绿幕背景视频
setReverbType	设置混响效果

setVoiceChangerType	设置变声类型
enableAudioEarMonitoring	开启（或关闭）耳返
enableAudioVolumeEvaluation	启用音量大小提示
enableAudioVolumeEvaluation	启用音量大小提示
switchCamera	切换摄像头
isCameraZoomSupported	查询当前摄像头是否支持缩放
setZoom	设置摄像头缩放倍数（焦距）
isCameraTorchSupported	查询是否支持开关闪光灯
enableTorch	开关/关闭闪光灯
isCameraFocusPositionInPreviewSupported	查询摄像头是否支持设置焦点
setFocusPosition	设置摄像头焦点坐标位置
isCameraAutoFocusFaceModeSupported	查询是否支持自动识别人脸位置
setSystemVolumeType	设置系统音量类型
checkAudioCapabilitySupport	查询是否支持音频某种能力（仅适用于Android）
startLocalAudio	设置音频质量
startRemoteView	开始显示远端视频画面
stopRemoteView	停止显示远端视频画面，同时不再拉取该远端用户的视频数据流
setLocalViewFillMode	设置本地图像的渲染模式
setLocalViewRotation	设置本地图像的顺时针旋转角度
setLocalViewMirror	设置本地摄像头预览画面的镜像模式
setRemoteViewFillMode	设置远端图像的渲染模式
setRemoteViewRotation	设置远端图像的顺时针旋转角度
startRemoteSubStreamView	开始显示远端用户的辅路画面
stopRemoteSubStreamView	停止显示远端用户的辅路画面

版权所有：腾讯云计算（北京）有限责任公司

setRemoteSubStreamViewFillMode	设置辅路画面的填充模式
setRemoteSubStreamViewRotation	设置辅路画面的顺时针旋转角度
setAudioQuality	设置音频质量
setPriorRemoteVideoStreamType	设定优先观看大画面还是小画面
setMicVolumeOnMixing	设置麦克风音量大小
playBGM	启动播放背景音乐
stopBGM	停止播放背景音乐
pauseBGM	停止播放背景音乐
resumeBGM	停止播放背景音乐
getBGMDuration	获取背景音乐总时长（单位：毫秒）
setBGMPosition	设置背景音乐的播放进度
setBGMVolume	设置背景音乐的音量大小
setBGMPlayoutVolume	设置背景音乐的本地播放音量
setBGMPublishVolume	设置背景音乐的远端播放音量
playAudioEffect	播放音效
setAudioEffectVolume	设置音效音量
stopAudioEffect	停止播放音效
stopAllAudioEffects	停止所有音效
setAllAudioEffectsVolume	设置所有音效音量
pauseAudioEffect	暂停音效
resumeAudioEffect	暂停音效
enableCustomVideoCapture	启用视频自定义采集模式
sendCustomVideoData	投送自己采集的视频数据

版权所有：腾讯云计算（北京）有限责任公司

muteLocalVideo	暂停/恢复发布本地的视频流
muteRemoteVideoStream	暂停 / 恢复订阅远端用户的视频流
snapshotVideo	视频画面截图
startSpeedTest	开始进行网络测速（进入房间前使用）
startScreenCapture	启动屏幕分享
setVideoEncoderRotation	设置视频编码器输出的画面方向
setVideoEncoderMirror	设置编码器输出的画面镜像模式
setGSensorMode	设置重力感应的适配模式

setListener

setListener

void setListener (TRTCCloudListener listener)

设置 TRTC 事件回调

@deprecated v11.4 版本开始不推荐使用，建议使用 addListener 替代之。

setBeautyStyle

setBeautyStyle

void setBeautyStyle (int beautyStyle
int beautyLevel
int whitenessLevel
int ruddinessLevel)

设置美颜、美白以及红润效果级别

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setEyeScaleLevel

setEyeScaleLevel

void setEyeScaleLevel (int eyeScaleLevel)

设置大眼级别

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setFaceSlimLevel

setFaceSlimLevel

void setFaceSlimLevel (int faceScaleLevel)

设置瘦脸级别

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setFaceVLevel

setFaceVLevel

void setFaceVLevel (int faceVLevel)

设置 V 脸级别

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setChinLevel

setChinLevel

void setChinLevel (int chinLevel)

设置下巴拉伸或收缩幅度

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setFaceShortLevel

setFaceShortLevel

void setFaceShortLevel (int faceShortlevel)

设置短脸级别

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setNoseSlimLevel

setNoseSlimLevel

void setNoseSlimLevel (int noseSlimLevel)

设置瘦鼻级别

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

selectMotionTmpl

selectMotionTmpl

void selectMotionTmpl (String motionPath)

设置动效贴纸

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setMotionMute

setMotionMute

void setMotionMute (boolean motionMute)

设置动效静音

@deprecated v6.9 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setFilter

setFilter

void setFilter (Bitmap image)

设置色彩滤镜效果

@deprecated v7.2 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setFilterConcentration

setFilterConcentration

void setFilterConcentration (float concentration)

设置色彩滤镜浓度

@deprecated v7.2 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setGreenScreenFile

setGreenScreenFile

boolean setGreenScreenFile (String file)

设置绿幕背景视频

@deprecated v7.2 版本开始不推荐使用，建议使用 getBeautyManager 替代之。

setReverbType

setReverbType

void setReverbType	(int reverbType)

设置混响效果

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setVoiceReverbType 替代之。

setVoiceChangerType

setVoiceChangerType

boolean setVoiceChangerType	(int voiceChangerType)

设置变声类型

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setVoiceChangerType 替代之。

enableAudioEarMonitoring

enableAudioEarMonitoring

void enableAudioEarMonitoring	(boolean enable)

开启（或关闭）耳返

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setVoiceEarMonitor 替代之。

enableAudioVolumeEvaluation

enableAudioVolumeEvaluation

void	(int interval)

enableAudioVolumeEvaluation

启用音量大小提示

@deprecated v10.1 版本开始不推荐使用，建议使用 enableAudioVolumeEvaluation(enable, params) 替代之。

enableAudioVolumeEvaluation

enableAudioVolumeEvaluation

void enableAudioVolumeEvaluation	(int interval boolean enable_vad)

启用音量大小提示

@deprecated v11.2 版本开始不推荐使用，建议使用 enableAudioVolumeEvaluation(enable, params) 替代之。

switchCamera

switchCamera

切换摄像头

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 switchCamera 接口替代之。

isCameraZoomSupported

isCameraZoomSupported

查询当前摄像头是否支持缩放

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 isCameraZoomSupported 接口替代之。

setZoom

setZoom

void setZoom	(int distance)

设置摄像头缩放倍数（焦距）

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCameraZoomRatio 接口替代之。

isCameraTorchSupported

isCameraTorchSupported

查询是否支持开关闪光灯

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 isCameraTorchSupported 接口替代之。

enableTorch

enableTorch

boolean enableTorch	(boolean enable)

开关/关闭闪光灯

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 enableCameraTorch 接口替代之。

isCameraFocusPositionInPreviewSupported

isCameraFocusPositionInPreviewSupported

查询摄像头是否支持设置焦点

@deprecated v8.0 版本开始不推荐使用。

setFocusPosition

setFocusPosition

void setFocusPosition (int x
int y)

设置摄像头焦点坐标位置

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCameraFocusPosition 接口替代之。

isCameraAutoFocusFaceModeSupported

isCameraAutoFocusFaceModeSupported

查询是否支持自动识别人脸位置

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 isAutoFocusEnabled 接口替代之。

setSystemVolumeType

setSystemVolumeType

void setSystemVolumeType (int type)

设置系统音量类型

@deprecated v8.0 版本开始不推荐使用，建议使用 startLocalAudio(quality) 替代之，通过 quality 参数来决策音质。

checkAudioCapabilitySupport

checkAudioCapabilitySupport

int checkAudioCapabilitySupport (int capabilityType)

查询是否支持音频某种能力（仅适用于Android）

@deprecated v10.1 版本开始不推荐使用

- TRTCAudioCapabilityLowLatencyChorus，低延时合唱能力。
- TRTCAudioCapabilityLowLatencyEarMonitor，低延时耳返能力。

参数	描述
capabilityType	音频能力类型.

返回值说明：

0：不支持此能力；1：支持此能力。

startLocalAudio

startLocalAudio

设置音频质量

@deprecated v8.0 版本开始不推荐使用，建议使用 startLocalAudio 替代之。

startRemoteView

startRemoteView

void startRemoteView (String userId, TXCloudVideoView view)

开始显示远端视频画面

@deprecated v8.0 版本开始不推荐使用，建议使用 startRemoteView 替代之。

stopRemoteView

stopRemoteView

void stopRemoteView (String userId)

停止显示远端视频画面，同时不再拉取该远端用户的视频数据流

@deprecated v8.0 版本开始不推荐使用，建议使用 stopRemoteView 替代之。

setLocalViewFillMode

setLocalViewFillMode

void setLocalViewFillMode (int mode)

设置本地图像的渲染模式

@deprecated v8.0 版本开始不推荐使用，建议使用 setLocalRenderParams 替代之。

setLocalViewRotation

setLocalViewRotation

void setLocalViewRotation (int rotation)

设置本地图像的顺时针旋转角度

@deprecated v8.0 版本开始不推荐使用，建议使用 setLocalRenderParams 替代之。

setLocalViewMirror

setLocalViewMirror

void setLocalViewMirror (int mirrorType)

设置本地摄像头预览画面的镜像模式

@deprecated v8.0 版本开始不推荐使用，建议使用 setLocalRenderParams 替代之。

setRemoteViewFillMode

setRemoteViewFillMode	
void setRemoteViewFillMode	(String userId int mode)

设置远端图像的渲染模式

@deprecated v8.0 版本开始不推荐使用，建议使用 setRemoteRenderParams 替代之。

setRemoteViewRotation

setRemoteViewRotation	
void setRemoteViewRotation	(String userId int rotation)

设置远端图像的顺时针旋转角度

@deprecated v8.0 版本开始不推荐使用，建议使用 setRemoteRenderParams 替代之。

startRemoteSubStreamView

startRemoteSubStreamView	
void startRemoteSubStreamView	(String userId TXCloudVideoView view)

开始显示远端用户的辅路画面

@deprecated v8.0 版本开始不推荐使用，建议使用 startRemoteView 替代之。

stopRemoteSubStreamView

stopRemoteSubStreamView

void stopRemoteSubStreamView	(String userId)

停止显示远端用户的辅路画面

@deprecated v8.0 版本开始不推荐使用，建议使用 stopRemoteView 替代之。

setRemoteSubStreamViewFillMode

setRemoteSubStreamViewFillMode

void setRemoteSubStreamViewFillMode	(String userId int mode)

设置辅路画面的填充模式

@deprecated v8.0 版本开始不推荐使用，建议使用 setRemoteRenderParams 替代之。

setRemoteSubStreamViewRotation

setRemoteSubStreamViewRotation

void setRemoteSubStreamViewRotation	(final String userId final int rotation)

设置辅路画面的顺时针旋转角度

@deprecated v8.0 版本开始不推荐使用，建议使用 setRemoteRenderParams:streamType:params: 替代之。

setAudioQuality

setAudioQuality

void setAudioQuality (int quality)

设置音频质量

@deprecated v8.0 版本开始不推荐使用，建议使用 startLocalAudio 替代之。

setPriorRemoteVideoStreamType

setPriorRemoteVideoStreamType

int setPriorRemoteVideoStreamType (int streamType)

设定优先观看大画面还是小画面

@deprecated v8.0 版本开始不推荐使用，建议使用 startRemoteView:streamType:view: 替代之。

setMicVolumeOnMixing

setMicVolumeOnMixing

void setMicVolumeOnMixing (int volume)

设置麦克风音量大小

playBGM

playBGM

void playBGM	(String path TRTCCloud.BGMNotify notify)

启动播放背景音乐

@deprecated v7.3 版本开始不推荐使用，建议使用 getAudioEffectManager 替代之。

stopBGM

stopBGM

停止播放背景音乐

@deprecated v7.3 版本开始不推荐使用，建议使用 getAudioEffectManager 替代之。

pauseBGM

pauseBGM

停止播放背景音乐

@deprecated v7.3 版本开始不推荐使用，建议使用 getAudioEffectManager 替代之。

resumeBGM

resumeBGM

停止播放背景音乐

@deprecated v7.3 版本开始不推荐使用，建议使用 getAudioEffectManager 替代之。

getBGMDuration

getBGMDuration

int getBGMDuration	(String path)

获取背景音乐总时长（单位：毫秒）

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 getMusicDurationInMS 替代之。

setBGMPosition

setBGMPosition

int setBGMPosition	(int pos)

设置背景音乐的播放进度

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 seekMusicToPosInMS 替代之。

setBGMVolume

setBGMVolume

void setBGMVolume	(int volume)

设置背景音乐的音量大小

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setMusicVolume 替代之。

setBGMPlayoutVolume

setBGMPlayoutVolume

void setBGMPlayoutVolume	(int volume)

设置背景音乐的本地播放音量

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setMusicPlayoutVolume 替代之。

setBGMPublishVolume

setBGMPublishVolume

void setBGMPublishVolume	(int volume)

设置背景音乐的远端播放音量

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setBGMPublishVolume 替代之。

playAudioEffect

playAudioEffect

void playAudioEffect	(TRTCCloudDef.TRTCAudioEffectParam effect)

播放音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 startPlayMusic 替代之。

setAudioEffectVolume

setAudioEffectVolume

void setAudioEffectVolume	(int effectId int volume)

设置音效音量

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setMusicPublishVolume 和 setMusicPlayoutVolume 替代之。

stopAudioEffect

stopAudioEffect

void stopAudioEffect (int effectId)

停止播放音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 stopPlayMusic 替代之。

stopAllAudioEffects

stopAllAudioEffects

停止所有音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 stopPlayMusic 替代之。

setAllAudioEffectsVolume

setAllAudioEffectsVolume

void setAllAudioEffectsVolume (int volume)

设置所有音效音量

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setMusicPublishVolume 和 setMusicPlayoutVolume 替代之。

pauseAudioEffect

pauseAudioEffect

void pauseAudioEffect (int effectId)

暂停音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 pauseAudioEffect 替代之。

resumeAudioEffect

resumeAudioEffect

void resumeAudioEffect (int effectId)

暂停音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 resumePlayMusic 替代之。

enableCustomVideoCapture

enableCustomVideoCapture

void enableCustomVideoCapture (boolean enable)

启用视频自定义采集模式

@deprecated v8.5 版本开始不推荐使用，建议使用 enableCustomVideoCapture(streamType, enable) 接口替代之。

sendCustomVideoData

sendCustomVideoData

void sendCustomVideoData (TRTCCloudDef.TRTCVideoFrame frame)

投送自己采集的视频数据

@deprecated v8.5 版本开始不推荐使用，建议使用 sendCustomVideoData(streamType, TRTCVideoFrame) 接口替代之。

muteLocalVideo

muteLocalVideo

void muteLocalVideo	(boolean mute)

暂停/恢复发布本地的视频流

@deprecated v8.9 版本开始不推荐使用，建议使用 muteLocalVideo(streamType, mute) 接口替代之。

muteRemoteVideoStream

muteRemoteVideoStream

void muteRemoteVideoStream	(String userId boolean mute)

暂停 / 恢复订阅远端用户的视频流

@deprecated v8.9 版本开始不推荐使用，建议使用 muteRemoteVideoStream(userId, streamType, mute) 接口替代之。

snapshotVideo

snapshotVideo

void snapshotVideo	(String userId int streamType TRTCCloudListener.TRTCSnapshotListener listener)

视频画面截图

@deprecated v11.0 版本开始不推荐使用，建议使用 snapshotVideo(userId, streamType, sourceType, listener) 接口替代之。

startSpeedTest

startSpeedTest

```
void startSpeedTest (int sdkAppId
                     String userId
                     String userSig)
```

开始进行网络测速（进入房间前使用）

@deprecated v9.2 版本开始不推荐使用，建议使用 startSpeedTest(params) 接口替代之。

startScreenCapture

startScreenCapture

```
void startScreenCapture (TRTCCloudDef.TRTCVideoEncParam encParams
                         TRTCCloudDef.TRTCScreenShareParams shareParams)
```

启动屏幕分享

@deprecated v7.2 版本开始不推荐使用，建议使用 startScreenCapture 替代之。

setVideoEncoderRotation

setVideoEncoderRotation

```
void setVideoEncoderRotation (int rotation)
```

设置视频编码器输出的画面方向

@deprecated v11.7 版本开始不推荐使用。

setVideoEncoderMirror

setVideoEncoderMirror

void setVideoEncoderMirror (boolean mirror)

设置编码器输出的画面镜像模式

@deprecated v11.7 版本开始不推荐使用。

setGSensorMode

setGSensorMode

void setGSensorMode (int mode)

设置重力感应的适配模式

@deprecated v11.7 版本开始不推荐使用。建议使用 setGravitySensorAdaptiveMode 接口替代之。

错误码表

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTC 错误码表

Function: 用于通知客户 TRTC 在使用过程中出现的警告和错误

错误码表

枚举类型

枚举类型	描述
TXLiteAVError	错误码
TXLiteAVWarning	警告码

TXLiteAVError

TXLiteAVError

错误码

枚举	取值	描述
ERR_NULL	0	无错误
ERR_FAILED	-1	暂未归类的通用错误
ERR_INVALID_PARAMETER	-2	调用 API 时，传入的参数不合法
ERR_REFUSED	-3	API 调用被拒绝
ERR_NOT_SUPPORTED	-4	当前 API 不支持调用

ERR_INVALID_LICENSE	-5	license 不合法，调用失败
ERR_REQUEST_SERVER_TIMEOUT	-6	请求服务器超时
ERR_SERVER_PROCESS_FAILED	-7	服务器无法处理您的请求
ERR_DISCONNECTED	-8	断开连接
ERR_CAMERA_START_FAIL	-1301	打开摄像头失败，例如在 Windows 或 Mac 设备，摄像头的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_CAMERA_NOT_AUTHORIZED	-1314	摄像头设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_CAMERA_SET_PARAM_FAIL	-1315	摄像头参数设置出错（参数不支持或其它）
ERR_CAMERA_OCCUPY	-1316	摄像头正在被占用中，可尝试打开其他摄像头
ERR_SCREEN_CAPTURE_START_FAIL	-1308	开始录屏失败，如果在移动设备出现，可能是权限被用户拒绝了，如果在 Windows 或 Mac 系统的设备出现，请检查录屏接口的参数是否符合要求
ERR_SCREEN_CAPTURE_UNSURPORT	-1309	录屏失败，在 Android 平台，需要 5.0以上的系统，在 iOS 平台，需要 11.0以上的系统
ERR_SCREEN_CAPTURE_STOPPED	-7001	录屏被系统中止
ERR_SCREEN_SHARE_NOT_AUTHORIZED	-102015	没有权限上行辅路
ERR_SCREEN_SHRAE_OCCUPIED_BY_OTHER	-102016	其他用户正在上行辅路
ERR_VIDEO_ENCODE_FAIL	-1303	视频帧编码失败，例如 iOS 设备切换到其他应用时，硬编码器可能被系统释放，再切换回来时，硬编码器重启前，可能会抛出

ERR_UNSUPPORTED_RESOLUTION	-1305	不支持的视频分辨率
ERR_PIXEL_FORMAT_UNSUPPORTED	-1327	自定视频采集：设置的 pixel format 不支持
ERR_BUFFER_TYPE_UNSUPPORTED	-1328	自定视频采集：设置的 buffer type 不支持
ERR_NO_AVAILABLE_HEVC_DECODERS	-2304	找不到可用的 HEVC 解码器
ERR_MIC_START_FAIL	-1302	打开麦克风失败，例如在 Windows 或 Mac 设备，麦克风的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_MIC_NOT_AUTHORIZED	-1317	麦克风设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_MIC_SET_PARAM_FAIL	-1318	麦克风设置参数失败
ERR_MIC_OCCUPY	-1319	麦克风正在被占用中，例如移动设备正在通话时，打开麦克风会失败
ERR_MIC_STOP_FAIL	-1320	停止麦克风失败
ERR_SPEAKER_START_FAIL	-1321	打开扬声器失败，例如在 Windows 或 Mac 设备，扬声器的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_SPEAKER_SET_PARAM_FAIL	-1322	扬声器设置参数失败
ERR_SPEAKER_STOP_FAIL	-1323	停止扬声器失败
ERR_AUDIO_PLUGIN_START_FAIL	-1330	开启系统声音录制失败，例如音频驱动插件不可用
ERR_AUDIO_PLUGIN_INSTALL_NOT_AUTHORIZED	-1331	安装音频驱动插件未授权

ERR_AUDIO_PLUGIN_INSTALL_FAILED	-1332	安装音频驱动插件失败
ERR_AUDIO_PLUGIN_INSTALLED_BUT_NEED_RESTART	-1333	安装虚拟声卡插件成功，但首次安装后功能暂时不可用，此为 Mac 系统限制，请在收到此错误码后提示用户重启当前 APP
ERR_AUDIO_ENCODE_FAIL	-1304	音频帧编码失败，例如传入自定义音频数据，SDK 无法处理
ERR_UNSUPPORTED_SAMPLERATE	-1306	不支持的音频采样率
ERR_TRTC_ENTER_ROOM_FAILED	-3301	进入房间失败，请查看 onError 中的 -3301 对应的 msg 提示确认失败原因
ERR_TRTC_REQUEST_IP_TIMEOUT	-3307	请求 IP 和 sig 超时，请检查网络是否正常，或网络防火墙是否放行 UDP。可尝试访问下列 IP：162.14.22.165:8000 162.14.6.105:8000 和域名：default-query.trtc.tencent-cloud.com:8000
ERR_TRTC_CONNECT_SERVER_TIMEOUT	-3308	请求进房超时，请检查是否断网或者是否开启vpn，您也可以切换4G进行测试确认
ERR_TRTC_ROOM_PARAM_NULL	-3316	进房参数为空，请检查：enterRoom:appScene: 接口调用是否传入有效的 param
ERR_TRTC_INVALID_SDK_APPID	-3317	进房参数 sdkAppId 错误，请检查 TRTCParams.sdkAppId 是否为空
ERR_TRTC_INVALID_ROOM_ID	-3318	进房参数 roomId 错误，请检查 TRTCParams.roomId 或 TRTCParams.strRoomId 是否为空，注意 roomId 和 strRoomId 不可混用
ERR_TRTC_INVALID_USER_ID	-3319	进房参数 userId 不正确，请检查 TRTCParams.userId 是否为空

ERR_TRTC_INVALID_USER_SIG	-3320	进房参数 userSig 不正确，请检查 TRTCParams.userSig 是否为空
ERR_TRTC_ENTER_ROOM_REFUSED	-3340	进房请求被拒绝，请检查是否连续调用 enterRoom 进入相同 Id 的房间
ERR_TRTC_INVALID_PRIVATE_MAPKEY	-100006	您开启了高级权限控制，但参数 TRTCParams.privateMapKey 校验失败， 您可参考 高级权限控制 进行检查
ERR_TRTC_SERVICE_SUSPENDED	-100013	服务不可用。请检查：套餐包剩余分钟数是否大于0，腾讯云账号是否欠费。 您可参考 套餐包管理 进行查看与配置
ERR_TRTC_USER_SIG_CHECK_FAILED	-100018	UserSig 校验失败，请检查参数 TRTCParams.userSig 是否填写正确，或是否已经过期。 您可参考 UserSig 生成与校验 进行校验
ERR_TRTC_PUSH_THIRD_PARTY_CLOUD_TIMEOUT	-3321	旁路转推请求超时
ERR_TRTC_MIX_TRANSCODING_TIMEOUT	-3322	云端混流请求超时
ERR_TRTC_PUSH_THIRD_PARTY_CLOUD_FAILED	-3323	旁路转推回包异常
ERR_TRTC_MIX_TRANSCODING_FAILED	-3324	云端混流回包异常
ERR_TRTC_START_PUBLISHING_TIMEOUT	-3333	开始向腾讯云的直播 CDN 推流信令超时
ERR_TRTC_START_PUBLISHING_FAILED	-3334	开始向腾讯云的直播 CDN 推流信令异常
ERR_TRTC_STOP_PUBLISHING_TIMEOUT	-3335	停止向腾讯云的直播 CDN 推流信令超时
ERR_TRTC_STOP_PUBLISHING_FAILED	-3336	停止向腾讯云的直播 CDN 推流信令异常
ERR_TRTC_CONNECT_OTHER_ROOM	-33	请求连麦超时

_TIMEOUT	26	
ERR_TRTC_DISCONNECT_OTHER_ROOM_TIMEOUT	-3327	请求退出连麦超时
ERR_TRTC_CONNECT_OTHER_ROOM_INVALID_PARAMETER	-3328	无效参数
ERR_TRTC_CONNECT_OTHER_ROOM_AS_AUDIENCE	-3330	当前是观众角色，不能请求或断开跨房连麦，需要先 `switchRole` 到主播
ERR_BGM_OPEN_FAILED	-4001	打开音频文件失败
ERR_BGM_DECODE_FAILED	-4002	音频文件解码失败
ERR_BGM_OVER_LIMIT	-4003	数量超过限定值，如同时预加载两首背景音乐
ERR_BGM_INVALID_OPERATION	-4004	无效操作，如开始播放后再调用预加载操作

TXLiteAVWarning

TXLiteAVWarning

警告码

枚举	取值	描述
WARNING_HW_ENCODER_START_FAIL	1103	硬编码启动出现问题，自动切换到软编码
WARNING_CURRENT_ENCODE_TYPE_CHANGED	1104	表示编码器发生改变，可以通过 onWarning 函数的扩展信息中的相关字段来获取当前的编码格式和类型。 type 字段值为 0 代表 H.264 编码，1 代表 H.265 编码 hardware 字段值为 0 代表软件编码，1 代表硬件编码 stream 字段值为 0 代表大流，1 代表小流，2 代表辅流

WARNING_VIDEO_ENCODER_SW_TO_HW	1107	当前 CPU 使用率太高，无法满足软件编码需求，自动切换到硬件编码
WARNING_INSUFFICIENT_CAPTURE_FPS	1108	摄像头采集帧率不足，部分自带美颜算法的 Android 手机上会出现
WARNING_SW_ENCODER_START_FAIL	1109	软编码启动失败
WARNING_REDUCE_CAPTURE_RESOLUTION	1110	摄像头采集分辨率被降低，以满足当前帧率和性能最优解。
WARNING_CAMERA_DEVICE_EMPTY	1111	没有检测到可用的摄像头设备
WARNING_CAMERA_NOT_AUTHORIZED	1112	用户未授权当前应用使用摄像头
WARNING_OUT_OF_MEMORY	1113	内存不足，部分功能可能不正常。
WARNING_CAMERA_IS_OCCUPIED	1114	摄像头被占用.
WARNING_CAMERA_DEVICE_ERROR	1115	摄像头设备异常.
WARNING_CAMERA_DISCONNECTED	1116	摄像头无法连接.
WARNING_CAMERA_START_FAILED	1117	摄像头启动失败.
WARNING_CAMERA_SERVER_DIED	1118	系统异常.
WARNING_SCREEN_CAPTURE_NOT_AUTHORIZED	1206	用户未授权当前应用使用屏幕录制
WARNING_CURRENT_DECODE_TYPE_CHANGED	2008	表示解码器发生改变，可以通过 onWarning 函数的扩展信息中的 type 字段来获取当前的解码格式。其中 1 代表 265 解码，0 代表 264 解码。注意 Windows 端不支持此错误码的扩展信息。
WARNING_VIDEO_FRAME_DECODE_F	210	当前视频帧解码失败

AIL	1	
WARNING_HW_DECODER_START_FAIL	2106	硬解启动失败，采用软解码
WARNING_VIDEO_DECODER_HW_TO_SW	2108	当前流硬解第一个 I 帧失败，SDK 自动切软解
WARNING_SW_DECODER_START_FAIL	2109	软解码器启动失败
WARNING_VIDEO_RENDER_FAIL	2110	视频渲染失败
WARNING_MICROPHONE_DEVICE_EMPTY	1201	没有检测到可用的麦克风设备
WARNING_SPEAKER_DEVICE_EMPTY	1202	没有检测到可用的扬声器设备
WARNING_MICROPHONE_NOT_AUTHORIZED	1203	用户未授权当前应用使用麦克风
WARNING_MICROPHONE_DEVICE_ABNORMAL	1204	音频采集设备不可用（例如被占用或者PC判定无效设备）
WARNING_SPEAKER_DEVICE_ABNORMAL	1205	音频播放设备不可用（例如被占用或者PC判定无效设备）
WARNING_BLUETOOTH_DEVICE_CONNECT_FAIL	1207	蓝牙设备连接失败（例如其他应用通过设置通话音量占用音频通道）
WARNING_MICROPHONE_IS_OCCUPIED	1208	音频采集设备被占用
WARNING_AUDIO_FRAME_DECODE_FAIL	2102	当前音频帧解码失败
WARNING_AUDIO_RECORDING_WRITE_FAIL	7001	音频录制写入文件失败
WARNING_MICROPHONE_HOWLING_DETECTED	7002	录制音频时监测到啸叫
WARNING_IGNORE_UPSTREAM_FOR_AUDIENCE	6001	当前是观众角色，不支持发布音视频，需要先切换成主播角色
WARNING_UPSTREAM_AUDIO_AND_	600	音视频发送时间戳异常，可能引起音画

VIDEO_OUT_OF_SYNC	6	不同步问题

全平台（C++）
API 概览

最近更新时间：2024-03-06 14:51:32

API OVERVIEW

创建实例和事件回调

函数列表	描述
getTRTCShareInstance	创建 TRTCCloud 实例（单例模式）
destroyTRTCShareInstance	销毁 TRTCCloud 实例（单例模式）
addCallback	添加 TRTC 事件回调
removeCallback	移除 TRTC 事件回调

房间相关接口函数

函数列表	描述
enterRoom	进入房间
exitRoom	离开房间
switchRole	切换角色
switchRoom	切换房间
connectOtherRoom	请求跨房通话
disconnectOtherRoom	退出跨房通话
setDefaultStreamRecvMode	设置订阅模式（需要在进入房前设置才能生效）
createSubCloud	创建子房间实例（用于多房间并发观看）
destroySubCloud	销毁子房间实例

版权所有：腾讯云计算（北京）有限责任公司

updateOtherRoomForwardMode	更改跨房主播在本房间的上行能力

CDN 相关接口函数

函数列表	描述
startPublishing	开始向腾讯云直播 CDN 上发布音视频流
stopPublishing	停止向腾讯云直播 CDN 上发布音视频流
startPublishCDNStream	开始向非腾讯云 CDN 上发布音视频流
stopPublishCDNStream	停止向非腾讯云 CDN 上发布音视频流
setMixTranscodingConfig	设置云端混流的排版布局和转码参数
startPublishMediaStream	开始发布媒体流
updatePublishMediaStream	更新发布媒体流
stopPublishMediaStream	停止发布媒体流

视频相关接口函数

函数列表	描述
startLocalPreview	开启本地摄像头的预览画面（移动端）
updateLocalView	更新本地摄像头的预览画面
stopLocalPreview	停止摄像头预览
muteLocalVideo	暂停/恢复发布本地的视频流
setVideoMuteImage	设置本地画面被暂停期间的替代图片
startRemoteView	订阅远端用户的视频流，并绑定视频渲染控件
updateRemoteView	更新远端用户的视频渲染控件

stopRemoteView	停止订阅远端用户的视频流，并释放渲染控件
stopAllRemoteView	停止订阅所有远端用户的视频流，并释放全部渲染资源
muteRemoteVideoStream	暂停/恢复订阅远端用户的视频流
muteAllRemoteVideoStreams	暂停/恢复订阅所有远端用户的视频流
setVideoEncoderParam	设置视频编码器的编码参数
setNetworkQosParam	设置网络质量控制的相关参数
setLocalRenderParams	设置本地画面的渲染参数
setRemoteRenderParams	设置远端画面的渲染模式
enableSmallVideoStream	开启大小画面双路编码模式
setRemoteVideoStreamType	切换指定远端用户的大小画面
snapshotVideo	视频画面截图
setGravitySensorAdaptiveMode	设置重力感应的适配模式（11.7 及以上版本）

音频相关接口函数

函数列表	描述
startLocalAudio	开启本地音频的采集和发布
stopLocalAudio	停止本地音频的采集和发布
muteLocalAudio	暂停/恢复发布本地的音频流
muteRemoteAudio	暂停/恢复播放远端的音频流
muteAllRemoteAudio	暂停/恢复播放所有远端用户的音频流
setRemoteAudioVolume	设定某一个远端用户的声音播放音量

setAudioCaptureVolume	设定本地音频的采集音量
getAudioCaptureVolume	获取本地音频的采集音量
setAudioPlayoutVolume	设定远端音频的播放音量
getAudioPlayoutVolume	获取远端音频的播放音量
enableAudioVolumeEvaluation	启用音量大小提示
startAudioRecording	开始录音
stopAudioRecording	停止录音
startLocalRecording	开启本地媒体录制
stopLocalRecording	停止本地媒体录制
setRemoteAudioParallelParams	设置远端音频流智能并发播放策略
enable3DSpatialAudioEffect	启用 3D 音效
updateSelf3DSpatialPosition	设置 3D 音效中自身坐标及朝向信息
updateRemote3DSpatialPosition	设置 3D 音效中远端用户坐标信息
set3DSpatialReceivingRange	设置指定用户所发出声音的可被接收范围

设备管理相关接口

函数列表	描述
getDeviceManager	获取设备管理类（TXDeviceManager）

美颜特效和图像水印

函数列表	描述
setBeautyStyle	设置美颜、美白、红润等特效
setWaterMark	添加水印

背景音乐和声音特效

函数列表	描述
getAudioEffectManager	获取音效管理类（TXAudioEffectManager）
startSystemAudioLoopback	开启系统声音采集（iOS 端暂未支持）
stopSystemAudioLoopback	停止系统声音采集（iOS 端暂未支持）
setSystemAudioLoopbackVolume	设置系统声音的采集音量

屏幕分享相关接口

函数列表	描述
startScreenCapture	启动屏幕分享
stopScreenCapture	停止屏幕分享
pauseScreenCapture	暂停屏幕分享
resumeScreenCapture	恢复屏幕分享
getScreenCaptureSources	枚举可分享的屏幕和窗口（该接口仅支持桌面系统）
selectScreenCaptureTarget	选取要分享的屏幕或窗口（该接口仅支持桌面系统）
setSubStreamEncoderParam	设置屏幕分享（即辅路）的视频编码参数（桌面系统和移动系统均已支持）
setSubStreamMixVolume	设置屏幕分享时的混音音量大小（该接口仅支持桌面系统）

addExcludedShareWindow	将指定窗口加入屏幕分享的排除列表中（该接口仅支持桌面系统）
removeExcludedShareWindow	将指定窗口从屏幕分享的排除列表中移除（该接口仅支持桌面系统）
removeAllExcludedShareWindow	将所有窗口从屏幕分享的排除列表中移除（该接口仅支持桌面系统）
addIncludedShareWindow	将指定窗口加入屏幕分享的包含列表中（该接口仅支持桌面系统）
removeIncludedShareWindow	将指定窗口从屏幕分享的包含列表中移除（该接口仅支持桌面系统）
removeAllIncludedShareWindow	将全部窗口从屏幕分享的包含列表中移除（该接口仅支持桌面系统）

自定义采集和自定义渲染

函数列表	描述
enableCustomVideoCapture	启用/关闭视频自定义采集模式
sendCustomVideoData	向 SDK 投送自己采集的视频帧
enableCustomAudioCapture	启用音频自定义采集模式
sendCustomAudioData	向 SDK 投送自己采集的音频数据
enableMixExternalAudioFrame	启用/关闭自定义音轨
mixExternalAudioFrame	向 SDK 混入自定义音轨
setMixExternalAudioVolume	设置推流时混入外部音频的推流音量和播放音量
generateCustomPTS	生成自定义采集时的时间戳
enableLocalVideoCustomProcess	.1 开启视频第三方美颜
setLocalVideoCustomProcessCallback	.2 设置第三方美颜的视频数据回调
setLocalVideoRenderCallback	设置本地视频自定义渲染回调
setRemoteVideoRenderCallback	设置远端视频自定义渲染回调

setAudioFrameCallback	设置音频数据自定义回调
setCapturedAudioFrameCallbackFormat	设置本地麦克风采集出的音频帧回调格式
setLocalProcessedAudioFrameCallbackFormat	设置经过前处理后的本地音频帧回调格式
setMixedPlayAudioFrameCallbackFormat	设置最终要由系统播放出的音频帧回调格式
enableCustomAudioRendering	开启音频自定义播放
getCustomAudioRenderingFrame	获取可播放的音频数据

自定义消息发送接口

函数列表	描述
sendCustomCmdMsg	使用 UDP 通道发送自定义消息给房间内所有用户
sendSEIMsg	使用 SEI 通道发送自定义消息给房间内所有用户

网络测试接口

函数列表	描述
startSpeedTest	开始进行网速测试（进入房间前使用）
stopSpeedTest	停止网络测速

调试相关接口

函数列表	描述
getSDKVersion	获取 SDK 版本信息
setLogLevel	设置 Log 输出级别
setConsoleEnabled	启用/禁用控制台日志打印
setLogCompressEnabled	启用/禁用日志的本地压缩

setLogDirPath	设置本地日志的保存路径
setLogCallback	设置日志回调
showDebugView	显示仪表盘
callExperimentalAPI	调用实验性接口

加密接口

函数列表	描述
enablePayloadPrivateEncryption	开启或关闭媒体流私有加密

错误和警告事件

函数列表	描述
onError	错误事件回调
onWarning	警告事件回调

房间相关事件回调

函数列表	描述
onEnterRoom	进入房间成功与否的事件回调
onExitRoom	离开房间的事件回调
onSwitchRole	切换角色的事件回调
onSwitchRoom	切换房间的结果回调
onConnectOtherRoom	请求跨房通话的结果回调
onDisconnectOtherRoom	结束跨房通话的结果回调
onUpdateOtherRoomForwardMode	更改跨房主播上行能力的结果回调

用户相关事件回调

函数列表	描述
onRemoteUserEnterRoom	有用户加入当前房间
onRemoteUserLeaveRoom	有用户离开当前房间
onUserVideoAvailable	某远端用户发布/取消了主路视频画面
onUserSubStreamAvailable	某远端用户发布/取消了辅路视频画面
onUserAudioAvailable	某远端用户发布/取消了自己的音频
onFirstVideoFrame	SDK 开始渲染自己本地或远端用户的首帧画面
onFirstAudioFrame	SDK 开始播放远端用户的首帧音频
onSendFirstLocalVideoFrame	自己本地的首个视频帧已被发布出去
onSendFirstLocalAudioFrame	自己本地的首个音频帧已被发布出去
onRemoteVideoStatusUpdated	远端视频状态变化的事件回调
onRemoteAudioStatusUpdated	远端音频状态变化的事件回调
onUserVideoSizeChanged	用户视频大小发生改变回调

网络和技术指标统计回调

函数列表	描述
onNetworkQuality	网络质量的实时统计回调
onStatistics	音视频技术指标的实时统计回调

onSpeedTestResult	网速测试的结果回调

与云端连接情况的事件回调

函数列表	描述
onConnectionLost	SDK 与云端的连接已经断开
onTryToReconnect	SDK 正在尝试重新连接到云端
onConnectionRecovery	SDK 与云端的连接已经恢复

硬件设备相关事件回调

函数列表	描述
onCameraDidReady	摄像头准备就绪
onMicDidReady	麦克风准备就绪
onUserVoiceVolume	音量大小的反馈回调
onDeviceChange	本地设备的通断状态发生变化（仅适用于桌面系统）
onAudioDeviceCaptureVolumeChanged	当前麦克风的系统采集音量发生变化
onAudioDevicePlayoutVolumeChanged	当前系统的播放音量发生变化
onSystemAudioLoopbackError	系统声音采集是否被成功开启的事件回调（仅适用于 Mac 系统）
onTestMicVolume	测试麦克风时的音量回调
onTestSpeakerVolume	测试扬声器时的音量回调

自定义消息的接收事件回调

函数列表	描述
onRecvCustomCmdMsg	收到自定义消息的事件回调

onMissCustomCmdMsg	自定义消息丢失的事件回调
onRecvSEIMsg	收到 SEI 消息的回调

CDN 相关事件回调

函数列表	描述
onStartPublishing	开始向腾讯云直播 CDN 上发布音视频流的事件回调
onStopPublishing	停止向腾讯云直播 CDN 上发布音视频流的事件回调
onStartPublishCDNStream	开始向非腾讯云 CDN 上发布音视频流的事件回调
onStopPublishCDNStream	停止向非腾讯云 CDN 上发布音视频流的事件回调
onSetMixTranscodingConfig	设置云端混流的排版布局和转码参数的事件回调
onStartPublishMediaStream	开始发布媒体流的事件回调
onUpdatePublishMediaStream	更新媒体流的事件回调
onStopPublishMediaStream	停止媒体流的事件回调
onCdnStreamStateChanged	RTMP/RTMPS 推流状态发生改变回调

屏幕分享相关事件回调

函数列表	描述
onScreenCaptureStarted	屏幕分享开启的事件回调
onScreenCapturePaused	屏幕分享暂停的事件回调
onScreenCapture	屏幕分享恢复的事件回调

Resumed	
onScreenCaptureStoped	屏幕分享停止的事件回调
onScreenCaptureCovered	屏幕分享的目标窗口被遮挡的事件回调（仅适用于 Windows 操作系统）

本地录制和本地截图的事件回调

函数列表	描述
onLocalRecordBegin	本地录制任务已经开始的事件回调
onLocalRecording	本地录制任务正在进行中的进展事件回调
onLocalRecordFragment	本地录制分片的事件回调
onLocalRecordComplete	本地录制任务已经结束的事件回调
onSnapshotComplete	本地截图完成的事件回调

废弃的事件回调

函数列表	描述
onUserEnter	有主播加入当前房间（已废弃）
onUserExit	有主播离开当前房间（已废弃）
onAudioEffectFinished	音效播放已结束（已废弃）
onPlayBGMBegin	开始播放背景音乐（已废弃）
onPlayBGMProgress	背景音乐的播放进度回调（已废弃）
onPlayBGMComplete	背景音乐播放已经结束（已废弃）
onSpeedTest	服务器测速的结果回调（已废弃）

视频数据自定义回调

函数列表	描述
onRenderVideoFrame	自定义视频渲染回调
onGLContextCreated	SDK 内部 OpenGL 环境已经创建的通知
onProcessVideoFrame	用于对接第三方美颜组件的视频处理回调
onGLContextDestroy	SDK 内部 OpenGL 环境被销毁的通知

音频数据自定义回调

函数列表	描述
onCapturedAudioFrame	本地采集并经过音频模块前处理后的音频数据回调
onLocalProcessedAudioFrame	本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调
onPlayAudioFrame	混音前的每一路远程用户的音频数据
onMixedPlayAudioFrame	将各路待播放音频混合之后并在最终提交系统播放之前的数据回调

更多事件回调接口

函数列表	描述
onLog	本地 LOG 的打印回调

背景音乐预加载事件回调

函数列表	描述
onLoadProgress	背景音乐预加载进度

onLoadError	背景音乐预加载出错

背景音乐的播放事件回调

函数列表	描述
onStart	背景音乐开始播放
onPlayProgress	背景音乐的播放进度
onComplete	背景音乐已经播放完毕

人声相关的特效接口

函数列表	描述
enableVoiceEarMonitor	开启耳返
setVoiceEarMonitorVolume	设置耳返音量
setVoiceReverbType	设置人声的混响效果
setVoiceChangerType	设置人声的变声特效
setVoiceCaptureVolume	设置语音音量
setVoicePitch	设置语音音调

背景音乐的相关接口

函数列表	描述
setMusicObserver	设置背景音乐的事件回调接口
startPlayMusic	开始播放背景音乐
stopPlayMusic	停止播放背景音乐
pausePlayMusic	暂停播放背景音乐

resumePlayMusic	恢复播放背景音乐
setAllMusicVolume	设置所有背景音乐的本地音量和远端音量的大小
setMusicPublishVolume	设置某一首背景音乐的远端音量的大小
setMusicPlayoutVolume	设置某一首背景音乐的本地音量的大小
setMusicPitch	调整背景音乐的音调高低
setMusicSpeedRate	调整背景音乐的变速效果
getMusicCurrentPosInMS	获取背景音乐的播放进度（单位：毫秒）
getMusicDurationInMS	获取背景音乐的总时长（单位：毫秒）
seekMusicToPosInTime	设置背景音乐的播放进度（单位：毫秒）
setMusicScratchSpeedRate	调整搓碟的变速效果
setPreloadObserver	设置预加载事件回调
preloadMusic	预加载背景音乐
getMusicTrackCount	获取背景音乐的音轨数量
setMusicTrack	指定背景音乐的播放音轨

设备操作接口

函数列表	描述
isFrontCamera	判断当前是否为前置摄像头（仅适用于移动端）
switchCamera	切换前置或后置摄像头（仅适用于移动端）
getCameraZoomMaxRatio	获取摄像头的最大缩放倍数（仅适用于移动端）
setCameraZoomRatio	设置摄像头的缩放倍数（仅适用于移动端）
isAutoFocusEnabled	查询是否支持自动识别人脸位置（仅适用于移动端）

enableCameraAutoFocus	开启自动对焦功能（仅适用于移动端）
setCameraFocusPosition	设置摄像头的对焦位置（仅适用于移动端）
enableCameraTorch	开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）
setAudioRoute	设置音频路由（仅适用于移动端）
getDevicesList	获取设备列表（仅适用于桌面端）
setCurrentDevice	设置当前要使用的设备（仅适用于桌面端）
getCurrentDevice	获取当前正在使用的设备（仅适用于桌面端）
setCurrentDeviceVolume	设置当前设备的音量（仅适用于桌面端）
getCurrentDeviceVolume	获取当前设备的音量（仅适用于桌面端）
setCurrentDeviceMute	设置当前设备的静音状态（仅适用于桌面端）
getCurrentDeviceMute	获取当前设备的静音状态（仅适用于桌面端）
enableFollowingDefaultAudioDevice	设置 SDK 使用的音频设备根据跟随系统默认设备（仅适用于桌面端）
startCameraDeviceTest	开始摄像头测试（仅适用于桌面端）
stopCameraDeviceTest	结束摄像头测试（仅适用于桌面端）
startMicDeviceTest	开始麦克风测试（仅适用于桌面端）
stopMicDeviceTest	结束麦克风测试（仅适用于桌面端）
startSpeakerDeviceTest	开始扬声器测试（仅适用于桌面端）
stopSpeakerDeviceTest	结束扬声器测试（仅适用于桌面端）
setApplicationPlayVolume	设置 Windows 系统音量合成器中当前进程的音量（仅适用于 Windows 系统）
getApplicationPlayVolume	获取 Windows 系统音量合成器中当前进程的音量（仅适用于 Windows 系统）
setApplicationMuteState	设置 Windows 系统音量合成器中当前进程的静音状态（仅适用于 Windows 系统）
getApplicationMuteState	获取 Windows 系统音量合成器中当前进程的静音状态（仅适用于 Windows 系统）

setCameraCapturerParam	设置摄像头采集偏好
setDeviceObserver	设置 onDeviceChanged 事件回调

弃用接口

函数列表	描述
setSystemVolumeType	设置系统音量类型（仅适用于移动端）

废弃接口

函数列表	描述
enableAudioVolumeEvaluation	启用音量大小提示
startLocalAudio	设置音频质量
startRemoteView	开始显示远端视频画面
stopRemoteView	停止显示远端视频画面，同时不再拉取该远端用户的视频数据流
setLocalViewFillMode	设置本地图像的渲染模式
setLocalViewRotation	设置本地图像的顺时针旋转角度
setLocalViewMirror	设置本地摄像头预览画面的镜像模式
setRemoteViewFillMode	设置远端图像的渲染模式
setRemoteViewRotation	设置远端图像的顺时针旋转角度
startRemoteSubStreamView	开始显示远端用户的辅路画面
stopRemoteSubStreamView	停止显示远端用户的辅路画面
setRemoteSubStreamViewFillMode	设置辅路画面的填充模式
setRemoteSubStreamViewRotation	设置辅路画面的顺时针旋转角度
setAudioQuality	设置音频质量

版权所有：腾讯云计算（北京）有限责任公司

setPriorRemoteVideoStreamType	设定优先观看大画面还是小画面
setMicVolumeOnMixing	设置麦克风音量大小
playBGM	启动播放背景音乐
stopBGM	停止播放背景音乐
pauseBGM	停止播放背景音乐
resumeBGM	停止播放背景音乐
getBGMDuration	获取背景音乐总时长（单位：毫秒）
setBGMPosition	设置背景音乐的播放进度
setBGMVolume	设置背景音乐的音量大小
setBGMPlayoutVolume	设置背景音乐的本地播放音量
setBGMPublishVolume	设置背景音乐的远端播放音量
playAudioEffect	播放音效
setAudioEffectVolume	设置音效音量
stopAudioEffect	停止播放音效
stopAllAudioEffects	停止所有音效
setAllAudioEffectsVolume	设置所有音效音量
pauseAudioEffect	暂停音效
resumeAudioEffect	暂停音效
enableCustomVideoCapture	启用视频自定义采集模式
sendCustomVideoData	投送自己采集的视频数据
muteLocalVideo	暂停/恢复发布本地的视频流
muteRemoteVideoStream	暂停 / 恢复订阅远端用户的视频流
startSpeedTest	开始进行网络测速（进入房间前使用）

版权所有：腾讯云计算（北京）有限责任公司

startScreenCapture	启动屏幕分享
setLocalVideoProcessCallback	设置第三方美颜的视频数据回调
getCameraDevicesList	获取摄像头设备列表
setCurrentCameraDevice	选定当前要使用的摄像头
getCurrentCameraDevice	获取当前使用的摄像头
getMicDevicesList	获取麦克风设备列表
getCurrentMicDevice	获取当前的麦克风设备
setCurrentMicDevice	选定当前使用的麦克风
getCurrentMicDeviceVolume	获取当前麦克风的设备音量
setCurrentMicDeviceVolume	设置当前麦克风的设备音量
setCurrentMicDeviceMute	设置系统当前麦克风设备的静音状态
getCurrentMicDeviceMute	获取系统当前麦克风设备是否静音
getSpeakerDevicesList	获取扬声器设备列表
getCurrentSpeakerDevice	获取当前的扬声器设备
setCurrentSpeakerDevice	设置要使用的扬声器
getCurrentSpeakerVolume	获取当前扬声器的设备音量
setCurrentSpeakerVolume	设置当前扬声器的设备音量
getCurrentSpeakerDeviceMute	获取系统当前扬声器设备是否静音
setCurrentSpeakerDevic	设置系统当前扬声器设备的静音状态

版权所有：腾讯云计算（北京）有限责任公司

eMute	
startCameraDeviceTest	开始进行摄像头测试
stopCameraDeviceTest	停止进行摄像头测试
startMicDeviceTest	开始进行麦克风测试
stopMicDeviceTest	开始进行麦克风测试
startSpeakerDeviceTest	开始进行扬声器测试
stopSpeakerDeviceTest	停止进行扬声器测试
selectScreenCaptureTarget	开始应用内的屏幕分享（iOS）
setVideoEncoderRotation	设置视频编码器输出的画面方向
setVideoEncoderMirror	设置编码器输出的画面镜像模式

ITRTCCloud

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTCCloud @ TXLiteAVSDK
Function: 腾讯云 TRTC 主功能接口
Version: 11.7

ITRTCCloud

ITRTCCloud

函数列表	描述
getTRTCShareInstance	创建 TRTCCloud 实例（单例模式）
destroyTRTCShareInstance	销毁 TRTCCloud 实例（单例模式）
addCallback	添加 TRTC 事件回调
removeCallback	移除 TRTC 事件回调
enterRoom	进入房间
exitRoom	离开房间
switchRole	切换角色
switchRole	切换角色（支持设置权限位）
switchRoom	切换房间
connectOtherRoom	请求跨房通话
disconnectOtherRoom	退出跨房通话
setDefaultStreamRecvMode	设置订阅模式（需要在进入房前设置才能生效）
createSubCloud	创建子房间实例（用于多房间并发观看）
destroySubCloud	销毁子房间实例

updateOtherRoomForwardMode	更改跨房主播在本房间的上行能力
startPublishing	开始向腾讯云直播 CDN 上发布音视频流
stopPublishing	停止向腾讯云直播 CDN 上发布音视频流
startPublishCDNStream	开始向非腾讯云 CDN 上发布音视频流
stopPublishCDNStream	停止向非腾讯云 CDN 上发布音视频流
setMixTranscodingConfig	设置云端混流的排版布局和转码参数
startPublishMediaStream	开始发布媒体流
updatePublishMediaStream	更新发布媒体流
stopPublishMediaStream	停止发布媒体流
startLocalPreview	开启本地摄像头的预览画面（移动端）
startLocalPreview	开启本地摄像头的预览画面（桌面端）
updateLocalView	更新本地摄像头的预览画面
stopLocalPreview	停止摄像头预览
muteLocalVideo	暂停/恢复发布本地的视频流
setVideoMuteImage	设置本地画面被暂停期间的替代图片
startRemoteView	订阅远端用户的视频流，并绑定视频渲染控件
updateRemoteView	更新远端用户的视频渲染控件
stopRemoteView	停止订阅远端用户的视频流，并释放渲染控件
stopAllRemoteView	停止订阅所有远端用户的视频流，并释放全部渲染资源
muteRemoteVideoStream	暂停/恢复订阅远端用户的视频流
muteAllRemoteVideoStreams	暂停/恢复订阅所有远端用户的视频流
setVideoEncoderParam	设置视频编码器的编码参数
setNetworkQosParam	设置网络质量控制的相关参数
setLocalRenderParams	设置本地画面的渲染参数
setRemoteRenderParams	设置远端画面的渲染模式

版权所有：腾讯云计算（北京）有限责任公司

enableSmallVideoStream	开启大小画面双路编码模式
setRemoteVideoStreamType	切换指定远端用户的大小画面
snapshotVideo	视频画面截图
setGravitySensorAdaptiveMode	设置重力感应的适配模式（11.7 及以上版本）
startLocalAudio	开启本地音频的采集和发布
stopLocalAudio	停止本地音频的采集和发布
muteLocalAudio	暂停/恢复发布本地的音频流
muteRemoteAudio	暂停/恢复播放远端的音频流
muteAllRemoteAudio	暂停/恢复播放所有远端用户的音频流
setRemoteAudioVolume	设定某一个远端用户的声音播放音量
setAudioCaptureVolume	设定本地音频的采集音量
getAudioCaptureVolume	获取本地音频的采集音量
setAudioPlayoutVolume	设定远端音频的播放音量
getAudioPlayoutVolume	获取远端音频的播放音量
enableAudioVolumeEvaluation	启用音量大小提示
startAudioRecording	开始录音
stopAudioRecording	停止录音
startLocalRecording	开启本地媒体录制
stopLocalRecording	停止本地媒体录制
setRemoteAudioParallelParams	设置远端音频流智能并发播放策略
enable3DSpatialAudioEffect	启用 3D 音效
updateSelf3DSpatialPosition	设置 3D 音效中自身坐标及朝向信息
updateRemote3DSpatialPosition	设置 3D 音效中远端用户坐标信息
set3DSpatialReceivingRange	设置指定用户所发出声音的可被接收范围
getDeviceManager	获取设备管理类（TXDeviceManager）

setBeautyStyle	设置美颜、美白、红润等特效
setWaterMark	添加水印
getAudioEffectManager	获取音效管理类（TXAudioEffectManager）
startSystemAudioLoopback	开启系统声音采集（iOS 端暂未支持）
stopSystemAudioLoopback	停止系统声音采集（iOS 端暂未支持）
setSystemAudioLoopbackVolume	设置系统声音的采集音量
startScreenCapture	启动屏幕分享
stopScreenCapture	停止屏幕分享
pauseScreenCapture	暂停屏幕分享
resumeScreenCapture	恢复屏幕分享
getScreenCaptureSources	枚举可分享的屏幕和窗口（该接口仅支持桌面系统）
selectScreenCaptureTarget	选取要分享的屏幕或窗口（该接口仅支持桌面系统）
setSubStreamEncoderParam	设置屏幕分享（即辅路）的视频编码参数（桌面系统和移动系统均已支持）
setSubStreamMixVolume	设置屏幕分享时的混音音量大小（该接口仅支持桌面系统）
addExcludedShareWindow	将指定窗口加入屏幕分享的排除列表中（该接口仅支持桌面系统）
removeExcludedShareWindow	将指定窗口从屏幕分享的排除列表中移除（该接口仅支持桌面系统）
removeAllExcludedShareWindow	将所有窗口从屏幕分享的排除列表中移除（该接口仅支持桌面系统）
addIncludedShareWindow	将指定窗口加入屏幕分享的包含列表中（该接口仅支持桌面系统）
removeIncludedShareWindow	将指定窗口从屏幕分享的包含列表中移除（该接口仅支持桌面系统）
removeAllIncludedShareWindow	将全部窗口从屏幕分享的包含列表中移除（该接口仅支持桌面系统）

enableCustomVideoCapture	启用/关闭视频自定义采集模式
sendCustomVideoData	向 SDK 投送自己采集的视频帧
enableCustomAudioCapture	启用音频自定义采集模式
sendCustomAudioData	向 SDK 投送自己采集的音频数据
enableMixExternalAudioFrame	启用/关闭自定义音轨
mixExternalAudioFrame	向 SDK 混入自定义音轨
setMixExternalAudioVolume	设置推流时混入外部音频的推流音量和播放音量
generateCustomPTS	生成自定义采集时的时间戳
enableLocalVideoCustomProcess	.1 开启视频第三方美颜
setLocalVideoCustomProcessCallback	.2 设置第三方美颜的视频数据回调
setLocalVideoRenderCallback	设置本地视频自定义渲染回调
setRemoteVideoRenderCallback	设置远端视频自定义渲染回调
setAudioFrameCallback	设置音频数据自定义回调
setCapturedAudioFrameCallbackFormat	设置本地麦克风采集出的音频帧回调格式
setLocalProcessedAudioFrameCallbackFormat	设置经过前处理后的本地音频帧回调格式
setMixedPlayAudioFrameCallbackFormat	设置最终要由系统播放出的音频帧回调格式
enableCustomAudioRendering	开启音频自定义播放
getCustomAudioRenderingFrame	获取可播放的音频数据
sendCustomCmdMsg	使用 UDP 通道发送自定义消息给房间内所有用户
sendSEIMsg	使用 SEI 通道发送自定义消息给房间内所有用户
startSpeedTest	开始进行网速测试（进入房间前使用）
stopSpeedTest	停止网络测速
getSDKVersion	获取 SDK 版本信息

版权所有：腾讯云计算（北京）有限责任公司

setLogLevel	设置 Log 输出级别
setConsoleEnabled	启用/禁用控制台日志打印
setLogCompressEnabled	启用/禁用日志的本地压缩
setLogDirPath	设置本地日志的保存路径
setLogCallback	设置日志回调
showDebugView	显示仪表盘
callExperimentalAPI	调用实验性接口
enablePayloadPrivateEncryption	开启或关闭媒体流私有加密

getTRTCShareInstance

getTRTCShareInstance

ITRTCCloud* getTRTCShareInstance	(void *context)

创建 TRTCCloud 实例（单例模式）

参数	描述
context	仅适用于 Android 平台，SDK 内部会将其转化为 Android 平台的 ApplicationContext 用于调用 Android System API。 如果传入的 context 参数为空，SDK 内部会自动获取当前进程的 ApplicationContext。

注意

1. 如果您使用 delete ITRTCCloud* 会导致编译错误，请使用 destroyTRTCCloud 释放对象指针。
2. 在 Windows、Mac 和 iOS 平台上，请调用 getTRTCShareInstance() 接口。
3. 在 Android 平台上，请调用 getTRTCShareInstance(void *context) 接口。

destroyTRTCShareInstance

destroyTRTCShareInstance

销毁 TRTCCloud 实例（单例模式）

addCallback

addCallback

void addCallback (ITRTCCloudCallback* callback)

添加 TRTC 事件回调

您可以通过 ITRTCCloudCallback 获得来自 SDK 的各类事件通知（比如：错误码，警告码，音视频状态参数等）。

removeCallback

removeCallback

void removeCallback (ITRTCCloudCallback* callback)

移除 TRTC 事件回调

enterRoom

enterRoom

void enterRoom (const TRTCParams& param

TRTCAppScene scene)

进入房间

TRTC 的所有用户都需要进入房间才能“发布”或“订阅”音视频流，“发布”是指将自己的音频和视频推送到云端，“订阅”是指从云端拉取房间里其他用户的音视频流。
调用该接口时，您需要指定您的应用场景 TRTCAppScene 以获取最佳的音视频传输体验，这些场景可以分成两大类：

实时通话：
包括 TRTCAppSceneVideoCall 和 TRTCAppSceneAudioCall 两个可选项，分别是视频通话和语音通话，该模式适合 1对1 的音视频通话，或者参会人数在 300 人以内的在线会议。

在线直播：
包括 TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom 两个可选项，分别是视频直播和语音直播，该模式适合十万人以内的在线直播场景，但需要您在接下来介绍的 TRTCParams 参数中指定 **角色(role)** 这个字段，也就是将房间中的用户区分为 **主播**（TRTCRoleAnchor）和 **观众**（TRTCRoleAudience）两种不同的角色。

调用该接口后，您会收到来自 ITRTCCloudCallback 中的 onEnterRoom(result) 回调：

- 如果进房成功，参数 result 会是一个正数（result > 0），表示从函数调用到进入房间所花费的时间，单位是毫秒（ms）。
- 如果进房失败，参数 result 会是一个负数（result < 0），表示进房失败的 错误码 。

参数	描述
param	进房参数，用于指定用户的身份、角色和安全票据等信息，详情请参见 TRTCParams 。
scene	应用场景，用于指定您的业务场景，同一个房间内的所有用户需要设定相同的 TRTCAppScene 。

注意
1. 同一个房间内的所有用户需要设定相同的 scene。不同的 scene 会导致偶现的异常问题。
2. 当您指定参数 scene 为 TRTCAppSceneLIVE 或 TRTCAppSceneVoiceChatRoom 时，您必须通过 TRTCParams 中的 “role” 字段为当前用户设定他/她在房间中的角色。
3. 请您尽量保证 enterRoom 与 exitRoom 前后配对使用，即保证”先退出前一个房间再进入下一个房间”，否则会导致很多异常问题。

exitRoom

exitRoom

离开房间

调用该接口会让用户离开自己所在的音视频房间，并释放摄像头、麦克风、扬声器等设备资源。
等资源释放完毕之后，SDK 会通过 ITRTCCloudCallback 中的 onExitRoom 回调向您通知。

如果您要再次调用 enterRoom 或者切换到其他的供应商的 SDK，建议等待 onExitRoom 回调到来之后再执行之后的操作，以避免摄像头或麦克风被占用的问题。

switchRole

switchRole

void switchRole （TRTCRoleType role）

切换角色

调用本接口可以实现用户在 主播 和 观众 两种角色之间来回切换。

由于视频直播和语音聊天室需要支持多达10万名观众同时观看，所以设定了 **只有主播才能发布自己的音视频** 的规则。因此，当有些观众希望发布自己的音视频流（以便能跟主播互动）时，就需要先把自己的角色切换成 **主播**。

您可以在进入房间时通过 TRTCParams 中的 role 字段事先确定用户的角色，也可以在进入房间后通过 switchRole 接口动态切换角色。

参数	描述
role	角色，默认为 **主播**。 • TRTCRoleAnchor ：主播，可以发布自己的音视频，同一个房间里最多支持50个主播同时发布音视频。 • TRTCRoleAudience ：观众，不能发布自己的音视频流，只能观看房间中其他主播的音视频。如果要发布自己的音视频，需要先通过 switchRole 切换成 **主播**，同一个房间内同时最多可以容纳 10 万名观众。

注意
1. 该接口仅适用于视频直播（TRTCAppSceneLIVE）和语音聊天室（TRTCAppSceneVoiceChatRoom）这两个场景。
2. 如果您在 enterRoom 时指定的 scene 为 TRTCAppSceneVideoCall 或 TRTCAppSceneAudioCall，请不要调用这个接口。

版权所有：腾讯云计算（北京）有限责任公司

switchRole

switchRole

```
void switchRole (TRTCRoleType role
                 const char* privateMapKey)
```

切换角色（支持设置权限位）

调用本接口可以实现用户在 `主播` 和 `观众` 两种角色之间来回切换。

由于视频直播和语音聊天室需要支持多达10万名观众同时观看，所以设定了“只有主播才能发布自己的音视频”的规则。
因此，当有些观众希望发布自己的音视频流（以便能跟主播互动）时，就需要先把自己的角色切换成“主播”。
您可以在进入房间时通过 TRTCParams 中的 role 字段事先确定用户的角色，也可以在进入房间后通过 switchRole 接口动态切换角色。

参数	描述
privateMapKey	用于权限控制的权限票据，当您希望某个房间只能让特定的 userId 进入或者上行视频时，需要使用 privateMapKey 进行权限保护。 • 仅建议有高级别安全需求的客户使用，更多详情请参见 开启高级权限控制 。
role	角色，默认为“主播”： • TRTCRoleAnchor ：主播，可以发布自己的音视频，同一个房间里最多支持50个主播同时发布音视频。 • TRTCRoleAudience ：观众，不能发布自己的音视频流，只能观看房间中其他主播的音视频。如果要发布自己的音视频，需要先通过 switchRole 切换成“主播”，同一个房间内同时最多可以容纳 10 万名观众。

注意

1. 该接口仅适用于视频直播（TRTCAppSceneLIVE）和语音聊天室（TRTCAppSceneVoiceChatRoom）这两个场景。
2. 如果您在 enterRoom 时指定的 scene 为 TRTCAppSceneVideoCall 或 TRTCAppSceneAudioCall，请不要调用这个接口。

switchRoom

switchRoom

void switchRoom	(const TRTCSwitchRoomConfig & config)

切换房间

使用该接口可以让用户可以快速从一个房间切换到另一个房间。

- 如果用户的身份是“观众”，该接口的调用效果等同于 exitRoom(当前房间) + enterRoom（新的房间）。
- 如果用户的身份是“主播”，该接口在切换房间的同时还会保持自己的音视频发布状态，因此在房间切换过程中，摄像头的预览和声音的采集都不会中断。

该接口适用于在线教育场景中，监课老师在多个房间中进行快速切换的场景。在该场景下使用 switchRoom 可以获得比 exitRoom+enterRoom 更好的流畅性和更少的代码量。
接口调用结果会通过 ITRTCCloudCallback 中的 onSwitchRoom(errCode, errMsg) 回调。

参数	描述
config	房间参数，详情请参见 TRTCSwitchRoomConfig 。

注意
由于对老版本 SDK 兼容的需求，参数 config 中同时包含 roomId 与 strRoomId 两个参数，这两个参数的填写格外讲究，请注意如下事项：
1. 若您选用 strRoomId，则 roomId 需要填写为0。若两者都填，将优先选用 roomId。
2. 所有房间需要同时使用 strRoomId 或同时使用 roomId，不可混用，否则将会出现很多预期之外的 bug。

connectOtherRoom

connectOtherRoom

void connectOtherRoom	(const char* param)

请求跨房通话

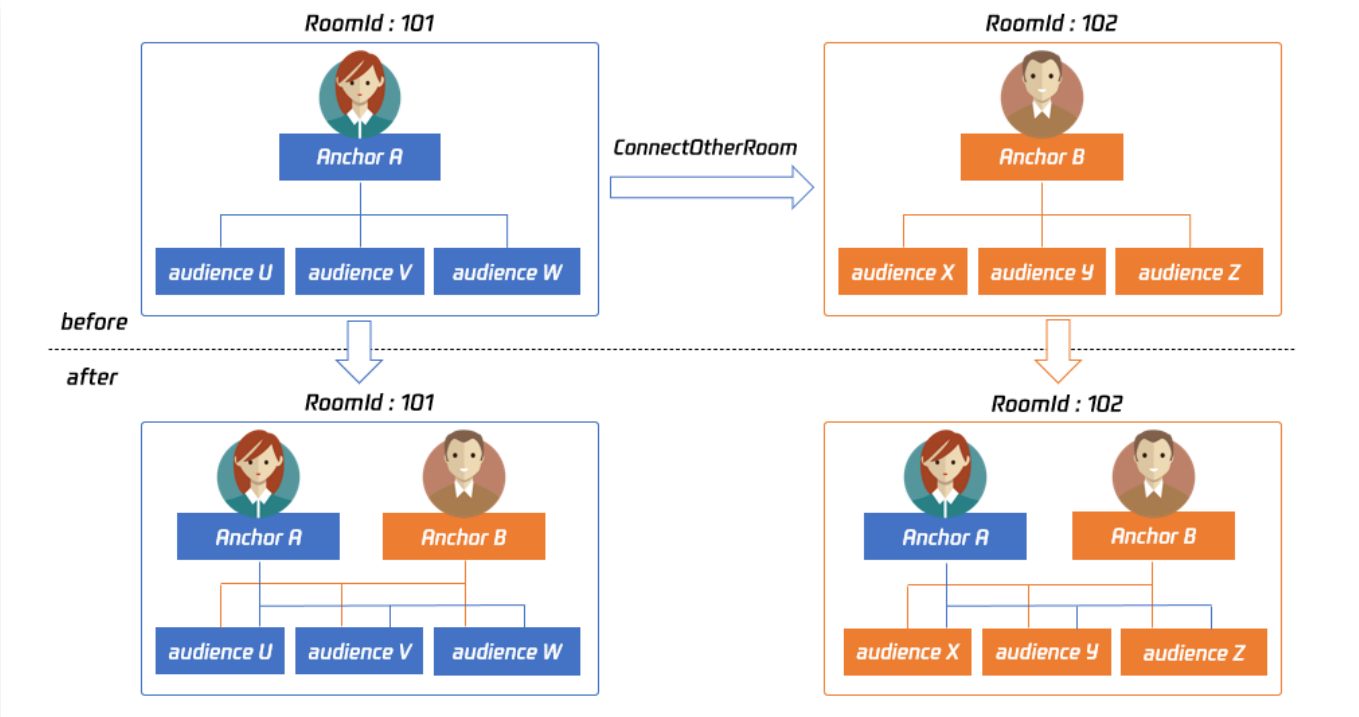
默认情况下，只有同一个房间中的用户之间可以进行音视频通话，不同的房间之间的音视频流是相互隔离的。但您可以通过调用该接口，将另一个房间中某个主播音视频流发布到自己所在的房间中，与此同时，该接口也会将自己的音视频流发布到目标主播的房间中。

也就是说，您可以使用该接口让身处两个不同房间中的主播进行跨房间的音视频流分享，从而让每个房间中的观众都能观看到这两个主播的音视频。该功能可以用来实现主播之间的 PK 功能。

跨房通话的请求结果会通过 ITRTCCloudCallback 中的 onConnectOtherRoom 回调通知给您。

例如：当房间“101”中的主播 A 通过 connectOtherRoom() 跟房间“102”中的主播 B 建立跨房通话后，

- 房间“101”中的用户都会收到主播 B 的 onRemoteUserEnterRoom(B) 和 onUserVideoAvailable(B,true) 这两个事件回调，即房间“101”中的用户都可以订阅主播 B 的音视频。
- 房间“102”中的用户都会收到主播 A 的 onRemoteUserEnterRoom(A) 和 onUserVideoAvailable(A,true) 这两个事件回调，即房间“102”中的用户都可以订阅主播 A 的音视频。



跨房通话的参数考虑到后续扩展字段的兼容性问题，暂时采用了 JSON 格式的参数：

情况一：数字房间号

如果房间“101”中的主播 A 要跟房间“102”中的主播 B 连麦，那么主播 A 调用该接口时需要传入：

```
{"roomId": 102, "userId": "userB"}
```

示例代码如下：

```
Json::Value jsonObj;
jsonObj["roomId"] = 102;
jsonObj["userId"] = "userB";
Json::FastWriter writer;
std::string params = writer.write(jsonObj);
trtc.ConnectOtherRoom(params.c_str());
```

情况二：字符串房间号

如果您使用的是字符串房间号，务必请将 json 中的 “roomId” 替换成 “strRoomId”：{"strRoomId": "102", "userId": "userB"}

示例代码如下：

```
Json::Value jsonObj;
jsonObj["strRoomId"] = "102";
jsonObj["userId"] = "userB";
Json::FastWriter writer;
std::string params = writer.write(jsonObj);
trtc.ConnectOtherRoom(params.c_str());
```

参数	描述
param	需要你传入 JSON 格式的字符串参数，roomId 代表数字格式的房间号，strRoomId 代表字符串格式的房间号，userId 代表目标主播的用户 ID。

disconnectOtherRoom

disconnectOtherRoom

退出跨房通话

退出结果会通过 ITRTCCloudCallback 中的 onDisconnectOtherRoom 回调通知给您。

setDefaultStreamRecvMode

setDefaultStreamRecvMode

void setDefaultStreamRecvMode	(bool autoRecvAudio bool autoRecvVideo)

设置订阅模式（需要在进入房前设置才能生效）

您可以通过该接口在“自动订阅”和“手动订阅”两种模式下进行切换：

- 自动订阅：默认模式，用户在进入房间后会立刻接收到该房间中的音视频流，音频会自动播放，视频会自动开始解码（依然需要您通过 startRemoteView 接口绑定渲染控件）。
- 手动订阅：在用户进入房间后，需要手动调用 startRemoteView 接口才能启动视频流的订阅和解码，需要手动调用 muteRemoteAudio (false) 接口才能启动声音的播放。

在绝大多数场景下，用户进入房间后都会订阅房间中所有主播的音视频流，因此 TRTC 默认采用了自动订阅模式，以求得最佳的“秒开体验”。
如果您的应用场景中每个房间同时会有很多路音视频流在发布，而每个用户只想选择性地订阅其中的 1-2 路，则推荐使用“手动订阅”模式以节省流量费用。

参数	描述
autoRecvAudio	true：自动订阅音频；false：需手动调用 muteRemoteAudio(false) 订阅音频。默认值：true。
autoRecvVideo	true：自动订阅视频；false：需手动调用 startRemoteView 订阅视频。默认值：true。

注意

1. 需要在进入房间前调用该接口进行设置才能生效。
2. 在自动订阅模式下，如果用户在进入房间后没有调用 startRemoteView 订阅视频流，SDK 会自动停止订阅视频流，以便达到节省流量的目的。

createSubCloud

createSubCloud

创建子房间实例（用于多房间并发观看）

TRTCCloud 一开始被设计成单例模式，限制了多房间并发观看的能力。
通过调用该接口，您可以创建出多个 TRTCCloud 实例，以便同时进入多个不同的房间观看音视频流。

但需要注意的是，您在多个 TRTCCloud 实例中发布音视频流的能力会受到一定限制。

该功能主要用于在线教育场景中一种被称为“超级小班课”的业务场景中，用于解决“每个 TRTC 的房间中最多只能有 50 人同时发布自己音视频流”的限制。

示例代码如下：

```
    //In the small room that needs interaction, enter the room as an anchor and
push audio and video streams
    ITRTCCloud *mainCloud = getTRTCShareInstance();
    TRTCParams mainParams;
    //Fill your params
    mainParams.role = TRTCRoleAnchor;
    mainCloud->enterRoom(mainParams, TRTCAppSceneLIVE);
    //...
    mainCloud->startLocalAudio(TRTCAudioQualityDefault);
    mainCloud->startLocalPreview(renderView);

    //In the large room that only needs to watch, enter the room as an audience
and pull audio and video streams
    ITRTCCloud *subCloud = mainCloud->createSubCloud();
    TRTCParams subParams;
    //Fill your params
    subParams.role = TRTCRoleAudience;
    subCloud->enterRoom(subParams, TRTCAppSceneLIVE);
    //...
    subCloud->startRemoteView(userId, TRTCVideoStreamTypeBig, renderView);
    //...
    //Exit from new room and release it.
    subCloud->exitRoom();
    mainCloud->destroySubCloud(subCloud);
```

注意

- 同一个用户，可以使用同一个 userId 进入多个不同 roomId 的房间。
- 两台不同的终端设备不可以同时使用同一个 userId 进入同一个 roomId 的房间。
- 您可以分别为不同实例分别设置 ITRTCCloudCallback 获取各自的事件通知。
- 同一个用户可以在多个 TRTCCloud 实例中推流，也可以调用子实例中与本地音视频相关的接口。但需要注意：
- 多实例的音频需要同时为麦克风采集或自定义采集，而且与音频设备相关的接口调用会以最后一次为准；

- 与摄像头相关的调用会以最后一次为准：startLocalPreview 。

返回值说明：

子 TRTCCloud 实例

destroySubCloud

destroySubCloud

void destroySubCloud (ITRTCCloud *subCloud)

销毁子房间实例

参数	描述
subCloud	子房间实例

updateOtherRoomForwardMode

updateOtherRoomForwardMode

void updateOtherRoomForwardMode (const char* param)

更改跨房主播在本房间的上行能力

通常情况下，在调用 connectOtherRoom 接口与另一个房间的主播进行跨房通话后，本房间内的所有观众都将收到该主播发布的音视频流。
您可以通过调用该接口，限制跨房主播在本房间内的上行能力，禁止或允许跨房主播发布音频/主路视频/辅路视频，该行为会影响房间内的所有用户。
在禁用跨房主播某种上行能力后，本房间内所有用户将无法收到对应音视频流，且无法再订阅对应的音视频。

需要注意的是，该接口只能由进行跨房通话的主播调用，且通过该接口设置的限制会因为跨房通话的中断或对应主播退房重置。

该接口的调用结果会通过 ITRTCCloudCallback 中的 onUpdateOtherRoomForwardMode 回调通知给您。

例如：
房间“101”中有主播 A 与观众 B，房间“102”中有主播 C，主播 C 正常发布音视频流。主播 A 通过 connectOtherRoom() 跟主播 C 建立跨房通话。

- 此时，主播 A 与观众 B 都会收到主播 C 的 onRemoteUserEnterRoom(C) 和 onUserVideoAvailable(C,true)、onUserAudioAvailable(C,true) 这三个事件回调，可以订阅主播 C 的音视频。

之后，主播 A 调用该接口禁用主播 C 在本房间发布音频的能力。

- 此后，主播 C 的音频流将无法发布至房间“101”，主播 A 与观众 B 都将收到 onUserAudioAvailable(C,false) 事件回调，且无法再通过 muteRemoteAudio(C,false) 订阅主播 C 的音频。
- 主播 C 的视频流将不受影响。房间“102”中的其他观众也不受影响，可以正常订阅主播 C 的音频。

该接口的参数考虑到后续扩展字段的兼容性问题，暂时采用了 JSON 格式的参数，示例如下：
情况一：数字房间号

```
{
 "roomId":102,
 "userId":"userC",
 "muteAudio":false,
 "muteVideo":true,
 "muteSubStream":false
}
```

情况二：字符串房间号

```
{
 "strRoomId":"102",
 "userId":"userC",
 "muteAudio":false,
 "muteVideo":true,
 "muteSubStream":false
}
```

参数	描述

param	需要你传入 JSON 格式的字符串参数，roomId 代表数字格式的房间号，strRoomId 代表字符串格式的房间号，userId 代表目标主播的用户 ID，muteAudio、muteVideo、muteSubStream均为可选项，分表代表禁止或允许跨房主播发布音频/主路视频/辅路视频的能力。

startPublishing

startPublishing

```
void startPublishing (const char* streamId
                      TRTCVideoStreamType streamType)
```

开始向腾讯云直播 CDN 上发布音视频流

该接口会向 TRTC 服务器发送指令，要求其将当前用户的音视频流旁路到直播 CDN 上。您可以通过参数 streamId 设定直播流的 StreamId，从而可以指定该用户的音视频流对应在直播 CDN 上的播放地址。

例如：您可以通过该接口将当前用户的直播流 ID 指定为 user_stream_001，那么该用户音视频流对应的 CDN 播放地址为：
“http://yourdomain/live/user_stream_001.flv”，其中 yourdomain 为您自己备案的播放域名，

您可以在直播控制台 配置您的播放域名，腾讯云不提供默认的播放域名。

您也可以在设置 enterRoom 的参数 TRTCParams 时指定 streamId, 而且我们更推荐您采用这种方案。

参数	描述
streamId	自定义流 ID。
streamType	仅支持 TRTCVideoStreamTypeBig 和 TRTCVideoStreamTypeSub 。

注意

您需要提前在 实时音视频控制台 中的功能配置页面上开启“启用旁路推流”才能生效。

- 若您选择“指定流旁路”，则您可以通过该接口将对应音视频流推送到腾讯云 CDN 且指定为填写的流 ID。
- 若您选择“全局自动旁路”，则您可以通过该接口调整默认的流 ID。

stopPublishing

stopPublishing

停止向腾讯云直播 CDN 上发布音视频流

startPublishCDNStream

startPublishCDNStream

void startPublishCDNStream (const TRTCPublishCDNParam & param)

开始向非腾讯云 CDN 上发布音视频流

该接口跟 startPublishing 功能类似，不同之处在于，startPublishing 仅支持向腾讯云的 CDN 发布，而本接口支持向非腾讯云的直播 CDN 上转推音视频流。

参数	描述
param	CDN 转推参数，详情请参见 TRTCPublishCDNParam

注意

- 使用 startPublishing 接口向腾讯云的直播 CDN 上发布音视频流不会收取额外费用。
- 使用 startPublishCDNStream 接口向非腾讯云的直播 CDN 上发布音视频流，需要收取额外的转推带宽费用。

stopPublishCDNStream

stopPublishCDNStream

停止向非腾讯云 CDN 上发布音视频流

setMixTranscodingConfig

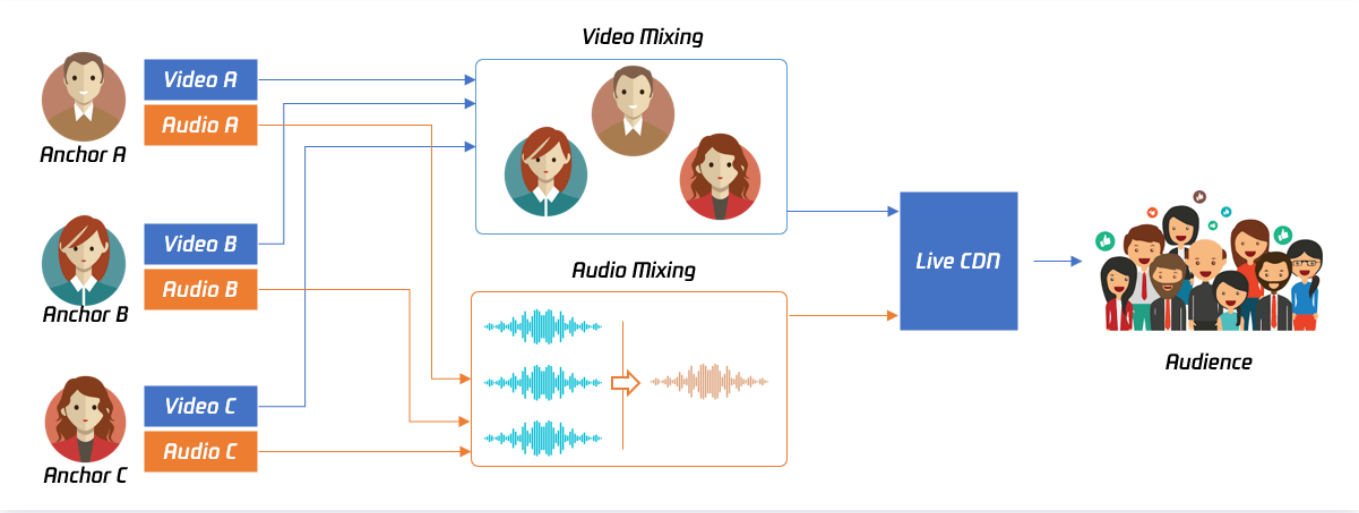
setMixTranscodingConfig

void setMixTranscodingConfig (TRTCTranscodingConfig* config)

设置云端混流的排版布局和转码参数

在一个直播间中可能同时会有多个主播发布自己的音视频流，但对于直播 CDN 上的观众而言，只需要观看一条 HTTP-FLV 或 HLS 格式的视频流即可。
当您调用本接口函数时，SDK 会向腾讯云的 TRTC 混流服务器发送一条指令，混流服务器会将房间里的多路音视频流混合成一路。
您可以通过 TRTCTranscodingConfig 参数来调整每一路画面的排版布局，也可以设置混合后的音视频流的各项编码参数。

参考文档：云端混流转码 。



参数	描述
config	如果 config 不为空，则开启云端混流，如果 config 为空则停止云端混流。详情请参见 TRTCTranscodingConfig 。

注意

关于云端混流的注意事项：

- 混流转码为收费功能，调用接口将产生云端混流转码费用，详见 云端混流转码计费说明 。
- 调用该接口的用户，如果没设定 config 参数中的 streamId 字段，TRTC 会将房间中的多路画面混合到当前用户所对应的音视频流上，即 A + B => A。
- 调用该接口的用户，如果设定了 config 参数中的 streamId 字段，TRTC 会将房间中的多路画

面混合到您指定的 streamId 上，即 A + B => streamId。

- 请注意，若您还在房间中且不再需要混流，请务必再次调用本接口并将 config 设置为空以进行取消，不及时取消混流可能会引起不必要的计费损失。
- 请放心，当您退房时 TRTC 会自动取消混流状态。

startPublishMediaStream

startPublishMediaStream

```
void startPublishMediaStream (TRTCPublishTarget * target
                              TRTCStreamEncoderParam * params
                              TRTCStreamMixingConfig * config)
```

开始发布媒体流

该接口会向 TRTC 服务器发送指令，要求其将当前用户的音视频流转推/转码到直播 CDN 或者回推到 TRTC 房间中您可以通过 TRTCPublishTarget 配置中的 TRTCPublishMode 指定具体的发布模式

参数	描述
config	媒体流转码配置参数。具体配置参考 TRTCStreamMixingConfig 。转码和回推到 TRTC 房间中时为必填项，您需要指定您预期的转码配置参数。转推模式下则无效。
params	媒体流编码输出参数，具体配置参考 TRTCStreamEncoderParam 。转码和回推到 TRTC 房间中时为必填项，您需要指定您预期的转码输出参数。在转推时，为了更好的转推稳定性和 CDN 兼容性，也建议您进行配置。
target	媒体流发布的目标地址，具体配置参考 TRTCPublishTarget 。支持转推/转码到腾讯或者第三方 CDN，也支持转码回推到 TRTC 房间中。

注意

1. SDK 会通过回调 onStartPublishMediaStream 带给您后台启动的任务标识（即 taskId）。
2. 同一个任务（TRTCPublishMode 与 TRTCPublishCdnUrl 均相同）仅支持启动一次。若您后续需要更新或者停止该项任务，需要记录并使用返回的 taskId，通过 updatePublishMediaStream 或者 stopPublishMediaStream 来操作。
3. target 支持同时配置多个 CDN URL（最多同时 10 个）。若您的同一个转推/转码任务需要发布至多路 CDN，则仅需要在 target 中配置多个 CDN URL 即可。同一个转码任务即使有多个转推地址，对应的转码计费仍只收取一份。

4. 使用时需要注意不要多个任务同时往相同的 URL 地址推送，以免引起异常推流状态。一种推荐的方案是 URL 中使用 “sdkappid_roomid_userid_main” 作为区分标识，这种命名方式容易辨认且不会在您的多个应用中发生冲突。

updatePublishMediaStream

updatePublishMediaStream

```
void updatePublishMediaStream(const char* taskId
                              TRTCPublishTarget * target
                              TRTCStreamEncoderParam * params
                              TRTCStreamMixingConfig * config)
```

更新发布媒体流

该接口会向 TRTC 服务器发送指令，更新通过 startPublishMediaStream 启动的媒体流

参数	描述
config	媒体流转码配置参数。具体配置参考 TRTCStreamMixingConfig 。转码和回推到 TRTC 房间中时为必填项，您需要指定您预期的转码配置参数。转推模式下则无效。
params	媒体流编码输出参数，具体配置参考 TRTCStreamEncoderParam 。转码和回推到 TRTC 房间中时为必填项，您需要指定您预期的转码输出参数。在转推时，为了更好的转推稳定性和 CDN 兼容性，也建议您进行配置。
target	媒体流发布的目标地址，具体配置参考 TRTCPublishTarget 。支持转推/转码到腾讯或者第三方 CDN，也支持转码回推到 TRTC 房间中。
taskId	通过回调 onStartPublishMediaStream 带给您后台启动的任务标识（即 taskId）

注意

1. 您可以通过本接口来更新发布的 CDN URL（支持增删，最多同时 10 个），但您使用时需要注意不要多个任务同时往相同的 URL 地址推送，以免引起异常推流状态。
2. 您可以通过 taskId 来更新调整转推/转码任务。例如在 pk 业务中，您可以先通过 startPublishMediaStream 发起转推，接着在主播发起 pk 时，通过 taskId 和本接口将转推更

新为转码任务。此时，CDN 播放将连续并且不会发生断流（您需要保持媒体流编码输出参数 param 一致）。
3. 同一个任务不支持纯音频、音视频、纯视频之间的切换。

stopPublishMediaStream

stopPublishMediaStream

```
void stopPublishMediaStream (const char* taskId)
```

停止发布媒体流

该接口会向 TRTC 服务器发送指令，停止通过 startPublishMediaStream 启动的媒体流

参数	描述
taskId	通过回调 onStartPublishMediaStream 带给您后台启动的任务标识（即 taskId）

注意
1. 若您的业务后台并没有保存该 taskId，在您的主播异常退房重进后，如果您需要重新获取 taskId，您可以再次调用 startPublishMediaStream 启动任务。此时 TRTC 后台会返回任务启动失败，同时带给您上一次启动的 taskId
2. 若 taskId 填空字符串，将会停止该用户所有通过 startPublishMediaStream 启动的媒体流，如果您只启动了一个媒体流或者想停止所有通过您启动的媒体流，推荐使用这种方式。

startLocalPreview

startLocalPreview

```
void startLocalPreview (bool frontCamera
                        TXView view)
```

开启本地摄像头的预览画面（移动端）

在 enterRoom 之前调用此函数，SDK 只会开启摄像头，并一直等到您调用 enterRoom 之后才开始推流。
在 enterRoom 之后调用此函数，SDK 会开启摄像头并自动开始视频推流。当开始渲染首帧摄像头画面时，您会收到 ITRTCCloudCallback 中的 onCameraDidReady 回调通知。

参数	描述
frontCamera	true：前置摄像头；false：后置摄像头。
view	承载视频画面的控件。

注意
如果希望开播前预览摄像头画面并通过 BeautyManager 调节美颜参数，您可以：
- 方案一：在调用 enterRoom 之前调用 startLocalPreview。
- 方案二：在调用 enterRoom 之后调用 startLocalPreview + muteLocalVideo(true)。

startLocalPreview

startLocalPreview

```
void startLocalPreview        (TXView view)
```

开启本地摄像头的预览画面（桌面端）

在调用该接口之前，您可以先调用 setCurrentCameraDevice 选择使用 Mac 自带摄像头或外接摄像头。
在 enterRoom 之前调用此函数，SDK 只会开启摄像头，并一直等到您调用 enterRoom 之后才开始推流。
在 enterRoom 之后调用此函数，SDK 会开启摄像头并自动开始视频推流。
当开始渲染首帧摄像头画面时，您会收到 TRTCCloudDelegate 中的 onCameraDidReady 回调通知。

参数	描述
view	承载视频画面的控件。

注意
如果希望开播前预览摄像头画面并通过 BeautyManager 调节美颜参数，您可以：
- 方案一：在调用 enterRoom 之前调用 startLocalPreview。

- 方案二：在调用 enterRoom 之后调用 startLocalPreview + muteLocalVideo(true)。

updateLocalView

updateLocalView

void updateLocalView	(TXView view)

更新本地摄像头的预览画面

stopLocalPreview

stopLocalPreview

停止摄像头预览

muteLocalVideo

muteLocalVideo

void muteLocalVideo	(TRTCVideoStreamType streamType bool mute)

暂停/恢复发布本地的视频流

该接口可以暂停（或恢复）发布本地的视频画面，暂停之后，同一房间中的其他用户将无法继续看到自己画面。
该接口在指定 TRTCVideoStreamTypeBig 时等效于 start/stopLocalPreview 这两个接口，但具有更好的响应速度。
因为 start/stopLocalPreview 需要打开和关闭摄像头，而打开和关闭摄像头都是硬件设备相关的操作，非常耗时。

相比之下，muteLocalVideo 只需要在软件层面对数据流进行暂停或者放行即可，因此效率更高，也更适合需要频繁打开关闭的场景。

当暂停/恢复发布指定 TRTCVideoStreamTypeBig 后，同一房间中的其他用户将会收到 onUserVideoAvailable 回调通知。
当暂停/恢复发布指定 TRTCVideoStreamTypeSub 后，同一房间中的其他用户将会收到 onUserSubStreamAvailable 回调通知。

参数	描述
mute	true：暂停；false：恢复。
streamType	要暂停/恢复的视频流类型（仅支持 TRTCVideoStreamTypeBig 和 TRTCVideoStreamTypeSub ）。

setVideoMuteImage

setVideoMuteImage

```
void setVideoMuteImage(TRTCImageBuffer* image
                       int fps)
```

设置本地画面被暂停期间的替代图片

当您调用 muteLocalVideo(true) 暂停本地画面时，您可以通过调用本接口设置一张替代图片，设置后，房间中的其他用户会看到这张替代图片，而不是黑屏画面。

参数	描述
fps	设置替代图片帧率，最小值为5，最大值为10，默认5。
image	设置替代图片，空值代表在 muteLocalVideo 之后不再发送视频流数据，默认值为空。

startRemoteView

startRemoteView

void startRemoteView	(const char* userId TRTCVideoStreamType streamType TXView view)

订阅远端用户的视频流，并绑定视频渲染控件

调用该接口可以让 SDK 拉取指定 userid 的视频流，并渲染到参数 view 指定的渲染控件上。您可以通过 setRemoteRenderParams 设置画面的显示模式。

- 如果您已经知道房间中有视频流的用户的 userid，可以直接调用 startRemoteView 订阅该用户的画面。
- 如果您不知道房间中有哪些用户在发布视频，您可以在 enterRoom 之后等待来自 onUserVideoAvailable 的通知。

调用本接口只是启动视频流的拉取，此时画面还需要加载和缓冲，当缓冲完毕后您会收到来自 onFirstVideoFrame 的通知。

参数	描述
streamType	指定要观看 userId 的视频流类型。 • 高清大画面：TRTCVideoStreamTypeBig 。 • 低清小画面：TRTCVideoStreamTypeSmall（需要远端用户通过 enableSmallVideoStream 开启双路编码后才有效果）。 • 辅流画面（常用于屏幕分享）：TRTCVideoStreamTypeSub 。
userId	指定远端用户的 ID。
view	用于承载视频画面的渲染控件。

注意

注意几点规则需要您关注：

1. SDK 支持同时观看某 userid 的大画面和辅路画面，或者同时观看某 userid 的小画面和辅路画面，但不支持同时观看大画面和小画面。
2. 只有当指定的 userid 通过 enableSmallVideoStream 开启双路编码后，才能观看该用户的小画面。
3. 当指定的 userid 的小画面不存在时，SDK 默认切换到该用户的大画面。

updateRemoteView

updateRemoteView

```
void updateRemoteView(const char* userId, TRTCVideoStreamType streamType, TXView view)
```

更新远端用户的视频渲染控件

该接口可用于更新远端视频画面的渲染控件，常被用于切换显示区域的交互场景中。

参数	描述
streamType	要设置预览窗口的流类型（仅支持 TRTCVideoStreamTypeBig 和 TRTCVideoStreamTypeSub ）。
userId	指定远端用户的 ID。
view	承载视频画面的控件。

stopRemoteView

stopRemoteView

```
void stopRemoteView(const char* userId, TRTCVideoStreamType streamType)
```

停止订阅远端用户的视频流，并释放渲染控件

调用此接口会让 SDK 停止接收该用户的视频流，并释放该路视频流的解码和渲染资源。

参数	描述
streamType	指定要观看 userId 的视频流类型。 • 高清大画面：TRTCVideoStreamTypeBig 。 • 低清小画面：TRTCVideoStreamTypeSmall 。

	• 辅流画面（常用于屏幕分享）：TRTCVideoStreamTypeSub 。
userId	指定远端用户的 ID。

stopAllRemoteView

stopAllRemoteView

停止订阅所有远端用户的视频流，并释放全部渲染资源

调用此接口会让 SDK 停止接收所有来自远端的视频流，并释放全部的解码和渲染资源。

> **注意**
> 如果当前有正在显示的辅路画面（屏幕分享）也会一并被停止。

muteRemoteVideoStream

muteRemoteVideoStream

```
void muteRemoteVideoStream (const char* userId
                            TRTCVideoStreamType streamType
                            bool mute)
```

暂停/恢复订阅远端用户的视频流

该接口仅暂停/恢复接收指定用户的视频流，但并不释放显示资源，视频画面会被冻屏在接口调用时的最后一帧。

参数	描述
mute	是否暂停接收。
streamType	要暂停/恢复的视频流类型。 • 高清大画面：TRTCVideoStreamTypeBig 。 • 低清小画面：TRTCVideoStreamTypeSmall 。 • 辅流画面（常用于屏幕分享）：TRTCVideoStreamTypeSub 。

userId	指定远端用户的 ID。

注意
该接口支持您在进入房间（enterRoom）前调用，暂停状态在退出房间（exitRoom）之后会被重置。
调用该接口暂停接收指定用户的视频流后，只调用 startRemoteView 接口无法播放指定用户的视频，需要调用 muteRemoteVideoStream(false) 或 muteAllRemoteVideoStreams(false) 才能恢复。

muteAllRemoteVideoStreams

muteAllRemoteVideoStreams

```
void muteAllRemoteVideoStreams (bool mute)
```

暂停/恢复订阅所有远端用户的视频流

该接口仅暂停/恢复接收所有用户的视频流，但并不释放显示资源，视频画面会被冻屏在接口调用时的最后一帧。

参数	描述
mute	是否暂停接收。

注意
该接口支持您在进入房间（enterRoom）前调用，暂停状态在退出房间（exitRoom）之后会被重置。
调用该接口暂停接收所有用户的视频流后，只调用 startRemoteView 接口无法播放指定用户的视频，需要调用 muteRemoteVideoStream(false) 或 muteAllRemoteVideoStreams(false) 才能恢复。

setVideoEncoderParam

setVideoEncoderParam

void setVideoEncoderParam	(const TRTCVideoEncParam & param)

设置视频编码器的编码参数

该设置能够决定远端用户看到的画面质量，同时也能决定云端录制出的视频文件的画面质量。

参数	描述
param	用于设置视频编码器的相关参数，详情请参见 TRTCVideoEncParam 。

注意
从v11.5版本开始，编码输出分辨率会按照宽8高2字节对齐，并且是向下调整，eg:输入分辨率540x960，实际编码输出分辨率536x960。

setNetworkQosParam

setNetworkQosParam

void setNetworkQosParam	(const TRTCNetworkQosParam & param)

设置网络质量控制的相关参数

该设置决定在差网络环境下的质量调控策略，如“画质优先”或“流畅优先”等策略。

参数	描述
param	用于设置网络质量控制的相关参数，详情请参见 TRTCNetworkQosParam 。

setLocalRenderParams

setLocalRenderParams

void setLocalRenderParams	(const TRTCRenderParams &params)

设置本地画面的渲染参数

可设置的参数包括有：画面的旋转角度、填充模式以及左右镜像等。

参数	描述
params	画面渲染参数，详情请参见 TRTCRenderParams 。

setRemoteRenderParams

setRemoteRenderParams

```
void setRemoteRenderParams(const char* userId, TRTCVideoStreamType streamType, const TRTCRenderParams &params)
```

设置远端画面的渲染模式

可设置的参数包括有：画面的旋转角度、填充模式以及左右镜像等。

参数	描述
params	画面渲染参数，详情请参见 TRTCRenderParams 。
streamType	可以设置为主路画面（ TRTCVideoStreamTypeBig ）或辅路画面（ TRTCVideoStreamTypeSub ）。
userId	指定远端用户的 ID。

enableSmallVideoStream

enableSmallVideoStream

```
void enableSmallVideoStream(bool enable
```

const TRTCVideoEncParam & smallVideoEncParam)

开启大小画面双路编码模式

开启双路编码模式后，当前用户的编码器会同时输出【高清大画面】和【低清小画面】两路视频流（但只有一路音频流）。
如此以来，房间中的其他用户就可以根据自身的网络情况或屏幕大小选择订阅【高清大画面】或是【低清小画面】。

参数	描述
enable	是否开启小画面编码，默认值：false。
smallVideoEncParam	小流的视频参数。

注意
双路编码开启后，会消耗更多的 CPU 和 网络带宽，所以 Mac、Windows 或者高性能 Pad 可以考虑开启，不建议手机端开启。

返回值说明：

0：成功；-1：当前大画面已被设置为较低画质，开启双路编码已无必要。

setRemoteVideoStreamType

setRemoteVideoStreamType

void setRemoteVideoStreamType (const char* userId
TRTCVideoStreamType streamType)

切换指定远端用户的大小画面

当房间中某个主播开启了双路编码之后，房间中其他用户通过 startRemoteView 订阅到的画面默认会是【高清大画面】。
您可以通过此接口选定希望订阅的画面是大画面还是小画面，该接口在 startRemoteView 之前和之后调用均可生效。

参数	描述

streamType	视频流类型，即选择看大画面还是小画面，默认为大画面。
userId	指定远端用户的 ID。

注意
此功能需要目标用户已经通过 enableSmallVideoStream 提前开启了双路编码模式，否则此调用无实际效果。

snapshotVideo

snapshotVideo

```
void snapshotVideo(const char* userId,
                   TRTCVideoStreamType streamType,
                   TRTCSnapshotSourceType sourceType)
```

视频画面截图

您可以通过本接口截取本地的视频画面，远端用户的主路画面以及远端用户的辅路（屏幕分享）画面。

参数	描述
sourceType	画面来源，可选择截取视频流画面（TRTCSnapshotSourceTypeStream）或视频渲染画面（TRTCSnapshotSourceTypeView），前者一般更清晰。
streamType	视频流类型，可选择截取主路画面（TRTCVideoStreamTypeBig，常用于摄像头）或辅路画面（TRTCVideoStreamTypeSub，常用于屏幕分享）。
userId	用户 ID，如指定空置表示截取本地的视频画面。

注意
Windows 平台目前仅支持截取 TRTCSnapshotSourceTypeStream 来源的视频画面。

setGravitySensorAdaptiveMode

setGravitySensorAdaptiveMode

void setGravitySensorAdaptiveMode (TRTCGravitySensorAdaptiveMode mode)

设置重力感应的适配模式（11.7 及以上版本）

开启重力感应后，如果采集端的设备发生旋转，采集端和观众端的画面都会进行相应地渲染以确保视野中的画面始终朝上。
只在sdk内部摄像头采集场景生效，并且只在移动端生效。
1. 该接口仅对采集端起作用，如果只是观看房间中的画面，开启此接口是无效的
2. 当采集端设备发生 90 度或 270 度旋转时，采集端或者观众看到的画面可能会被裁剪以保持比例的协调

参数	描述
mode	重力感应模式，详情请参见 TRTCGravitySensorAdaptiveMode_Disable 、TRTCGravitySensorAdaptiveMode_FillByCenterCrop 和 TRTCGravitySensorAdaptiveMode_FitWithBlackBorder 默认值：TRTCGravitySensorAdaptiveMode_Disable 。

startLocalAudio

startLocalAudio

void startLocalAudio (TRTCAudioQuality quality)

开启本地音频的采集和发布

SDK 默认不开启麦克风，当用户需要发布本地音频时，需要调用该接口开启麦克风采集，并将音频编码并发布到当前的房间中。
开启本地音频的采集和发布后，房间中的其他用户会收到 onUserAudioAvailable(userId, true) 的通知。

参数	描述
quality	声音音质 • TRTCAudioQualitySpeech，流畅：采样率：16k；单声道；音频裸码率：16kbps；适合语音通话为主的场景，比如在线会议，语音通话。 • TRTCAudioQualityDefault，默认：采样率：48k；单声道；音频裸码率：

50kbps；SDK 默认的音频质量，如无特殊需求推荐选择之。
- TRTCAudioQualityMusic，高音质：采样率：48k；双声道 + 全频带；音频裸码率：128kbps；适合需要高保真传输音乐的场景，比如在线K歌、音乐直播等。

注意
该函数会检查麦克风的使用权限，如果当前 App 没有麦克风权限，SDK 会自动向用户申请麦克风使用权限。

stopLocalAudio

stopLocalAudio

停止本地音频的采集和发布

停止本地音频的采集和发布后，房间中的其他用户会收到 onUserAudioAvailable(userId, false) 的通知。

muteLocalAudio

muteLocalAudio

void muteLocalAudio (bool mute)

暂停/恢复发布本地的音频流

当您暂停发布本地音频流之后，房间中的其他用户会收到 onUserAudioAvailable(userId, false) 的通知。
当您恢复发布本地音频流之后，房间中的其他用户会收到 onUserAudioAvailable(userId, true) 的通知。

与 stopLocalAudio 的不同之处在于，muteLocalAudio(true) 并不会释放麦克风权限，而是继续发送码率极低的静音包。
这对于需要云端录制的场景非常适用，因为 MP4 等格式的视频文件，对于音频数据的连续性要求很高，使用 stopLocalAudio 会导致录制出的 MP4 文件不易播放。
因此在对录制文件的质量要求较高的场景中，建议选择 muteLocalAudio 而不建议使用 stopLocalAudio。

参数	描述
mute	true：静音；false：恢复。

muteRemoteAudio

muteRemoteAudio

```
void muteRemoteAudio (const char* userId
                      bool mute)
```

暂停/恢复播放远端的音频流

当您静音某用户的远端音频时，SDK 会停止播放指定用户的声音，同时也会停止拉取该用户的音频数据。

参数	描述
mute	true：静音；false：取消静音。
userId	用于指定远端用户的 ID。

注意
在进入房间（enterRoom）之前或之后调用本接口均生效，静音状态在退出房间（exitRoom）之后会被重置为 false。

muteAllRemoteAudio

muteAllRemoteAudio

```
void muteAllRemoteAudio (bool mute)
```

暂停/恢复播放所有远端用户的音频流

当您静音所有用户的远端音频时，SDK 会停止播放所有来自远端的音频流，同时也会停止拉取所有用户的音频数据。

参数	描述
mute	true：静音；false：取消静音。

注意
在进入房间（enterRoom）之前或之后调用本接口均生效，静音状态在退出房间（exitRoom）之后会被重置为 false。

setRemoteAudioVolume

setRemoteAudioVolume

```
void setRemoteAudioVolume (const char *userId
                           int volume)
```

设定某一个远端用户的声音播放音量

您可以通过 setRemoteAudioVolume(userId, 0) 将某一个远端用户的声音静音。

参数	描述
userId	用于指定远端用户的 ID。
volume	音量大小，取值范围为0 - 100，默认值：100。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setAudioCaptureVolume

setAudioCaptureVolume

void setAudioCaptureVolume (int volume)

设定本地音频的采集音量

参数	描述
volume	音量大小，取值范围为0 – 100；默认值：100。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

getAudioCaptureVolume

getAudioCaptureVolume

获取本地音频的采集音量

setAudioPlayoutVolume

setAudioPlayoutVolume

void setAudioPlayoutVolume (int volume)

设定远端音频的播放音量

该接口会控制 SDK 最终交给系统播放的声音音量，调节效果会影响到本地音频录制文件的音量大小，但不会影响到耳返的音量大小。

参数	描述
volu	音量大小，取值范围为0 – 100，默认值：100。

me	

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

getAudioPlayoutVolume

getAudioPlayoutVolume

获取远端音频的播放音量

enableAudioVolumeEvaluation

enableAudioVolumeEvaluation

```
void enableAudioVolumeEvaluation(bool enable
                                 const TRTCAudioVolumeEvaluateParams & params)
```

启用音量大小提示

开启此功能后，SDK 会在 ITRTCCloudCallback 中的 onUserVoiceVolume 回调中反馈本地或远端用户的音频音量评估信息，包括音量大小、人声检测、音频频谱等。

参数	描述
enable	是否启用音量提示，默认为关闭状态。
params	音量评估等相关参数，请参见 TRTCAudioVolumeEvaluateParams 。

startAudioRecording

startAudioRecording

int startAudioRecording (const TRTCAudioRecordingParams & param)

开始录音

当您调用该接口后， SDK 会将本地和远端的所有音频（包括本地音频，远端音频，背景音乐和音效等）混合并录制到一个本地文件中。
该接口在进入房间前后调用均可生效，如果录制任务在退出房间前尚未通过 stopAudioRecording 停止，则退出房间后录制任务会自动被停止。
本次录制的启动、完成状态会通过本地录制相关回调进行通知。参见 TRTCCloud 相关回调。

参数	描述
param	录音参数，请参见 TRTCAudioRecordingParams 。

注意
自 v11.5 版本，音频录制的状态结果由返回值统一调整为异步回调进行通知。参见 TRTCCloud 相关回调。

返回值说明：

0：成功；-1：录音已开始；-2：文件或目录创建失败；-3：后缀指定的音频格式不支持。

stopAudioRecording

stopAudioRecording

停止录音

如果录制任务在退出房间前尚未通过本接口停止，则退出房间后录音任务会自动被停止。

startLocalRecording

startLocalRecording

void startLocalRecording (const TRTCLocalRecordingParams & params)

开启本地媒体录制

开启后把直播过程中的音视频内容录制到本地的一个文件中。

参数	描述
params	录制参数，请参见 TRTCLocalRecordingParams 。

stopLocalRecording

stopLocalRecording

停止本地媒体录制

如果录制任务在退出房间前尚未通过本接口停止，则退出房间后录音任务会自动被停止。

setRemoteAudioParallelParams

setRemoteAudioParallelParams

void setRemoteAudioParallelParams	(const TRTCAudioParallelParams & params)

设置远端音频流智能并发播放策略

设置远端音频流智能并发播放策略，适用于上麦人数比较多的场景。

参数	描述
params	音频并发参数，请参见 TRTCAudioParallelParams 。

enable3DSpatialAudioEffect

enable3DSpatialAudioEffect

void enable3DSpatialAudioEffect	(bool enabled)

启用 3D 音效

启用 3D 音效。注意需使用流畅音质 TRTCAudioQualitySpeech 或默认音质 TRTCAudioQualityDefault 。

参数	描述
enabled	是否启用 3D 音效，默认为关闭状态。

updateSelf3DSpatialPosition

updateSelf3DSpatialPosition

```
void updateSelf3DSpatialPosition (int position[3]
                                  float axisForward[3]
                                  float axisRight[3]
                                  float axisUp[3])
```

设置 3D 音效中自身坐标及朝向信息

更新自身在世界坐标系中的位置和朝向， SDK 会根据该方法参数计算自身和远端用户之间的相对位置，进而渲染出空间音效。注意各参数应分别传入长度为 3 的数组。

参数	描述
axisForward	自身坐标系前轴在世界坐标系中的单位向量，三个值依次表示前、右、上坐标值。
axisRight	自身坐标系右轴在世界坐标系中的单位向量，三个值依次表示前、右、上坐标值。
axisUp	自身坐标系上轴在世界坐标系中的单位向量，三个值依次表示前、右、上坐标值。
position	自身在世界坐标系中的坐标，三个值依次表示前、右、上坐标值。

注意
请适当限制调用频率，推荐两次坐标设置至少间隔 100ms。

updateRemote3DSpatialPosition

updateRemote3DSpatialPosition

```
void updateRemote3DSpatialPosition (const char* userId
                                    int position[3])
```

设置 3D 音效中远端用户坐标信息

更新远端用户在世界坐标系中的位置，SDK 会根据自身和远端用户之间的相对位置，进而渲染出空间音效。注意参数为长度等于 3 的数组。

参数	描述
position	该远端用户在世界坐标系中的坐标，三个值依次表示前、右、上坐标值。
userId	指定远端用户的 ID。

注意
请适当限制调用频率，推荐同一远端用户两次坐标设置至少间隔 100ms。

set3DSpatialReceivingRange

set3DSpatialReceivingRange

```
void set3DSpatialReceivingRange (const char* userId
                                 int range)
```

设置指定用户所发出声音的可被接收范围

设置该范围大小之后，该指定用户的声音将在该范围内可被听见，超出该范围将被衰减为 0。

参数	描述
range	声音最大可被接收范围。
userId	指定远端用户的 ID。

getDeviceManager

getDeviceManager

获取设备管理类（TXDeviceManager）

setBeautyStyle

setBeautyStyle

```
void setBeautyStyle (TRTCBeautyStyle style
                     uint32_t beautyLevel
                     uint32_t whitenessLevel
                     uint32_t ruddinessLevel)
```

设置美颜、美白、红润等特效

SDK 内部集成了两套风格不同的磨皮算法：
- “光滑”：算法比较激进，磨皮效果比较明显，适用于秀场直播。
- “自然”：算法更多地保留了面部细节，磨皮效果更加自然，适用于绝大多数直播场景。

参数	描述
beautyLevel	美颜级别，取值范围0 - 9，0表示关闭，1 - 9值越大，效果越明显。
ruddinessLevel	红润级别，取值范围0 - 9，0表示关闭，1 - 9值越大，效果越明显。
style	磨皮算法，有“光滑”和“自然”两种算法。
whitenessLevel	美白级别，取值范围0 - 9，0表示关闭，1 - 9值越大，效果越明显。

setWaterMark

setWaterMark

```
void
setWaterMark
```
(TRTCVideoStreamType streamType
const char* srcData
TRTCWaterMarkSrcType srcType
uint32_t nWidth
uint32_t nHeight
float xOffset
float yOffset
float fWidthRatio
bool isVisibleOnLocalPreview = false)

添加水印

水印的位置是通过 xOffset, yOffset, fWidthRatio 来指定的。

- xOffset：水印的坐标，取值范围为0 – 1的浮点数。
- yOffset：水印的坐标，取值范围为0 – 1的浮点数。
- fWidthRatio：水印的大小比例，取值范围为0 – 1的浮点数。

参数	描述
fWidthRatio	水印显示的宽度占画面宽度比例（水印按该参数等比例缩放显示）。
isVisibleOnLocalPreview	true：本地预览显示水印；false：本地预览不显示水印，仅在win/mac下生效。
nHeight	水印图片像素高度（源数据为文件路径时忽略该参数）。
nWidth	水印图片像素宽度（源数据为文件路径时忽略该参数）。
srcData	水印图片源数据（传 nullptr 表示去掉水印）。
srcType	水印图片源数据类型。
streamType	要设置水印的流类型（TRTCVideoStreamTypeBig、

版权所有：腾讯云计算（北京）有限责任公司

	TRTCVideoStreamTypeSub）。
xOffset	水印显示的左上角 x 轴偏移。
yOffset	水印显示的左上角 y 轴偏移。

注意
本接口只支持给主路视频添加图片水印。

getAudioEffectManager

getAudioEffectManager

获取音效管理类（TXAudioEffectManager）

TXAudioEffectManager 是音效管理接口，您可以通过该接口实现如下功能：

- 背景音乐：支持在线音乐和本地音乐，支持变速、变调等特效、支持原生和伴奏并播放和循环播放。
- 耳机耳返：麦克风捕捉的声音实时通过耳机播放，常用于音乐直播。
- 混响效果：KTV、小房间、大会堂、低沉、洪亮...。
- 变声特效：萝莉、大叔、重金属...。
- 短音效：鼓掌声、欢笑声等简短的音效文件（对于小于10秒的文件，请将 isShortFile 参数设置为 true）。

startSystemAudioLoopback

startSystemAudioLoopback

```
void startSystemAudioLoopback (const char* deviceName = nullptr)
```

开启系统声音采集（iOS 端暂未支持）

该接口会从电脑的声卡中采集音频数据，并将其混入到 SDK 当前的音频数据流中，从而使房间中的其他用户也能听到主播的电脑所播放出的声音。
在线教育场景中，老师可以使用此功能让 SDK 采集教学影片中的声音，并广播给同房间中的学生。
音乐直播场景中，主播可以使用此功能让 SDK 采集音乐播放器中的音乐，从而为自己的直播间增加背景音乐。

参数	描述
deviceName	您可以指定该参数为空值（nullptr），代表让 SDK 采集整个系统的声音。

注意

在 Windows 平台下，您也可以将参数 deviceName 设置为某个应用程序的可执行文件（如 QQMuisc.exe）的绝对路径，此时 SDK 只会采集该应用程序的声音（仅支持 32 位版本的 SDK）。

您也可以指定该参数为某个扬声器设备的名称来采集特定扬声器声音（通过接口 TXDeviceManager 中的 getDevicesList 接口，可以获取类型为 TXMediaDeviceTypeSpeaker 的扬声器设备）。

stopSystemAudioLoopback

stopSystemAudioLoopback

停止系统声音采集（iOS 端暂未支持）

setSystemAudioLoopbackVolume

setSystemAudioLoopbackVolume

void setSystemAudioLoopbackVolume (uint32_t volume)

设置系统声音的采集音量

参数	描述
volume	设置的音量大小，范围是：[0 ~ 150]，默认值为100。

startScreenCapture

startScreenCapture

```
void startScreenCapture (TXView view
                         TRTCVideoStreamType streamType
                         TRTCVideoEncParam* encParam)
```

启动屏幕分享

该接口可以抓取整个屏幕的内容，或抓取您指定的某个应用的窗口内容，并将其分享给同房间中的其他用户。

参数	描述
encParam	屏幕分享的画面编码参数，SDK 会优先使用您通过此接口设置的编码参数： • 如果您设置 encParam 为空值，且您已通过 setSubStreamEncoderParam 设置过辅路视频编码参数，SDK 将使用您设置过的辅路编码参数进行屏幕分享。 • 如果您设置 encParam 为空值，且您未通过 setSubStreamEncoderParam 设置过辅路视频编码参数，SDK 将自动选择一个最佳的编码参数进行屏幕分享。
streamType	屏幕分享使用的线路，可以设置为主路（TRTCVideoStreamTypeBig）或者辅路（TRTCVideoStreamTypeSub），推荐使用辅路。
view	渲染控件所在的父控件，可以设置为空值，表示不显示屏幕分享的预览效果。

注意

1. 同一个用户同时最多只能发布一路主路（TRTCVideoStreamTypeBig）画面和一路辅路（TRTCVideoStreamTypeSub）画面。
2. 默认情况下，屏幕分享使用辅路画面。如果使用主路做屏幕分享，您需要提前停止摄像头采集（stopLocalPreview）以避免相互冲突。
3. 同一个房间中同时只能有一个用户使用辅路做屏幕分享，也就是说，同一个房间中同时只允许一个用户开启辅路。
4. 当房间中已经有其他用户在使用辅路分享屏幕时，此时调用该接口会收到来自 ITRTCCloudCallback 的 onError(ERR_SERVER_CENTER_ANOTHER_USER_PUSH_SUB_VIDEO) 回调。

stopScreenCapture

stopScreenCapture

停止屏幕分享

pauseScreenCapture

pauseScreenCapture

暂停屏幕分享

> **注意**
> 从v11.5版本开始，暂停屏幕采集会使用最后一帧按照1fps帧率输出。

resumeScreenCapture

resumeScreenCapture

恢复屏幕分享

getScreenCaptureSources

getScreenCaptureSources

ITRTCScreenCaptureSourceList*
getScreenCaptureSources (const SIZE &thumbnailSize, const SIZE &iconSize)

枚举可分享的屏幕和窗口（该接口仅支持桌面系统）

当您在对接桌面端系统的屏幕分享功能时，一般都需要展示一个选择分享目标的界面，这样用户能够使用这个界面选择是分享整个屏幕还是某个窗口。
通过本接口，您就可以查询到当前系统中可用于分享的窗口的 ID、名称以及缩略图。我们在 Demo 中提供了一份默认的界面实现供您参考。

参数	描述
iconSize	指定要获取的窗口图标大小。
thumbnailSize	指定要获取的窗口缩略图大小，缩略图可用于绘制在窗口选择界面上。

注意

1. 返回的列表中包含屏幕和应用窗口，屏幕是列表中的第一个元素。如果用户有多个显示器，那么每个显示器都是一个分享目标。
2. 请不要使用 delete ITRTCScreenCaptureSourceList* 删除 SourceList，这很容易导致崩溃，请使用 ITRTCScreenCaptureSourceList 中的 release 方法释放列表。

返回值说明：

窗口列表包括屏幕。

selectScreenCaptureTarget

selectScreenCaptureTarget

```
void selectScreenCaptureTarget (const TRTCScreenCaptureSourceInfo &source
                                const RECT& captureRect
                                const TRTCScreenCaptureProperty &property)
```

选取要分享的屏幕或窗口（该接口仅支持桌面系统）

当您通过 getScreenCaptureSources 获取到可以分享的屏幕和窗口之后，您可以调用该接口选定期望分享的目标屏幕或目标窗口。
在屏幕分享的过程中，您也可以随时调用该接口以切换分享目标。

支持如下四种情况：

- 共享整个屏幕：sourceInfoList 中 type 为 Screen 的 source，captureRect 设为 { 0, 0, 0, 0 }
- 共享指定区域：sourceInfoList 中 type 为 Screen 的 source，captureRect 设为非 nullptr，例如 { 100, 100, 300, 300 }
- 共享整个窗口：sourceInfoList 中 type 为 Window 的 source，captureRect 设为 { 0, 0, 0, 0 }
- 共享窗口区域：sourceInfoList 中 type 为 Window 的 source，captureRect 设为非 nullptr，例如 { 100, 100, 300, 300 }

参数	描述
captureRect	指定捕获的区域
property	指定屏幕分享目标的属性，包括捕获鼠标，高亮捕获窗口等，详情参考 TRTCScreenCaptureProperty 定义
source	指定分享源

注意
设置高亮边框颜色、宽度参数在 Mac 平台不生效。

setSubStreamEncoderParam

setSubStreamEncoderParam

void setSubStreamEncoderParam (const TRTCVideoEncParam & param)

设置屏幕分享（即辅路）的视频编码参数（桌面系统和移动系统均已支持）

该接口可以设定远端用户所看到的屏幕分享（即辅路）的画面质量，同时也能决定云端录制出的视频文件中屏幕分享的画面质量。
请注意如下两个接口的差异：

- setVideoEncoderParam 用于设置主路画面（TRTCVideoStreamTypeBig，一般用于摄像头）的视频编码参数。
- setSubStreamEncoderParam 用于设置辅路画面（TRTCVideoStreamTypeSub，一般用于屏幕分享）的视频编码参数。

参数	描述
param	辅流编码参数，详情请参见 TRTCVideoEncParam。

setSubStreamMixVolume

setSubStreamMixVolume

void setSubStreamMixVolume	(uint32_t volume)

设置屏幕分享时的混音音量大小（该接口仅支持桌面系统）

这个数值越高，屏幕分享音量的占比就越高，麦克风音量占比就越小，所以不推荐设置得太大，否则麦克风的声音就被压制了。

参数	描述
volume	设置的混音音量大小，范围0 - 100。

addExcludedShareWindow

addExcludedShareWindow

void addExcludedShareWindow	(TXView windowID)

将指定窗口加入屏幕分享的排除列表中（该接口仅支持桌面系统）

加入排除列表中的窗口不会被分享出去，常见的用法是将某个应用的窗口加入到排除列表中以避免隐私问题。支持启动屏幕分享前设置过滤窗口，也支持屏幕分享过程中动态添加过滤窗口。

参数	描述
windowID	不希望分享出去的窗口

注意

1. 该接口只有在 TRTCScreenCaptureSourceInfo 中的 type 指定为 TRTCScreenCaptureSourceTypeScreen 时生效，即只有在分享整个屏幕内容时，排除指定窗口的功能才生效。
2. 使用该接口添加到排除列表中的窗口会在退出房间后被 SDK 自动清除。
3. Mac 平台下请传入窗口 ID（即 CGWindowID），您可以通过 TRTCScreenCaptureSourceInfo 中的 sourceId 成员获得。

removeExcludedShareWindow

removeExcludedShareWindow

void removeExcludedShareWindow (TXView windowID)

将指定窗口从屏幕分享的排除列表中移除（该接口仅支持桌面系统）

参数	描述
windowID	要排除的窗口 id。

removeAllExcludedShareWindow

removeAllExcludedShareWindow

将所有窗口从屏幕分享的排除列表中移除（该接口仅支持桌面系统）

addIncludedShareWindow

addIncludedShareWindow

void addIncludedShareWindow (TXView windowID)

将指定窗口加入屏幕分享的包含列表中（该接口仅支持桌面系统）

该接口只有在 TRTCScreenCaptureSourceInfo 中的 type 指定为 TRTCScreenCaptureSourceTypeWindow 时生效。即只有在分享窗口内容时，额外包含指定窗口的功能才生效。
您在 startScreenCapture 之前和之后调用均可。

参数	描述
window	希望被分享出去的窗口（Windows 平台下为窗口句柄： HWND ）

ID	

注意
通过该方法添加到包含列表中的窗口，会在退出房间后被 SDK 自动清除。

removeIncludedShareWindow

removeIncludedShareWindow

```
void removeIncludedShareWindow (TXView windowID)
```

将指定窗口从屏幕分享的包含列表中移除（该接口仅支持桌面系统）

该接口只有在 TRTCScreenCaptureSourceInfo 中的 type 指定为 TRTCScreenCaptureSourceTypeWindow 时生效。
即只有在分享窗口内容时，额外包含指定窗口的功能才生效。

参数	描述
windowID	希望被分享出去的窗口（Mac 平台：窗口 ID；Windows 平台：HWND）

removeAllIncludedShareWindow

removeAllIncludedShareWindow

将全部窗口从屏幕分享的包含列表中移除（该接口仅支持桌面系统）

该接口只有在 TRTCScreenCaptureSourceInfo 中的 type 指定为 TRTCScreenCaptureSourceTypeWindow 时生效。
即只有在分享窗口内容时，额外包含指定窗口的功能才生效。

enableCustomVideoCapture

enableCustomVideoCapture

```
void enableCustomVideoCapture(TRTCVideoStreamType streamType, bool enable)
```

启用/关闭视频自定义采集模式

开启该模式后，SDK 不在运行原有的视频采集流程，即不再继续从摄像头采集数据和美颜，而是只保留视频编码和发送能力。
您需要通过 sendCustomVideoData 不断地向 SDK 塞入自己采集的视频画面。

参数	描述
enable	是否启用，默认值：false。
streamType	用于指定视频流类型，TRTCVideoStreamTypeBig：高清大画面；TRTCVideoStreamTypeSub：辅路画面。

sendCustomVideoData

sendCustomVideoData

```
void sendCustomVideoData(TRTCVideoStreamType streamType, TRTCVideoFrame* frame)
```

向 SDK 投送自己采集的视频帧

使用此接口可以向 SDK 投送自己采集的视频帧，SDK 会将视频帧进行编码并通过自身的网络模块传输出去。

参数 TRTCVideoFrame 推荐下列填写方式（其他字段不需要填写）：

- pixelFormat：Windows 和 Android 平台仅支持 TRTCVideoPixelFormat_I420，iOS 和 Mac平台支持 TRTCVideoPixelFormat_I420 和 TRTCVideoPixelFormat_BGRA32。
- bufferType：推荐选择 TRTCVideoBufferType_Buffer。
- data：用于承载视频帧数据的 buffer。
- length：视频帧数据长度，如果 pixelFormat 设定为 I420 格式，length 可以按照如下公式计算：length = width × height × 3 / 2。

- width：视频图像的宽度，如 640 px。
- height：视频图像的高度，如 480 px。
- timestamp：时间戳，单位为毫秒（ms），请使用视频帧在采集时被记录下来的时间戳（可以在采集到一帧视频帧之后，通过调用 generateCustomPTS 获取时间戳）。

参考文档：自定义采集和渲染。

参数	描述
frame	视频数据，支持 I420 格式数据。
streamType	用于指定视频流类型，TRTCVideoStreamTypeBig：高清大画面；TRTCVideoStreamTypeSub：辅路画面。

注意

1. 推荐您在采集到的一帧视频帧后，即调用 generateCustomPTS 接口获取该帧的 timestamp 数值，这样可以获得最佳的音画同步效果。
2. SDK 最终编码出的视频帧率并不是由您调用本接口的频率决定的，而是由您在 setVideoEncoderParam 中所设置的 FPS 决定的。
3. 请尽量保持本接口的调用间隔是均匀的，否则会导致编码器输出帧率不稳或者音画不同步等问题。
4. iOS 和 Mac平台目前支持传入 TRTCVideoPixelFormat_I420 或 TRTCVideoPixelFormat_BGRA32 格式的视频帧。
5. Windows 和 Android 平台目前仅支持传入 TRTCVideoPixelFormat_I420 格式的视频帧。

enableCustomAudioCapture

enableCustomAudioCapture

```
void enableCustomAudioCapture (bool enable)
```

启用音频自定义采集模式

开启该模式后，SDK 不在运行原有的音频采集流程，即不再继续从麦克风采集音频数据，而是只保留音频编码和发送能力。
您需要通过 sendCustomAudioData 不断地向 SDK 塞入自己采集的音频数据。

参数	描述

enable	是否启用，默认值：false。

注意
由于回声抵消（AEC）需要严格的控制声音采集和播放的时间，所以开启自定义音频采集后，AEC 能力可能会失效。

sendCustomAudioData

sendCustomAudioData

```
void sendCustomAudioData (TRTCAudioFrame* frame)
```

向 SDK 投送自己采集的音频数据

参数 TRTCAudioFrame 推荐下列填写方式（其他字段不需要填写）：

- audioFormat：音频数据格式，仅支持 TRTCAudioFrameFormatPCM。
- data：音频帧 buffer。音频帧数据只支持 PCM 格式，支持[5ms ~ 100ms]帧长，推荐使用 20ms 帧长，长度计算方法：【48000采样率、单声道的帧长度：48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节】。
- sampleRate：采样率，支持：16000、24000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- timestamp：时间戳，单位为毫秒（ms），请使用音频帧在采集时被记录下来的时间戳（可以在采集到一帧音频帧之后，通过调用 generateCustomPTS 获取时间戳）。

参考文档：自定义采集和渲染 。

参数	描述
frame	音频数据

注意
请您精准地按每帧时长的间隔调用本接口，数据投送间隔不均匀时极易触发声音卡顿。

enableMixExternalAudioFrame

enableMixExternalAudioFrame

```
void enableMixExternalAudioFrame (bool enablePublish
                                  bool enablePlayout)
```

启用/关闭自定义音轨

开启后，您可以通过本接口向 SDK 混入一条自定义的音轨。通过两个布尔型参数，您可以控制该音轨是否要在远端和本地播放。

参数	描述
enablePlayout	控制混入的音轨是否要在本地播放，默认值：false。
enablePublish	控制混入的音轨是否要在远端播放，默认值：false。

注意
如果您指定参数 enablePublish 和 enablePlayout 均为 false，代表完全关闭您的自定义音轨。

mixExternalAudioFrame

mixExternalAudioFrame

```
int mixExternalAudioFrame (TRTCAudioFrame* frame)
```

向 SDK 混入自定义音轨

调用该接口之前，您需要先通过 enableMixExternalAudioFrame 开启自定义音轨，之后就可以通过本接口将自己的音轨以 PCM 格式混入到 SDK 中。
理想情况下，我们期望您的代码能够以非常均匀的速度向 SDK 提供音轨数据。但我们也非常清楚，完美的调用间隔是一个巨大的挑战。
所以 SDK 内部会开启一个音轨数据的缓冲区，该缓冲区的作用类似一个“蓄水池”，它能够暂存您传入的音轨数据，平抑由于接口调用间隔的抖动问题。
本接口的返回值代表这个音轨缓冲区的大小，单位是毫秒（ms），比如：如果该接口返回 50，则代表当前的音轨缓冲区有 50ms 的音轨数据。因此只要您在 50ms 内再次调用本接口，SDK 就能保证您混入的音轨数

据是连续的。
当您调用该接口后，如果发现返回值 > 100ms，则可以等待一帧音频帧的播放时间之后再次调用；如果返回值 < 100ms，则代表缓冲区比较小，您可以再次混入一些音轨数据以确保音轨缓冲区的大小维持在“安全水位”以上。

参数 TRTCAudioFrame 推荐下列填写方式（其他字段不需要填写）：

- data：音频帧 buffer。音频帧数据只支持 PCM 格式，支持[5ms ~ 100ms]帧长，推荐使用 20ms 帧长，长度计算方法：【48000采样率、单声道的帧长度：48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节】。
- sampleRate：采样率，支持：16000、24000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- timestamp：时间戳，单位为毫秒（ms），请使用音频帧在采集时被记录下来的时间戳（可以在获得一帧音频帧之后，通过调用 generateCustomPTS 获得时间戳）。

参数	描述
frame	音频数据

注意
请您精准地按每帧时长的间隔调用本接口，数据投送间隔不均匀时极易触发声音卡顿。

返回值说明：

>= 0 缓冲的长度，单位：ms。< 0 错误（-1 未启用 mixExternalAudioFrame）

setMixExternalAudioVolume

setMixExternalAudioVolume

```
void setMixExternalAudioVolume(int publishVolume, int playoutVolume)
```

设置推流时混入外部音频的推流音量和播放音量

参数	描述
playoutVolum	设置的播放音量大小，范围0 – 100，-1表示不改变。

e	
publishVolume	设置的推流音量大小，范围0 – 100，-1表示不改变。

generateCustomPTS

generateCustomPTS

生成自定义采集时的时间戳

本接口仅适用于自定义采集模式，用于解决音视频帧的采集时间（capture time）和投送时间（send time）不一致所导致的音画不同步问题。

当您通过 sendCustomVideoData 或 sendCustomAudioData 等接口进行自定义视频或音频采集时，请按照如下操作使用该接口：
1. 首先，在采集到一帧视频或音频帧时，通过调用本接口获得当时的 PTS 时间戳。
2. 之后可以将该视频或音频帧送入您使用的前处理模块（如第三方美颜组件，或第三方音效组件）。
3. 在真正调用 sendCustomVideoData 或 sendCustomAudioData 进行投送时，请将该帧在采集时记录的 PTS 时间戳赋值给 TRTCVideoFrame 或 TRTCAudioFrame 中的 timestamp 字段。

返回值说明：

时间戳（单位：ms）

enableLocalVideoCustomProcess

enableLocalVideoCustomProcess

int enableLocalVideoCustomProcess	(bool enable TRTCVideoPixelFormat pixelFormat TRTCVideoBufferType bufferType)

.1 开启视频第三方美颜

开启后，您可以通过 ITRTCVideoFrameCallback 获取到指定像素格式和视频数据结构类型的图像帧。

参数	描述
bufferType	指定视频数据结构类型。
enable	是否启用视频第三方美颜，默认为关闭状态。
pixelFormat	指定回调的像素格式。

返回值说明：

0：成功；<0：错误

setLocalVideoCustomProcessCallback

setLocalVideoCustomProcessCallback

```
void setLocalVideoCustomProcessCallback(ITRTCVideoFrameCallback* callback)
```

.2 设置第三方美颜的视频数据回调

设置该回调之后，SDK 会把采集到的视频帧通过您设置的 callback 回调出来，用于第三方美颜组件进行二次处理，之后 SDK 会将处理后的视频帧进行编码和发送。

参数	描述
callback	自定义美颜回调，详见 ITRTCVideoFrameCallback 。

setLocalVideoRenderCallback

setLocalVideoRenderCallback

```
int setLocalVideoRenderCallback(TRTCVideoPixelFormat pixelFormat
                                TRTCVideoBufferType bufferType
                                ITRTCVideoRenderCallback* callback)
```

设置本地视频自定义渲染回调

设置该回调之后，SDK 内部会跳过原来的渲染流程，并把采集到的数据回调出来，您需要自己完成画面渲染。

- 您可以通过调用 setLocalVideoRenderCallback(TRTCVideoPixelFormat_Unknown, TRTCVideoBufferType_Unknown, nullptr) 停止回调。
- iOS、Mac、Windows 平台目前仅支持回调 TRTCVideoPixelFormat_I420 或 TRTCVideoPixelFormat_BGRA32 像素格式的视频帧。
- Android 平台目前仅支持传入 TRTCVideoPixelFormat_I420，TRTCVideoPixelFormat_RGBA32 或 TRTCVideoPixelFormat_Texture_2D 像素格式的视频帧。

参数	描述
bufferType	指定视频数据结构类型，目前只支持 TRTCVideoBufferType_Buffer
callback	自定义渲染回调
pixelFormat	指定回调的像素格式

返回值说明：

0：成功；<0：错误

setRemoteVideoRenderCallback

setRemoteVideoRenderCallback

```
int setRemoteVideoRenderCallback(const char* userId,
                                 TRTCVideoPixelFormat pixelFormat,
                                 TRTCVideoBufferType bufferType,
                                 ITRTCVideoRenderCallback* callback)
```

设置远端视频自定义渲染回调

设置该回调之后，SDK 内部会跳过原来的渲染流程，并把采集到的数据回调出来，您需要自己完成画面渲染。

- 您可以通过调用 setRemoteVideoRenderCallback(TRTCVideoPixelFormat_Unknown, TRTCVideoBufferType_Unknown, nullptr) 停止回调。

- iOS、Mac、Windows 平台目前仅支持回调 TRTCVideoPixelFormat_I420 或 TRTCVideoPixelFormat_BGRA32 像素格式的视频帧。
- Android 平台目前仅支持传入 TRTCVideoPixelFormat_I420，TRTCVideoPixelFormat_RGBA32 或 TRTCVideoPixelFormat_Texture_2D 像素格式的视频帧。

参数	描述
bufferType	指定视频数据结构类型，目前只支持 TRTCVideoBufferType_Buffer
callback	自定义渲染回调
pixelFormat	指定回调的像素格式
userId	远端用户id

注意
实际使用时，需要先调用 startRemoteView(userid, nullptr) 来获取远端用户的视频流（view 设置为 nullptr 即可），否则不会有数据回调出来。

返回值说明：

0：成功；<0：错误

setAudioFrameCallback

setAudioFrameCallback

int setAudioFrameCallback (ITRTCAudioFrameCallback* callback)

设置音频数据自定义回调

设置该回调之后，SDK 内部会把音频数据（PCM 格式）回调出来，包括：

- onCapturedAudioFrame：本地麦克风采集到的音频数据回调
- onLocalProcessedAudioFrame：本地采集并经过音频模块前处理后的音频数据回调
- onPlayAudioFrame：混音前的每一路远程用户的音频数据
- onMixedPlayAudioFrame：将各路音频混合之后并最终要由系统播放出的音频数据回调

注意
设置回调为空即代表停止自定义音频回调，反之，设置回调不为空则代表启动自定义音频回调。

setCapturedAudioFrameCallbackFormat

setCapturedAudioFrameCallbackFormat

int setCapturedAudioFrameCallbackFormat (TRTCAudioFrameCallbackFormat* format)

设置本地麦克风采集出的音频帧回调格式

本接口用于设置 onCapturedAudioFrame 回调出来的 AudioFrame 的格式：

- sampleRate：采样率，支持：16000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- samplesPerCall：采样点数，定义回调数据帧长。帧长必须为 10ms 的整数倍。

如果希望用毫秒数计算回调帧长，则将毫秒数转换成采样点数的公式为：采样点数 = 毫秒数 * 采样率 / 1000。
举例：48000 采样率希望回调 20ms 帧长的数据，则采样点数应该填：960 = 20 * 48000 / 1000。

参数	描述
format	音频数据回调格式。

注意
最终回调的帧长度是以字节为单位，采样点数转换成字节数的计算公式为：字节数 = 采样点数 * channel * 2（位宽）举例：48000 采样率，双声道，20ms 帧长，采样点数为 960，字节数为 3840 = 960 * 2 * 2

返回值说明：

0：成功；<0：错误

setLocalProcessedAudioFrameCallbackFormat

setLocalProcessedAudioFrameCallbackFormat

int (TRTCAudioFrameCallbackFormat* f

setLocalProcessedAudioFrameCallbackFormat

设置经过前处理后的本地音频帧回调格式

本接口用于设置 onLocalProcessedAudioFrame 回调出来的 AudioFrame 的格式：

- sampleRate：采样率，支持：16000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。
- samplesPerCall：采样点数，定义回调数据帧长。帧长必须为 10ms 的整数倍。

如果希望用毫秒数计算回调帧长，则将毫秒数转换成采样点数的公式为：采样点数 = 毫秒数 * 采样率 / 1000。
举例：48000 采样率希望回调20ms帧长的数据，则采样点数应该填: 960 = 20 * 48000 / 1000。

参数	描述
format	音频数据回调格式。

注意
最终回调的帧长度是以字节为单位，采样点数转换成字节数的计算公式为：字节数 = 采样点数 * channel * 2（位宽）举例：48000 采样率，双声道，20ms 帧长，采样点数为 960，字节数为 3840 = 960 * 2 * 2

返回值说明：

0：成功；<0：错误

setMixedPlayAudioFrameCallbackFormat

setMixedPlayAudioFrameCallbackFormat

int setMixedPlayAudioFrameCallbackFormat (TRTCAudioFrameCallbackFormat* format

设置最终要由系统播放出的音频帧回调格式

本接口用于设置 onMixedPlayAudioFrame 回调出来的 AudioFrame 的格式：

- sampleRate：采样率，支持：16000、32000、44100、48000。
- channel：声道数（如果是立体声，数据是交叉的），单声道：1； 双声道：2。

- samplesPerCall：采样点数，定义回调数据帧长。帧长必须为 10ms 的整数倍。

如果希望用毫秒数计算回调帧长，则将毫秒数转换成采样点数的公式为：采样点数 = 毫秒数 * 采样率 / 1000。
举例：48000 采样率希望回调20ms帧长的数据，则采样点数应该填: 960 = 20 * 48000 / 1000。

参数	描述
format	音频数据回调格式。

注意
最终回调的帧长度是以字节为单位，采样点数转换成字节数的计算公式为：字节数 = 采样点数 * channel * 2（位宽）举例：48000 采样率，双声道，20ms 帧长，采样点数为 960，字节数为 3840 = 960 * 2 * 2

返回值说明：

0：成功；<0：错误

enableCustomAudioRendering

enableCustomAudioRendering

```
void enableCustomAudioRendering (bool enable)
```

开启音频自定义播放

如果您需要外接一些特定的音频设备，或者希望自己掌控音频的播放逻辑，您可以通过该接口启用音频自定义播放。
启用音频自定义播放后，SDK 将不再调用系统的音频接口播放数据，您需要通过 getCustomAudioRenderingFrame 获取 SDK 要播放的音频帧并自行播放。

参数	描述
enable	是否启用音频自定义播放，默认为关闭状态。

注意

需要您在进入房间前设置才能生效，暂不支持进入房间后再设置。

getCustomAudioRenderingFrame

getCustomAudioRenderingFrame

void getCustomAudioRenderingFrame (TRTCAudioFrame* audioFrame)

获取可播放的音频数据

调用该接口之前，您需要先通过 enableCustomAudioRendering 开启音频自定义播放。

参数 TRTCAudioFrame 推荐下列填写方式（其他字段不需要填写）：

- sampleRate：采样率，必填，支持 16000、24000、32000、44100、48000。
- channel：声道数，必填，单声道请填1，双声道请填2，双声道时数据是交叉的。
- data：用于获取音频数据的 buffer。需要您根据一帧音频帧的帧长度分配好 data 的内存大小。获取的 PCM 数据支持 10ms 或 20ms 两种帧长，推荐使用 20ms 的帧长。

计算公式为：采样率 x 播放时长 x 声道数量 x 2（每个样点的字节数），例如：48000采样率、单声道、且播放时长为 20ms 的一帧音频帧的 buffer 大小为 48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节。

参数	描述
audioFrame	音频数据帧。

注意

1. 参数 audioFrame 中的 sampleRate、channel 需提前设置好，同时分配好所需读取帧长的 data 空间。
2. SDK 内部会根据 sampleRate 和 channel 自动填充 data 数据。
3. 建议由系统的音频播放线程直接驱动该函数的调用，在播放完一帧音频之后，即调用该接口获取下一帧可播放的音频数据。

sendCustomCmdMsg

sendCustomCmdMsg

```
bool sendCustomCmdMsg(uint32_t cmdId
                      const uint8_t* data
                      uint32_t dataSize
                      bool reliable
                      bool ordered)
```

使用 UDP 通道发送自定义消息给房间内所有用户

该接口可以让您借助 TRTC 的 UDP 通道，向当前房间里的其他用户广播自定义数据，以达到传输信令的目的。
房间中的其他用户可以通过 ITRTCCloudCallback 中的 onRecvCustomCmdMsg 回调接收消息。

参数	描述
cmdID	消息 ID，取值范围为1 – 10。
data	待发送的消息，单个消息的最大长度被限制为 1KB。
ordered	是否要求有序，即是否要求接收端的数据包顺序和发送端的数据包顺序一致（这会带来一定的接收延时）。
reliable	是否可靠发送，可靠发送可以获得更高的发送成功率，但可靠发送比不可靠发送会带来更大的接收延迟。

注意

1. 发送消息到房间内所有用户（暂时不支持 Web/小程序端），每秒最多能发送30条消息。
2. 每个包最大为 1KB，超过则很有可能会被中间路由器或者服务器丢弃。
3. 每个客户端每秒最多能发送总计 8KB 数据。
4. 请将 reliable 和 ordered 同时设置为 true 或同时设置为 false，暂不支持交叉设置。
5. 强烈建议您将不同类型的消息设定为不同的 cmdID，这样可以在要求有序的情况下减小消息时延。
6. 目前仅支持主播身份。

返回值说明：

true：消息已经发出；false：消息发送失败。

sendSEIMsg

sendSEIMsg

```
bool sendSEIMsg(const uint8_t* data,
                uint32_t dataSize,
                int32_t repeatCount)
```

使用 SEI 通道发送自定义消息给房间内所有用户

该接口可以让您借助 TRTC 的 SEI 通道，向当前房间里的其他用户广播自定义数据，已达到传输信令的目的。

视频帧的头部有一个叫做 SEI 的头部数据块，该接口的原理就是利用这个被称为 SEI 的头部数据块，将您要发送的自定义信令嵌入其中，使其同视频帧一并发送出去。
因此，与 sendCustomCmdMsg 相比，SEI 通道传输的信令具有更好的兼容性：信令可以伴随着视频帧一直传输到直播 CDN 上。
不过，由于视频帧头部的数据块不能太大，建议您使用该接口时，尽量将信令控制在几个字节的大小。

最常见的用法是把自定义的时间戳（timestamp）用本接口嵌入视频帧中，实现消息和画面的完美对齐（比如：教育场景下的课件和视频信号的对齐）。
房间中的其他用户可以通过 ITRTCCloudCallback 中的 onRecvSEIMsg 回调接收消息。

参数	描述
data	待发送的数据，最大支持 1KB（1000字节）的数据大小
repeatCount	发送数据次数

注意
本接口有以下限制：
1. 数据在接口调用完后不会被即时发送出去，而是从下一帧视频帧开始与视频帧一起发送。
2. 发送消息到房间内所有用户，每秒最多能发送 30 条消息（与 sendCustomCmdMsg 共享限制）。
3. 每个包最大为 1KB，若发送大量数据，会导致视频码率增大，可能导致视频画质下降甚至卡顿（与 sendCustomCmdMsg 共享限制）。

4. 每个客户端每秒最多能发送总计8KB数据（与 sendCustomCmdMsg 共享限制）。
5. 若指定多次发送（repeatCount > 1），则数据会被带在后续的连续 repeatCount 个视频帧中发送出去，同样会导致视频码率增大。
6. 如果 repeatCount > 1，多次发送，接收消息 onRecvSEIMsg 回调也可能会收到多次相同的消息，需要去重。

返回值说明：

true：消息已通过限制，等待后续视频帧发送；false：消息被限制发送

startSpeedTest

startSpeedTest

int startSpeedTest (const TRTCSpeedTestParams & params)

开始进行网速测试（进入房间前使用）

参数	描述
params	测速选项

注意

1. 测速过程将产生少量的基础服务费用，详见 计费概述 > 基础服务 文档说明。
2. 请在进入房间前进行网速测试，在房间中网速测试会影响正常的音视频传输效果，而且由于干扰过多，网速测试结果也不准确。
3. 同一时间只允许一项网速测试任务运行。

返回值说明：

接口调用结果，< 0：失败

stopSpeedTest

stopSpeedTest

停止网络测速

getSDKVersion

getSDKVersion

获取 SDK 版本信息

setLogLevel

setLogLevel

void setLogLevel （TRTCLogLevel level)

设置 Log 输出级别

参数	描述
level	参见 TRTCLogLevel，默认值：TRTCLogLevelNone

setConsoleEnabled

setConsoleEnabled

void setConsoleEnabled (bool enabled)

启用/禁用控制台日志打印

参数	描述
enabled	指定是否启用，默认：禁止状态。

setLogCompressEnabled

setLogCompressEnabled

void setLogCompressEnabled (bool enabled)

启用/禁用日志的本地压缩

开启压缩后，Log 存储体积明显减小，但需要腾讯云提供的 Python 脚本解压后才能阅读。
禁用压缩后，Log 采用明文存储，可以直接用记事本打开阅读，但占用空间较大。

参数	描述
enabled	指定是否启用，默认为启动状态

setLogDirPath

setLogDirPath

void setLogDirPath (const char* path)

设置本地日志的保存路径

通过该接口您可以更改 SDK 本地日志的默认存储路径，SDK 默认的本地日志的存储位置：

- Windows 平台：在 C:/Users/[系统用户名]/AppData/Roaming/liteav/log，即 %appdata%/liteav/log 下。
- iOS 或 Mac 平台：在 sandbox Documents/log 下。
- Android 平台：在 /app私有目录/files/log/liteav/ 下。

参数	描述
path	存储日志的路径

注意
请务必在所有其他接口之前调用，并且保证您指定的目录是存在的，并且您的应用程序拥有对该目录的读写权限。

setLogCallback

setLogCallback

void setLogCallback (ITRTCLogCallback* callback)

设置日志回调

showDebugView

showDebugView

void showDebugView (int showType)

显示仪表盘

“仪表盘”是位于视频渲染控件之上的一个半透明的调试信息浮层，用于展示音视频信息和事件信息，便于对接和调试。

参数	描述
showType	0：不显示；1：显示精简版（仅显示音视频信息）；2：显示完整版（包含音视频信息和事件信息）。

callExperimentalAPI

callExperimentalAPI

char* callExperimentalAPI (const char *jsonStr)

调用实验性接口

enablePayloadPrivateEncryption

enablePayloadPrivateEncryption

int enablePayloadPrivateEncryption	(bool enabled const TRTCPayloadPrivateEncryptionConfig & cor

开启或关闭媒体流私有加密

在安全要求较高的场景下，TRTC 建议您在加入房间前，调用 enablePayloadPrivateEncryption 方法开启媒体流私有加密。
用户退出房间后，SDK 会自动关闭私有加密。如需重新开启私有加密，您需要在用户再次加入房间前调用该方法。

参数	描述
config	配置媒体流私有加密的算法和密钥，参见 TRTCPayloadPrivateEncryptionConfig 。
enabled	是否开启媒体流私有加密。

注意
TRTC 已经内置对媒体流进行加密后再传输，启用媒体流私有加密后将使用您传入的密匙与初始向量进行再次加密。

返回值说明：

接口调用结果，0: 方法调用成功， -1: 传入参数无效， -2: 功能已过期。若需解锁：请前往中国大陆站点开通 TRTC 旗舰版套餐 ，并填写 申请 ，审核通过后方可使用。

TRTCCloudCallback

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: ITRTCCloudCallback @ TXLiteAVSDK

Function: 腾讯云实时音视频的事件回调接口

TRTCCloudCallback

ITRTCCloudCallback

函数列表	描述
onError	错误事件回调
onWarning	警告事件回调
onEnterRoom	进入房间成功与否的事件回调
onExitRoom	离开房间的事件回调
onSwitchRole	切换角色的事件回调
onSwitchRoom	切换房间的结果回调
onConnectOtherRoom	请求跨房通话的结果回调
onDisconnectOtherRoom	结束跨房通话的结果回调
onUpdateOtherRoomForwardMode	更改跨房主播上行能力的结果回调
onRemoteUserEnterRoom	有用户加入当前房间
onRemoteUserLeaveRoom	有用户离开当前房间
onUserVideoAvailable	某远端用户发布/取消了主路视频画面
onUserSubStreamAvailable	某远端用户发布/取消了辅路视频画面

onUserAudioAvailable	某远端用户发布/取消了自己的音频
onFirstVideoFrame	SDK 开始渲染自己本地或远端用户的首帧画面
onFirstAudioFrame	SDK 开始播放远端用户的首帧音频
onSendFirstLocalVideoFrame	自己本地的首个视频帧已被发布出去
onSendFirstLocalAudioFrame	自己本地的首个音频帧已被发布出去
onRemoteVideoStatusUpdated	远端视频状态变化的事件回调
onRemoteAudioStatusUpdated	远端音频状态变化的事件回调
onUserVideoSizeChanged	用户视频大小发生改变回调
onNetworkQuality	网络质量的实时统计回调
onStatistics	音视频技术指标的实时统计回调
onSpeedTestResult	网速测试的结果回调
onConnectionLost	SDK 与云端的连接已经断开
onTryToReconnect	SDK 正在尝试重新连接到云端
onConnectionRecovery	SDK 与云端的连接已经恢复
onCameraDidReady	摄像头准备就绪
onMicDidReady	麦克风准备就绪
onUserVoiceVolume	音量大小的反馈回调
onDeviceChange	本地设备的通断状态发生变化（仅适用于桌面系统）
onAudioDeviceCaptureVolumeChanged	当前麦克风的系统采集音量发生变化
onAudioDevicePlayoutVolumeChanged	当前系统的播放音量发生变化
onSystemAudioLoopbackError	系统声音采集是否被成功开启的事件回调（仅适用于 Mac 系统）

onTestMicVolume	测试麦克风时的音量回调
onTestSpeakerVolume	测试扬声器时的音量回调
onRecvCustomCmdMsg	收到自定义消息的事件回调
onMissCustomCmdMsg	自定义消息丢失的事件回调
onRecvSEIMsg	收到 SEI 消息的回调
onStartPublishing	开始向腾讯云直播 CDN 上发布音视频流的事件回调
onStopPublishing	停止向腾讯云直播 CDN 上发布音视频流的事件回调
onStartPublishCDNStream	开始向非腾讯云 CDN 上发布音视频流的事件回调
onStopPublishCDNStream	停止向非腾讯云 CDN 上发布音视频流的事件回调
onSetMixTranscodingConfig	设置云端混流的排版布局和转码参数的事件回调
onStartPublishMediaStream	开始发布媒体流的事件回调
onUpdatePublishMediaStream	更新媒体流的事件回调
onStopPublishMediaStream	停止媒体流的事件回调
onCdnStreamStateChanged	RTMP/RTMPS 推流状态发生改变回调
onScreenCaptureStarted	屏幕分享开启的事件回调
onScreenCapturePaused	屏幕分享暂停的事件回调
onScreenCaptureResumed	屏幕分享恢复的事件回调
onScreenCaptureStoped	屏幕分享停止的事件回调
onScreenCaptureCovered	屏幕分享的目标窗口被遮挡的事件回调（仅适用于 Windows 操作系统）
onLocalRecordBegin	本地录制任务已经开始的事件回调
onLocalRecording	本地录制任务正在进行中的进展事件回调
onLocalRecordFragment	本地录制分片的事件回调

版权所有：腾讯云计算（北京）有限责任公司

onLocalRecordComplete	本地录制任务已经结束的事件回调
onSnapshotComplete	本地截图完成的事件回调
onUserEnter	有主播加入当前房间（已废弃）
onUserExit	有主播离开当前房间（已废弃）
onAudioEffectFinished	音效播放已结束（已废弃）
onPlayBGMBegin	开始播放背景音乐（已废弃）
onPlayBGMProgress	背景音乐的播放进度回调（已废弃）
onPlayBGMComplete	背景音乐播放已经结束（已废弃）
onSpeedTest	服务器测速的结果回调（已废弃）

ITRTCVideoRenderCallback

函数列表	描述
onRenderVideoFrame	自定义视频渲染回调

ITRTCVideoFrameCallback

函数列表	描述
onGLContextCreated	SDK 内部 OpenGL 环境已经创建的通知
onProcessVideoFrame	用于对接第三方美颜组件的视频处理回调
onGLContextDestroy	SDK 内部 OpenGL 环境被销毁的通知

ITRTCAudioFrameCallback

函数列表	描述
onCapturedAudioFrame	本地采集并经过音频模块前处理后的音频数据回调
onLocalProcessedAudioFrame	本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调

onPlayAudioFrame	混音前的每一路远程用户的音频数据
onMixedPlayAudioFrame	将各路待播放音频混合之后并在最终提交系统播放之前的数据回调

ITRTCLogCallback

函数列表	描述
onLog	本地 LOG 的打印回调

onError

onError

```
void onError (TXLiteAVError errCode
              const char* errMsg
              void* extraInfo)
```

错误事件回调

错误事件，表示 SDK 抛出的不可恢复的错误，比如进入房间失败或设备开启失败等。
参考文档：错误码表

参数	描述
errCode	错误码
errMsg	错误信息
extInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题

onWarning

onWarning

void onWarning	(TXLiteAVWarning warningCode const char* warningMsg void* extraInfo)

警告事件回调

警告事件，表示 SDK 抛出的提示性问题，比如视频出现卡顿或 CPU 使用率太高等。
参考文档：错误码表

参数	描述
extInfo	扩展信息字段，个别警告码可能会带额外的信息帮助定位问题
warningCode	警告码
warningMsg	警告信息

onEnterRoom

onEnterRoom

void onEnterRoom	(int result)

进入房间成功与否的事件回调

调用 TRTCCloud 中的 enterRoom() 接口执行进房操作后，会收到来自 ITRTCCloudCallback 的 onEnterRoom(result) 回调：

- 如果加入成功，回调 result 会是一个正数（result > 0），代表进入房间所消耗的时间，单位是毫秒（ms）。
- 如果加入失败，回调 result 会是一个负数（result < 0），代表失败原因的错误码。

进房失败的错误码含义请参见错误码表 。

参数	描述
resul	result > 0 时为进房耗时（ms），result < 0 时为进房错误码。

t	

注意

1. 在 Ver6.6 之前的版本，只有进房成功会抛出 onEnterRoom(result) 回调，进房失败由 onError() 回调抛出。
2. 在 Ver6.6 之后的版本：无论进房成功或失败，均会抛出 onEnterRoom(result) 回调，同时进房失败也会有 onError() 回调抛出。

onExitRoom

onExitRoom

```
void onExitRoom(int reason)
```

离开房间的事件回调

调用 TRTCCloud 中的 exitRoom 接口会执行退出房间的相关逻辑，例如释放音视频设备资源和编解码器资源等。
待 SDK 占用的所有资源释放完毕后，SDK 会抛出 onExitRoom() 回调通知到您。

如果您要再次调用 enterRoom 或者切换到其他的音视频 SDK，请等待 onExitRoom 回调到来后再执行相关操作。否则可能会遇到例如摄像头、麦克风设备被强占等各种异常问题。

参数	描述
reason	离开房间原因，0：主动调用 exitRoom 退出房间；1：被服务器踢出当前房间；2：当前房间整个被解散。

onSwitchRole

onSwitchRole

```
void onSwitchRole(TXLiteAVError errCode, const char* errMsg)
```

切换角色的事件回调

调用 TRTCCloud 中的 switchRole() 接口会切换主播和观众的角色，该操作会伴随一个线路切换的过程，待 SDK 切换完成后，会抛出 onSwitchRole() 事件回调。

参数	描述
errCode	错误码，ERR_NULL 代表切换成功，其他请参见 错误码 。
errMsg	错误信息。

onSwitchRoom

onSwitchRoom

```
void onSwitchRoom(TXLiteAVError errCode, const char* errMsg)
```

切换房间的结果回调

调用 TRTCCloud 中的 switchRoom() 接口可以让用户快速地从一个房间切换到另一个房间，待 SDK 切换完成后，会抛出 onSwitchRoom() 事件回调。

参数	描述
errCode	错误码，ERR_NULL 代表切换成功，其他请参见 错误码 。
errMsg	错误信息。

onConnectOtherRoom

onConnectOtherRoom

```
void onConnectOtherRoom(const char* userId, TXLiteAVError errCode)
```

const char* errMsg)

请求跨房通话的结果回调

调用 TRTCCloud 中的 connectOtherRoom() 接口会将两个不同房间中的主播拉通视频通话，也就是所谓的“主播PK”功能。
调用者会收到 onConnectOtherRoom() 回调来获知跨房通话是否成功，
如果成功，两个房间中的所有用户都会收到来自另一个房间中的 PK 主播的 onUserVideoAvailable() 回调。

参数	描述
errCode	错误码，ERR_NULL 代表切换成功，其他请参见 错误码 。
errMsg	错误信息。
userId	要跨房通话的另一个房间中的主播的用户 ID。

onDisconnectOtherRoom

onDisconnectOtherRoom

void onDisconnectOtherRoom (TXLiteAVError errCode, const char* errMsg)

结束跨房通话的结果回调

onUpdateOtherRoomForwardMode

onUpdateOtherRoomForwardMode

void onUpdateOtherRoomForwardMode (TXLiteAVError errCode, const char* errMsg)

更改跨房主播上行能力的结果回调

onRemoteUserEnterRoom

onRemoteUserEnterRoom

void onRemoteUserEnterRoom	(const char* userId)

有用户加入当前房间

出于性能方面的考虑，在 TRTC 两种不同的应用场景（即 AppScene，在 enterRoom 时通过第二个参数指定）下，该通知的行为会有差别：

- 直播类场景（TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom）：该场景下的用户区分主播和观众两种角色，只有主播进入房间时才会触发该通知，观众进入房间时不会触发该通知。
- 通话类场景（TRTCAppSceneVideoCall 和 TRTCAppSceneAudioCall）：该场景下的用户没有角色的区分（可认为都是主播），任何用户进入房间都会触发该通知。

参数	描述
userId	远端用户的用户标识。

注意

1. 事件回调 onRemoteUserEnterRoom 和 onRemoteUserLeaveRoom 只适用于维护当前房间里的“用户列表”，有此事件回调不代表一定有视频画面。
2. 如果需要显示远程画面，请监听代表某个用户是否有视频画面的 onUserVideoAvailable() 事件回调。

onRemoteUserLeaveRoom

onRemoteUserLeaveRoom

void onRemoteUserLeaveRoom	(const char* userId int reason)

有用户离开当前房间

与 onRemoteUserEnterRoom 相对应，在两种不同的应用场景（即 AppScene，在 enterRoom 时通过第二个参数指定）下，该通知的行为会有差别：

- 直播类场景（TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom）：只有主播离开房间时才会触发该通知，观众离开房间不会触发该通知。
- 通话类场景（TRTCAppSceneVideoCall 和 TRTCAppSceneAudioCall）：该场景下用户没有角色的区别，任何用户的离开都会触发该通知。

参数	描述
reason	离开原因，0表示用户主动退出房间，1表示用户超时退出，2表示被踢出房间，3表示主播因切换到观众退出房间。
userId	远端用户的用户标识。

onUserVideoAvailable

onUserVideoAvailable

```
void onUserVideoAvailable (const char* userId
                           bool available)
```

某远端用户发布/取消了主路视频画面

主路画面 一般被用于承载摄像头画面。当您收到 onUserVideoAvailable(userId, true) 通知时，表示该路画面已经有可播放的视频帧到达。
此时，您需要调用 startRemoteView 接口订阅该用户的远程画面，订阅成功后，您会继续收到该用户的首帧画面渲染回调 onFirstVideoFrame(userid)。

当您收到 onUserVideoAvailable(userId, false) 通知时，表示该路远程画面已经被关闭，关闭的原因可能是该用户调用了 muteLocalVideo 或 stopLocalPreview 。

参数	描述
available	该用户是否发布（或取消发布）了主路视频画面，true：发布；false：取消发布。
userId	远端用户的用户标识。

onUserSubStreamAvailable

onUserSubStreamAvailable

```
void
onUserSubStreamAvailable        (const char* userId

                                 bool available)
```

某远端用户发布/取消了辅路视频画面

“辅路画面”一般被用于承载屏幕分享的画面。当您收到 onUserSubStreamAvailable(userId, true) 通知时，表示该路画面已经有可播放的视频帧到达。
此时，您需要调用 startRemoteView 接口订阅该用户的远程画面，订阅成功后，您会继续收到该用户的首帧画面渲染回调 onFirstVideoFrame(userid)。

参数	描述
available	该用户是否发布（或取消发布）了辅路视频画面，true: 发布；false：取消发布。
userId	远端用户的用户标识。

注意
显示辅路画面使用的函数是 startRemoteView 而非 startRemoteSubStreamView，startRemoteSubStreamView 已经被废弃。

onUserAudioAvailable

onUserAudioAvailable

```
void onUserAudioAvailable       (const char* userId

                                 bool available)
```

某远端用户发布/取消了自己的音频

当您收到 onUserAudioAvailable(userId, true) 通知时，表示该用户发布了自己的声音，此时 SDK 的表现为：

- 在自动订阅模式下，您无需做任何操作，SDK 会自动播放该用户的声音。

- 在手动订阅模式下，您可以通过 muteRemoteAudio(userid, false) 来播放该用户的声音。

参数	描述
available	该用户是否发布（或取消发布）了自己的音频，true: 发布；false：取消发布。
userId	远端用户的用户标识。

注意
SDK 默认使用自动订阅模式，您可以通过 setDefaultStreamRecvMode 设置为手动订阅，但需要在您进入房间之前调用才生效。

onFirstVideoFrame

onFirstVideoFrame

```
void onFirstVideoFrame (const char* userId
                        const TRTCVideoStreamType streamType
                        const int width
                        const int height)
```

SDK 开始渲染自己本地或远端用户的首帧画面

SDK 会在渲染自己本地或远端用户的首帧画面时抛出该事件，您可以通过回调事件中的 userId 参数来判断事件来自于“本地”还是来自于“远端”。

- 如果 userId 为空值，代表 SDK 已经开始渲染自己本地的视频画面，不过前提是您已经调用了 startLocalPreview 或 startScreenCapture。
- 如果 userId 不为空，代表 SDK 已经开始渲染远端用户的视频画面，不过前提是您已经调用了 startRemoteView 订阅了该用户的视频画面。

参数	描述
height	画面的高度。
streamType	视频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。
userId	本地或远端的用户标识，如果 userId 为空值代表自己本地的首帧画面已到来，

	userId 不为空则代表远端用户的首帧画面已到来。
width	画面的宽度。

注意

1. 只有当您调用了 startLocalPreview 或 startScreenCapture 之后，才会触发自己本地的首帧画面事件回调。
2. 只有当您调用了 startRemoteView 或 startRemoteSubStreamView 之后，才会触发远端用户的首帧画面事件回调。

onFirstAudioFrame

onFirstAudioFrame

void onFirstAudioFrame (const char* userId)

SDK 开始播放远端用户的首帧音频

SDK 会在播放远端用户的首帧音频时抛出该事件，本地音频的首帧事件暂不抛出。

参数	描述
userId	远端用户的用户标识。

onSendFirstLocalVideoFrame

onSendFirstLocalVideoFrame

void onSendFirstLocalVideoFrame (const TRTCVideoStreamType streamType)

自己本地的首个视频帧已被发布出去

当您成功进入房间并通过 startLocalPreview 或 startScreenCapture 开启本地视频采集之后（开启采集和进入房间的先后顺序无影响），SDK 就会开始进行视频编码，并通过自身的网络模块向云端发布自己本地的视频数据。当 SDK 成功地向云端送出自己的第一帧视频数据帧以后，就会抛出 onSendFirstLocalVideoFrame 事件回调。

版权所有：腾讯云计算（北京）有限责任公司

参数	描述
streamType	视频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。

onSendFirstLocalAudioFrame

onSendFirstLocalAudioFrame

自己本地的首个音频帧已被发布出去

当您成功进入房间并通过 startLocalAudio 开启本地音频采集之后（开启采集和进入房间的先后顺序无影响），SDK 就会开始进行音频编码，并通过自身的网络模块向云端发布自己本地的音频数据。当 SDK 成功地向云端送出自己的第一帧音频数据帧以后，就会抛出 onSendFirstLocalAudioFrame 事件回调。

onRemoteVideoStatusUpdated

onRemoteVideoStatusUpdated

```
void onRemoteVideoStatusUpdated(const char* userId,
                                TRTCVideoStreamType streamType,
                                TRTCAVStatusType status,
                                TRTCAVStatusChangeReason reason,
                                void *extrainfo)
```

远端视频状态变化的事件回调

您可以通过此事件回调获取远端每一路画面的播放状态（包括 Playing、Loading 和 Stopped 三种状态），从而进行相应的 UI 展示。

参数	描述
extrainfo	额外信息。
reason	视频状态改变的原因。

status	视频状态：包括 Playing、Loading 和 Stopped 三种状态。
streamType	视频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。
userId	用户标识。

onRemoteAudioStatusUpdated

onRemoteAudioStatusUpdated

```
void onRemoteAudioStatusUpdated(const char* userId,
                                TRTCAVStatusType status,
                                TRTCAVStatusChangeReason reason,
                                void *extrainfo)
```

远端音频状态变化的事件回调

您可以通过此事件回调获取远端每一路音频的播放状态（包括 Playing、Loading 和 Stopped 三种状态），从而进行相应的 UI 展示。

参数	描述
extrainfo	额外信息。
reason	音频状态改变的原因。
status	音频状态：包括 Playing、Loading 和 Stopped 三种状态。
userId	用户标识。

onUserVideoSizeChanged

onUserVideoSizeChanged

```
void onUserVideoSizeChanged(const char* userId
```

```
                                TRTCVideoStreamType streamType
                                int newWidth
                                int newHeight)
```

用户视频大小发生改变回调

当您收到 onUserVideoSizeChanged(userId, streamtype, newWidth, newHeight) 通知时，表示该路画面大小发生了调整，调整的原因可能是该用户调用了 setVideoEncoderParam 或者 setSubStreamEncoderParam 重新设置了画面尺寸。

参数	描述
newHeight	视频流的高度（像素）。
newWidth	视频流的宽度（像素）。
streamType	视频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。
userId	用户标识。

onNetworkQuality

onNetworkQuality

```
void onNetworkQuality (TRTCQualityInfo localQuality
                       TRTCQualityInfo* remoteQuality
                       uint32_t remoteQualityCount)
```

网络质量的实时统计回调

该统计回调每间隔2秒抛出一次，用于通知 SDK 感知到的当前网络的上行和下行质量。
SDK 会使用一组内嵌的自研算法对当前网络的延迟高低、带宽大小以及稳定情况进行评估，并计算出一个评估结果：如果评估结果为 1（Excellent）代表当前的网络情况非常好，如果评估结果为 6（Down）代表当前网络无法支撑 TRTC 的正常通话。

参数	描述

localQuality	上行网络质量。
remoteQuality	下行网络质量，代表数据流历经“远端->云端->本端”这样一条完整的传输链路后，最终在本端统计到的数据质量。因此，这里的下行网络质量代表的是远端上行链路与本端下行链路共同影响的结果。

注意

暂时无法通过该接口单独判定远端用户的上行质量。

onStatistics

onStatistics

void onStatistics (const TRTCStatistics & statistics)

音视频技术指标的实时统计回调

该统计回调每间隔2秒抛出一次，用于通知 SDK 内部音频、视频以及网络相关的专业技术指标，这些信息在 TRTCStatistics 均有罗列：

- 视频统计信息：视频的分辨率（resolution）、帧率（FPS）和比特率（bitrate）等信息。
- 音频统计信息：音频的采样率（samplerate）、声道（channel）和比特率（bitrate）等信息。
- 网络统计信息：SDK 和云端一次往返（SDK > Cloud > SDK）的网络耗时（rtt）、丢包率（loss）、上行流量（sentBytes）和下行流量（receivedBytes）等信息。

参数	描述
statistics	统计数据，包括自己本地的统计信息和远端用户的统计信息，详情请参考 TRTCStatistics 。

注意

如果您只需要获知当前网络质量的好坏，并不需要花太多时间研究本统计回调，更推荐您使用 onNetworkQuality 来解决问题。

onSpeedTestResult

onSpeedTestResult

void onSpeedTestResult	(const TRTCSpeedTestResult & result)

网速测试的结果回调

该统计回调由 startSpeedTest: 触发。

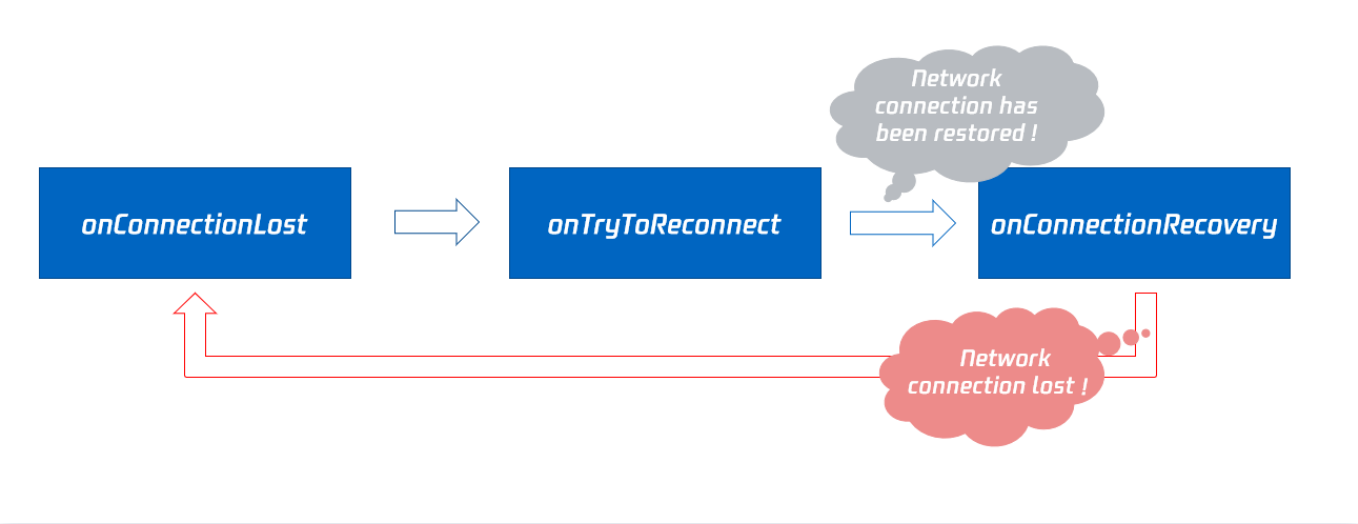
参数	描述
result	网速测试结果，包括丢包、往返延迟、上下行的带宽速率，详情请参见 TRTCSpeedTestResult 。

onConnectionLost

onConnectionLost

SDK 与云端的连接已经断开

SDK 会在跟云端的连接断开时抛出此事件回调，导致断开的原因大多是网络不可用或者网络切换所致，比如用户在通话中走进电梯时就可能会遇到此事件。
在抛出此事件之后，SDK 会努力跟云端重新建立连接，重连过程中会抛出 onTryToReconnect ，连接恢复后会抛出 onConnectionRecovery 。
所以，SDK 会在如下三个连接相关的事件中按如下规律切换：



onTryToReconnect

onTryToReconnect

SDK 正在尝试重新连接到云端

SDK 会在跟云端的连接断开时抛出 onConnectionLost，之后会努力跟云端重新建立连接并抛出本事件，连接恢复后会抛出 onConnectionRecovery。

onConnectionRecovery

onConnectionRecovery

SDK 与云端的连接已经恢复

SDK 会在跟云端的连接断开时抛出 onConnectionLost，之后会努力跟云端重新建立连接并抛出 onTryToReconnect，连接恢复后会抛出本事件回调。

onCameraDidReady

onCameraDidReady

摄像头准备就绪

当您调用 startLocalPreivew 之后，SDK 会尝试启动摄像头，如果摄像头能够启动成功就会抛出本事件。如果启动失败，大概率是因为当前应用没有获得访问摄像头的权限，或者摄像头当前正在被其他程序独占使用中。
您可以通过捕获 onError 事件回调获知这些异常情况并通过 UI 界面提示用户。

onMicDidReady

onMicDidReady

麦克风准备就绪

当您调用 startLocalAudio 之后，SDK 会尝试启动麦克风，如果麦克风能够启动成功就会抛出本事件。如果启动失败，大概率是因为当前应用没有获得访问麦克风的权限，或者麦克风当前正在被其他程序独占使用中。
您可以通过捕获 onError 事件回调获知这些异常情况并通过 UI 界面提示用户。

版权所有：腾讯云计算（北京）有限责任公司

onUserVoiceVolume

onUserVoiceVolume

```
void onUserVoiceVolume(TRTCVolumeInfo* userVolumes,
                       uint32_t userVolumesCount,
                       uint32_t totalVolume)
```

音量大小的反馈回调

SDK 可以评估每一路音频的音量大小，并每隔一段时间抛出该事件回调，您可以根据音量大小在 UI 上做出相应的提示，比如 波形图 或 音量槽 。
要完成这个功能， 您需要先调用 enableAudioVolumeEvaluation 开启这个能力并设定事件抛出的时间间隔。
需要补充说明的是，无论当前房间中是否有人说话，SDK 都会按照您设定的时间间隔定时抛出此事件回调。

参数	描述
totalVolume	所有远端用户的总音量大小，取值范围 0 – 100。
userVolumes	是一个数组，用于承载所有正在说话的用户的音量大小，取值范围 0 – 100。

注意
userVolumes 为一个数组，对于数组中的每一个元素，当 userId 为空时表示本地麦克风采集的音量大小，当 userId 不为空时代表远端用户的音量大小。

onDeviceChange

onDeviceChange

```
void onDeviceChange(const char* deviceId,
                    TRTCDeviceType type)
```

TRTCDeviceState state)

本地设备的通断状态发生变化（仅适用于桌面系统）

当本地设备（包括摄像头、麦克风以及扬声器）被插入或者拔出时，SDK 便会抛出此事件回调。

参数	描述
deviceId	设备 ID。
deviceType	设备类型。
state	通断状态，0：设备已添加；1：设备已被移除；1：设备已启用。

onAudioDeviceCaptureVolumeChanged

onAudioDeviceCaptureVolumeChanged

void onAudioDeviceCaptureVolumeChanged (uint32_t volume

bool muted)

当前麦克风的系统采集音量发生变化

在 Mac 或 Windows 这样的桌面操作系统上，用户可以在设置中心找到声音相关的设置面板，并设置麦克风的采集音量大小。
用户将麦克风的采集音量设置得越大，麦克风采集到的声音的原始音量也就会越大，反之就会越小。
在有些型号的键盘以及笔记本电脑上，用户还可以通过按下 禁用麦克风 按钮（图标是一个 话筒 叠加了一道代表禁用的斜线）来将麦克风静音。

当用户通过系统设置界面或者通过键盘上的快捷键设定操作系统的麦克风采集音量时，SDK 便会抛出此事件。

参数	描述
muted	麦克风是否被用户禁用了：true 被禁用，false 被启用。
volu	系统采集音量，取值范围 0 - 100，用户可以在系统的声音设置面板上进行拖拽调整。

me	

注意
您需要调用 enableAudioVolumeEvaluation 接口并设定 interval 大于 0 开启次事件回调，设定 interval 等于 0 关闭此事件回调。

onAudioDevicePlayoutVolumeChanged

onAudioDevicePlayoutVolumeChanged

```
void onAudioDevicePlayoutVolumeChanged(uint32_t volume
                                       bool muted)
```

当前系统的播放音量发生变化

在 Mac 或 Windows 这样的桌面操作系统上，用户可以在设置中心找到声音相关的设置面板，并设置系统的播放音量大小。
在有些型号的键盘以及笔记本电脑上，用户还可以通过按下“静音”按钮（图标是一个喇叭上叠加了一道代表禁用的斜线）来将系统静音。

当用户通过系统设置界面或者通过键盘上的快捷键设定操作系统的播放音量时，SDK 便会抛出此事件。

参数	描述
muted	系统是否被用户静音了：true 被静音，false 已恢复。
volume	系统播放音量，取值范围 0 – 100，用户可以在系统的声音设置面板上进行拖拽调整。

注意
您需要调用 enableAudioVolumeEvaluation 接口并设定 interval 大于 0 开启次事件回调，设定 interval 等于 0 关闭此事件回调。

onSystemAudioLoopbackError

onSystemAudioLoopbackError

void onSystemAudioLoopbackError (TXLiteAVError errCode)

系统声音采集是否被成功开启的事件回调（仅适用于 Mac 系统）

在 Mac 系统上，您可以通过调用 startSystemAudioLoopback 为当前系统安装一个音频驱动，并让 SDK 通过该音频驱动捕获当前 Mac 系统播放出的声音。
当用于播片教学或音乐直播中，比如老师端可以使用此功能，让 SDK 能够采集老师所播放的电影中的声音，使同房间的学生端也能听到电影中的声音。
SDK 会将系统声音采集是否被成功开启的结果，通过本事件回调抛出，需要您关注参数中的错误码。

参数	描述
err	ERR_NULL 表示成功，其余值表示失败。

onTestMicVolume

onTestMicVolume

void onTestMicVolume (uint32_t volume)

测试麦克风时的音量回调

当您调用 startMicDeviceTest 测试麦克风是否正常工作时，SDK 会不断地抛出此回调，参数中的 volume 代表当前麦克风采集到的音量大小。
如果在测试期间 volume 出现了大小波动的情况，说明麦克风状态健康；如果 volume 的数值始终是 0，说明麦克风的状态异常，需要提示用户进行更换。

参数	描述
volume	麦克风采集到的音量值，取值范围0 – 100。

onTestSpeakerVolume

onTestSpeakerVolume

```
void
onTestSpeakerVolume
(uint32_t volume)
```

测试扬声器时的音量回调

当您调用 startSpeakerDeviceTest 测试扬声器是否正常工作时，SDK 会不断地抛出此回调。
参数中的 volume 代表的是 SDK 提交给系统扬声器去播放的声音的音量值大小，如果该数值持续变化，但用户反馈听不到声音，则说明扬声器状态异常。

参数	描述
volume	SDK 提交给扬声器去播放的声音的音量，取值范围0 – 100。

onRecvCustomCmdMsg

onRecvCustomCmdMsg

```
void
onRecvCustomCmdMsg
(const char* userId
int32_t cmdID
uint32_t seq
const uint8_t* message
uint32_t messageSize)
```

收到自定义消息的事件回调

当房间中的某个用户使用 sendCustomCmdMsg 发送自定义 UDP 消息时，房间中的其他用户可以通过 onRecvCustomCmdMsg 事件回调接收到该条消息。

参数	描述
cmdID	命令 ID。
messa	消息数据。

ge	
seq	消息序号。
userId	用户标识。

onMissCustomCmdMsg

onMissCustomCmdMsg

```
void
onMissCustomCmdMsg (const char* userId
                    int32_t cmdID
                    int32_t errCode
                    int32_t missed)
```

自定义消息丢失的事件回调

当您使用 sendCustomCmdMsg 发送自定义 UDP 消息时，即使设置了可靠传输（reliable），也无法确 100% 不丢失，只是丢消息概率极低，能满足常规可靠性要求。
在发送端设置了可靠运输（reliable）后，SDK 都会通过此回调通知过去时间段内（通常为5s）传输途中丢失的自定义消息数量统计信息。

参数	描述
cmdID	命令 ID。
errCode	错误码。
missed	丢失的消息数量。
userId	用户标识。

注意
只有在发送端设置了可靠传输（reliable），接收方才能收到消息的丢失回调。

onRecvSEIMsg

onRecvSEIMsg

```
void onRecvSEIMsg (const char* userId
                   const uint8_t* message
                   uint32_t messageSize)
```

收到 SEI 消息的回调

当房间中的某个用户使用 sendSEIMsg 借助视频数据帧发送 SEI 消息时，房间中的其他用户可以通过 onRecvSEIMsg 事件回调接收到该条消息。

参数	描述
message	数据。
userId	用户标识。

onStartPublishing

onStartPublishing

```
void onStartPublishing (int err
                        const char *errMsg)
```

开始向腾讯云直播 CDN 上发布音视频流的事件回调

当您调用 startPublishing 开始向腾讯云直播 CDN 上发布音视频流时，SDK 会立刻将这一指令同步给云端服务器。
随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	0表示成功，其余值表示失败。

errMsg	具体错误原因。

onStopPublishing

onStopPublishing

```
void onStopPublishing (int err
                       const char *errMsg)
```

停止向腾讯云直播 CDN 上发布音视频流的事件回调

当您调用 stopPublishing 停止向腾讯云直播 CDN 上发布音视频流时，SDK 会立刻将这一指令同步给云端服务器。
随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	0表示成功，其余值表示失败。
errMsg	具体错误原因。

onStartPublishCDNStream

onStartPublishCDNStream

```
void onStartPublishCDNStream (int errCode
                              const char* errMsg)
```

开始向非腾讯云 CDN 上发布音视频流的事件回调

当您调用 startPublishCDNStream 开始向非腾讯云直播 CDN 上发布音视频流时，SDK 会立刻将这一指令同步给云端服务器。随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	0表示成功，其余值表示失败。
errMsg	具体错误原因。

注意
当您收到成功的事件回调时，只是说明您的发布指令已经同步到腾讯云后台服务器，但如果目标 CDN 厂商的服务器不接收该条视频流，依然可能导致发布失败。

onStopPublishCDNStream

onStopPublishCDNStream

```
void onStopPublishCDNStream(int errCode
                            const char* errMsg)
```

停止向非腾讯云 CDN 上发布音视频流的事件回调

当您调用 stopPublishCDNStream 开始向非腾讯云直播 CDN 上发布音视频流时，SDK 会立刻将这一指令同步给云端服务器。随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	0表示成功，其余值表示失败。
errMsg	具体错误原因。

onSetMixTranscodingConfig

onSetMixTranscodingConfig

```
void onSetMixTranscodingConfig(int err
```

```
const char* errMsg)
```

设置云端混流的排版布局和转码参数的事件回调

当您调用 setMixTranscodingConfig 调整云端混流的排版布局和转码参数时，SDK 会立刻将这一调整指令同步给云端服务器。随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
err	错误码：0表示成功，其余值表示失败。
errMsg	具体的错误原因。

onStartPublishMediaStream

onStartPublishMediaStream

```
void onStartPublishMediaStream (const char* taskId
                                int code
                                const char* message
                                void* extraInfo)
```

开始发布媒体流的事件回调

当您调用 startPublishMediaStream 开始向 TRTC 后台服务发布媒体流时，SDK 会立刻将这一指令同步给云端服务器，随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
code	回调结果，0 表示成功，其余值表示失败。
extraInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。
message	具体回调信息。
taskId	当请求成功时，TRTC 后台会在回调中提供给您这项任务的 taskId，后续您可以通过

	该 taskId 结合 updatePublishMediaStream 和 stopPublishMediaStream 进行更新和停止。

onUpdatePublishMediaStream

onUpdatePublishMediaStream

```
void onUpdatePublishMediaStream(const char* taskId,
                                int code,
                                const char* message,
                                void* extraInfo)
```

更新媒体流的事件回调

当您调用媒体流发布接口（updatePublishMediaStream）开始向 TRTC 后台服务更新媒体流时，SDK 会立刻将这一指令同步给云端服务器，随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
code	回调结果，0 表示成功，其余值表示失败。
extraInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。
message	具体回调信息。
taskId	您调用媒体流发布接口（updatePublishMediaStream）时传入的 taskId，会通过此回调再带回给您，用于标识该回调属于哪一次更新请求。

onStopPublishMediaStream

onStopPublishMediaStream

```
void onStopPublishMediaStream(const char* taskId)
```

```
int code
const char* message
void* extraInfo)
```

停止媒体流的事件回调

当您调用停止发布媒体流（stopPublishMediaStream）开始向 TRTC 后台服务停止媒体流时，SDK 会立刻将这一指令同步给云端服务器，随后 SDK 会收到来自云端服务器的处理结果，并将指令的执行结果通过本事件回调通知给您。

参数	描述
code	回调结果，0 表示成功，其余值表示失败。
extraInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。
message	具体回调信息。
taskId	您调用停止发布媒体流（stopPublishMediaStream）时传入的 taskId，会通过此回调再带回给您，用于标识该回调属于哪一次停止请求。

onCdnStreamStateChanged

onCdnStreamStateChanged

```
void onCdnStreamStateChanged
(const char* cdnUrl
int status
int code
const char* msg
void* extraInfo)
```

RTMP/RTMPS 推流状态发生改变回调

当您调用 startPublishMediaStream 开始向 TRTC 后台服务发布媒体流时，SDK 会立刻将这一指令同步给云端服务。
若您在目标推流配置（TRTCPublishTarget）设置了向腾讯或者第三方 CDN 上发布音视频流的 URL 配置，则具体 RTMP 或者 RTMPS 推流状态将通过此回调同步给您。

参数	描述
cdnUrl	您调用 startPublishMediaStream 时通过目标推流配置（TRTCPublishTarget）传入的 url，在推流状态变更时，会通过此回调同步给您。
code	推流结果，0 表示成功，其余值表示出错。
extraInfo	扩展信息字段，个别错误码可能会带额外的信息帮助定位问题。
message	具体推流信息。
status	推流状态。 • 0：推流未开始或者已结束。在您调用 stopPublishMediaStream 时会返回该状态。 • 1：正在连接 TRTC 服务器和 CDN 服务器。若无法立刻成功，TRTC 后台服务会多次重试尝试推流，并返回该状态（5s回调一次）。如成功进行推流，则进入状态 2；如服务器出错或 60 秒内未成功推流，则进入状态 4。 • 2：CDN 推流正在进行。在成功推流后，会返回该状态。 • 3：TRTC 服务器和 CDN 服务器推流中断，正在恢复。当 CDN 出现异常，或推流短暂中断时，TRTC 后台服务会自动尝试恢复推流，并返回该状态（5s回调一次）。如成功恢复推流，则进入状态 2；如服务器出错或 60 秒内未成功恢复，则进入状态 4。 • 4：TRTC 服务器和 CDN 服务器推流中断，且恢复或连接超时。即此时推流失败，你可以再次调用 updatePublishMediaStream 尝试推流。 • 5：正在断开 TRTC 服务器和 CDN 服务器。在您调用 stopPublishMediaStream 时，TRTC 后台服务会依次同步状态 5 和状态 0。

onScreenCaptureStarted

onScreenCaptureStarted

屏幕分享开启的事件回调

当您通过 startScreenCapture 等相关接口启动屏幕分享时，SDK 便会抛出此事件回调。

onScreenCapturePaused

onScreenCapturePaused

```
void onScreenCapturePaused (int reason)
```

屏幕分享暂停的事件回调

当您通过 pauseScreenCapture 暂停屏幕分享时，SDK 便会抛出此事件回调。

参数	描述
reason	原因。 • 0：用户主动暂停。 • 1：注意此字段的含义在 MAC 和 Windows 平台有稍微差异。屏幕窗口不可见暂停（Mac）。表示设置屏幕分享参数导致的暂停（Windows）。 • 2：表示屏幕分享窗口被最小化导致的暂停（仅 Windows）。 • 3：表示屏幕分享窗口被隐藏导致的暂停（仅 Windows）。

onScreenCaptureResumed

onScreenCaptureResumed

```
void onScreenCaptureResumed (int reason)
```

屏幕分享恢复的事件回调

当您通过 resumeScreenCapture 恢复屏幕分享时，SDK 便会抛出此事件回调。

参数	描述
reason	恢复原因。 • 0：用户主动恢复。 • 1：注意此字段的含义在 MAC 和 Windows 平台有稍微差异。屏幕窗口恢复可见从而恢复分享（Mac）。屏幕分享参数设置完毕后自动恢复（Windows）。 • 2：表示屏幕分享窗口从最小化被恢复（仅 Windows）。 • 3：表示屏幕分享窗口从隐藏被恢复（仅 Windows）。

版权所有：腾讯云计算（北京）有限责任公司

onScreenCaptureStoped

onScreenCaptureStoped

void onScreenCaptureStoped	(int reason)

屏幕分享停止的事件回调

当您通过 stopScreenCapture 停止屏幕分享时，SDK 便会抛出此事件回调。

参数	描述
reason	停止原因，0：用户主动停止；1：屏幕窗口关闭导致停止；2：表示屏幕分享的显示屏状态变更（如接口被拔出、投影模式变更等）。

onScreenCaptureCovered

onScreenCaptureCovered

屏幕分享的目标窗口被遮挡的事件回调（仅适用于 Windows 操作系统）

当屏幕分享的目标窗口被遮挡无法正常捕获时，SDK 会抛出此事件回调，你可以在捕获到该事件回调后，通过 UI 上的一些变化来提示用户移开遮盖窗口。

onLocalRecordBegin

onLocalRecordBegin

void onLocalRecordBegin	(int errCode const char* storagePath)

本地录制任务已经开始的事件回调

当您调用 startLocalRecording 启动本地媒体录制任务时，SDK 会抛出该事件回调，用于通知您录制任务是否已经顺利启动。

参数	描述
errCode	状态码。 • 0：录制任务启动成功。 • -1：内部错误导致录制任务启动失败。 • -2：文件后缀名有误（比如不支持的录制格式）。 • -6：录制已经启动，需要先停止录制。 • -7：录制文件已存在，需要先删除文件。 • -8：录制目录无写入权限，请检查目录权限问题。
storagePath	录制文件存储路径。

onLocalRecording

onLocalRecording

```
void onLocalRecording (long duration
                       const char* storagePath)
```

本地录制任务正在进行中的进展事件回调

当您调用 startLocalRecording 成功启动本地媒体录制任务后，SDK 会定时地抛出本事件回调。
您可通过捕获该事件回调来获知录制任务的健康状况。
您可以在 startLocalRecording 时设定本事件回调的抛出间隔。

参数	描述
duration	已经录制的累计时长，单位毫秒。
storagePath	录制文件存储路径。

onLocalRecordFragment

onLocalRecordFragment

void onLocalRecordFragment	(const char* storagePath)

本地录制分片的事件回调

当您开启分片录制时，每完成一个分片，都会通过此接口回调。

参数	描述
storagePath	分片文件存储路径。

onLocalRecordComplete

onLocalRecordComplete

void onLocalRecordComplete	(int errCode const char* storagePath)

本地录制任务已经结束的事件回调

当您调用 stopLocalRecording 停止本地媒体录制任务时，SDK 会抛出该事件回调，用于通知您录制任务的最终结果。

参数	描述
errCode	状态码。 • 0：结束录制任务成功。 • -1：录制失败。 • -2：切换分辨率或横竖屏导致录制结束。 • -3：录制时间太短，或未采集到任何视频或音频数据，请检查录制时长，或是否已开启音、视频采集。
storagePath	录制文件存储路径。

onSnapshotComplete

onSnapshotComplete

```
void
onSnapshotComplete
(const char* userId
TRTCVideoStreamType type
char* data
uint32_t length
uint32_t width
uint32_t height
TRTCVideoPixelFormat format)
```

本地截图完成的事件回调

参数	描述
bmp	截图结果，如果 bmp 为 null 代表本次截图操作失败。
data	截图数据，为 nullptr 表示截图失败。
format	截图数据格式，目前只支持 TRTCVideoPixelFormat_BGRA32。
height	截图画面的高度。
length	截图数据长度，对于BGRA32而言，length = width * height * 4。
type	视频流类型。
userId	用户标识，如果 userId 为空字符串，则代表截取的是本地画面。
width	截图画面的宽度。

注意

> 全平台 C++ 接口和 Java 接口在参数上是不一样的，C++ 接口用 7 个参数描述一个截图画面，Java 接口只用一个 Bitmap 描述一个截图画面。

onUserEnter

onUserEnter

```
void
onUserEnter        (const char* userId)
```

有主播加入当前房间（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 onRemoteUserEnterRoom 替代之。

onUserExit

onUserExit

```
void onUserExit    (const char* userId
                    int reason)
```

有主播离开当前房间（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 onRemoteUserLeaveRoom 替代之。

onAudioEffectFinished

onAudioEffectFinished

```
void onAudioEffectFinished    (int effectId
                               int code)
```

音效播放已结束（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 ITXAudioEffectManager 接口替代之。
新的接口中不再区分背景音乐和音效，而是统一用 startPlayMusic 取代之。

onPlayBGMBegin

onPlayBGMBegin

```
void
onPlayBGMBegin
```
(TXLiteAVError errCode)

开始播放背景音乐（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 ITXMusicPlayObserver 接口替代之。
新的接口中不再区分背景音乐和音效，而是统一用 startPlayMusic 取代之。

onPlayBGMProgress

onPlayBGMProgress

```
void
onPlayBGMProgress
```
(uint32_t progressMS

uint32_t durationMS)

背景音乐的播放进度回调（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 ITXMusicPlayObserver 接口替代之。
新的接口中不再区分背景音乐和音效，而是统一用 startPlayMusic 取代之。

onPlayBGMComplete

onPlayBGMComplete

```
void
onPlayBGMComplete
```
(TXLiteAVError errCode)

背景音乐播放已经结束（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 ITXMusicPlayObserver 接口替代之。
新的接口中不再区分背景音乐和音效，而是统一用 startPlayMusic 取代之。

版权所有：腾讯云计算（北京）有限责任公司

onSpeedTest

onSpeedTest

```
void
onSpeedTest
```
(const TRTCSpeedTestResult& currentResult
uint32_t finishedCount
uint32_t totalCount)

服务器测速的结果回调（已废弃）

@deprecated 新版本开始不推荐使用，建议使用 onSpeedTestResult 接口替代之。

onRenderVideoFrame

onRenderVideoFrame

```
void
onRenderVideoFrame
```
(const char* userId
TRTCVideoStreamType streamType
TRTCVideoFrame* frame)

自定义视频渲染回调

当您设置了本地或者远端的视频自定义渲染回调之后，SDK 就会将原本要交给渲染控件进行渲染的视频帧通过此回调接口抛送给您，以便于您进行自定义渲染。

参数	描述
frame	待渲染的视频帧信息。
streamType	频流类型：主路（Main）一般用于承载摄像头画面，辅路（Sub）一般用于承载屏幕分享画面。
userId	视频源的 userId，如果是本地视频回调（setLocalVideoRenderDelegate），该参数可以忽略。

onGLContextCreated

onGLContextCreated

SDK 内部 OpenGL 环境已经创建的通知

onProcessVideoFrame

onProcessVideoFrame

int onProcessVideoFrame (TRTCVideoFrame *srcFrame, TRTCVideoFrame *dstFrame)

用于对接第三方美颜组件的视频处理回调

如果您选购了第三方美颜组件，就需要在 TRTCCloud 中设置第三方美颜回调，之后 TRTC 就会将原本要进行预处理的视频帧通过此回调接口抛送给您。
之后您就可以将 TRTC 抛出的视频帧交给第三方美颜组件进行图像处理，由于抛出的数据是可读且可写的，因此第三方美颜的处理结果也可以同步给 TRTC 进行后续的编码和发送。

情况一：美颜组件自身会产生新的纹理。
如果您使用的美颜组件会在处理图像的过程中产生一帧全新的纹理（用于承载处理后的图像），那请您在回调函数中将 dstFrame.textureId 设置为新纹理的 ID：

```
int onProcessVideoFrame(TRTCVideoFrame * srcFrame, TRTCVideoFrame
*dstFrame) {
    dstFrame->textureId = mFURenderer.onDrawFrameSingleInput(srcFrame-
>textureId);
    return 0;
}
```

情况二：美颜组件需要您提供目标纹理。
如果您使用的第三方美颜模块并不生成新的纹理，而是需要您设置给该模块一个输入纹理和一个输出纹理，则可以考虑如下方案：

```
int onProcessVideoFrame(TRTCVideoFrame *srcFrame, TRTCVideoFrame
*dstFrame) {
```

```
    thirdparty_process(srcFrame->textureId, srcFrame->width, srcFrame->height,
dstFrame->textureId);
    return 0;
}
```

参数	描述
dstFrame	用于接收第三方美颜处理过的视频画面。
srcFrame	用于承载 TRTC 采集到的摄像头画面。

注意
目前仅支持 OpenGL 纹理方案（ PC 仅支持 TRTCVideoBufferType_Buffer 格式）。

onGLContextDestroy

onGLContextDestroy

SDK 内部 OpenGL 环境被销毁的通知

onCapturedAudioFrame

onCapturedAudioFrame

void onCapturedAudioFrame (TRTCAudioFrame *frame)

本地采集并经过音频模块前处理后的音频数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把刚采集到并经过前处理(ANS、AEC、AGC）之后的数据，以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。
3. 此接口回调出的音频数据已经经过了前处理(ANS、AEC、AGC），但**不包含**背景音、音效、混响等前处理效果，延迟较低。

onLocalProcessedAudioFrame

onLocalProcessedAudioFrame

```
void onLocalProcessedAudioFrame (TRTCAudioFrame *frame)
```

本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把刚采集到并经过前处理、音效处理和混 BGM 之后的数据，在最终进行网络编码之前，以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

特殊说明：
您可以通过设置接口中的 `TRTCAudioFrame.extraData` 字段，达到传输信令的目的。由于音频帧头部的数据块不能太大，建议您写入 `extraData` 时，尽量将信令控制在几个字节的大小，如果超过 100 个字节，写入的数据不会被发送。

房间内其他用户可以通过 ITRTCAudioFrameCallback 中的 `onRemoteUserAudioFrame` 中的 `TRTCAudioFrame.extraData` 字段回调接收数据。

参数	描述

frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。
3. 此接口回调出的数据已经经过了前处理(ANS、AEC、AGC）、音效和混 BGM 处理，声音的延迟相比于 onCapturedAudioFrame 要高一些。

onPlayAudioFrame

onPlayAudioFrame

```
void onPlayAudioFrame(TRTCAudioFrame *frame
                      const char* userId)
```

混音前的每一路远程用户的音频数据

当您设置完音频数据自定义回调之后，SDK 内部会把远端的每一路原始数据，在最终混音之前，以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。
userId	用户标识。

注意

此接口回调出的音频数据是只读的，不支持修改。

onMixedPlayAudioFrame

onMixedPlayAudioFrame

```
void onMixedPlayAudioFrame (TRTCAudioFrame *frame)
```

将各路待播放音频混合之后并在最终提交系统播放之前的数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把各路待播放的音频混合之后的音频数据，在提交系统播放之前，以 PCM 格式的形式通过本接口回调给您。

- 此接口回调出的音频时间帧长固定为0.02s，格式为 PCM 格式。
- 由时间帧长转化为字节帧长的公式为 `采样率 × 时间帧长 × 声道数 × 采样点位宽` 。
- 以 TRTC 默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为 `48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节` 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。
3. 此接口回调出的是对各路待播放音频数据的混合，但其中并不包含耳返的音频数据。

onLog

onLog

```
void onLog (const char* log
```

```
TRTCLogLevel level
const char* module)
```

本地 LOG 的打印回调

如果您希望捕获 SDK 的本地日志打印行为，可以通过设置日志回调，让 SDK 将要打印的日志都通过本回调接口抛送给您。

参数	描述
level	日志等级请参见 TRTC_LOG_LEVEL。
log	日志内容。
module	保留字段，暂无具体意义，目前为固定值 TXLiteAVSDK。

ITRTCStatistics

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTC 音视频统计指标（只读）

Function: TRTC SDK 会以两秒钟一次的频率向您汇报当前实时的音视频指标（帧率、码率、卡顿情况等）

ITRTCStatistics

结构体类型

函数列表	描述
TRTCLocalStatistics	本地的音视频统计指标
TRTCRemoteStatistics	远端的音视频统计指标
TRTCStatistics	网络和性能的汇总统计指标

TRTCLocalStatistics

TRTCLocalStatistics

本地的音视频统计指标

枚举类型	描述
audioBitrate	【字段含义】本地音频的码率，即每秒钟新产生音频数据的多少，单位 Kbps。
audioCaptureState	【字段含义】音频设备采集状态（用于检测音频外设的健康度）。 0：采集设备状态正常；1：检测到长时间静音；2：检测到破音；3：检测到声音异常间断。

audioSampleRate	【字段含义】本地音频的采样率，单位 Hz。
frameRate	【字段含义】本地视频的帧率，即每秒钟会有多少视频帧，单位：FPS。
height	【字段含义】本地视频的高度，单位 px。
streamType	【字段含义】视频流类型（高清大画面|低清小画面|辅流画面）。
videoBitrate	【字段含义】本地视频的码率，即每秒钟新产生视频数据的多少，单位 Kbps。
width	【字段含义】本地视频的宽度，单位 px。

TRTCRemoteStatistics

TRTCRemoteStatistics

远端的音视频统计指标

枚举类型	描述
audioBitrate	【字段含义】本地音频的码率，单位 Kbps
audioBlockRate	【字段含义】音频播放卡顿率，单位 (%) 音频播放卡顿率（audioBlockRate）= 音频播放的累计卡顿时长（audioTotalBlockTime）/ 音频播放的总时长
audioPacketLoss	【字段含义】音频流的总丢包率（%）。 audioPacketLoss 代表音频流历经 `主播>云端>观众` 这样一条完整的传输链路后，最终在观众端统计到的丢包率。 audioPacketLoss 越小越好，丢包率为0即表示该路音频流的所有数据均已经完整地到达了观众端。 如果出现了 downLoss == 0 但 audioPacketLoss != 0 的情况，说明该路音频流在“云端=>观众”这一段链路上没有出现丢包，但是在 `主播>云端` 这一段链路上出现了不可恢复的丢包。
audioSampleRate	【字段含义】本地音频的采样率，单位 Hz。
audioTotalBlockTime	【字段含义】音频播放的累计卡顿时长，单位 ms
finalLoss	【字段含义】该路音视频流的总丢包率（%）。已废弃，不推荐使用；建议使用 audioPacketLoss、videoPacketLoss 替代。

frameRate	【字段含义】远端视频的帧率，单位：FPS。
height	【字段含义】远端视频的高度，单位 px。
jitterBufferDelay	【字段含义】播放延迟，单位 ms 为了避免网络抖动和网络包乱序导致的声音和画面卡顿，TRTC 会在播放端管理一个播放缓冲区，用于对接收到的网络数据包进行整理，该缓冲区的大小会根据当前的网络质量进行自适应调整，该缓冲区的大小折算成以毫秒为单位的时间长度，也就是 jitterBufferDelay。
point2PointDelay	【字段含义】端到端延迟，单位 ms point2PointDelay 代表 “主播=>云端=>观众” 的延迟，更准确地说，它代表了“采集=>编码=>网络传输=>接收=>缓冲=>解码=>播放” 全链路的延迟。 point2PointDelay 需要本地和远端的 SDK 均为 8.5 及以上的版本才生效，若远端用户为 8.5 以前的版本，此数值会一直为0，代表无意义。
remoteNetworkRTT	【字段含义】该用户从 SDK 到云端的往返延时，单位 ms 该数值代表从 SDK 发送一个网络包到云端，再从云端回送一个网络包到 SDK 的总计耗时，也就是一个网络包经历 “SDK=>云端=>SDK” 的总耗时。 该数值越小越好：如果 remoteNetworkRTT < 50ms，意味着较低的音视频通话延迟；如果 remoteNetworkRTT > 200ms，则意味着较高的音视频通话延迟。 需要特别解释的是，remoteNetworkRTT 代表 “SDK=>云端=>SDK” 的总耗时，所以不需要区分 remoteNetworkUpRTT 和 remoteNetworkDownRTT
remoteNetworkUplinkLoss	【字段含义】该用户从 SDK 到云端的上行丢包率，单位 (%) 该数值越小越好，如果 remoteNetworkUplinkLoss 为 0%，则意味着上行链路的网络质量很好，上传到云端的数据包基本不发生丢失。 如果 remoteNetworkUplinkLoss 为 30%，则意味着 SDK 向云端发送的音视频数据包中，会有 30% 丢失在传输链路中。
streamType	【字段含义】视频流类型（高清大画面|低清小画面|辅流画面）。
userId	【字段含义】用户 ID
videoBitrate	【字段含义】远端视频的码率，单位 Kbps。
videoBlockRate	【字段含义】视频播放卡顿率，单位 (%) 视频播放卡顿率（videoBlockRate）= 视频播放的累计卡顿时长（videoTotalBlockTime）/ 视频播放的总时长。
videoPacketLoss	【字段含义】该路视频流的总丢包率（%）。 videoPacketLoss 代表该路视频流历经 `主播>云端>观众` 这样一条完整的传输链路后，最终在观众端统计到的丢包率。

	videoPacketLoss 越小越好，丢包率为0即表示该路视频流的所有数据均已经完整地到达了观众端。 如果出现了 downLoss == 0 但 videoPacketLoss != 0 的情况，说明该路视频流在 `云端>观众` 这一段链路上没有出现丢包，但是在 `主播>云端` 这一段链路上出现了不可恢复的丢包。
videoTotalBlockTime	【字段含义】视频播放的累计卡顿时长，单位 ms
width	【字段含义】远端视频的宽度，单位 px。

TRTCStatistics

TRTCStatistics

网络和性能的汇总统计指标

枚举类型	描述
appCpu	【字段含义】当前应用的 CPU 使用率，单位 (%)，Android 8.0 以上不支持。
appMemoryUsageInMB	【字段含义】当前应用的内存占用，单位 (MB)。
downLoss	【字段含义】从云端到 SDK 的下行丢包率，单位 (%) 该数值越小越好，如果 downLoss 为 0%，则意味着下行链路的网络质量很好，从云端接收的数据包基本不发生丢失。 如果 downLoss 为 30%，则意味着云端向 SDK 传输的音视频数据包中，会有 30% 丢失在传输链路中。
gatewayRtt	【字段含义】从 SDK 到本地路由器的往返时延，单位 ms 该数值代表从 SDK 发送一个网络包到本地路由器网关，再从网关回送一个网络包到 SDK 的总计耗时，也就是一个网络包经历【SDK>网关>SDK【 的总耗时。 该数值越小越好：如果 gatewayRtt < 50ms，意味着较低的音视频通话延迟；如果 gatewayRtt > 200ms，则意味着较高的音视频通话延迟。 当网络类型为蜂窝网时，该值无效。
localStatisticsArray	【字段含义】本地的音视频统计信息 由于本地可能有三路音视频流（即高清大画面，低清小画面，以及辅流画面），因此本地的音视频统计信息是一个数组。

localStatisticsArraySize	【字段含义】数组 localStatisticsArray 的大小
receivedBytes	【字段含义】总接收字节数（包含信令数据和音视频数据），单位：字节数（Bytes）。
remoteStatisticsArray	【字段含义】远端的音视频统计信息 因为同时可能有多个远端用户，而且每个远端用户同时可能有多路音视频流（即高清大画面，低清小画面，以及辅流画面），因此远端的音视频统计信息是一个数组。
remoteStatisticsArraySize	【字段含义】数组 remoteStatisticsArray 的大小。
rtt	【字段含义】从 SDK 到云端的往返延时，单位 ms 该数值代表从 SDK 发送一个网络包到云端，再从云端回送一个网络包到 SDK 的总计耗时，也就是一个网络包经历 “SDK=>云端=>SDK” 的总耗时。 该数值越小越好：如果 rtt < 50ms，意味着较低的音视频通话延迟；如果 rtt > 200ms，则意味着较高的音视频通话延迟。 需要特别解释的是，rtt 代表 “SDK=>云端=>SDK” 的总耗时，所以不需要区分 upRtt 和 downRtt。
sentBytes	【字段含义】总发送字节数（包含信令数据和音视频数据），单位：字节数（Bytes）。
systemCpu	【字段含义】当前系统的 CPU 使用率，单位 (%)，Android 8.0 以上不支持。
systemMemoryInMB	【字段含义】当前系统的内存，单位 (MB)。
systemMemoryUsageInMB	【字段含义】当前系统内存占用，单位 (MB)，iOS/MAC 不支持 (MB)。
upLoss	【字段含义】从 SDK 到云端的上行丢包率，单位 (%) 该数值越小越好，如果 upLoss 为 0%，则意味着上行链路的网络质量很好，上传到云端的数据包基本不发生丢失。 如果 upLoss 为 30%，则意味着 SDK 向云端发送的音视频数据包中，会有 30% 丢失在传输链路中。

ITXAudioEffectManager

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: 背景音乐、短音效和人声特效的管理类

Function: 用于对背景音乐、短音效和人声特效进行设置的管理类

ITXAudioEffectManager

ITXMusicPreloadObserver

函数列表	描述
onLoadProgress	背景音乐预加载进度
onLoadError	背景音乐预加载出错

ITXMusicPlayObserver

函数列表	描述
onStart	背景音乐开始播放
onPlayProgress	背景音乐的播放进度
onComplete	背景音乐已经播放完毕

ITXAudioEffectManager

函数列表	描述
enableVoiceEarMonitor	开启耳返
setVoiceEarMonitor	设置耳返音量

Volume	
setVoiceReverbType	设置人声的混响效果
setVoiceChangerType	设置人声的变声特效
setVoiceCaptureVolume	设置语音音量
setVoicePitch	设置语音音调
setMusicObserver	设置背景音乐的事件回调接口
startPlayMusic	开始播放背景音乐
stopPlayMusic	停止播放背景音乐
pausePlayMusic	暂停播放背景音乐
resumePlayMusic	恢复播放背景音乐
setAllMusicVolume	设置所有背景音乐的本地音量和远端音量的大小
setMusicPublishVolume	设置某一首背景音乐的远端音量的大小
setMusicPlayoutVolume	设置某一首背景音乐的本地音量的大小
setMusicPitch	调整背景音乐的音调高低
setMusicSpeedRate	调整背景音乐的变速效果
getMusicCurrentPosInMS	获取背景音乐的播放进度（单位：毫秒）
getMusicDurationInMS	获取背景音乐的总时长（单位：毫秒）
seekMusicToPosInTime	设置背景音乐的播放进度（单位：毫秒）
setMusicScratchSpeedRate	调整搓碟的变速效果
setPreloadObserver	设置预加载事件回调

版权所有：腾讯云计算（北京）有限责任公司

preloadMusic	预加载背景音乐
getMusicTrackCount	获取背景音乐的音轨数量
setMusicTrack	指定背景音乐的播放音轨

结构体类型

函数列表	描述
AudioMusicParam	背景音乐的播放控制信息

枚举类型

枚举类型	描述
TXVoiceReverbType	混响特效
TXVoiceChangerType	变声特效

onLoadProgress

onLoadProgress

```
void onLoadProgress(int id, int progress)
```

背景音乐预加载进度

onLoadError

onLoadError

```
void (int id
```

```
onLoadError
                    int errorCode)
```

背景音乐预加载出错

参数	描述
errorCode	-4001:打开文件失败，如音频文件格式不支持，本地音频文件不存在，网络音频文件无法访问等；-4002:解码失败，如音频文件损坏，网络音频文件服务器无法访问等；-4003:预加载数量超上限，请先调用 stopPlayMusic 释放无用的预加载。

onStart

onStart

```
void onStart    (int id
                 int errCode)
```

背景音乐开始播放

在背景音乐开始播放成功或者失败时调用。

参数	描述
errCode	错误码。0: 开始播放成功；-4001: 打开文件失败，如音频文件格式不支持，本地音频文件不存在，网络音频文件无法访问等。
id	音乐 ID。

onPlayProgress

onPlayProgress

```
void onPlayProgress    (int id
                        long curPtsMS
                        long durationMS)
```

背景音乐的播放进度

onComplete

onComplete

```
void onComplete (int id
                 int errCode)
```

背景音乐已经播放完毕

在背景音乐播放完毕或播放错误时调用。

参数	描述
errCode	错误码。0: 播放结束；-4002: 解码失败，如音频文件损坏，网络音频文件服务器无法访问等。
id	音乐 ID。

enableVoiceEarMonitor

enableVoiceEarMonitor

```
void enableVoiceEarMonitor (bool enable)
```

开启耳返

主播开启耳返后，可以在耳机里听到麦克风采集到的自己发出的声音，该特效适用于主播唱歌的应用场景中。

需要您注意的是，由于蓝牙耳机的硬件延迟非常高，所以在主播佩戴蓝牙耳机时无法开启此特效，请尽量在用户界面上提示主播佩戴有线耳机。
同时也需要注意，并非所有的手机开启此特效后都能达到优秀的耳返效果，我们已经对部分耳返效果不佳的手机屏蔽了该特效。

参数	描述

enable	true：开启；false：关闭。

注意
仅在主播佩戴耳机时才能开启此特效，同时请您提示主播佩戴有线耳机。

setVoiceEarMonitorVolume

setVoiceEarMonitorVolume

```
void setVoiceEarMonitorVolume (int volume)
```

设置耳返音量

通过该接口您可以设置耳返特效中声音的音量大小。

参数	描述
volume	音量大小，取值范围为 0 – 100，默认值：100。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setVoiceReverbType

setVoiceReverbType

```
void setVoiceReverbType (TXVoiceReverbType type)
```

设置人声的混响效果

通过该接口您可以设置人声的混响效果，具体特效请参见枚举定义 TXVoiceReverbType 。

> **注意**
> 设置的效果在退出房间后会自动失效，如果下次进房还需要对应特效，需要调用此接口再次进行设置。

setVoiceChangerType

setVoiceChangerType

```
void setVoiceChangerType (TXVoiceChangerType type)
```

设置人声的变声特效

通过该接口您可以设置人声的变声特效，具体特效请参见枚举定义 TXVoiceChangeType 。

> **注意**
> 设置的效果在退出房间后会自动失效，如果下次进房还需要对应特效，需要调用此接口再次进行设置。

setVoiceCaptureVolume

setVoiceCaptureVolume

```
void setVoiceCaptureVolume (int volume)
```

设置语音音量

该接口可以设置语音音量的大小，一般配合音乐音量的设置接口 setAllMusicVolume 协同使用，用于调谐语音和音乐在混音前各自的音量占比。

参数	描述
volume	音量大小，取值范围为0 - 100，默认值：100。

> **注意**

> 如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setVoicePitch

setVoicePitch

```
void setVoicePitch (double pitch)
```

设置语音音调

该接口可以设置语音音调，用于实现变调不变速的目的。

参数	描述
pitch	音调，取值范围为-1.0f~1.0f，默认值：0.0f。

setMusicObserver

setMusicObserver

```
void setMusicObserver (int musicId
                       ITXMusicPlayObserver* observer)
```

设置背景音乐的事件回调接口

请在播放背景音乐之前使用该接口设置播放事件回调，以便感知背景音乐的播放进度。

参数	描述
musicId	音乐 ID。
observer	具体参考 ITXMusicPlayObserver 中定义接口。

> **注意**

1. 如果该 ID 不需要使用，observer 可设置为 NULL 彻底释放。

startPlayMusic

startPlayMusic

void startPlayMusic (AudioMusicParam musicParam)

开始播放背景音乐

每个音乐都需要您指定具体的 ID，您可以通过该 ID 对音乐的开始、停止、音量等进行设置。

参数	描述
musicParam	音乐参数。

注意

1. 如果要多次播放同一首背景音乐，请不要每次播放都分配一个新的 ID，我们推荐使用相同的 ID。
2. 若您希望同时播放多首不同的音乐，请为不同的音乐分配不同的 ID 进行播放。
3. 如果使用同一个 ID 播放不同音乐，SDK 会先停止播放旧的音乐，再播放新的音乐。

stopPlayMusic

stopPlayMusic

void stopPlayMusic (int id)

停止播放背景音乐

参数	描述
id	音乐 ID。

pausePlayMusic

pausePlayMusic

```
void pausePlayMusic (int id)
```

暂停播放背景音乐

参数	描述
id	音乐 ID。

resumePlayMusic

resumePlayMusic

```
void resumePlayMusic (int id)
```

恢复播放背景音乐

参数	描述
id	音乐 ID。

setAllMusicVolume

setAllMusicVolume

```
void setAllMusicVolume (int volume)
```

设置所有背景音乐的本地音量和远端音量的大小

该接口可以设置所有背景音乐的本地音量和远端音量。

- 本地音量：即主播本地可以听到的背景音乐的音量大小。
- 远端音量：即观众端可以听到的背景音乐的音量大小。

参数	描述

volume	音量大小，取值范围为0 – 100，默认值：60。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPublishVolume

setMusicPublishVolume

```
void setMusicPublishVolume(int id, int volume)
```

设置某一首背景音乐的远端音量的大小

该接口可以细粒度地控制每一首背景音乐的远端音量，也就是观众端可听到的背景音乐的音量大小。

参数	描述
id	音乐 ID。
volume	音量大小，取值范围为0 – 100；默认值：60。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPlayoutVolume

setMusicPlayoutVolume

```
void setMusicPlayoutVolume(int id
```

int volume)

设置某一首背景音乐的本地音量的大小

该接口可以细粒度地控制每一首背景音乐的本地音量，也就是主播本地可以听到的背景音乐的音量大小。

参数	描述
id	音乐 ID。
volume	音量大小，取值范围为0 - 100，默认值：60。

注意
如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPitch

setMusicPitch

void setMusicPitch (int id
float pitch)

调整背景音乐的音调高低

参数	描述
id	音乐 ID。
pitch	音调，默认值是0.0f，范围是：[-1 ~ 1] 之间的浮点数。

setMusicSpeedRate

setMusicSpeedRate

void (int id

setMusicSpeedRate

float speedRate)

调整背景音乐的变速效果

参数	描述
id	音乐 ID。
speedRate	速度，默认值是1.0f，范围是：[0.5 ~ 2] 之间的浮点数。

getMusicCurrentPosInMS

getMusicCurrentPosInMS

long getMusicCurrentPosInMS (int id)

获取背景音乐的播放进度（单位：毫秒）

参数	描述
id	音乐 ID。

返回值说明：

成功返回当前播放时间，单位：毫秒，失败返回 -1。

getMusicDurationInMS

getMusicDurationInMS

long getMusicDurationInMS (char* path)

获取背景音乐的总时长（单位：毫秒）

参数	描述
path	音乐文件路径。

返回值说明：

成功返回时长，失败返回 -1。

seekMusicToPosInTime

seekMusicToPosInTime

```
void seekMusicToPosInTime (int id
                           int pts)
```

设置背景音乐的播放进度（单位：毫秒）

参数	描述
id	音乐 ID。
pts	单位: 毫秒。

注意

请尽量避免过度频繁地调用该接口，因为该接口可能会再次读写音乐文件，耗时稍高。
因此，当用户拖拽音乐的播放进度条时，请在用户完成拖拽操作后再调用本接口。
因为 UI 上的进度条控件往往会以很高的频率反馈用户的拖拽进度，如不做频率限制，会导致较差的用户体验。

setMusicScratchSpeedRate

setMusicScratchSpeedRate

```
void setMusicScratchSpeedRate (int id
                               float scratchSpeedRate)
```

调整搓碟的变速效果

参数	描述
id	音乐 ID。
scratchSpeedRate	搓碟速度，默认值是1.0f，范围是：[-12.0 ~ 12.0] 之间的浮点数，速度值正/负表示方向正/反，绝对值大小表示速度快慢。

注意
前置条件 preloadMusic 成功。

setPreloadObserver

setPreloadObserver

void setPreloadObserver (ITXMusicPreloadObserver* observer)

设置预加载事件回调

请在预加载背景音乐之前使用该接口设置回调，以便感知背景音乐的预加载进度。

参数	描述
observer	具体参考 ITXMusicPreloadObserver 中定义接口。

preloadMusic

preloadMusic

void preloadMusic (AudioMusicParam preloadParam)

预加载背景音乐

每个音乐都需要您指定具体的 ID，您可以通过该 ID 对音乐的开始、停止、音量等进行设置。

参数	描述

preloadParam	预加载音乐参数。

注意

1. 预先加载最多同时支持2个不同 ID 的预加载，且预加载时长不超过10分钟，使用完需 stopPlayMusic，否则内存不释放。
2. 若该ID对应的音乐正在播放中，预加载会失败，需先调用 stopPlayMusic。
3. 当 musicParam 和传入 startPlayMusic 的 musicParam 完全相同时，预加载有效。

getMusicTrackCount

getMusicTrackCount

long getMusicTrackCount (int id)

获取背景音乐的音轨数量

参数	描述
id	音乐 ID。

setMusicTrack

setMusicTrack

void setMusicTrack (int id
int trackIndex)

指定背景音乐的播放音轨

参数	描述
id	音乐 ID。
inde	默认播放第一个音轨。取值范围[0, 音轨总数)。

x	

> **注意**
> 音轨总数量可通过 getMusicTrackCount 接口获取。

TXVoiceReverbType

TXVoiceReverbType

混响特效

混响特效可以作用于人声之上，通过声学算法对声音进行叠加处理，模拟出各种不同环境下的临场感受，目前支持如下几种混响效果：
0：关闭；1：KTV；2：小房间；3：大会堂；4：低沉；5：洪亮；6：金属声；7：磁性；8：空灵；9：录音棚；10：悠扬 11：录音棚2。

枚举	取值	描述
TXLiveVoiceReverbType_0	0	关闭特效
TXLiveVoiceReverbType_1	1	KTV
TXLiveVoiceReverbType_2	2	小房间
TXLiveVoiceReverbType_3	3	大会堂
TXLiveVoiceReverbType_4	4	低沉
TXLiveVoiceReverbType_5	5	洪亮
TXLiveVoiceReverbType_6	6	金属声
TXLiveVoiceReverbType_7	7	磁性
TXLiveVoiceReverbTy	8	空灵

pe_8		
TXLiveVoiceReverbType_9	9	录音棚
TXLiveVoiceReverbType_10	10	悠扬
TXLiveVoiceReverbType_11	11	录音棚2

TXVoiceChangeType

TXVoiceChangeType

变声特效

变声特效可以作用于人声之上，通过声学算法对人声进行二次处理，以获得与原始声音所不同的音色，目前支持如下几种变声特效：
0：关闭；1：熊孩子；2：萝莉；3：大叔；4：重金属；5：感冒；6：外语腔；7：困兽；8：肥宅；9：强电流；10：重机械；11：空灵。

枚举	取值	描述
TXVoiceChangerType_0	0	关闭
TXVoiceChangerType_1	1	熊孩子
TXVoiceChangerType_2	2	萝莉
TXVoiceChangerType_3	3	大叔
TXVoiceChangerType_4	4	重金属
TXVoiceChangerType_5	5	感冒
TXVoiceChangerType_6	6	外语腔

版权所有：腾讯云计算（北京）有限责任公司

TXVoiceChangerType_7	7	困兽
TXVoiceChangerType_8	8	肥宅
TXVoiceChangerType_9	9	强电流
TXVoiceChangerType_10	10	重机械
TXVoiceChangerType_11	11	空灵

TXAudioMusicParam

TXAudioMusicParam

背景音乐的播放控制信息

该信息用于在接口 startPlayMusic 中指定背景音乐的相关信息，包括播放 ID、文件路径和循环次数等：
1. 如果要多次播放同一首背景音乐，请不要每次播放都分配一个新的 ID，我们推荐使用相同的 ID。
2. 若您希望同时播放多首不同的音乐，请为不同的音乐分配不同的 ID 进行播放。
3. 如果使用同一个 ID 播放不同音乐，SDK 会先停止播放旧的音乐，再播放新的音乐。

枚举类型	描述
endTimeMS	【字段含义】音乐结束播放时间点，单位毫秒，0表示播放至文件结尾。
id	【字段含义】音乐 ID。 【特殊说明】SDK 允许播放多路音乐，因此需要使用 ID 进行标记，用于控制音乐的开始、停止、音量等。
isShortFile	【字段含义】播放的是否为短音乐文件。 【推荐取值】true：需要重复播放的短音乐文件；false：正常的音乐文件。默认值：false。
loopCount	【字段含义】音乐循环播放的次数。 【推荐取值】取值范围为0 - 任意正整数，默认值：0。0 表示播放音乐一次；1 表示播放音乐两次；以此类推。

path	【字段含义】音效文件的完整路径或 URL 地址。支持的音频格式包括 MP3、AAC、M4A、WAV。
publish	【字段含义】是否将音乐传到远端。 【推荐取值】true：音乐在本地播放的同时，远端用户也能听到该音乐；false：主播只能在本地听到该音乐，远端观众听不到。默认值：false。
startTimeMS	【字段含义】音乐开始播放时间点，单位：毫秒。

ITXDeviceManager

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: 音视频设备管理模块

Function: 用于管理摄像头、麦克风和扬声器等音视频相关的硬件设备

ITXDeviceManager

ITXDeviceManager

函数列表	描述
isFrontCamera	判断当前是否为前置摄像头（仅适用于移动端）
switchCamera	切换前置或后置摄像头（仅适用于移动端）
getCameraZoomMaxRatio	获取摄像头的最大缩放倍数（仅适用于移动端）
setCameraZoomRatio	设置摄像头的缩放倍数（仅适用于移动端）
isAutoFocusEnabled	查询是否支持自动识别人脸位置（仅适用于移动端）
enableCameraAutoFocus	开启自动对焦功能（仅适用于移动端）
setCameraFocusPosition	设置摄像头的对焦位置（仅适用于移动端）
enableCameraTorch	开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）
setAudioRoute	设置音频路由（仅适用于移动端）
getDevicesList	获取设备列表（仅适用于桌面端）
setCurrentDevice	设置当前要使用的设备（仅适用于桌面端）
getCurrentDevice	获取当前正在使用的设备（仅适用于桌面端）
setCurrentDeviceVolume	设置当前设备的音量（仅适用于桌面端）
getCurrentDeviceVolume	获取当前设备的音量（仅适用于桌面端）

setCurrentDeviceMute	设置当前设备的静音状态（仅适用于桌面端）
getCurrentDeviceMute	获取当前设备的静音状态（仅适用于桌面端）
enableFollowingDefaultAudioDevice	设置 SDK 使用的音频设备根据跟随系统默认设备（仅适用于桌面端）
startCameraDeviceTest	开始摄像头测试（仅适用于桌面端）
stopCameraDeviceTest	结束摄像头测试（仅适用于桌面端）
startMicDeviceTest	开始麦克风测试（仅适用于桌面端）
startMicDeviceTest	开始麦克风测试（仅适用于桌面端）
stopMicDeviceTest	结束麦克风测试（仅适用于桌面端）
startSpeakerDeviceTest	开始扬声器测试（仅适用于桌面端）
stopSpeakerDeviceTest	结束扬声器测试（仅适用于桌面端）
startCameraDeviceTest	开始摄像头测试（仅适用于桌面端）
setApplicationPlayVolume	设置 Windows 系统音量合成器中当前进程的音量（仅适用于 Windows 系统）
getApplicationPlayVolume	获取 Windows 系统音量合成器中当前进程的音量（仅适用于 Windows 系统）
setApplicationMuteState	设置 Windows 系统音量合成器中当前进程的静音状态（仅适用于 Windows 系统）
getApplicationMuteState	获取 Windows 系统音量合成器中当前进程的静音状态（仅适用于 Windows 系统）
setCameraCapturerParam	设置摄像头采集偏好
setDeviceObserver	设置 onDeviceChanged 事件回调
setSystemVolumeType	设置系统音量类型（仅适用于移动端）

结构体类型

函数列表	描述
TXCameraCaptureParam	摄像头采集参数
ITXDeviceInfo	音视频设备的相关信息（仅适用于桌面平台）

ITXDeviceCollection	设备信息列表（仅适用于桌面平台）

枚举类型

枚举类型	描述
TXSystemVolumeType	系统音量类型（仅适用于移动设备）
TXAudioRoute	音频路由（即声音的播放模式）
TXMediaDeviceType	设备类型（仅适用于桌面平台）
TXMediaDeviceState	设备操作
TXCameraCaptureMode	摄像头采集偏好

isFrontCamera

isFrontCamera

判断当前是否为前置摄像头（仅适用于移动端）

switchCamera

switchCamera

int switchCamera	(bool frontCamera)

切换前置或后置摄像头（仅适用于移动端）

getCameraZoomMaxRatio

getCameraZoomMaxRatio

获取摄像头的最大缩放倍数（仅适用于移动端）

setCameraZoomRatio

setCameraZoomRatio

int setCameraZoomRatio (float zoomRatio)

设置摄像头的缩放倍数（仅适用于移动端）

参数	描述
zoomRatio	取值范围1 – 5，取值为1表示最远视角（正常镜头），取值为5表示最近视角（放大镜头）。最大值推荐为5，若超过5，视频数据会变得模糊不清。

isAutoFocusEnabled

isAutoFocusEnabled

查询是否支持自动识别人脸位置（仅适用于移动端）

enableCameraAutoFocus

enableCameraAutoFocus

int enableCameraAutoFocus (bool enabled)

开启自动对焦功能（仅适用于移动端）

开启后，SDK 会自动检测画面中的人脸位置，并将摄像头的焦点始终对焦在人脸位置上。

setCameraFocusPosition

setCameraFocusPosition

```
int setCameraFocusPosition (float x
                            float y)
```

设置摄像头的对焦位置（仅适用于移动端）

您可以通过该接口实现如下交互：
1. 在本地摄像头的预览画面上，允许用户单击操作。
2. 在用户的单击位置显示一个矩形方框，以示摄像头会在此处对焦。
3. 随后将用户点击位置的坐标通过本接口传递给 SDK，之后 SDK 会操控摄像头按照用户期望的位置进行对焦。

参数	描述
position	对焦位置，请传入期望对焦点的坐标值

> **注意**
> 使用该接口的前提是先通过 enableCameraAutoFocus 关闭自动对焦功能。

返回值说明：

0：操作成功；负数：操作失败。

enableCameraTorch

enableCameraTorch

```
int enableCameraTorch (bool enabled)
```

开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）

setAudioRoute

setAudioRoute

int setAudioRoute （TXAudioRoute route）

设置音频路由（仅适用于移动端）

手机有两个音频播放设备：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。
设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。

getDevicesList

getDevicesList

ITXDeviceCollection* getDevicesList （TXMediaDeviceType type）

获取设备列表（仅适用于桌面端）

参数	描述
type	设备类型，指定需要获取哪种设备的列表。详见 TXMediaDeviceType 定义。

注意

- 使用完毕后请调用 release 方法释放资源，这样可以让 SDK 维护 ITXDeviceCollection 对象的生命周期。
- 不要使用 delete 释放返回的 Collection 对象，delete ITXDeviceCollection* 指针会导致异常崩溃。
- type 只支持 TXMediaDeviceTypeMic、TXMediaDeviceTypeSpeaker、TXMediaDeviceTypeCamera。
- 此接口只支持 Mac 和 Windows 平台。

setCurrentDevice

setCurrentDevice

int setCurrentDevice （TXMediaDeviceType type

const char* deviceId)

设置当前要使用的设备（仅适用于桌面端）

参数	描述
deviceId	设备ID，您可以通过接口 getDevicesList 获得设备 ID。
type	设备类型，详见 TXMediaDeviceType 定义。

返回值说明：

0：操作成功；负数：操作失败。

getCurrentDevice

getCurrentDevice

ITXDeviceInfo* getCurrentDevice (TXMediaDeviceType type)

获取当前正在使用的设备（仅适用于桌面端）

setCurrentDeviceVolume

setCurrentDeviceVolume

int setCurrentDeviceVolume (TXMediaDeviceType type

uint32_t volume)

设置当前设备的音量（仅适用于桌面端）

这里的音量指的是麦克风的采集音量或者扬声器的播放音量，摄像头是不支持设置音量的。

参数	描述
volume	音量大小，取值范围为0 – 100，默认值：100。

getCurrentDeviceVolume

getCurrentDeviceVolume

uint32_t getCurrentDeviceVolume (TXMediaDeviceType type)

获取当前设备的音量（仅适用于桌面端）

这里的音量指的是麦克风的采集音量或者扬声器的播放音量，摄像头是不支持获取音量的。

setCurrentDeviceMute

setCurrentDeviceMute

int setCurrentDeviceMute (TXMediaDeviceType type
bool mute)

设置当前设备的静音状态（仅适用于桌面端）

这里的音量指的是麦克风和扬声器，摄像头是不支持静音操作的。

getCurrentDeviceMute

getCurrentDeviceMute

bool getCurrentDeviceMute (TXMediaDeviceType type)

获取当前设备的静音状态（仅适用于桌面端）

这里的音量指的是麦克风和扬声器，摄像头是不支持静音操作的。

enableFollowingDefaultAudioDevice

enableFollowingDefaultAudioDevice

int enableFollowingDefaultAudioDevice (TXMediaDeviceType type

bool enable)

设置 SDK 使用的音频设备根据跟随系统默认设备（仅适用于桌面端）

仅支持设置麦克风和扬声器类型，摄像头暂不支持跟随系统默认设备

参数	描述
enable	是否跟随系统默认的音频设备。 • true：跟随。当系统默认音频设备发生改变或者有新音频设备插入时，SDK 立即切换音频设备。 • false：不跟随。当系统默认音频设备发生改变或者有新音频设备插入时，SDK 不会切换音频设备。
type	设备类型，详见 TXMediaDeviceType 定义。

startCameraDeviceTest

startCameraDeviceTest

int startCameraDeviceTest (void* view)

开始摄像头测试（仅适用于桌面端）

注意
在测试过程中可以使用 setCurrentDevice 接口切换摄像头。

stopCameraDeviceTest

stopCameraDeviceTest

结束摄像头测试（仅适用于桌面端）

startMicDeviceTest

startMicDeviceTest

int startMicDeviceTest (uint32_t interval)

开始麦克风测试（仅适用于桌面端）

该接口可以测试麦克风是否能正常工作，测试到的麦克风采集音量的大小，会以回调的形式通知给您，其中 volume 的取值范围为0 - 100。

参数	描述
interval	麦克风音量的回调间隔。

注意
该接口调用后默认会回播麦克风录制到的声音到扬声器中。

startMicDeviceTest

startMicDeviceTest

int startMicDeviceTest (uint32_t interval
bool playback)

开始麦克风测试（仅适用于桌面端）

该接口可以测试麦克风是否能正常工作，测试到的麦克风采集音量的大小，会以回调的形式通知给您，其中 volume 的取值范围为0 - 100。

参数	描述
interval	麦克风音量的回调间隔。
playback	是否开启回播麦克风声音，开启后用户测试麦克风时会听到自己的声音。

stopMicDeviceTest

stopMicDeviceTest

结束麦克风测试（仅适用于桌面端）

startSpeakerDeviceTest

startSpeakerDeviceTest

int startSpeakerDeviceTest (const char* filePath)

开始扬声器测试（仅适用于桌面端）

该接口通过播放指定的音频文件，用于测试播放设备是否能正常工作。如果用户在测试时能听到声音，说明播放设备能正常工作。

参数	描述
filePath	声音文件的路径

stopSpeakerDeviceTest

stopSpeakerDeviceTest

结束扬声器测试（仅适用于桌面端）

startCameraDeviceTest

startCameraDeviceTest

int startCameraDeviceTest (ITRTCVideoRenderCallback* callback)

开始摄像头测试（仅适用于桌面端）

该接口支持自定义渲染，即您可以通过接 ITRTCVideoRenderCallback 回调接口接管摄像头的渲染画面。

setApplicationPlayVolume

setApplicationPlayVolume

int setApplicationPlayVolume (int volume)

设置 Windows 系统音量合成器中当前进程的音量（仅适用于 Windows 系统）

getApplicationPlayVolume

getApplicationPlayVolume

获取 Windows 系统音量合成器中当前进程的音量（仅适用于 Windows 系统）

setApplicationMuteState

setApplicationMuteState

int setApplicationMuteState (bool bMute)

设置 Windows 系统音量合成器中当前进程的静音状态（仅适用于 Windows 系统）

getApplicationMuteState

getApplicationMuteState

获取 Windows 系统音量合成器中当前进程的静音状态（仅适用于 Windows 系统）

setCameraCapturerParam

setCameraCapturerParam

void (const TXCameraCaptureParam & params)

setCameraCapturerParam

设置摄像头采集偏好

setDeviceObserver

setDeviceObserver

void setDeviceObserver (ITXDeviceObserver* observer)

设置 onDeviceChanged 事件回调

setSystemVolumeType

setSystemVolumeType

int setSystemVolumeType (TXSystemVolumeType type)

设置系统音量类型（仅适用于移动端）

@deprecated v9.5 版本开始不推荐使用，建议使用 TRTCCloud 中的 startLocalAudio(quality) 接口替代之，通过 quality 参数来决策音质。

TXSystemVolumeType(Deprecated)

TXSystemVolumeType(Deprecated)

系统音量类型（仅适用于移动设备）

@deprecated v9.5 版本开始不推荐使用。

现代智能手机中一般都具备两套系统音量类型，即“通话音量”和“媒体音量”。

- 通话音量：手机专门为接打电话所设计的音量类型，自带回声抵消（AEC）功能，并且支持通过蓝牙耳机上的麦克风进行拾音，缺点是音质比较一般。

当您通过手机侧面的音量按键下调手机音量时，如果无法将其调至零（也就是无法彻底静音），说明您的手机当前处于通话音量。

- 媒体音量：手机专门为音乐场景所设计的音量类型，无法使用系统的 AEC 功能，并且不支持通过蓝牙耳机的麦克风进行拾音，但具备更好的音乐播放效果。

当您通过手机侧面的音量按键下调手机音量时，如果能够将手机音量调至彻底静音，说明您的手机当前处于媒体音量。

SDK 目前提供了三种系统音量类型的控制模式：自动切换模式、全程通话音量模式、全程媒体音量模式。

枚举	取值	描述
TXSystemVolumeTypeAuto	0	自动切换模式
TXSystemVolumeTypeMedia	1	全程媒体音量
TXSystemVolumeTypeVOIP	2	全程通话音量

TXAudioRoute

TXAudioRoute

音频路由（即声音的播放模式）

音频路由，即声音是从手机的扬声器还是从听筒中播放出来，因此该接口仅适用于手机等移动端设备。
手机有两个扬声器：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。

- 设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
- 设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。

枚举	取值	描述
TXAudioRouteSpeakerphone	0	Speakerphone：使用扬声器播放（即“免提”），扬声器位于手机底部，声音偏大，适合外放音乐。
TXAudioRouteEarpiece	1	Earpiece：使用听筒播放，听筒位于手机顶部，声音偏小，适合需要保护隐私的通话场景。

TXMediaDeviceType

TXMediaDeviceType

设备类型（仅适用于桌面平台）

该枚举值用于定义三种类型的音视频设备，即摄像头、麦克风和扬声器，以便让一套设备管理接口可以操控三种不同类型的设备。

枚举	取值	描述
TXMediaDeviceTypeUnknown	-1	未定义的设备类型
TXMediaDeviceTypeMic	0	麦克风类型设备
TXMediaDeviceTypeSpeaker	1	扬声器类型设备
TXMediaDeviceTypeCamera	2	摄像头类型设备

TXMediaDeviceState

TXMediaDeviceState

设备操作

该枚举值用于本地设备的状态变化通知 onDeviceChanged。

枚举	取值	描述
TXMediaDeviceStateAdd	0	设备已被插入
TXMediaDeviceStateRemove	1	设备已被移除
TXMediaDeviceStateActive	2	设备已启用
TXMediaDefaultDeviceChanged	3	系统默认设备变更

版权所有：腾讯云计算（北京）有限责任公司

TXCameraCaptureMode

TXCameraCaptureMode

摄像头采集偏好

该枚举类型用于摄像头采集参数设置。

枚举	取值	描述
TXCameraResolutionStrategyAuto	0	自动调整采集参数。 SDK 根据实际的采集设备性能及网络情况，选择合适的摄像头输出参数，在设备性能及视频预览质量之间，维持平衡。
TXCameraResolutionStrategyPerformance	1	优先保证设备性能。 SDK 根据用户设置编码器的分辨率和帧率，选择最接近的摄像头输出参数，从而保证设备性能。
TXCameraResolutionStrategyHighQuality	2	优先保证视频预览质量。 SDK选择较高的摄像头输出参数，从而提高预览视频的质量。在这种情况下，会消耗更多的 CPU 及内存做视频前处理。
TXCameraCaptureManual	3	允许用户设置本地摄像头采集的视频宽高。

TXCameraCaptureParam

TXCameraCaptureParam

摄像头采集参数

该设置能决定本地预览图像画质。

枚举类型	描述
height	**字段含义：** 采集图像宽度
mode	**字段含义：** 摄像头采集偏好，请参见 TXCameraCaptureMode

width	**字段含义：** 采集图像长度

TXMediaDeviceInfo

TXMediaDeviceInfo

音视频设备的相关信息（仅适用于桌面平台）

该结构体用于描述一个音视频设备的关键信息，比如设备 ID、设备名称等等，以便用户能够在用户界面上选择自己期望使用的音视频设备。

枚举类型	描述
getDeviceName()	设备名称（UTF-8）
getDevicePID()	设备 ID（UTF-8）

ITXDeviceCollection

ITXDeviceCollection

设备信息列表（仅适用于桌面平台）

此结构体的作用相当于 std::vector<ITXDeviceInfo>，用于解决不同版本的 STL 容器的二进制兼容问题。

枚举类型	描述
getCount()	设备数量
index)	设备信息（JSON 格式） **注意** 示例：{"SupportedResolution":[{"width":640,"height":480},{"width":320,"height":240}]} param index 设备索引，值为 [0,getCount)，return 返回 JSON 格式的设备信息

release()	释放设备列表，请不要使用 delete 释放资源 !!!

类型定义

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTC 关键类型定义

Function: 分辨率、质量等级等枚举和常量值的定义

类型定义

结构体类型

函数列表	描述
TRTCParams	进房参数
TRTCVideoEncParam	视频编码参数
TRTCNetworkQosParam	网络流控（Qos）参数集
TRTCRenderParams	视频画面的渲染参数
TRTCQualityInfo	网络质量
TRTCVolumeInfo	音量大小
TRTCSpeedTestParams	测速参数
TRTCSpeedTestResult	网络测速结果
TRTCTexture	视频纹理数据
TRTCVideoFrame	视频帧信息
TRTCAudioFrame	音频帧数据
TRTCMixUser	云端混流中各路画面的描述信息
TRTCTranscodingConfig	云端混流的排版布局和转码参数
TRTCPublishCDNParam	向非腾讯云 CDN 上发布音视频流时需设置的转推参数

TRTCAudioRecordingParams	本地音频文件的录制参数
TRTCLocalRecordingParams	本地媒体文件的录制参数
TRTCAudioEffectParam	音效参数（已废弃）
TRTCSwitchRoomConfig	房间切换参数
TRTCAudioFrameCallbackFormat	音频自定义回调的格式参数
TRTCImageBuffer	TRTC 屏幕分享图标信息以及 mute image 垫片
TRTCScreenCaptureSourceInfo	屏幕分享的目标信息（仅适用于桌面系统）
TRTCScreenCaptureProperty	屏幕分享的进阶控制参数
TRTCAudioParallelParams	远端音频流智能并发播放策略的参数
TRTCUser	媒体流发布相关配置的用户信息
TRTCPublishCdnUrl	向腾讯或者第三方 CDN 上发布音视频流时需设置的 url 配置
TRTCPublishTarget	目标推流配置
TRTCVideoLayout	转码视频布局
TRTCWatermark	水印布局
TRTCStreamEncoderParam	媒体流编码输出参数
TRTCStreamMixingConfig	媒体流转码配置参数
TRTCPayloadPrivateEncryptionConfig	媒体流私有加密配置
TRTCAudioVolumeEvaluateParams	音量评估等相关参数设置

枚举类型

枚举类型	描述
TRTCVideoResolution	视频分辨率

TRTCVideoResolutionMode	视频宽高比模式
TRTCVideoStreamType	视频流类型
TRTCVideoFillMode	视频画面填充模式
TRTCVideoRotation	视频画面旋转方向
TRTCBeautyStyle	美颜（磨皮）算法
TRTCVideoPixelFormat	视频像素格式
TRTCVideoBufferType	视频数据传递方式
TRTCVideoMirrorType	视频的镜像类型
TRTCSnapshotSourceType	本地视频截图的数据源
TRTCAppScene	应用场景
TRTCRoleType	角色
TRTCQosControlMode	流控模式（已废弃）
TRTCVideoQosPreference	画质偏好
TRTCQuality	网络质量
TRTCAVStatusType	视频状态类型
TRTCAVStatusChangeReason	视频状态变化原因类型
TRTCAudioQuality	声音音质
TRTCAudioFrameFormat	音频帧的内容格式
TRTCAudioFrameOperationMode	音频回调数据读写模式
TRTCLogLevel	Log 级别
TRTCScreenCaptureSourceType	屏幕分享的目标类型（仅适用于桌面端）

TRTCTranscodingConfigMode	云端混流的排版模式
TRTCLocalRecordType	媒体录制类型
TRTCMixInputType	混流输入类型
TRTCWaterMarkSrcType	水印图片的源类型
TRTCAudioRecordingContent	音频录制内容类型
TRTCPublishMode	媒体流发布模式
TRTCEncryptionAlgorithm	加密算法
TRTCSpeedTestScene	测速场景
TRTCGravitySensorAdaptiveMode	设置重力感应的适配模式（仅适用于移动端）

TRTCVideoResolution

TRTCVideoResolution

视频分辨率

此处仅定义横屏分辨率（如 640 × 360），如需使用竖屏分辨率（如360 × 640），需要同时指定 TRTCVideoResolutionMode 为 Portrait。

枚举	取值	描述
TRTCVideoResolution_120_120	1	宽高比 1:1；分辨率 120x120；建议码率（VideoCall）80kbps; 建议码率（LIVE）120kbps。
TRTCVideoResolution_160_160	3	宽高比 1:1 分辨率 160x160；建议码率（VideoCall）100kbps; 建议码率（LIVE）150kbps。
TRTCVideoResolution_270_270	5	宽高比 1:1；分辨率 270x270；建议码率（VideoCall）200kbps; 建议码率（LIVE）

		300kbps。
TRTCVideoResolution_480_480	7	宽高比 1:1；分辨率 480x480；建议码率（VideoCall）350kbps；建议码率（LIVE）500kbps。
TRTCVideoResolution_160_120	50	宽高比4:3；分辨率 160x120；建议码率（VideoCall）100kbps；建议码率（LIVE）150kbps。
TRTCVideoResolution_240_180	52	宽高比 4:3；分辨率 240x180；建议码率（VideoCall）150kbps；建议码率（LIVE）250kbps。
TRTCVideoResolution_280_210	54	宽高比 4:3；分辨率 280x210；建议码率（VideoCall）200kbps；建议码率（LIVE）300kbps。
TRTCVideoResolution_320_240	56	宽高比 4:3；分辨率 320x240；建议码率（VideoCall）250kbps；建议码率（LIVE）375kbps。
TRTCVideoResolution_400_300	58	宽高比 4:3；分辨率 400x300；建议码率（VideoCall）300kbps；建议码率（LIVE）450kbps。
TRTCVideoResolution_480_360	60	宽高比 4:3；分辨率 480x360；建议码率（VideoCall）400kbps；建议码率（LIVE）600kbps。
TRTCVideoResolution_640_480	62	宽高比 4:3；分辨率 640x480；建议码率（VideoCall）600kbps；建议码率（LIVE）900kbps。
TRTCVideoResolution_960_720	64	宽高比 4:3；分辨率 960x720；建议码率（VideoCall）1000kbps；建议码率（LIVE）1500kbps。
TRTCVideoResolution_160_90	100	宽高比 16:9；分辨率 160x90；建议码率（VideoCall）150kbps；建议码率（LIVE）250kbps。
TRTCVideoResolution_256_144	102	宽高比 16:9；分辨率 256x144；建议码率（VideoCall）200kbps；建议码率（LIVE）300kbps。
TRTCVideoResolution_320_180	104	宽高比 16:9；分辨率 320x180；建议码率（VideoCall）250kbps；建议码率（LIVE）

		400kbps。
TRTCVideoResolution_480_270	106	宽高比 16:9；分辨率 480x270；建议码率（VideoCall）350kbps; 建议码率（LIVE）550kbps。
TRTCVideoResolution_640_360	108	宽高比 16:9；分辨率 640x360；建议码率（VideoCall）500kbps; 建议码率（LIVE）900kbps。
TRTCVideoResolution_960_540	110	宽高比 16:9；分辨率 960x540；建议码率（VideoCall）850kbps; 建议码率（LIVE）1300kbps。
TRTCVideoResolution_1280_720	112	宽高比 16:9；分辨率 1280x720；建议码率（VideoCall）1200kbps; 建议码率（LIVE）1800kbps。
TRTCVideoResolution_1920_1080	114	宽高比 16:9；分辨率 1920x1080；建议码率（VideoCall）2000kbps; 建议码率（LIVE）3000kbps。

TRTCVideoResolutionMode

TRTCVideoResolutionMode

视频宽高比模式

TRTCVideoResolution 中仅定义了横屏分辨率（如 640 × 360），如需使用竖屏分辨率（如360 × 640），需要同时指定 TRTCVideoResolutionMode 为 Portrait。

枚举	取值	描述
TRTCVideoResolutionModeLandscape	0	横屏分辨率，例如：TRTCVideoResolution_640_360 + TRTCVideoResolutionModeLandscape = 640 × 360。
TRTCVideoResolutionModePortrait	1	竖屏分辨率，例如：TRTCVideoResolution_640_360 + TRTCVideoResolutionModePortrait = 360 × 640。

TRTCVideoStreamType

TRTCVideoStreamType

视频流类型

TRTC 内部有三种不同的视频流，分别是：

- 高清大画面：一般用来传输摄像头的视频数据。
- 低清小画面：小画面和大画面的内容相互，但是分辨率和码率都比大画面低，因此清晰度也更低。
- 辅流画面：一般用于屏幕分享，同一时间在同一个房间中只允许一个用户发布辅流视频，其他用户必须要等该用户关闭之后才能发布自己的辅流。

> **注意**
> 不支持单独开启低清小画面，小画面必须依附于大画面而存在，SDK 会自动设定低清小画面的分辨率和码率。

枚举	取值	描述
TRTCVideoStreamTypeBig	0	高清大画面，一般用来传输摄像头的视频数据。
TRTCVideoStreamTypeSmall	1	低清小画面：小画面和大画面的内容相互，但是分辨率和码率都比大画面低，因此清晰度也更低。
TRTCVideoStreamTypeSub	2	辅流画面：一般用于屏幕分享，同一时间在同一个房间中只允许一个用户发布辅流视频，其他用户必须要等该用户关闭之后才能发布自己的辅流。

TRTCVideoFillMode

TRTCVideoFillMode

视频画面填充模式

如果视频显示区域的宽高比不等于视频内容的宽高比时，需要指定画面的填充模式。

枚举	取值	描述
TRTCVideoFillMode	0	填充模式：即将画面内容居中等比缩放以充满整个显示区域，

_Fill		超出显示区域的部分将会被裁剪掉，此模式下画面可能不完整。
TRTCVideoFillMode_Fit	1	适应模式：即按画面长边进行缩放以适应显示区域，短边部分会被填充为黑色，此模式下图像完整但可能留有黑边。

TRTCVideoRotation

TRTCVideoRotation

视频画面旋转方向

TRTC 提供了对本地和远程画面的旋转角度设置 API，下列的旋转角度都是指顺时针方向的。

枚举	取值	描述
TRTCVideoRotation0	0	不旋转。
TRTCVideoRotation90	1	顺时针旋转90度。
TRTCVideoRotation180	2	顺时针旋转180度。
TRTCVideoRotation270	3	顺时针旋转270度。

TRTCBeautyStyle

TRTCBeautyStyle

美颜（磨皮）算法

TRTC 内置多种不同的磨皮算法，您可以选择最适合您产品定位的方案。

枚举	取值	描述
TRTCBeautyStyleSmooth	0	光滑，算法比较激进，磨皮效果比较明显，适用于秀场直播。

TRTCBeautyStyleNature	1	自然，算法更多地保留了面部细节，磨皮效果更加自然，适用于绝大多数直播场景。

TRTCVideoPixelFormat

TRTCVideoPixelFormat

视频像素格式

TRTC 提供针对视频的自定义采集和自定义渲染功能：

- 在自定义采集功能中，您可以用下列枚举值描述您采集的视频像素格式。
- 在自定义渲染功能中，您可以指定您期望 SDK 回调出的视频像素格式。

枚举	取值	描述
TRTCVideoPixelFormat_Unknown	0	未定义的格式。
TRTCVideoPixelFormat_I420	1	YUV420P(I420) 格式。
TRTCVideoPixelFormat_Texture_2D	2	OpenGL 2D 纹理格式。
TRTCVideoPixelFormat_BGRA32	3	BGRA 格式
TRTCVideoPixelFormat_NV21	4	NV21 格式。
TRTCVideoPixelFormat_RGBA32	5	RGBA 格式。

TRTCVideoBufferType

TRTCVideoBufferType

视频数据传递方式

在自定义采集和自定义渲染功能，您需要用到下列枚举值来指定您希望以什么方式传递视频数据：

- 方案一：使用内存 Buffer 传递视频数据，该方案在 iOS 效率尚可，但在 Android 系统上效率较差，Windows 暂时仅支持内存 Buffer 的传递方式。
- 方案二：使用 Texture 纹理传递视频数据，该方案在 iOS 和 Android 系统下均有较高的效率，Windows 暂不支持，需要您有一定的 OpenGL 编程基础。

枚举	取值	描述
TRTCVideoBufferType_Unknown	0	未定义的传递方式。
TRTCVideoBufferType_Buffer	1	使用内存 Buffer 传递视频数据，iOS：PixelBuffer；Android：用于 JNI 层的 Direct Buffer；Win：内存数据块。
TRTCVideoBufferType_Texture	3	使用 Texture 纹理传递视频数据。

TRTCVideoMirrorType

TRTCVideoMirrorType

视频的镜像类型

视频的镜像是指对视频内容进行左右翻转，尤其是对本地的摄像头预览视频，开启镜像后能给主播带来熟悉的“照镜子”体验。

枚举	取值	描述
TRTCVideoMirrorType_Auto	0	自动模式：如果正使用前置摄像头则开启镜像，如果是后置摄像头则不开启镜像（仅适用于移动设备）。
TRTCVideoMirrorType_Enable	1	强制开启镜像，不论当前使用的是前置摄像头还是后置摄像头。
TRTCVideoMirrorType_Disable	2	强制关闭镜像，不论当前使用的是前置摄像头还是后置摄像头。

TRTCSnapshotSourceType

TRTCSnapshotSourceType

本地视频截图的数据源

SDK 支持从如下两种数据源中截取图片并保存成本地文件：

- 视频流：从视频流中截取原生的视频内容，截取的内容不受渲染控件的显示控制。
- 渲染层：从渲染控件中截取显示的视频内容，可以做到用户所见即所得的效果，但如果显示区域过小，截取出的图片也会很小。

枚举	取值	描述
TRTCSnapshotSourceTypeStream	0	从视频流中截取原生的视频内容，截取的内容不受渲染控件的显示控制。
TRTCSnapshotSourceTypeView	1	从渲染控件中截取显示的视频内容，可以做到用户所见即所得的效果，但如果显示区域过小，截取出的图片也会很小。

TRTCAppScene

TRTCAppScene

应用场景

TRTC 针对常见的音视频应用场景都进行了定向优化，以满足各种垂直场景下的差异化要求，主要场景可以分为如下两类：

- 直播（LIVE）场景：包括 LIVE 和 VoiceChatRoom，前者是音频+视频，后者是纯音频。

直播场景下，用户被分成“主播”和“观众”两种角色，单个房间中同时最多支持10万人在线，适合于观众人数众多的直播场景。

- 实时（RTC）场景：包括 VideoCall 和 AudioCall，前者是音频+视频，后者是纯音频。

实时场景下，用户没有角色的差异，但单个房间中同时最多支持 300 人在线，适合于小范围实时通信的场景。

枚举	取值	描述
TRTCAppSceneVideoCall	0	视频通话场景，支持720P、1080P高清画质，单个房间最多支持300人同时在线，最高支持50人同时发言。 适用于[1对1视频通话]、[300人视频会议]、[在线问诊]、[教育小班课]、[远程面试]等业务场景。
TRTCAppSceneLIVE	1	视频互动直播，支持平滑上下麦，切换过程无需等待，主播延时小于300ms；支持十万级别观众同时播放，播放延时

		低至1000ms。 适用于[低延时互动直播]、[大班课]、[主播PK]、[视频相亲]、[在线互动课堂]、[远程培训]、[超大型会议]等业务场景。 **注意** 此场景下，您必须通过 TRTCParams 中的 role 字段指定当前用户的角色。
TRTCAppSceneAudioCall	2	语音通话场景，默认采用 SPEECH 音质，单个房间最多支持300人同时在线，最高支持50人同时发言。 适用于 [1对1语音通话]、[300人语音会议]、[语音聊天]、[语音会议]、[在线狼人杀] 等业务场景。
TRTCAppSceneVoiceChatRoom	3	语音互动直播，支持平滑上下麦，切换过程无需等待，主播延时小于 300ms；支持十万级别观众同时播放，播放延时低至 1000ms。 适用于 [语音俱乐部]、[在线K歌房]、[音乐直播间]、[FM电台] 等业务场景。 **注意** 此场景下，您必须通过 TRTCParams 中的 role 字段指定当前用户的角色。

TRTCRoleType

TRTCRoleType

角色

仅适用于直播类场景（即 TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom），把用户区分成两种不同的身份：

- 主播：可以随时发布自己的音视频流，但人数有限制，同一个房间中最多只允许 50 个主播同时发布自己的音视频流。
- 观众：只能观看其他用户的音视频流，要发布音视频流，需要先通过 switchRole 切换成主播，同一个房间中最多能容纳10万观众。

枚举	取值	描述

TRTCRoleAnchor	20	主播：可以随时发布自己的音视频流，但人数有限制，同一个房间中最多只允许 50 个主播同时发布自己的音视频流。
TRTCRoleAudience	21	观众：只能观看其他用户的音视频流，要发布音视频流，需要先通过 switchRole 切换成主播，同一个房间中最多能容纳10万观众。

TRTCQosControlMode(Deprecated)

TRTCQosControlMode(Deprecated)

流控模式（已废弃）

枚举	取值	描述
TRTCQosControlModeClient	0	本地控制，用于 SDK 开发内部调试，客户请勿使用。
TRTCQosControlModeServer	1	云端控制，默认模式，推荐选择。

TRTCVideoQosPreference

TRTCVideoQosPreference

画质偏好

TRTC 在弱网络环境下有两种调控模式：“优先保证画面清晰”或“优先保证画面流畅”，两种模式均会优先保障声音数据的传输。

枚举	取值	描述
TRTCVideoQosPreferenceSmooth	1	流畅优先：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的流畅性，代价就是画面会比较模糊且伴随有较多的马赛克。
TRTCVideoQosPreferenceClear	2	清晰优先（默认值）：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的清晰度，代价就是画面会比较卡顿。

TRTCQuality

TRTCQuality

网络质量

TRTC 会每隔两秒对当前的网络质量进行评估，评估结果为六个等级：Excellent 表示最好，Down 表示最差。

枚举	取值	描述
TRTCQuality_Unknown	0	未定义。
TRTCQuality_Excellent	1	当前网络非常好。
TRTCQuality_Good	2	当前网络比较好。
TRTCQuality_Poor	3	当前网络一般。
TRTCQuality_Bad	4	当前网络较差。
TRTCQuality_Vbad	5	当前网络很差。
TRTCQuality_Down	6	当前网络不满足 TRTC 的最低要求。

TRTCAVStatusType

TRTCAVStatusType

视频状态类型

该枚举类型用于视频状态变化回调接口 onRemoteVideoStatusUpdated ，用于指定当前的视频状态。

枚举	取值	描述
TRTCAVStatusStopped	0	停止播放。
TRTCAVStatusPlaying	1	正在播放。

TRTCAVStatusLoading	2	正在加载。

TRTCAVStatusChangeReason

TRTCAVStatusChangeReason

视频状态变化原因类型

该枚举类型用于视频状态变化回调接口 onRemoteVideoStatusUpdated ，用于指定当前的视频状态原因。

枚举	取值	描述
TRTCAVStatusChangeReasonInternal	0	缺省值。
TRTCAVStatusChangeReasonBufferingBegin	1	网络缓冲。
TRTCAVStatusChangeReasonBufferingEnd	2	结束缓冲。
TRTCAVStatusChangeReasonLocalStarted	3	本地启动视频流播放。
TRTCAVStatusChangeReasonLocalStopped	4	本地停止视频流播放。
TRTCAVStatusChangeReasonRemoteStarted	5	远端视频流开始（或继续）。
TRTCAVStatusChangeReasonRemoteStopped	6	远端视频流停止（或中断）。

TRTCAudioQuality

TRTCAudioQuality

声音音质

TRTC 提供了三种精心校调好的模式，用来满足各种垂直场景下对音质的差异化追求：

- 人声模式（Speech）：适用于以人声沟通为主的应用场景，该模式下音频传输的抗性较强，TRTC 会通过各种人声处理技术保障在弱网络环境下的流畅度最佳。
- 音乐模式（Music）：适用于对声乐要求很苛刻的场景，该模式下音频传输的数据量很大，TRTC 会通过各项技术确保音乐信号在各频段均能获得高保真的细节还原度。
- 默认模式（Default）：介于 Speech 和 Music 之间的档位，对音乐的还原度比人声模式要好，但传输数据量比音乐模式要低很多，对各种场景均有不错的适应性。

枚举	取值	描述
TRTCAudioQualitySpeech	1	人声模式：采样率：16k；单声道；编码码率：16kbps；具备几个模式中最强的网络抗性，适合语音通话为主的场景，比如在线会议，语音通话等。
TRTCAudioQualityDefault	2	默认模式：采样率：48k；单声道；编码码率：50kbps；介于 Speech 和 Music 之间的档位，SDK 默认档位，推荐选择。
TRTCAudioQualityMusic	3	音乐模式：采样率：48k；全频带立体声；编码码率：128kbps；适合需要高保真传输音乐的场景，比如在线K歌、音乐直播等。

TRTCAudioFrameFormat

TRTCAudioFrameFormat

音频帧的内容格式

枚举	取值	描述
TRTCAudioFrameFormatNone	0	None
TRTCAudioFrameFormatPCM	Not Defined	PCM 格式的音频数据。

TRTCAudioFrameOperationMode

TRTCAudioFrameOperationMode

音频回调数据读写模式

TRTC 提供了两种音频回调数据的操作模式。

- 读写模式（ReadWrite）：可以获取并修改回调的音频数据，默认模式。
- 只读模式（ReadOnly）：仅从回调中获取音频数据。

枚举	取值	描述
TRTCAudioFrameOperationModeReadWrite	0	读写模式：可以获取并修改回调的音频数据。
TRTCAudioFrameOperationModeReadOnly	1	只读模式：仅从回调中获取音频数据。

TRTCLogLevel

TRTCLogLevel

Log 级别

不同的日志等级定义了不同的详实程度和日志数量，推荐一般情况下将日志等级设置为：TRTCLogLevelInfo。

枚举	取值	描述
TRTCLogLevelVerbose	0	输出所有级别的 Log。
TRTCLogLevelDebug	1	输出 DEBUG，INFO，WARNING，ERROR 和 FATAL 级别的 Log。
TRTCLogLevelInfo	2	输出 INFO，WARNING，ERROR 和 FATAL 级别的 Log。
TRTCLogLevelWarn	3	输出WARNING，ERROR 和 FATAL 级别的 Log。
TRTCLogLevelError	4	输出ERROR 和 FATAL 级别的 Log。

版权所有：腾讯云计算（北京）有限责任公司

TRTCLogLevelFatal	5	仅输出 FATAL 级别的 Log。
TRTCLogLevelNone	6	不输出任何 SDK Log。

TRTCScreenCaptureSourceType

TRTCScreenCaptureSourceType

屏幕分享的目标类型（仅适用于桌面端）

枚举	取值	描述
TRTCScreenCaptureSourceTypeUnknown	-1	未定义。
TRTCScreenCaptureSourceTypeWindow	0	该分享目标是某一个应用的窗口。
TRTCScreenCaptureSourceTypeScreen	1	该分享目标是某一台显示器的屏幕。
TRTCScreenCaptureSourceTypeCustom	2	该分享目标是用户自定义的数据源。

TRTCTranscodingConfigMode

TRTCTranscodingConfigMode

云端混流的排版模式

TRTC 的云端混流服务能够将房间中的多路音视频流混合成一路，因此您需要指定画面的排版方案，我们提供了如下几种排版模式。

枚举	取值	描述
TRTCTranscodingConfigMode_Unknown	0	未定义。
TRTCTranscodingConfigMode_Manual	1	全手动排版模式。

		该模式下，您需要指定每一路画面的精确排版位置。该模式的自由度最高，但易用性也最差： • 您需要填写 TRTCTranscodingConfig 中的所有参数，包括每一路画面（TRTCMixUser）的位置坐标。 • 您需要监听 ITRTCCloudCallback 中的 onUserVideoAvailable() 和 onUserAudioAvailable() 事件回调，并根据当前房间中各个麦上用户的音视频状态不断地调整 mixUsers 参数。
TRTCTranscodingConfigMode_Template_PureAudio	2	纯音频模式。 该模式适用于语音通话（AudioCall）和语音聊天室（VoiceChatRoom）等纯音频的应用场景。 • 您只需要在进入房间后，通过 setMixTranscodingConfig 接口设置一次，之后 SDK 就会自动把房间内所有上麦用户的声音混流到当前用户的直播流上。 • 您无需设置 TRTCTranscodingConfig 中的 mixUsers 参数，只需设置 audioSampleRate、audioBitrate 和 audioChannels 等参数即可。
TRTCTranscodingConfigMode_Template_PresetLayout	3	预排版模式。 最受欢迎的排版模式，因为该模式支持您通过占位符提前对各路画面的位置进行设定，之后 SDK 会自动根据房间中画面的路数动态进行适配调整。 此模式下，您依然需要设置 mixUsers 参数，但可以将 userId 设置为“占位符”，可选的占位符有： • "$PLACE_HOLDER_REM

		OTE$" ：指代远程用户的画面，可以设置多个。 • "$PLACE_HOLDER_LOCAL_MAIN$"：指代本地摄像头画面，只允许设置一个。 • "$PLACE_HOLDER_LOCAL_SUB$"：指代本地屏幕分享画面，只允许设置一个。 此模式下，您不需要监听 ITRTCCloudCallback 中的 onUserVideoAvailable 和 onUserAudioAvailable 回调进行实时调整，只需要在进房成功后调用一次 setMixTranscodingConfig 即可，之后 SDK 会自动将真实的 userId 补位到您设置的占位符上。
TRTCTranscodingConfigMode_Template_ScreenSharing	4	屏幕分享模式。 适用于在线教育场景等以屏幕分享为主的应用场景，仅支持 Windows 和 Mac 两个平台的 SDK。 该模式下，SDK 会先根据您通过 videoWidth 和 videoHeight 参数设置的目标分辨率构建一张画布， • 当老师未开启屏幕分享时，SDK 会将老师的摄像头画面等比例拉伸绘制到该画布上； • 当老师开启屏幕分享之后，SDK 会将屏幕分享画面绘制到同样的画布上。 此种排版模式的目的是为了确保混流模块的输出分辨率一致，避免课程回放和网页观看的花屏问题（网页播放器不支持可变分辨率）。 同时，连麦学生的声音也会被默认混合到老师的音视频流中。 由于教学模式下的视频内容以屏幕分享为主，因此同时传输摄像头画面和屏幕分享画面是非常浪费带宽的。 推荐的做法是直接将摄像头画面通过 setLocalVideoRenderCallback 接口自定义绘制到当前屏幕上。

		在该模式下，您无需设置 TRTCTranscodingConfig 中的 mixUsers 参数，SDK 不会混合学生的画面，以免干扰屏幕分享的效果。 您可以将 TRTCTranscodingConfig 中的 width x height 设为 0px × 0px，SDK 会自动根据用户当前屏幕的宽高比计算出一个合适的分辨率： • 如果老师当前屏幕宽度 <= 1920px，SDK 会使用老师当前屏幕的实际分辨率。 • 如果老师当前屏幕宽度 > 1920px，SDK 会根据当前屏幕宽高比，选择 1920x1080(16:9)、1920x1200(16:10)、1920x1440(4:3) 三种分辨率中的一种。

TRTCRecordType

TRTCRecordType

媒体录制类型

该枚举类型用于本地媒体录制接口 startLocalRecording，用于指定是录制音视频文件还是纯音频文件。

枚举	取值	描述
TRTCLocalRecordType_Audio	0	仅录制音频。
TRTCLocalRecordType_Video	1	仅录制视频。
TRTCLocalRecordType_Both	2	同时录制音频和视频。

版权所有：腾讯云计算（北京）有限责任公司

TRTCMixInputType

TRTCMixInputType

混流输入类型

枚举	取值	描述
TRTCMixInputTypeUndefined	0	默认值，考虑到针对老版本的兼容性，如果您指定了 inputType 为 Undefined，SDK 会根据另一个参数 pureAudio 的数值决定混流输入类型。
TRTCMixInputTypeAudioVideo	1	混入音频和视频。
TRTCMixInputTypePureVideo	2	只混入视频。
TRTCMixInputTypePureAudio	3	只混入音频。
TRTCMixInputTypeWatermark	4	混入水印，此时您无需指定 userId 字段，但需要指定 image 字段，推荐使用 png 格式的图片。

TRTCWaterMarkSrcType

TRTCWaterMarkSrcType

水印图片的源类型

枚举	取值	描述
TRTCWaterMarkSrcTypeFile	0	图片文件路径，支持 BMP、GIF、JPEG、PNG、TIFF、Exif、WMF 和 EMF 文件格式。
TRTCWaterMarkSrcTypeBGRA32	1	BGRA32格式内存块
TRTCWaterMarkSrcTypeRGBA32	2	RGBA32格式内存块。

版权所有：腾讯云计算（北京）有限责任公司

TRTCAudioRecordingContent

TRTCAudioRecordingContent

音频录制内容类型

该枚举类型用于音频录制接口 startAudioRecording，用于指定录制音频的内容。

枚举	取值	描述
TRTCAudioRecordingContentAll	0	录制本地和远端所有音频。
TRTCAudioRecordingContentLocal	1	仅录制本地音频。
TRTCAudioRecordingContentRemote	2	仅录制远端音频。

TRTCPublishMode

TRTCPublishMode

媒体流发布模式

该枚举类型用于媒体流发布接口 startPublishMediaStream TRTC 的媒体流发布服务能够将房间中的多路音视频流混合成一路发布至 CDN 或者回推到房间内，也可以将您当前的这路音视频发布到腾讯或者第三方 CDN 因此您需要指定对应媒体流的发布模式，我们提供了如下几种模式。

枚举	取值	描述
TRTCPublishModeUnknown	0	未定义。
TRTCPublishBigStreamToCdn	1	您可以通过设置该参数将您房间内的主路流（TRTCVideoStreamTypeBig）发布到腾讯或者第三方直播 CDN 服务商（仅支持标准 RTMP 协议）。
TRTCPublishSubStreamToCdn	2	您可以通过设置该参数将您房间内的辅路流（TRTCVideoStreamTypeSub）发布到腾讯或者第三方直播 CDN 服务商（仅支持标准 RTMP 协议）。

TRTCPublishMixStreamToCdn	3	您可以通过设置该参数，配合编码输出参数（TRTCStreamEncoderParam）和混流转码参数（TRTCStreamMixingConfig），将您指定的多路音视频流进行转码并发布到腾讯或者第三方直播 CDN 服务商（仅支持标准 RTMP 协议）。
TRTCPublishMixStreamToRoom	4	您可以通过设置该参数，配合媒体流编码输出参数（TRTCStreamEncoderParam）和混流转码参数（TRTCStreamMixingConfig），将您指定的多路音视频流进行转码并发布到您指定的房间中。 • 通过 TRTCPublishTarget 中的 TRTCUser 进行指定回推房间的机器人信息。

TRTCEncryptionAlgorithm

TRTCEncryptionAlgorithm

加密算法

该枚举类型用于媒体流私有加密算法选择。

枚举	取值	描述
TRTCEncryptionAlgorithmAes128Gcm	0	AES GCM 128。
TRTCEncryptionAlgorithmAes256Gcm	1	AES GCM 256。

TRTCSpeedTestScene

TRTCSpeedTestScene

测速场景

该枚举类型用于测速场景选择。

枚举	取值	描述
TRTCSpeedTestScene_DelayTesting	1	延迟测试。

TRTCSpeedTestScene_DelayAndBandwidthTesting	2	延迟与带宽测试。
TRTCSpeedTestScene_OnlineChorusTesting	3	在线合唱测试。

TRTCGravitySensorAdaptiveMode

TRTCGravitySensorAdaptiveMode

设置重力感应的适配模式（仅适用于移动端）

枚举	取值	描述
TRTCGravitySensorAdaptiveMode_Disable	0	关闭重力感应，根据当前采集分辨率与设置的编码分辨率决策，如果两者不一致，则通过旋转90度，保证最大画幅。
TRTCGravitySensorAdaptiveMode_FillByCenterCrop	1	开启重力感应，始终保证远端画面图像为正，中间过程需要处理分辨率不一致时，采用居中裁剪模式。
TRTCGravitySensorAdaptiveMode_FitWithBlackBorder	2	开启重力感应，始终保证远端画面图像为正，中间过程需要处理分辨率不一致时，采用叠加黑边模式。

TRTCParams

TRTCParams

进房参数

作为 TRTC SDK 的进房参数，只有该参数填写正确，才能顺利进入 roomId 或者 strRoomId 所指定的音视频房间。
由于历史原因，TRTC 支持数字和字符串两种类型的房间号，分别是 roomId 和 strRoomId。
请注意：不要混用 roomId 和 strRoomId，因为它们之间是不互通的，比如数字 123 和字符串 “123” 在 TRTC 看来是两个完全不同的房间。

枚举类型	描述
businessInfo	【字段含义】业务数据字段（选填），部分高级特性才需要用到此字段。 【推荐取值】请不要自行设置该字段。
privateMapKey	【字段含义】用于权限控制的权限票据（选填），当您希望某个房间只能让特定的 userId 进入时，需要使用 privateMapKey 进行权限保护。 【推荐取值】仅建议有高级别安全需求的客户使用，更多详情请参见 进房权限保护 。
role	【字段含义】直播场景下的角色，仅适用于直播场景（ TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom ），通话场景下指定该参数是无效的。 【推荐取值】默认值：主播（ TRTCRoleAnchor ）。
roomId	【字段含义】数字房间号，在同一个房间里的用户（userId）可以彼此看到对方并进行音视频通话。 【推荐取值】取值范围：1 - 4294967294。 【特别说明】roomId 与 strRoomId 是互斥的，若您选用 strRoomId，则 roomId 需要填写为 0。若两者都填，SDK 将优先选用 roomId。 【请您注意】不要混用 roomId 和 strRoomId，因为它们之间是不互通的，比如数字 123 和字符串 `123` 在 TRTC 看来是两个完全不同的房间。
sdkAppId	【字段含义】应用标识（必填），腾讯云基于 sdkAppId 完成计费统计。 【推荐取值】在 实时音视频控制台 创建应用后可以在账号信息页面中得到该 ID。
strRoomId	【字段含义】字符串房间号，在同一个房间里的用户（userId）可以彼此看到对方并进行音视频通话。 【特别说明】roomId 与 strRoomId 是互斥的，若您选用 strRoomId，则 roomId 需要填写为0。若两者都填，SDK 将优先选用 roomId。 【请您注意】不要混用 roomId 和 strRoomId，因为它们之间是不互通的，比如数字 123 和字符串 `123` 在 TRTC 看来是两个完全不同的房间。 【推荐取值】限制长度为 64 字节。以下为支持的字符集范围（共 89 个字符）： • 大小写英文字母（a-zA-Z）； • 数字（0-9）； • 空格、 `!` 、 `#` 、 `$` 、 `%` 、 `&` 、 `(` 、 `)` 、 `+` 、 `-` 、 `:` 、 `;` 、 `<` 、 `=` 、 `.` 、 `>` 、 `?` 、 `@` 、 `[` 、 `]` 、 `^` 、 `_` 、 `{` 、 `}` 、 `|` 、 `~` 、 `,` 。
streamId	【字段含义】用于指定在腾讯云直播平台上的 streamId（选填），设置之

	后，您可以在腾讯云直播 CDN 上通过标准拉流方案（FLV 或 HLS）播放该用户的音视频流。 【推荐取值】限制长度为64字节，可以不填写，一种推荐的方案是使用 `sdkappid_roomid_userid_main` 作为 streamid，这种命名方式容易辨认且不会在您的多个应用中发生冲突。 【特殊说明】要使用腾讯云直播 CDN，您需要先在 控制台 中的功能配置页开启“启动自动旁路直播”开关。 【参考文档】CDN 旁路直播 。
userDefineRecordId	【字段含义】云端录制开关（选填），用于指定是否要在云端将该用户的音视频流录制下来。 【参考文档】云端录制 。 【推荐取值】限制长度为64字节，只允许包含大小写英文字母（a-zA-Z）、数字（0-9）及下划线和连词符。 方案一：手动录制方案： 1. 在 控制台>应用管理>云端录制配置 中开启云端录制。 2. 设置 `录制形式` 为 `手动录制` 。 3. 设置手动录制后，在一个 TRTC 房间中只有设置了 userDefineRecordId 参数的用户才会在云端录制出视频文件，不指定该参数的用户不会产生录制行为。 4. 云端会以 “userDefineRecordId_起始时间_结束时间” 的格式命名录制下来的文件。 方案二：自动录制方案： 1. 需要在 控制台>应用管理>云端录制配置 中开启云端录制。 2. 设置 `录制形式` 为 `自动录制` 。 3. 设置自动录制后，在一个 TRTC 房间中的任何一个有音视频上行的用户，均会在云端录制出视频文件。 4. 文件会以 `userDefineRecordId_起始时间_结束时间` 的格式命名，如果不指定 userDefineRecordId，则文件会以 `streamId_起始时间_结束时间` 命名。
userId	【字段含义】用户标识（必填），当前用户的 userId，相当于用户名，使用 UTF-8 编码。 【推荐取值】如果一个用户在您的账号系统中的 ID 为“mike”，则 userId 即可设置为“mike”。
userSig	【字段含义】用户签名（必填），当前 userId 对应的验证签名，相当于使用云服务的登录密码。 【推荐取值】具体计算方法请参见 如何计算UserSig 。

TRTCVideoEncParam

TRTCVideoEncParam

视频编码参数

该设置决定远端用户看到的画面质量，同时也决定了云端录制出的视频文件的画面质量。

枚举类型	描述
enableAdjustRes	【字段含义】是否允许动态调整分辨率（开启后会对云端录制产生影响）。 【推荐取值】该功能适用于不需要云端录制的场景，开启后 SDK 会根据当前网络情况，智能选择出一个合适的分辨率，避免出现“大分辨率+小码率”的低效编码模式。 【特别说明】默认值：关闭。如有云端录制的需求，请不要开启此功能，因为如果视频分辨率发生变化后，云端录制出的 MP4 在普通的播放器上无法正常播放。
minVideoBitrate	【字段含义】最低视频码率，SDK 会在网络不佳的情况下主动降低视频码率以保持流畅度，最低会降至 minVideoBitrate 所设定的数值。 【特别说明】默认值：0，此时最低码率由 SDK 会根据您指定的分辨率，自动计算出合适的数值。 【推荐取值】您可以通过同时设置 videoBitrate 和 minVideoBitrate 两个参数，用于约束 SDK 对视频码率的调整范围： • 如果您追求 弱网络下允许卡顿但要保持清晰 的效果，可以设置 minVideoBitrate 为 videoBitrate 的 60%。 • 如果您追求 弱网络下允许模糊但要保持流畅 的效果，可以设置 minVideoBitrate 为一个较低的数值（比如 100kbps）。 • 如果您将 videoBitrate 和 minVideoBitrate 设置为同一个值，等价于关闭 SDK 对视频码率的自适应调节能力。
resMode	【字段含义】分辨率模式（横屏分辨率 or 竖屏分辨率）。 【推荐取值】手机平台（iOS、Android）建议选择 Portrait，桌面平台（Windows、Mac）建议选择 Landscape。 【特别说明】如需使用竖屏分辨率，请指定 resMode 为 Portrait，例如：640 × 360 + Portrait = 360 × 640。
videoBitrate	【字段含义】目标视频码率，SDK 会按照目标码率进行编码，只有在弱网络环境下才会主动降低视频码率。 【推荐取值】请参见本 TRTCVideoResolution 在各档位注释的最佳码率，也可以在此基础上适当调高。比如：TRTCVideoResolution_1280_720 对应 1200kbps 的目标码率，您也可以设置为 1500kbps 用来获得更好的观感清晰度。 【特别说明】您可以通过同时设置 videoBitrate 和 minVideoBitrate 两个参数，用于约束 SDK 对视频码率的调整范围： • 如果您追求“弱网络下允许卡顿但要保持清晰”的效果，可以设置 minVideoBitrate 为 videoBitrate 的 60%。 • 如果您追求“弱网络下允许模糊但要保持流畅”的效果，可以设置

	minVideoBitrate 为一个较低的数值（比如 100kbps）。 • 如果您将 videoBitrate 和 minVideoBitrate 设置为同一个值，等价于关闭 SDK 对视频码率的自适应调节能力。
videoFps	【字段含义】视频采集帧率。 【推荐取值】15fps或20fps。5fps以下，卡顿感明显。10fps以下，会有轻微卡顿感。20fps以上，会浪费带宽（电影的帧率为24fps）。 【特别说明】部分 Android 手机的前置摄像头并不支持15fps以上的采集帧率，部分主打美颜功能的 Android 手机的前置摄像头的采集帧率可能低于10fps。
videoResolution	【字段含义】视频分辨率。 【特别说明】如需使用竖屏分辨率，请指定 resMode 为 Portrait，例如：640 × 360 + Portrait = 360 × 640。 【推荐取值】 • 手机视频通话：建议选择 360 × 640 及以下分辨率，resMode 选择 Portrait，即竖屏分辨率。 • 手机在线直播：建议选择 540 × 960，resMode 选择 Portrait，即竖屏分辨率。 • 桌面平台（Win + Mac）：建议选择 640 × 360 及以上分辨率，resMode 选择 Landscape，即横屏分辨率。

TRTCNetworkQosParam

TRTCNetworkQosParam

网络流控（Qos）参数集

网络流控相关参数，该设置决定 SDK 在弱网络环境下的调控策略（例如：“清晰优先”或“流畅优先”）。

枚举类型	描述
controlMode	【字段含义】流控模式（已废弃）。 【推荐取值】云端控制。 【特别说明】请设置为云端控制模式（TRTCQosControlModeServer）。
preference	【字段含义】清晰优先还是流畅优先。 【推荐取值】清晰优先。 【特别说明】该参数主要影响 TRTC 在较差网络环境下的音视频表现： • 流畅优先：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的流畅性，代价就是画面会比较模糊且伴随有较多的马赛克。参见 TRTCVideoQosPreferenceSmooth

	• 清晰优先（默认值）：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的清晰度，代价就是画面会比较卡顿。参见 TRTCVideoQosPreferenceClear

TRTCRenderParams

TRTCRenderParams

视频画面的渲染参数

您可以通过设置此参数来控制画面的旋转角度、填充模式和左右镜像模式。

枚举类型	描述
fillMode	【字段含义】画面填充模式。 【推荐取值】填充（画面可能会被拉伸裁剪）或适应（画面可能会有黑边），默认值：TRTCVideoFillMode_Fill。
mirrorType	【字段含义】画面镜像模式。 【推荐取值】默认值：TRTCVideoMirrorType_Auto。
rotation	【字段含义】图像的顺时针旋转角度。 【推荐取值】支持 90、180 以及 270 旋转角度，默认值：TRTCVideoRotation0。

TRTCQualityInfo

TRTCQualityInfo

网络质量

表征网络质量的好坏，您可以通过该数值在用户界面上展示每个用户的网络质量。

枚举类型	描述
quality	网络质量
userId	用户 ID

TRTCVolumeInfo

TRTCVolumeInfo

音量大小

表征语音音量的评估值，您可以通过该数值在用户界面上展示每个用户的音量大小。

枚举类型	描述
pitch	本地用户的人声频率（单位：Hz），取值范围[0 – 4000]，对于远端用户，该值始终为0。
spectrumData	音频频谱数据是将音频数据在频率域中的分布，划分为 256 个频率段，使用 spectrumData 记录各个频率段的能量值，每个能量值的取值范围为 [−300, 0]，单位为 dBFS。 **注意** 本地频谱使用编码前的音频数据计算，会受到本地采集音量、BGM 等影响；远端频谱使用接收到的音频数据计算，本地调整远端播放音量等操作不会对其产生影响。
spectrumDataLength	spectrumDataLength 记录音频频谱数据的长度，为 256。
userId	说话者的 userId, 如果 userId 为空则代表是当前用户自己。
vad	是否检测到人声，0：非人声 1：人声。
volume	说话者的音量大小, 取值范围[0 – 100]。

TRTCSpeedTestParams

TRTCSpeedTestParams

测速参数

您可以在用户进入房间前通过 startSpeedTest 接口测试网速（注意：请不要在通话中调用）。

枚举类型	描述

expectedDownBandwidth	预期的下行带宽（kbps，取值范围： 10 ~ 5000，为 0 时不测试）。 **注意** 当参数 scene 设置为 TRTCSpeedTestScene_OnlineChorusTesting 时，为了获取更准确的 rtt / jitter 等信息，取值范围限制为 10 ~ 1000。
expectedUpBandwidth	预期的上行带宽（kbps，取值范围： 10 ~ 5000，为 0 时不测试）。 **注意** 当参数 scene 设置为 TRTCSpeedTestScene_OnlineChorusTesting 时，为了获取更准确的 rtt / jitter 等信息，取值范围限制为 10 ~ 1000。
scene	测速场景。
sdkAppId	应用标识，请参见 TRTCParams 中的相关说明。
userId	用户标识，请参见 TRTCParams 中的相关说明。
userSig	用户签名，请参见 TRTCParams 中的相关说明。

TRTCSpeedTestResult

TRTCSpeedTestResult

网络测速结果

您可以在用户进入房间前通过 startSpeedTest 接口进行测速（注意：请不要在通话中调用）。

枚举类型	描述
availableDownBandwidth	下行带宽（kbps，-1：无效值）。
availableUpBandwidth	上行带宽（kbps，-1：无效值）。
downJitter	下行数据包抖动（ms），指用户当前网络环境下数据通信的稳定性，该值越小越好，正常数值范围是0ms - 100ms，-1 代表此次测速没有成功

	测得有效值，一般情况下 WiFi 网络的 Jitter 会比 4G/5G 环境稍大。
downLostRate	下行丢包率，取值范围是 [0 - 1.0]，例如 0.2 表示每从服务器收取 10 个数据包可能会在中途丢失 2 个。
errMsg	带宽测试错误信息。
ip	服务器 IP 地址。
quality	内部通过评估算法测算出的网络质量，更多信息请参见 TRTCQuality 。
rtt	延迟（毫秒），指当前设备到 TRTC 服务器的一次网络往返时间，该值越小越好，正常数值范围是10ms - 100ms。
success	测试是否成功。
upJitter	上行数据包抖动（ms），指用户当前网络环境下数据通信的稳定性，该值越小越好，正常数值范围是0ms - 100ms，-1 代表此次测速没有成功测得有效值，一般情况下 WiFi 网络的 Jitter 会比 4G/5G 环境稍大。
upLostRate	上行丢包率，取值范围是 [0 - 1.0]，例如 0.3 表示每向服务器发送 10 个数据包可能会在中途丢失 3 个。

TRTCTexture

TRTCTexture

视频纹理数据

枚举类型	描述
glContext	【字段含义】纹理所在的 OpenGL 上下文环境。
glTextureId	【字段含义】视频纹理 ID。

TRTCVideoFrame

TRTCVideoFrame

视频帧信息

TRTCVideoFrame 用来描述一帧视频画面的裸数据，也就是编码前或者解码后的视频画面数据。

枚举类型	描述
bufferType	【字段含义】视频数据结构类型。
data	【字段含义】bufferType 为 TRTCVideoBufferType_Buffer 时的视频数据，承载用于 C++ 层的内存数据块。
height	【字段含义】视频高度。
length	【字段含义】视频数据的长度，单位是字节。对于 i420 而言：length = width * height * 3 / 2；对于 BGRA32 而言：length = width * height * 4。
rotation	【字段含义】视频像素的顺时针旋转角度。
texture	【字段含义】bufferType 为 TRTCVideoBufferType_Texture 时的视频数据，承载用于 OpenGL 渲染的纹理数据。
timestamp	【字段含义】视频帧的时间戳，单位毫秒。 【推荐取值】自定义视频采集时可以设置为0。若该参数为0，SDK 会自定填充 timestamp 字段，但请“均匀”地控制 sendCustomVideoData 的调用间隔。
videoFormat	【字段含义】视频的像素格式。
width	【字段含义】视频宽度。

TRTCAudioFrame

TRTCAudioFrame

音频帧数据

枚举类型	描述
audioFormat	【字段含义】音频帧的格式。
channel	【字段含义】声道数。
data	【字段含义】音频数据。
extraData	【字段含义】音频额外数据，远端用户通过 `onLocalProcessedAudioFrame` 写入的数据会通过该字段回调。

extraDataLength	【字段含义】音频消息数据的长度。
length	【字段含义】音频数据的长度。
sampleRate	【字段含义】采样率。
timestamp	【字段含义】时间戳，单位ms。

TRTCMixUser

TRTCMixUser

云端混流中各路画面的描述信息

TRTCMixUser 用于指定云端混流中每一路视频画面的位置、大小、图层以及流类型等信息。

枚举类型	描述
image	【字段含义】占位图或水印图 • 占位图是指当对应 userId 混流内容为纯音频时，混合后的画面中显示的是占位图片。 • 水印图是指一张贴在混合后画面中的半透明图片，这张图片会一直覆盖于混合后的画面上。 • 当指定 inputType 为 TRTCMixInputTypePureAudio 时，image 为占位图，此时需要您指定 userId。 • 当指定 inputType 为 TRTCMixInputTypeWatermark 时，image 为水印图，此时不需要您指定 userId。 【推荐取值】默认值：空值，即不设置占位图或者水印图。 【特别说明】 • 您可以将 image 设置为控制台中的某一个素材 ID，这需要您事先在 “控制台 => 应用管理 => 功能配置 => 素材管理” 中单击 [新增图片] 按钮进行上传。 • 上传成功后可以获得对应的“图片ID”，然后将“图片ID”转换成字符串类型并设置给 image 字段即可（比如假设“图片ID” 为 63，可以设置 image = @"63"）。 • 您也可以将 image 设置为图片的 URL 地址，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式，推荐使用 png 格式的半透明图片作为水印。 • image 仅在 inputType 为 TRTCMixInputTypePureAudio 或者

	TRTCMixInputTypeWatermark 时才生效。
inputType	【字段含义】指定该路流的混合内容（只混音频、只混视频、混合音视频、混入水印）。 【默认取值】默认值：TRTCMixInputTypeUndefined。 【特别说明】 • 当指定 inputType 为 TRTCMixInputTypeUndefined 并设置 pureAudio 为 YES 时，等效于设置 inputType 为 TRTCMixInputTypePureAudio。 • 当指定 inputType 为 TRTCMixInputTypeUndefined 并设置 pureAudio 为 NO 时，等效于设置 inputType 为 TRTCMixInputTypeAudioVideo。 • 当指定 inputType 为 TRTCMixInputTypeWatermark 时，您可以不指定 userId 字段，但需要指定 image 字段。
pureAudio	【字段含义】指定该路流是不是只混合声音。 【推荐取值】默认值：false。 【特别说明】已废弃，推荐使用8.5版本开始新引入的字段：inputType。
rect	【字段含义】指定该路画面的坐标区域（单位：像素）。
renderMode	【字段含义】该画面在输出时的显示模式。 【推荐取值】默认值：视频流默认为0。0为裁剪，1为缩放，2为缩放并显示黑底。 【特别说明】水印图和占位图暂时不支持设置 renderMode，默认强制拉伸处理。
roomId	【字段含义】该路音视频流所在的房间号（设置为空值代表当前用户所在的房间号）。
soundLevel	【字段含义】该路音频参与混音时的音量等级（取值范围：0 - 100）。 【默认取值】默认值：100。
streamType	【字段含义】指定该路画面是主路画面（TRTCVideoStreamTypeBig）还是辅路画面（TRTCVideoStreamTypeSub）。
userId	【字段含义】用户 ID
zOrder	【字段含义】指定该路画面的层级（取值范围：1 - 15，不可重复）。

TRTCTranscodingConfig

TRTCTranscodingConfig

云端混流的排版布局和转码参数

用于指定混流时各路画面的排版位置信息和云端转码的编码参数。

枚举类型	描述
appId	【字段含义】腾讯云直播服务的 AppID。 【推荐取值】请在 实时音视频控制台 依次单击【应用管理】=>【应用信息】，并在【旁路直播信息】中获取 appid。
audioBitrate	【字段含义】指定云端转码的目标音频码率。 【推荐取值】默认值：64kbps，取值范围是 [32，192]。
audioChannels	【字段含义】指定云端转码的音频声道数。 【推荐取值】默认值：1，代表单声道。可设定的数值只有两个数字：1-单声道，2-双声道。
audioCodec	【字段含义】指定云端转码的输出流音频编码类型。 【推荐取值】默认值：0，代表LC-AAC。可设定的数值只有三个数字：0 - LC-AAC，1 - HE-AAC，2 - HE-AACv2。 【特别说明】HE-AAC 和 HE-AACv2 支持的输出流音频采样率范围为 [48000, 44100, 32000, 24000, 16000]。 【特别说明】当音频编码设置为 HE-AACv2 时，只支持输出流音频声道数为双声道。 【特别说明】HE-AAC 和 HE-AACv2 取值仅在输出流为您额外设置的 streamId 上时才生效。
audioSampleRate	【字段含义】指定云端转码的目标音频采样率。 【推荐取值】默认值：48000Hz。支持12000HZ、16000HZ、22050HZ、24000HZ、32000HZ、44100HZ、48000HZ。
backgroundColor	【字段含义】指定混合画面的底色颜色。 【推荐取值】默认值：0x000000 代表黑色。格式为十六进制数字，比如：“0x61B9F1” 代表 RGB 分别为(97,158,241)。
backgroundImage	【字段含义】指定混合画面的背景图片。 【推荐取值】默认值：空值，即不设置背景图片。 【特别说明】 • 您可以将 image 设置为控制台中的某一个素材 ID，这需要您事先在 “控制台 => 应用管理 => 功能配置 => 素材管理” 中单击 [新增图片] 按钮进行上传。 • 上传成功后可以获得对应的“图片ID”，然后将“图片ID”转换成字符串类型并设置给 image 字段即可（比如假设“图片ID” 为 63，可以设置 image = @"63"）。 • 您也可以将 image 设置为图片的 URL 地址，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式。
bizId	【字段含义】腾讯云直播服务的 bizid。

	【推荐取值】请在 实时音视频控制台 依次单击【应用管理】=>【应用信息】，并在【旁路直播信息】中获取 bizid。
mixUsersArray	【字段含义】指定云端混流中每一路视频画面的位置、大小、图层以及流类型等信息。 【推荐取值】该字段是一个 TRTCMixUser 类型的数组，数组中的每一个元素都用来代表每一路画面的信息。
mixUsersArraySize	【字段含义】数组 mixUsersArray 的元素个数。
mode	【字段含义】排版模式。 【推荐取值】请根据您的业务场景要求自行选择，预排版模式是适用性较好的一种模式。
streamId	【字段含义】输出到 CDN 上的直播流 ID。 【推荐取值】默认值：空值，即房间里的多路音视频流最终会混合到接口调用者的那一路音视频流上。 • 如不设置该参数，SDK 会执行默认逻辑，即房间里的多路音视频流会混合到该接口调用者的那一路音视频流上，也就是 A + B => A。 • 如您设置该参数，SDK 会将房间里的多路音视频流混合到您指定的直播流上，也就是 A + B => C（C 代表您指定的 streamId）。
videoBitrate	【字段含义】指定云端转码的目标视频码率（kbps）。 【推荐取值】如果填0，TRTC 会根据 videoWidth 和 videoHeight 估算出一个合理的码率值，您也可以参考视频分辨率枚举定义中所推荐的码率值（见注释部分）。
videoFramerate	【字段含义】指定云端转码的目标视频帧率（FPS）。 【推荐取值】默认值：15fps，取值范围是 (0,30]。
videoGOP	【字段含义】指定云端转码的目标视频关键帧间隔（GOP）。 【推荐取值】默认值：2，单位为秒，取值范围是 [1,8]。
videoHeight	【字段含义】指定云端转码的目标分辨率（高度）。 【推荐取值】单位：像素值，推荐值：640，如果你只混合音频流，请将 width 和 height 均设置为 0，否则混流转码后的直播流中会有黑色背景。
videoSeiParams	【字段含义】混流 SEI 参数，默认不填写。 【特别说明】参数以 json 字符串形式传入，示例如下： `{ "payLoadContent":"xxx", "payloadType":5, "payloadUuid":"1234567890abcdef1234567890abcdef",`

```
  "interval":1000,
  "followIdr":false
}
```

当前支持的字段及含义：

- payloadContent: 必填。透传 sei 的 payload 内容，不能为空；
- payloadType: 必填。sei 消息的类型，取值范围: 5 或 [100, 254] 范围内的整数（244 除外，244 是内部自定义的时间戳 sei）；
- payloadUuid: 当 payloadType 为 5 时必须填写，其他情况下该值会被忽略。该值必须是长度为 32 的十六进制数字；
- interval: 选填，默认 1000。sei 的发送间隔，单位毫秒；
- followIdr: 选填，默认 false。该值为 true 时，发送关键帧时会确保带 sei，否则不确保。

videoWidth	【字段含义】指定云端转码的目标分辨率（宽度）。 【推荐取值】单位：像素值，推荐值：360，如果你只混合音频流，请将 width 和 height 均设置为 0，否则混流转码后的直播流中会有黑色背景。

TRTCPublishCDNParam

TRTCPublishCDNParam

向非腾讯云 CDN 上发布音视频流时需设置的转推参数

TRTC 的后台服务支持通过标准 RTMP 协议，将其中的音视频流发布到第三方直播 CDN 服务商。
如果您使用腾讯云直播 CDN 服务，可无需关注此参数，直接使用 startPublish 接口即可。

枚举类型	描述
appId	【字段含义】腾讯云直播服务的 AppID。 【推荐取值】请在 实时音视频控制台 依次单击【应用管理】=>【应用信息】，并在【旁路直播信息】中获取 appid。
bizId	【字段含义】腾讯云直播服务的 bizid。 【推荐取值】请在 实时音视频控制台 依次单击【应用管理】=>【应用信息】，并在【旁路直播信息】中获取 bizid。
streamId	【字段含义】需要转推的 streamId。 【推荐取值】默认值：空值。如果不填写，则默认转推调用者的旁路流。
url	【字段含义】指定该路音视频流在第三方直播服务商的推流地址（RTMP 格式）。

	【推荐取值】各家服务商的推流地址规则差异较大，请根据目标服务商的要求填写合法的推流 URL，TRTC 的后台服务器会按照您填写的 URL 向第三方服务商推送标准格式音视频流。 【特别说明】推流 URL 必须为 RTMP 格式，必须符合您的目标直播服务商的规范要求，否则目标服务商会拒绝来自 TRTC 后台服务的推流请求。

TRTCAudioRecordingParams

TRTCAudioRecordingParams

本地音频文件的录制参数

该参数用于在音频录制接口 startAudioRecording 中指定录制参数。

枚举类型	描述
filePath	【字段含义】录音文件的保存路径（必填）。 【特别说明】该路径需精确到文件名及格式后缀，格式后缀用于决定录音文件的格式，目前支持的格式有 PCM、WAV 和 AAC。 例如：假如您指定路径为 "mypath/record/audio.aac"，代表您希望 SDK 生成一个 AAC 格式的音频录制文件。 请您指定一个有读写权限的合法路径，否则录音文件无法生成。
maxDurationPerFile	【字段含义】maxDurationPerFile 录制文件分片时长，单位毫秒，最小值10000。默认值为0，表示不分片。
recordingContent	【字段含义】音频录制内容类型。 【特别说明】默认录制所有本地和远端音频。

TRTCLocalRecordingParams

TRTCLocalRecordingParams

本地媒体文件的录制参数

该参数用于在本地媒体文件的录制接口 startLocalRecording 中指定录制相关参数。
接口 startLocalRecording 是接口 startAudioRecording 的能力加强版本，前者可以录制视频文件，后者只能录制音频文件。

枚举类型	描述

filePath	【字段含义】录制的文件地址（必填），请确保路径有读写权限且合法，否则录制文件无法生成。 【特别说明】该路径需精确到文件名及格式后缀，格式后缀用于决定录制出的文件格式，目前支持的格式暂时只有 MP4。 例如：假如您指定路径为 "mypath/record/test.mp4"，代表您希望 SDK 生成一个 MP4 格式的本地视频文件。 请您指定一个有读写权限的合法路径，否则录制文件无法生成。
interval	【字段含义】interval 录制信息更新频率，单位毫秒，有效范围：1000-10000。默认值为-1，表示不回调。
maxDurationPerFile	【字段含义】maxDurationPerFile 录制文件分片时长，单位毫秒，最小值10000。默认值为0，表示不分片。
recordType	【字段含义】媒体录制类型，默认值：TRTCRecordTypeBoth，即同时录制音频和视频。

TRTCSwitchRoomConfig

TRTCSwitchRoomConfig

房间切换参数

该参数用于切换房间接口 switchRoom，可以让用户从一个房间快速切换到另一个房间。

枚举类型	描述
privateMapKey	【字段含义】用于权限控制的权限票据（选填），当您希望某个房间只能让特定的 userId 进入时，需要使用 privateMapKey 进行权限保护。 【推荐取值】仅建议有高级别安全需求的客户使用，更多详情请参见 进房权限保护 。
roomId	【字段含义】数字房间号码 [选填]，在同一个房间内的用户可以看到彼此并能够进行音视频通话。 【推荐取值】取值范围：1 - 4294967294。 【特别说明】roomId 和 strRoomId 必须并且只能填一个。若两者都填，则优先选择 roomId。
strRoomId	【字段含义】字符串房间号码 [选填]，在同一个房间内的用户可以看到彼此并能够进行音视频通话。 【特别说明】roomId 和 strRoomId 必须并且只能填一个。若两者都填，则优先选择 roomId。

userSig	【字段含义】用户签名 [选填]，当前 userId 对应的验证签名，相当于登录密码。 如果您在切换房间时不指定新计算出的 userSig，SDK 会继续使用您在进入房间时（enterRoom）时所指定的 userSig。 这就需要您必须保证旧的 userSig 在切换房间的那一刻仍在签名允许的效期内，否则会导致房间切换失败。 【推荐取值】具体计算方法请参见 如何计算UserSig 。

TRTCAudioFrameDelegateFormat

TRTCAudioFrameDelegateFormat

音频自定义回调的格式参数

该参数用于在音频自定义回调相关的接口中，设置 SDK 回调出来的音频数据的相关格式（包括采样率、声道数等）。

枚举类型	描述
channel	【字段含义】声道数。 【推荐取值】默认值：1，代表单声道。可设定的数值只有两个数字：1-单声道，2-双声道。
mode	【字段含义】回调数据读写模式 【推荐取值】TRTCAudioFrameOperationModeReadOnly：仅从回调中获取音频数据。可设定的模式有 TRTCAudioFrameOperationModeReadOnly，TRTCAudioFrameOperationModeReadWrite。
sampleRate	【字段含义】采样率。 【推荐取值】默认值：48000Hz。支持 16000, 32000, 44100, 48000。
samplesPerCall	【字段含义】采样点数。 【推荐取值】取值必须是 sampleRate/100 的整数倍。

TRTCImageBuffer

TRTCImageBuffer

TRTC 屏幕分享图标信息以及 mute image 垫片

枚举类型	描述
buffer	图像存储的内容，一般为 BGRA 结构。
height	图像的高度。
length	图像数据的大小。
width	图像的宽度。

TRTCScreenCaptureSourceInfo

TRTCScreenCaptureSourceInfo

屏幕分享的目标信息（仅适用于桌面系统）

在用户进行屏幕分享时，可以选择抓取整个桌面，也可以仅抓取某个程序的窗口。
TRTCScreenCaptureSourceInfo 用于描述待分享目标的信息，包括 ID、名称、缩略图等，该结构体中的字段信息均是只读的。

枚举类型	描述
height	【字段含义】屏幕/窗口高，单位:像素点。
iconBGRA	【字段含义】分享窗口的图标。
isMainScreen	【字段含义】是否为主显示屏（适用于多显示器的情况）。
isMinimizeWindow	【字段含义】是否为最小化窗口。
sourceId	【字段含义】采集源的ID，对于窗口，该字段代表窗口的 ID；对于屏幕，该字段代表显示器的 ID。
sourceName	【字段含义】采集源名称（采用 UTF8 编码）。
thumbBGRA	【字段含义】分享窗口的缩略图。
type	【字段含义】采集源类型（是分享整个屏幕？还是分享某个窗口？）。
width	【字段含义】屏幕/窗口宽，单位:像素点。
x	【字段含义】屏幕/窗口 x 坐标，单位:像素点。
y	【字段含义】屏幕/窗口 y 坐标，单位:像素点。

版权所有：腾讯云计算（北京）有限责任公司

TRTCScreenCaptureProperty

TRTCScreenCaptureProperty

屏幕分享的进阶控制参数

该参数用于屏幕分享相关的接口selectScreenCaptureTarget，用于在指定分享目标时设定一系列进阶控制参数。
比如：是否采集鼠标、是否要采集子窗口、是否要在被分享目标周围绘制一个边框等。

枚举类型	描述
enableCaptureChildWindow	【字段含义】窗口采集时是否采集子窗口（需要子窗口与被采集窗口具有 Owner 或 Popup 属性），默认为 false。
enableCaptureMouse	【字段含义】是否采集目标内容的同时采集鼠标，默认为 true。
enableHighLight	【字段含义】是否高亮正在共享的窗口（在被分享目标周围绘制一个边框），默认为 true。
enableHighPerformance	【字段含义】是否开启高性能模式（只会在分享屏幕时会生效），默认为 true。 【特殊说明】开启后屏幕采集性能最佳，但会丧失抗遮挡能力，如果您同时开启 enableHighLight + enableHighPerformance，远端用户可以看到高亮的边框。
highLightColor	【字段含义】指定高亮边框的颜色，RGB 格式，传入 0 时代表采用默认颜色，默认颜色为 #FFE640。
highLightWidth	【字段含义】指定高亮边框的宽度，传入0时采用默认描边宽度，默认宽度为 5px，您可以设置的最大值为 50。

TRTCAudioParallelParams

TRTCAudioParallelParams

远端音频流智能并发播放策略的参数

该参数用于设置远端音频流智能并发播放策略。

枚举类型	描述
includeUsersCount	【字段含义】指定用户必定能并发播放。 【特殊说明】指定必定并发播放的用户 ID 列表。这些用户不参与智能选择。includeUsers 的数量必须小于 maxCount，否则本次并发播放设置失效。includeUsers 仅在 maxCount > 0 时有效。当 includeUsers 生效时，参与智能并发选择的最大播放数 = maxCount − 有效 includeUsers 的数量。
maxCount	【字段含义】最大并发播放数。默认值：0。 • 如果 maxCount > 0，且实际人数 > maxCount，会实时智能选出 maxCount 路数据进行播放，这会极大的降低性能消耗。 • 如果 maxCount = 0，SDK 不限制并发播放数，在上麦人数比较多的房间可能会引发性能问题。

TRTCUser

TRTCUser

媒体流发布相关配置的用户信息

您可以通过设置该参数，配合媒体流目标发布参数（TRTCPublishTarget）和混流转码参数（TRTCStreamMixingConfig），将您指定的多路音视频流进行转码并发布到您填写的目标发布地址中

枚举类型	描述
intRoomId	【字段含义】数字房间号，需要和您的进房参数（TRTCParams）中的房间号类型相匹配。 【推荐取值】取值范围：1 − 4294967294。 【特别说明】intRoomId 与 strRoomId 是互斥的，若您进房参数中选用 strRoomId，则 intRoomId 需要填写为0。若两者都填，SDK 将优先选用 intRoomId。
strRoomId	【字段含义】字符串房间号，需要和您的进房参数（TRTCParams）中的房间号类型相匹配。 【特别说明】intRoomId 与 strRoomId 是互斥的，若您进房参数中选用 roomId，则 strRoomId 无需填写。若两者都填，SDK 将优先选用 intRoomId。 【推荐取值】限制长度为 64 字节。以下为支持的字符集范围（共 89 个字符）： • 大小写英文字母（a-zA-Z）； • 数字（0-9）； • 空格、 ! 、 # 、 $ 、 % 、 & 、 (、) 、 + 、 - 、 : 、 ; 、 < 、 = 、 . 、 > 、 ? 、 @ 、 [、] 、 ^ 、 _ 、 { 、

	} 、 | 、 ~ 、 , 。
userId	【字段含义】用户标识，当前用户的 userId，相当于用户名，使用 UTF-8 编码。 【推荐取值】如果一个用户在您的账号系统中的 ID 为“mike”，则 userId 即可设置为“mike”。

TRTCPublishCdnUrl

TRTCPublishCdnUrl

向腾讯或者第三方 CDN 上发布音视频流时需设置的 url 配置

该配置用于媒体流发布接口 startPublishMediaStream 中的目标推流配置（TRTCPublishTarget）

枚举类型	描述
isInternalLine	【字段含义】指定该路音视频流是否发布至腾讯云。 【推荐取值】默认值：true。 【特别说明】若您的目标直播服务商为腾讯，请将此参数设置为 true，此时后台计费系统不会对此计算转推服务费。
rtmpUrl	【字段含义】指定该路音视频流在腾讯或者第三方直播服务商的推流地址（RTMP 格式）。 【推荐取值】各家服务商的推流地址规则差异较大，请根据目标服务商的要求填写合法的推流 URL，TRTC 的后台服务器会按照您填写的 URL 向第三方服务商推送标准格式音视频流。 【特别说明】推流 URL 必须为 RTMP 格式，必须符合您的目标直播服务商的规范要求，否则目标服务商会拒绝来自 TRTC 后台服务的推流请求。

TRTCPublishTarget

TRTCPublishTarget

目标推流配置

该配置用于媒体流发布接口 startPublishMediaStream 。

枚举类型	描述
cdnUrlList	【字段含义】发布至腾讯或者第三方直播服务商的推流地址（RTMP 格

	式）。 【特别说明】若您的 mode 选择为 TRTCPublishMixStreamToRoom，此时您不需要设置该参数。
cdnUrlListSize	【字段含义】数组 cdnUrlList 的元素个数。 【特别说明】若您的 mode 选择为 TRTCPublishMixStreamToRoom，此时您不需要设置该参数。
mixStreamIdentity	【字段含义】回推房间机器人信息。 【特别说明】仅当您的 mode 选择为 TRTCPublishMixStreamToRoom 时，您需要设置该参数。 【特别说明】设置后，该路转码音视频数据将回推到您指定的房间中。建议设置为特殊的 userId，以避免难以区分回推机器人和您通过 TRTC SDK 进房的主播。 【特别说明】参与混流的用户不支持订阅该转码流。 【特别说明】当您进房前设置的订阅模式（setDefaultStreamRecvMode）均为手动时，您需要自行管理您想要拉取的音视频流（通常当您拉取回推房间的转码流时，您应该取消订阅参与转码的对应音视频单流）。 【特别说明】当您进房前设置的订阅模式（setDefaultStreamRecvMode）均为自动时，不参与转码的用户将自动收到后台下发的转码流并不再继续接收参与转码的音视频单流。除非您明确进行取消订阅（muteRemoteVideoStream 和 muteRemoteAudio），否则转码流数据将持续下发。
mode	【字段含义】媒体流发布模式。 【推荐取值】请根据您的业务场景要求自行选择，TRTC 支持转推、转码和回推到 RTC 房间的模式。 【特别说明】若您的业务场景需要多个发布模式，您可以通过多次调用媒体流发布接口（startPublishMediaStream）并分别设置不同的 TRTCPublishTarget。 【特别说明】同一个 mode 请对应一个媒体流发布接口（startPublishMediaStream），并在后续需要调整时，使用 updatePublishCDNStream 进行更新。

TRTCVideoLayout

TRTCVideoLayout

转码视频布局

该配置用于媒体流发布接口（startPublishMediaStream）中的转码配置（TRTCStreamMixingConfig）。
用于指定转码流中每一路视频画面的位置、大小、图层以及流类型等信息。

枚举类型	描述
backgroundColor	【字段含义】指定混合画面的底色颜色。 【推荐取值】默认值：0x000000 代表黑色。格式为十六进制数字，比如：“0x61B9F1” 代表 RGB 分别为 (97,158,241)。
fillMode	【字段含义】画面填充模式。 【推荐取值】填充（画面可能会被拉伸裁剪）或适应（画面可能会有黑边），默认值：TRTCVideoFillMode_Fill。
fixedVideoStreamType	【字段含义】指定该路画面是主路画面（TRTCVideoStreamTypeBig）还是辅路画面（TRTCVideoStreamTypeSub）。
fixedVideoUser	【字段含义】参与转码的用户信息。 【特别说明】用户信息（TRTCUser）支持不填写（即 userId、intRoomId 和 strRoomId 均不填写）。此时当发起混流的房间中有主播上行音视频数据时，TRTC 后台服务器将自动将对应主播音视频填充到您指定的布局中。
placeHolderImage	【字段含义】占位图 URL，即当指定用户暂时仅上行音频时，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 【推荐取值】默认值：空值，即不设置占位图。 【特别说明】此时需要您指定用户信息 fixedVideoUser 中的 userId。 【特别说明】 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式，推荐使用 png 格式的半透明图片作为占位图。
rect	【字段含义】指定该路画面的坐标区域（单位：像素）。
zOrder	【字段含义】指定该路画面的层级（取值范围：0 – 15，不可重复）。

TRTCWatermark

TRTCWatermark

水印布局

该配置用于媒体流发布接口（startPublishMediaStream）中的转码配置（TRTCStreamMixingConfig）

枚举类型	描述

rect	【字段含义】指定该路水印画面的坐标区域（单位：像素）。
watermarkUrl	【字段含义】水印 URL，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 【特别说明】 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式，推荐使用 png 格式的半透明图片作为水印。
zOrder	【字段含义】指定该路水印画面的层级（取值范围：0 - 15，不可重复）。

TRTCStreamEncoderParam

TRTCStreamEncoderParam

媒体流编码输出参数

【字段含义】该配置用于媒体流发布接口（startPublishMediaStream）。
【特别说明】当您的发布目标（TRTCPublishTarget）中的 mode 配置为 TRTCPublish_MixStream_ToCdn 或者 TRTCPublish_MixStream_ToRoom 时，该参数为必填。
【特别说明】当您使用转推服务（mode 为 TRTCPublish_BigStream_ToCdn 或者 TRTCPublish_SubStream_ToCdn）时，为了更好的转推稳定性以及更好的 CDN 播放兼容性，也建议您填写该配置的具体参数。

枚举类型	描述
audioEncodedChannelNum	【字段含义】指定媒体发布流的目标音频声道数。 【推荐取值】默认值：1，代表单声道。可设定的数值只有两个数字：1-单声道，2-双声道。
audioEncodedCodecType	【字段含义】指定媒体发布流的目标音频编码类型。 【推荐取值】默认值：0，代表 LC-AAC。可设定的数值只有三个数字：0 - LC-AAC，1 - HE-AAC，2 - HE-AACv2。 【特别说明】HE-AAC 和 HE-AACv2 支持的输出流音频采样率范围为 [48000, 44100, 32000, 24000, 16000]。 【特别说明】当音频编码设置为 HE-AACv2 时，只支持输出流音频声道数为双声道。
audioEncodedKbps	【字段含义】指定媒体发布流的目标音频码率（kbps）。 【推荐取值】默认值：50kbps，取值范围是 [32，192]。

audioEncodedSampleRate	【字段含义】指定媒体发布流的目标音频采样率。 【推荐取值】默认值：48000Hz。取值为 [48000, 44100, 32000, 24000, 16000, 8000]，单位是 Hz。
videoEncodedCodecType	【字段含义】指定媒体发布流的目标视频编码类型。 【推荐取值】默认值：0，代表 H264。可设定的数值只有两个数字：0 - H264，1 - H265。
videoEncodedFPS	【字段含义】指定媒体发布流的目标视频帧率（FPS）。 【推荐取值】推荐值：20fps，取值范围是 (0,30]。
videoEncodedGOP	【字段含义】指定媒体发布流的目标视频关键帧间隔（GOP）。 【推荐取值】推荐值：3，单位为秒，取值范围是 [1,5]。
videoEncodedHeight	【字段含义】指定媒体发布流的目标分辨率（高度）。 【推荐取值】单位：像素值，推荐值：640，如果你只混合音频流，请将 width 和 height 均设置为 0，否则混流转码后的直播流中会有黑色背景。
videoEncodedKbps	【字段含义】指定媒体发布流的目标视频码率（kbps）。 【推荐取值】如果填 0，TRTC 会根据 videoEncodedWidth 和 videoEncodedHeight 估算出一个合理的码率值，您也可以参考视频分辨率枚举定义中所推荐的码率值（见注释部分）。
videoEncodedWidth	【字段含义】指定媒体发布流的目标分辨率（宽度）。 【推荐取值】单位：像素值，推荐值：368，如果你只混合音频流，请将 width 和 height 均设置为 0，否则混流转码后的直播流中会有黑色背景。

videoSeiParams	【字段含义】混流 SEI 参数，默认不填写。
	【特别说明】参数以 json 字符串形式传入，示例如下：
	<pre>{
	"payLoadContent":"xxx",
	"payloadType":5,
	
	"payloadUuid":"1234567890abcdef1234567890abcdef",
	"interval":1000,
	"followIdr":false
	}</pre>
	当前支持的字段及含义：
	• payloadContent: 必填。透传 sei 的 payload 内容，不能为空；
	• payloadType: 必填。sei 消息的类型，取值范围: 5 或 [100, 254] 范围内的整数（244 除外，244 是内部自定义的时间戳 sei）；
	• payloadUuid: 当 payloadType 为 5 时必须填写，其他情况下该值会被忽略。该值必须是长度为 32 的十六进制数字；

	• interval: 选填，默认 1000。sei 的发送间隔，单位毫秒； • followIdr: 选填，默认 false。该值为 true 时，发送关键帧时会确保带 sei，否则不确保。

TRTCStreamMixingConfig

TRTCStreamMixingConfig

媒体流转码配置参数

该配置用于媒体流发布接口（ startPublishMediaStream ）。
用于指定转码时各路画面的排版位置信息和输入的音频信息。

枚举类型	描述
audioMixUserList	【字段含义】指定转码流中的每一路输入音频的信息。 【推荐取值】该字段是一个 TRTCUser 类型的数组，数组中的每一个元素都用来代表每一路输入音频的信息。 【特别说明】用户信息支持不填写（即 audioMixUserList 为空）。此时若设置了 TRTCStreamEncoderParam 中音频相关编码输出参数，TRTC 后台服务器将自动将所有主播的音频混合输出（当前仅支持最高 16 路音视频输入）。
audioMixUserListSize	【字段含义】数组 audioMixUserList 的元素个数。
backgroundColor	【字段含义】指定混合画面的底色颜色。 【推荐取值】默认值：0x000000 代表黑色。格式为十六进制数字，比如：“0x61B9F1” 代表 RGB 分别为（97、158、241）。
backgroundImage	【字段含义】指定混合画面的背景图 URL，腾讯云的后台服务器会将该 URL 地址指定的图片混合到最终的画面中。 【推荐取值】默认值：空值，即不设置背景图片。 【特别说明】 • URL 链接长度限制为 512 字节。图片大小限制不超过 2MB。 • 图片格式支持 png、jpg、jpeg、bmp 格式，推荐使用 png 格式的半透明图片作为背景图。
videoLayoutList	【字段含义】指定混合画面中的每一路视频画面的位置、大小、图层以及流类型等信息。 【推荐取值】该字段是一个 TRTCVideoLayout 类型的数组，数组中的每一个元素都用来代表每一路画面的信息。

videoLayoutListSize	【字段含义】数组 videoLayoutList 的元素个数。
watermarkList	【字段含义】指定混合画面中的每一路水印画面的位置、大小、图层等信息。 【推荐取值】该字段是一个 TRTCWatermark 类型的数组，数组中的每一个元素都用来代表每一路水印的信息。
watermarkListSize	【字段含义】数组 watermarkList 的元素个数。

TRTCPayloadPrivateEncryptionConfig

TRTCPayloadPrivateEncryptionConfig

媒体流私有加密配置

该配置用于设置媒体流私有加密的算法和密钥。

枚举类型	描述
encryptionAlgorithm	【字段含义】加密算法，默认为 TRTCEncryptionAlgorithmAes128Gcm。
encryptionKey	【字段含义】加密用密钥，字符串类型。 【推荐取值】若加密算法为 TRTCEncryptionAlgorithmAes128Gcm，密匙长度需为 16 字节，若加密算法为 TRTCEncryptionAlgorithmAes256Gcm，密匙长度需为 32 字节。
encryptionSalt[32]	【字段含义】盐，加密用初始向量。 【推荐取值】需确保填入该参数的数组不为空、不全为 0 且数据长度为 32 字节。

TRTCAudioVolumeEvaluateParams

TRTCAudioVolumeEvaluateParams

音量评估等相关参数设置

该设置用于开启人声检测、声音频谱计算。

枚举类型	描述
enablePitchCalculation	【字段含义】是否开启本地人声频率计算
enableSpectrumCalculation	【字段含义】是否开启声音频谱计算。
enableVadDetection	【字段含义】是否开启本地人声检测。 【请您注意】在 startLocalAudio 之前调用才可以生效。
interval	【字段含义】设置 onUserVoiceVolume 回调的触发间隔，单位为毫秒，最小间隔为 100ms，如果小于等于 0 则会关闭回调。 【推荐取值】推荐值：300，单位为毫秒。 【特别说明】interval 大于 0 时，音量提示将默认开启，无需另外设置。

废弃接口

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2022 Tencent. All rights reserved.

Module: DeprecatedTRTCCloud @ TXLiteAVSDK

Function: TRTC 废弃接口

废弃接口

IDeprecatedTRTCCloud

函数列表	描述
enableAudioVolumeEvaluation	启用音量大小提示
enableAudioVolumeEvaluation	启用音量大小提示
startLocalAudio	设置音频质量
startRemoteView	开始显示远端视频画面
stopRemoteView	停止显示远端视频画面，同时不再拉取该远端用户的视频数据流
setLocalViewFillMode	设置本地图像的渲染模式
setLocalViewRotation	设置本地图像的顺时针旋转角度
setLocalViewMirror	设置本地摄像头预览画面的镜像模式
setRemoteViewFillMode	设置远端图像的渲染模式
setRemoteViewRotation	设置远端图像的顺时针旋转角度
startRemoteSubStreamView	开始显示远端用户的辅路画面
stopRemoteSubStreamView	停止显示远端用户的辅路画面

setRemoteSubStreamViewFillMode	设置辅路画面的填充模式
setRemoteSubStreamViewRotation	设置辅路画面的顺时针旋转角度
setAudioQuality	设置音频质量
setPriorRemoteVideoStreamType	设定优先观看大画面还是小画面
setMicVolumeOnMixing	设置麦克风音量大小
playBGM	启动播放背景音乐
stopBGM	停止播放背景音乐
pauseBGM	停止播放背景音乐
resumeBGM	停止播放背景音乐
getBGMDuration	获取背景音乐总时长（单位：毫秒）
setBGMPosition	设置背景音乐的播放进度
setBGMVolume	设置背景音乐的音量大小
setBGMPlayoutVolume	设置背景音乐的本地播放音量
setBGMPublishVolume	设置背景音乐的远端播放音量
playAudioEffect	播放音效
setAudioEffectVolume	设置音效音量
stopAudioEffect	停止播放音效
stopAllAudioEffects	停止所有音效
setAllAudioEffectsVolume	设置所有音效音量
pauseAudioEffect	暂停音效
resumeAudioEffect	暂停音效
enableCustomVideoCapture	启用视频自定义采集模式
sendCustomVideoData	投送自己采集的视频数据

版权所有：腾讯云计算（北京）有限责任公司

muteLocalVideo	暂停/恢复发布本地的视频流
muteRemoteVideoStream	暂停 / 恢复订阅远端用户的视频流
startSpeedTest	开始进行网络测速（进入房间前使用）
startScreenCapture	启动屏幕分享
setLocalVideoProcessCallback	设置第三方美颜的视频数据回调
getCameraDevicesList	获取摄像头设备列表
setCurrentCameraDevice	选定当前要使用的摄像头
getCurrentCameraDevice	获取当前使用的摄像头
getMicDevicesList	获取麦克风设备列表
getCurrentMicDevice	获取当前的麦克风设备
setCurrentMicDevice	选定当前使用的麦克风
getCurrentMicDeviceVolume	获取当前麦克风的设备音量
setCurrentMicDeviceVolume	设置当前麦克风的设备音量
setCurrentMicDeviceMute	设置系统当前麦克风设备的静音状态
getCurrentMicDeviceMute	获取系统当前麦克风设备是否静音
getSpeakerDevicesList	获取扬声器设备列表
getCurrentSpeakerDevice	获取当前的扬声器设备
setCurrentSpeakerDevice	设置要使用的扬声器
getCurrentSpeakerVolume	获取当前扬声器的设备音量

版权所有：腾讯云计算（北京）有限责任公司

setCurrentSpeakerVolume	设置当前扬声器的设备音量
getCurrentSpeakerDeviceMute	获取系统当前扬声器设备是否静音
setCurrentSpeakerDeviceMute	设置系统当前扬声器设备的静音状态
startCameraDeviceTest	开始进行摄像头测试
startCameraDeviceTest	开始进行摄像头测试
stopCameraDeviceTest	停止进行摄像头测试
startMicDeviceTest	开始进行麦克风测试
stopMicDeviceTest	开始进行麦克风测试
startSpeakerDeviceTest	开始进行扬声器测试
stopSpeakerDeviceTest	停止进行扬声器测试
selectScreenCaptureTarget	开始应用内的屏幕分享（iOS）
setVideoEncoderRotation	设置视频编码器输出的画面方向
setVideoEncoderMirror	设置编码器输出的画面镜像模式

enableAudioVolumeEvaluation

enableAudioVolumeEvaluation

void enableAudioVolumeEvaluation	(uint32_t interval)

启用音量大小提示

@deprecated v10.1 版本开始不推荐使用，建议使用 enableAudioVolumeEvaluation(enable, params) 替代之。

enableAudioVolumeEvaluation

enableAudioVolumeEvaluation

void enableAudioVolumeEvaluation	(uint32_t interval bool enable_vad)

启用音量大小提示

@deprecated v11.2 版本开始不推荐使用，建议使用 enableAudioVolumeEvaluation(enable, params) 替代之。

startLocalAudio

startLocalAudio

设置音频质量

@deprecated v8.0 版本开始不推荐使用，建议使用 startLocalAudio 替代之。

startRemoteView

startRemoteView

void startRemoteView	(const char* userId TXView rendView)

开始显示远端视频画面

@deprecated v8.0 版本开始不推荐使用，建议使用 startRemoteView 替代之。

stopRemoteView

stopRemoteView

void stopRemoteView	(const char* userId)

停止显示远端视频画面，同时不再拉取该远端用户的视频数据流

@deprecated v8.0 版本开始不推荐使用，建议使用 stopRemoteView 替代之。

setLocalViewFillMode

setLocalViewFillMode

void setLocalViewFillMode (TRTCVideoFillMode mode)

设置本地图像的渲染模式

@deprecated v8.0 版本开始不推荐使用，建议使用 setLocalRenderParams 替代之。

setLocalViewRotation

setLocalViewRotation

void setLocalViewRotation (TRTCVideoRotation rotation)

设置本地图像的顺时针旋转角度

@deprecated v8.0 版本开始不推荐使用，建议使用 setLocalRenderParams 替代之。

setLocalViewMirror

setLocalViewMirror

void setLocalViewMirror (bool mirror)

设置本地摄像头预览画面的镜像模式

@deprecated v8.0 版本开始不推荐使用，建议使用 setLocalRenderParams 替代之。

setRemoteViewFillMode

setRemoteViewFillMode

```
void
setRemoteViewFillMode
(const char* userId
TRTCVideoFillMode mode)
```

设置远端图像的渲染模式

@deprecated v8.0 版本开始不推荐使用，建议使用 setRemoteRenderParams 替代之。

setRemoteViewRotation

setRemoteViewRotation

```
void
setRemoteViewRotation
(const char* userId
TRTCVideoRotation rotation)
```

设置远端图像的顺时针旋转角度

@deprecated v8.0 版本开始不推荐使用，建议使用 setRemoteRenderParams 替代之。

startRemoteSubStreamView

startRemoteSubStreamView

```
void
startRemoteSubStreamView
(const char* userId
TXView rendView)
```

开始显示远端用户的辅路画面

@deprecated v8.0 版本开始不推荐使用，建议使用 startRemoteView 替代之。

stopRemoteSubStreamView

stopRemoteSubStreamView

```
void
stopRemoteSubStreamView (const char* userId)
```

停止显示远端用户的辅路画面

@deprecated v8.0 版本开始不推荐使用，建议使用 stopRemoteView 替代之。

setRemoteSubStreamViewFillMode

setRemoteSubStreamViewFillMode

```
void
setRemoteSubStreamViewFillMode (const char* userId
                                TRTCVideoFillMode mode)
```

设置辅路画面的填充模式

@deprecated v8.0 版本开始不推荐使用，建议使用 setRemoteRenderParams 替代之。

setRemoteSubStreamViewRotation

setRemoteSubStreamViewRotation

```
void
setRemoteSubStreamViewRotation (const char* userId
                                TRTCVideoRotation rotation)
```

设置辅路画面的顺时针旋转角度

@deprecated v8.0 版本开始不推荐使用，建议使用 setRemoteRenderParams:streamType:params: 替代之。

setAudioQuality

setAudioQuality

void setAudioQuality （TRTCAudioQuality quality）

设置音频质量

@deprecated v8.0 版本开始不推荐使用，建议使用 startLocalAudio 替代之。

setPriorRemoteVideoStreamType

setPriorRemoteVideoStreamType

void setPriorRemoteVideoStreamType （TRTCVideoStreamType type）

设定优先观看大画面还是小画面

@deprecated v8.0 版本开始不推荐使用，建议使用 startRemoteView:streamType:view: 替代之。

setMicVolumeOnMixing

setMicVolumeOnMixing

void setMicVolumeOnMixing (uint32_t volume)

设置麦克风音量大小

playBGM

playBGM

void playBGM (const char* path)

启动播放背景音乐

@deprecated v7.3 版本开始不推荐使用，建议使用 getAudioEffectManager 替代之。

stopBGM

stopBGM

停止播放背景音乐

@deprecated v7.3 版本开始不推荐使用，建议使用 getAudioEffectManager 替代之。

pauseBGM

pauseBGM

停止播放背景音乐

@deprecated v7.3 版本开始不推荐使用，建议使用 getAudioEffectManager 替代之。

resumeBGM

resumeBGM

停止播放背景音乐

@deprecated v7.3 版本开始不推荐使用，建议使用 getAudioEffectManager 替代之。

getBGMDuration

getBGMDuration

uint32_t getBGMDuration (const char* path)

获取背景音乐总时长（单位：毫秒）

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 getMusicDurationInMS 替代之。

setBGMPosition

setBGMPosition

void
setBGMPosition (uint32_t pos)

设置背景音乐的播放进度

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 seekMusicToPosInMS 替代之。

setBGMVolume

setBGMVolume

void
setBGMVolume (uint32_t volume)

设置背景音乐的音量大小

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setMusicVolume 替代之。

setBGMPlayoutVolume

setBGMPlayoutVolume

void
setBGMPlayoutVolume (uint32_t volume)

设置背景音乐的本地播放音量

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setMusicPlayoutVolume 替代之。

setBGMPublishVolume

setBGMPublishVolume

void setBGMPublishVolume (uint32_t volume)

设置背景音乐的远端播放音量

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setBGMPublishVolume 替代之。

playAudioEffect

playAudioEffect

void playAudioEffect (TRTCAudioEffectParam * effect)

播放音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 startPlayMusic 替代之。

setAudioEffectVolume

setAudioEffectVolume

void setAudioEffectVolume (int effectId

int volume)

设置音效音量

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setMusicPublishVolume 和 setMusicPlayoutVolume 替代之。

stopAudioEffect

stopAudioEffect

void stopAudioEffect (int effectId)

停止播放音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 stopPlayMusic 替代之。

stopAllAudioEffects

stopAllAudioEffects

停止所有音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 stopPlayMusic 替代之。

setAllAudioEffectsVolume

setAllAudioEffectsVolume

void setAllAudioEffectsVolume (int volume)

设置所有音效音量

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 setMusicPublishVolume 和 setMusicPlayoutVolume 替代之。

pauseAudioEffect

pauseAudioEffect

void pauseAudioEffect (int effectId)

暂停音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 pauseAudioEffect 替代之。

resumeAudioEffect

resumeAudioEffect

void resumeAudioEffect (int effectId)

暂停音效

@deprecated v7.3 版本开始不推荐使用，建议使用 TXAudioEffectManager 中的 resumePlayMusic 替代之。

enableCustomVideoCapture

enableCustomVideoCapture

void enableCustomVideoCapture (bool enable)

启用视频自定义采集模式

@deprecated v8.5 版本开始不推荐使用，建议使用 enableCustomVideoCapture(streamType, enable) 接口替代之。

sendCustomVideoData

sendCustomVideoData

void sendCustomVideoData (TRTCVideoFrame* frame)

投送自己采集的视频数据

@deprecated v8.5 版本开始不推荐使用，建议使用 sendCustomVideoData(streamType, TRTCVideoFrame) 接口替代之。

muteLocalVideo

muteLocalVideo

void muteLocalVideo	(bool mute)

暂停/恢复发布本地的视频流

@deprecated v8.9 版本开始不推荐使用，建议使用 muteLocalVideo(streamType, mute) 接口替代之。

muteRemoteVideoStream

muteRemoteVideoStream

void muteRemoteVideoStream	(const char* userId bool mute)

暂停 / 恢复订阅远端用户的视频流

@deprecated v8.9 版本开始不推荐使用，建议使用 muteRemoteVideoStream(userId, streamType, mute) 接口替代之。

startSpeedTest

startSpeedTest

void startSpeedTest	(uint32_t sdkAppId const char* userId const char* userSig)

开始进行网络测速（进入房间前使用）

@deprecated v9.2 版本开始不推荐使用，建议使用 startSpeedTest(params) 接口替代之。

startScreenCapture

startScreenCapture

void startScreenCapture (TXView rendView)

启动屏幕分享

@deprecated v7.2 版本开始不推荐使用，建议使用 startScreenCapture 替代之。

setLocalVideoProcessCallback

setLocalVideoProcessCallback

int setLocalVideoProcessCallback (TRTCVideoPixelFormat pixelFormat
TRTCVideoBufferType bufferType
ITRTCVideoFrameCallback* callback)

设置第三方美颜的视频数据回调

@deprecated v11.4 版本开始不推荐使用，建议使用 enableLocalVideoCustomProcess 和 setLocalVideoCustomProcessCallback 替代之。

getCameraDevicesList

getCameraDevicesList

获取摄像头设备列表

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getDevicesList 接口替代之。

setCurrentCameraDevice

setCurrentCameraDevice

void setCurrentCameraDevice (const char* deviceId)

选定当前要使用的摄像头

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDevice 接口替代之。

getCurrentCameraDevice

getCurrentCameraDevice

获取当前使用的摄像头

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDevice 接口替代之。

getMicDevicesList

getMicDevicesList

获取麦克风设备列表

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getDevicesList 接口替代之。

getCurrentMicDevice

getCurrentMicDevice

获取当前的麦克风设备

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDevice 接口替代之。

setCurrentMicDevice

setCurrentMicDevice

```
void setCurrentMicDevice       (const char* micId)
```

选定当前使用的麦克风

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDevice 接口替代之。

getCurrentMicDeviceVolume

getCurrentMicDeviceVolume

获取当前麦克风的设备音量

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDeviceVolume 接口替代之。

setCurrentMicDeviceVolume

setCurrentMicDeviceVolume

void setCurrentMicDeviceVolume	(uint32_t volume)

设置当前麦克风的设备音量

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDeviceVolume 接口替代之。

setCurrentMicDeviceMute

setCurrentMicDeviceMute

void setCurrentMicDeviceMute	(bool mute)

设置系统当前麦克风设备的静音状态

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDeviceMute 接口替代之。

版权所有：腾讯云计算（北京）有限责任公司

getCurrentMicDeviceMute

getCurrentMicDeviceMute

获取系统当前麦克风设备是否静音

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDeviceMute 接口替代之。

getSpeakerDevicesList

getSpeakerDevicesList

获取扬声器设备列表

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getDevicesList 接口替代之。

getCurrentSpeakerDevice

getCurrentSpeakerDevice

获取当前的扬声器设备

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDevice 接口替代之。

setCurrentSpeakerDevice

setCurrentSpeakerDevice

void setCurrentSpeakerDevice	(const char* speakerId)

设置要使用的扬声器

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDevice 接口替代之。

getCurrentSpeakerVolume

getCurrentSpeakerVolume

获取当前扬声器的设备音量

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDeviceVolume 接口替代之。

setCurrentSpeakerVolume

setCurrentSpeakerVolume

void setCurrentSpeakerVolume	(uint32_t volume)

设置当前扬声器的设备音量

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDeviceVolume 接口替代之。

getCurrentSpeakerDeviceMute

getCurrentSpeakerDeviceMute

获取系统当前扬声器设备是否静音

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 getCurrentDeviceMute 接口替代之。

setCurrentSpeakerDeviceMute

setCurrentSpeakerDeviceMute

void setCurrentSpeakerDeviceMute	(bool mute)

设置系统当前扬声器设备的静音状态

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 setCurrentDeviceMute 接口替代之。

startCameraDeviceTest

startCameraDeviceTest

void startCameraDeviceTest	(TXView renderView)

开始进行摄像头测试

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 startCameraDeviceTest 接口替代之。

startCameraDeviceTest

startCameraDeviceTest

void startCameraDeviceTest	(ITRTCVideoRenderCallback* callback)

开始进行摄像头测试

stopCameraDeviceTest

stopCameraDeviceTest

停止进行摄像头测试

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 stopCameraDeviceTest 接口替代之。

startMicDeviceTest

startMicDeviceTest

void startMicDeviceTest (uint32_t interval)

开始进行麦克风测试

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 startMicDeviceTest 接口替代之。

stopMicDeviceTest

stopMicDeviceTest

开始进行麦克风测试

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 stopMicDeviceTest 接口替代之。

startSpeakerDeviceTest

startSpeakerDeviceTest

void startSpeakerDeviceTest (const char* testAudioFilePath)

开始进行扬声器测试

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 startSpeakerDeviceTest 接口替代之。

stopSpeakerDeviceTest

stopSpeakerDeviceTest

停止进行扬声器测试

@deprecated v8.0 版本开始不推荐使用，建议使用 TXDeviceManager 中的 stopSpeakerDeviceTest 接口替代之。

selectScreenCaptureTarget

selectScreenCaptureTarget

```
void selectScreenCaptureTarget (const TRTCScreenCaptureSourceInfo & source
                                const RECT& captureRect
                                bool captureMouse = true
                                bool highlightWindow = true)
```

开始应用内的屏幕分享（iOS）

@deprecated v8.6 版本开始不推荐使用，建议使用 startScreenCaptureInApp 接口替代之。

setVideoEncoderRotation

setVideoEncoderRotation

```
void setVideoEncoderRotation (TRTCVideoRotation rotation)
```

设置视频编码器输出的画面方向

@deprecated v11.7 版本开始不推荐使用。

setVideoEncoderMirror

setVideoEncoderMirror

```
void setVideoEncoderMirror (bool mirror)
```

设置编码器输出的画面镜像模式

@deprecated v11.7 版本开始不推荐使用。

版权所有：腾讯云计算（北京）有限责任公司

错误码表

最近更新时间：2024-03-06 14:51:32

Copyright (c) 2021 Tencent. All rights reserved.

Module: TRTC 错误码表

Function: 用于通知客户 TRTC 在使用过程中出现的警告和错误

错误码表

枚举类型

枚举类型	描述
TXLiteAVError	错误码
TXLiteAVWarning	警告码

TXLiteAVError

TXLiteAVError

错误码

枚举	取值	描述
ERR_NULL	0	无错误
ERR_FAILED	-1	暂未归类的通用错误
ERR_INVALID_PARAMETER	-2	调用 API 时，传入的参数不合法
ERR_REFUSED	-3	API 调用被拒绝
ERR_NOT_SUPPORTED	-4	当前 API 不支持调用

ERR_INVALID_LICENSE	-5	license 不合法，调用失败
ERR_REQUEST_SERVER_TIMEOUT	-6	请求服务器超时
ERR_SERVER_PROCESS_FAILED	-7	服务器无法处理您的请求
ERR_DISCONNECTED	-8	断开连接
ERR_CAMERA_START_FAIL	-1301	打开摄像头失败，例如在 Windows 或 Mac 设备，摄像头的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_CAMERA_NOT_AUTHORIZED	-1314	摄像头设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_CAMERA_SET_PARAM_FAIL	-1315	摄像头参数设置出错（参数不支持或其它）
ERR_CAMERA_OCCUPY	-1316	摄像头正在被占用中，可尝试打开其他摄像头
ERR_SCREEN_CAPTURE_START_FAIL	-1308	开始录屏失败，如果在移动设备出现，可能是权限被用户拒绝了，如果在 Windows 或 Mac 系统的设备出现，请检查录屏接口的参数是否符合要求
ERR_SCREEN_CAPTURE_UNSURPORT	-1309	录屏失败，在 Android 平台，需要5.0以上的系统，在 iOS 平台，需要11.0以上的系统
ERR_SCREEN_CAPTURE_STOPPED	-7001	录屏被系统中止
ERR_SCREEN_SHARE_NOT_AUTHORIZED	-102015	没有权限上行辅路
ERR_SCREEN_SHRAE_OCCUPIED_BY_OTHER	-102016	其他用户正在上行辅路
ERR_VIDEO_ENCODE_FAIL	-1303	视频帧编码失败，例如 iOS 设备切换到其他应用时，硬编码器可能被系统释放，再切换回来时，硬编码器重启前，可能会抛出

ERR_UNSUPPORTED_RESOLUTION	-1305	不支持的视频分辨率
ERR_PIXEL_FORMAT_UNSUPPORTED	-1327	自定视频采集：设置的 pixel format 不支持
ERR_BUFFER_TYPE_UNSUPPORTED	-1328	自定视频采集：设置的 buffer type 不支持
ERR_NO_AVAILABLE_HEVC_DECODERS	-2304	找不到可用的 HEVC 解码器
ERR_MIC_START_FAIL	-1302	打开麦克风失败，例如在 Windows 或 Mac 设备，麦克风的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_MIC_NOT_AUTHORIZED	-1317	麦克风设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_MIC_SET_PARAM_FAIL	-1318	麦克风设置参数失败
ERR_MIC_OCCUPY	-1319	麦克风正在被占用中，例如移动设备正在通话时，打开麦克风会失败
ERR_MIC_STOP_FAIL	-1320	停止麦克风失败
ERR_SPEAKER_START_FAIL	-1321	打开扬声器失败，例如在 Windows 或 Mac 设备，扬声器的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_SPEAKER_SET_PARAM_FAIL	-1322	扬声器设置参数失败
ERR_SPEAKER_STOP_FAIL	-1323	停止扬声器失败
ERR_AUDIO_PLUGIN_START_FAIL	-1330	开启系统声音录制失败，例如音频驱动插件不可用
ERR_AUDIO_PLUGIN_INSTALL_NOT_AUTHORIZED	-1331	安装音频驱动插件未授权

ERR_AUDIO_PLUGIN_INSTALL_FAILED	-1332	安装音频驱动插件失败
ERR_AUDIO_PLUGIN_INSTALLED_BUT_NEED_RESTART	-1333	安装虚拟声卡插件成功，但首次安装后功能暂时不可用，此为 Mac 系统限制，请在收到此错误码后提示用户重启当前 APP
ERR_AUDIO_ENCODE_FAIL	-1304	音频帧编码失败，例如传入自定义音频数据，SDK 无法处理
ERR_UNSUPPORTED_SAMPLERATE	-1306	不支持的音频采样率
ERR_TRTC_ENTER_ROOM_FAILED	-3301	进入房间失败，请查看 onError 中的 -3301 对应的 msg 提示确认失败原因
ERR_TRTC_REQUEST_IP_TIMEOUT	-3307	请求 IP 和 sig 超时，请检查网络是否正常，或网络防火墙是否放行 UDP。可尝试访问下列 IP：162.14.22.165:8000 162.14.6.105:8000 和域名：default-query.trtc.tencent-cloud.com:8000
ERR_TRTC_CONNECT_SERVER_TIMEOUT	-3308	请求进房超时，请检查是否断网或者是否开启vpn，您也可以切换4G进行测试确认
ERR_TRTC_ROOM_PARAM_NULL	-3316	进房参数为空，请检查：enterRoom:appScene: 接口调用是否传入有效的 param
ERR_TRTC_INVALID_SDK_APPID	-3317	进房参数 sdkAppId 错误，请检查 TRTCParams.sdkAppId 是否为空
ERR_TRTC_INVALID_ROOM_ID	-3318	进房参数 roomId 错误，请检查 TRTCParams.roomId 或 TRTCParams.strRoomId 是否为空，注意 roomId 和 strRoomId 不可混用
ERR_TRTC_INVALID_USER_ID	-3319	进房参数 userId 不正确，请检查 TRTCParams.userId 是否为空

ERR_TRTC_INVALID_USER_SIG	-3320	进房参数 userSig 不正确，请检查 TRTCParams.userSig 是否为空
ERR_TRTC_ENTER_ROOM_REFUSED	-3340	进房请求被拒绝，请检查是否连续调用 enterRoom 进入相同 Id 的房间
ERR_TRTC_INVALID_PRIVATE_MAPKEY	-100006	您开启了高级权限控制，但参数 TRTCParams.privateMapKey 校验失败， 您可参考 高级权限控制 进行检查
ERR_TRTC_SERVICE_SUSPENDED	-100013	服务不可用。请检查：套餐包剩余分钟数是否大于0，腾讯云账号是否欠费。 您可参考 套餐包管理 进行查看与配置
ERR_TRTC_USER_SIG_CHECK_FAILED	-100018	UserSig 校验失败，请检查参数 TRTCParams.userSig 是否填写正确，或是否已经过期。 您可参考 UserSig 生成与校验 进行校验
ERR_TRTC_PUSH_THIRD_PARTY_CLOUD_TIMEOUT	-3321	旁路转推请求超时
ERR_TRTC_MIX_TRANSCODING_TIMEOUT	-3322	云端混流请求超时
ERR_TRTC_PUSH_THIRD_PARTY_CLOUD_FAILED	-3323	旁路转推回包异常
ERR_TRTC_MIX_TRANSCODING_FAILED	-3324	云端混流回包异常
ERR_TRTC_START_PUBLISHING_TIMEOUT	-3333	开始向腾讯云的直播 CDN 推流信令超时
ERR_TRTC_START_PUBLISHING_FAILED	-3334	开始向腾讯云的直播 CDN 推流信令异常
ERR_TRTC_STOP_PUBLISHING_TIMEOUT	-3335	停止向腾讯云的直播 CDN 推流信令超时
ERR_TRTC_STOP_PUBLISHING_FAILED	-3336	停止向腾讯云的直播 CDN 推流信令异常
ERR_TRTC_CONNECT_OTHER_ROOM	-33	请求连麦超时

_TIMEOUT	26	
ERR_TRTC_DISCONNECT_OTHER_ROOM_TIMEOUT	-3327	请求退出连麦超时
ERR_TRTC_CONNECT_OTHER_ROOM_INVALID_PARAMETER	-3328	无效参数
ERR_TRTC_CONNECT_OTHER_ROOM_AS_AUDIENCE	-3330	当前是观众角色，不能请求或断开跨房连麦，需要先 `switchRole` 到主播
ERR_BGM_OPEN_FAILED	-4001	打开音频文件失败
ERR_BGM_DECODE_FAILED	-4002	音频文件解码失败
ERR_BGM_OVER_LIMIT	-4003	数量超过限定值，如同时预加载两首背景音乐
ERR_BGM_INVALID_OPERATION	-4004	无效操作，如开始播放后再调用预加载操作

TXLiteAVWarning

TXLiteAVWarning

警告码

枚举	取值	描述
WARNING_HW_ENCODER_START_FAIL	1103	硬编码启动出现问题，自动切换到软编码
WARNING_CURRENT_ENCODE_TYPE_CHANGED	1104	表示编码器发生改变，可以通过 onWarning 函数的扩展信息中的相关字段来获取当前的编码格式和类型。 type 字段值为 0 代表 H.264 编码，1 代表 H.265 编码 hardware 字段值为 0 代表软件编码，1 代表硬件编码 stream 字段值为 0 代表大流，1 代表小流，2 代表辅流

WARNING_VIDEO_ENCODER_SW_TO_HW	1107	当前 CPU 使用率太高，无法满足软件编码需求，自动切换到硬件编码
WARNING_INSUFFICIENT_CAPTURE_FPS	1108	摄像头采集帧率不足，部分自带美颜算法的 Android 手机上会出现
WARNING_SW_ENCODER_START_FAIL	1109	软编码启动失败
WARNING_REDUCE_CAPTURE_RESOLUTION	1110	摄像头采集分辨率被降低，以满足当前帧率和性能最优解。
WARNING_CAMERA_DEVICE_EMPTY	1111	没有检测到可用的摄像头设备
WARNING_CAMERA_NOT_AUTHORIZED	1112	用户未授权当前应用使用摄像头
WARNING_OUT_OF_MEMORY	1113	内存不足，部分功能可能不正常。
WARNING_CAMERA_IS_OCCUPIED	1114	摄像头被占用.
WARNING_CAMERA_DEVICE_ERROR	1115	摄像头设备异常.
WARNING_CAMERA_DISCONNECTED	1116	摄像头无法连接.
WARNING_CAMERA_START_FAILED	1117	摄像头启动失败.
WARNING_CAMERA_SERVER_DIED	1118	系统异常.
WARNING_SCREEN_CAPTURE_NOT_AUTHORIZED	1206	用户未授权当前应用使用屏幕录制
WARNING_CURRENT_DECODE_TYPE_CHANGED	2008	表示解码器发生改变，可以通过 onWarning 函数的扩展信息中的 type 字段来获取当前的解码格式。其中 1 代表 265 解码，0 代表 264 解码。注意 Windows 端不支持此错误码的扩展信息。
WARNING_VIDEO_FRAME_DECODE_F	210	当前视频帧解码失败

AIL	1	
WARNING_HW_DECODER_START_FAIL	2106	硬解启动失败，采用软解码
WARNING_VIDEO_DECODER_HW_TO_SW	2108	当前流硬解第一个 I 帧失败，SDK 自动切软解
WARNING_SW_DECODER_START_FAIL	2109	软解码器启动失败
WARNING_VIDEO_RENDER_FAIL	2110	视频渲染失败
WARNING_MICROPHONE_DEVICE_EMPTY	1201	没有检测到可用的麦克风设备
WARNING_SPEAKER_DEVICE_EMPTY	1202	没有检测到可用的扬声器设备
WARNING_MICROPHONE_NOT_AUTHORIZED	1203	用户未授权当前应用使用麦克风
WARNING_MICROPHONE_DEVICE_ABNORMAL	1204	音频采集设备不可用（例如被占用或者PC判定无效设备）
WARNING_SPEAKER_DEVICE_ABNORMAL	1205	音频播放设备不可用（例如被占用或者PC判定无效设备）
WARNING_BLUETOOTH_DEVICE_CONNECT_FAIL	1207	蓝牙设备连接失败（例如其他应用通过设置通话音量占用音频通道）
WARNING_MICROPHONE_IS_OCCUPIED	1208	音频采集设备被占用
WARNING_AUDIO_FRAME_DECODE_FAIL	2102	当前音频帧解码失败
WARNING_AUDIO_RECORDING_WRITE_FAIL	7001	音频录制写入文件失败
WARNING_MICROPHONE_HOWLING_DETECTED	7002	录制音频时监测到啸叫
WARNING_IGNORE_UPSTREAM_FOR_AUDIENCE	6001	当前是观众角色，不支持发布音视频，需要先切换成主播角色
WARNING_UPSTREAM_AUDIO_AND_	600	音视频发送时间戳异常，可能引起音画

VIDEO_OUT_OF_SYNC	6	不同步问题

Windows（C#）
API 概览

最近更新时间：2022-10-10 11:06:33

ITRTCCloud @ TXLiteAVSDK

创建与销毁 ITRTCCloud 单例

API	描述
getTRTCShareInstance	获取 ITRTCCloud 单例对象。
destroyTRTCShareInstance	释放 ITRTCCloud 单例对象。

设置 TRTCCloudCallback 回调

API	描述
addCallback	设置回调接口 ITRTCCloudCallback 。
removeCallback	移除事件回调。

房间相关接口函数

API	描述
enterRoom	进入房间。
exitRoom	离开房间。
switchRole	切换角色，仅适用于直播场景（TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom）。
connectOtherRoom	请求跨房通话（主播 PK）。
disconnectOtherRoom	关闭跨房连麦。
setDefaultStreamRecvMode	设置音视频数据接收模式（需要在进房前设置才能生效）。
switchRoom	切换房间。

版权所有：腾讯云计算（北京）有限责任公司

CDN 相关接口函数

API	描述
startPublishing	开始向腾讯云的直播 CDN 推流。
stopPublishing	停止向腾讯云的直播 CDN 推流。
startPublishCDNStream	开始向友商云的直播 CDN 转推。
stopPublishCDNStream	停止向非腾讯云地址转推。
setMixTranscodingConfig	设置云端的混流转码参数。

视频相关接口函数

API	描述
startLocalPreview	开启本地视频的预览画面。
updateLocalView	更新本地视频预览画面的窗口。
stopLocalPreview	停止本地视频采集及预览。
muteLocalVideo	暂停/恢复推送本地的视频数据。
startRemoteView	开始拉取并显示指定用户的远端画面。
updateRemoteView	更新远端视频渲染的窗口。
stopRemoteView	停止显示远端视频画面，同时不再拉取远端用户的视频数据流。
stopAllRemoteView	停止显示所有远端视频画面，同时不再拉取远端用户的视频数据流。
muteRemoteVideoStream	暂停/恢复接收指定的远端视频流。
muteAllRemoteVideoStreams	暂停/恢复接收所有远端视频流。
setVideoEncoderParam	设置视频编码器相关参数。
setNetworkQosParam	设置网络流控相关参数。
setLocalRenderParams	设置本地图像（主流）的渲染参数。
setVideoEncoderRotation	设置视频编码输出的画面方向，即设置远端用户观看到的和服务器录制的画面方向。

setVideoEncoderMirror	设置编码器输出的画面镜像模式。
setRemoteRenderParams	设置远端图像的渲染模式。
enableSmallVideoStream	开启大小画面双路编码模式。
setRemoteVideoStreamType	选定观看指定 userId 的大画面还是小画面。

音频相关接口函数

API	描述
startLocalAudio	开启本地音频的采集和上行。
stopLocalAudio	关闭本地音频的采集和上行。
muteLocalAudio	静音/取消静音本地的音频。
muteRemoteAudio	静音/取消静音指定的远端用户的声音。
muteAllRemoteAudio	静音/取消静音所有用户的声音。
setAudioCaptureVolume	设置 SDK 采集音量。
getAudioCaptureVolume	获取 SDK 采集音量。
setAudioPlayoutVolume	设置 SDK 播放音量。
getAudioPlayoutVolume	获取 SDK 播放音量。
enableAudioVolumeEvaluation	启用或关闭音量大小提示。
startAudioRecording	开始录音。
stopAudioRecording	停止录音。

设备相关接口函数

API	描述
getDeviceManager	获取设备管理模块。

美颜特效和图像水印

API	描述
setBeautyStyle	设置美颜、美白、红润效果级别。
setWaterMark	设置水印。

音乐特效和人声特效

API	描述
getAudioEffectManager	获取音效管理类 ITXAudioEffectManager。
startSystemAudioLoopback	打开系统声音采集。
stopSystemAudioLoopback	关闭系统声音采集。
setSystemAudioLoopbackVolume	设置系统声音采集的音量。

屏幕分享相关接口函数

API	描述
startScreenCapture	启动屏幕分享。
stopScreenCapture	停止屏幕采集。
pauseScreenCapture	暂停屏幕分享。
resumeScreenCapture	恢复屏幕分享。
getScreenCaptureSources	枚举可分享的屏幕窗口，建议在 startScreenCapture 之前调用。
selectScreenCaptureTarget	设置屏幕共享参数，该方法在屏幕共享过程中也可以调用。
setSubStreamEncoderParam	设置屏幕分享的编码器参数。
setSubStreamMixVolume	设置屏幕分享的混音音量大小。
addExcludedShareWindow	将指定窗口加入屏幕分享的排除列表中，加入排除列表中的窗口不会被分享出去。
removeExcludedShareWindow	将指定窗口从屏幕分享的排除列表中移除。
removeAllExcludedShareWindow	将所有窗口从屏幕分享的排除列表中移除。

自定义采集和渲染

API	描述
enableCustomVideoCapture	启用视频自定义采集模式。
sendCustomVideoData	向 SDK 投送自己采集的视频数据。
enableCustomAudioCapture	启用音频自定义采集模式。
sendCustomAudioData	向 SDK 投送自己采集的音频数据。
setLocalVideoRenderCallback	设置本地视频自定义渲染。
setRemoteVideoRenderCallback	设置远端视频自定义渲染。
setAudioFrameCallback	设置音频数据回调。

自定义消息发送

API	描述
sendCustomCmdMsg	发送自定义消息给房间内所有用户。
sendSEIMsg	将小数据量的自定义数据嵌入视频帧中。

网络测试

API	描述
startSpeedTest	开始进行网络测速（视频通话期间请勿测试，以免影响通话质量）。
stopSpeedTest	停止网络测速。

LOG 相关接口函数

API	描述
getSDKVersion	获取 SDK 版本信息。
setLogLevel	设置 Log 输出级别。
setConsoleEnabled	启用或禁用控制台日志打印。
setLogCompressEnabled	启用或禁用 Log 的本地压缩。
setLogDirPath	设置日志保存路径。

版权所有：腾讯云计算（北京）有限责任公司

setLogCallback	设置日志回调。
showDebugView	显示仪表盘。
callExperimentalAPI	调用实验性 API 接口。

弃用接口函数

API	描述
setMicVolumeOnMixing	设置麦克风的音量大小。
startScreenCapture	启动屏幕分享。
playBGM	启动播放背景音乐。
stopBGM	停止播放背景音乐。
pauseBGM	暂停播放背景音乐。
resumeBGM	继续播放背景音乐。
getBGMDuration	获取音乐文件总时长，单位毫秒。
setBGMPosition	设置 BGM 播放进度。
setBGMVolume	设置背景音乐播放音量的大小。
setBGMPlayoutVolume	设置背景音乐本地播放音量的大小。
setBGMPublishVolume	设置背景音乐远端播放音量的大小。
playAudioEffect	播放音效。
setAudioEffectVolume	设置音效音量。
stopAudioEffect	停止音效。
stopAllAudioEffects	停止所有音效。
setAllAudioEffectsVolume	设置所有音效的音量。
pauseAudioEffect	暂停音效。
resumeAudioEffect	恢复音效。
selectScreenCaptureTarget	设置屏幕共享参数。

startRemoteView	开始显示远端视频画面。
stopRemoteView	停止显示远端视频画面，同时不再拉取远端用户的视频数据流。
setLocalViewFillMode	设置本地图像的渲染模式。
setRemoteViewFillMode	设置远端图像的渲染模式。
setLocalViewRotation	设置本地图像的顺时针旋转角度。
setRemoteViewRotation	设置远端图像的顺时针旋转角度。
setLocalViewMirror	设置本地摄像头预览画面的镜像模式。
startRemoteSubStreamView	开始显示远端用户的辅路画面（TRTCVideoStreamTypeSub，一般用于屏幕分享）。
stopRemoteSubStreamView	停止显示远端用户的辅路画面（TRTCVideoStreamTypeSub，一般用于屏幕分享）。
setRemoteSubStreamViewFillMode	设置辅路画面（TRTCVideoStreamTypeSub，一般用于屏幕分享）的显示模式。
setRemoteSubStreamViewRotation	设置辅路画面（TRTCVideoStreamTypeSub，一般用于屏幕分享）的顺时针旋转角度。
setPriorRemoteVideoStreamType	设定观看方优先选择的视频质量。
setAudioQuality	设置音频质量。
startLocalAudio	开启本地音频的采集和上行。
getCameraDevicesList	获取摄像头设备列表。
setCurrentCameraDevice	设置要使用的摄像头。
getCurrentCameraDevice	获取当前使用的摄像头。
getMicDevicesList	获取麦克风设备列表。
getCurrentMicDevice	获取当前选择的麦克风。
setCurrentMicDevice	设置要使用的麦克风。
getCurrentMicDeviceVolume	获取系统当前麦克风设备音量。
setCurrentMicDeviceVolu	设置系统当前麦克风设备的音量。

版权所有：腾讯云计算（北京）有限责任公司

me	
getSpeakerDevicesList	获取扬声器设备列表。
getCurrentSpeakerDevice	获取当前的扬声器设备。
setCurrentSpeakerDevice	设置要使用的扬声器。
getCurrentSpeakerVolume	获取系统当前扬声器设备音量。
setCurrentSpeakerVolume	设置系统当前扬声器设备音量。
startCameraDeviceTest	开始进行摄像头测试。
stopCameraDeviceTest	停止摄像头测试。
startMicDeviceTest	开启麦克风测试。
stopMicDeviceTest	停止麦克风测试。
startSpeakerDeviceTest	开启扬声器测试。
stopSpeakerDeviceTest	停止扬声器测试。

TRTCCloudCallback @ TXLiteAVSDK

腾讯云视频通话功能的回调接口类。

错误事件和警告事件

API	描述
onError	错误回调，SDK 不可恢复的错误，一定要监听，并分情况给用户适当的界面提示。
onWarning	警告回调：用于告知您一些非严重性问题，例如出现了卡顿或者可恢复的解码失败。

房间事件回调

API	描述
onEnterRoom	已加入房间的回调。
onExitRoom	离开房间的事件回调。
onSwitchRole	切换角色结果回调。

onConnectOtherRoom	请求跨房通话（主播 PK）的结果回调。
onDisconnectOtherRoom	结束跨房通话（主播 PK）的结果回调。
onSwitchRoom	切换房间 (switchRoom) 的结果回调。

成员事件回调

API	描述
onRemoteUserEnterRoom	有用户加入当前房间。
onRemoteUserLeaveRoom	有用户离开当前房间。
onUserVideoAvailable	用户是否开启摄像头视频。
onUserSubStreamAvailable	用户是否开启屏幕分享。
onUserAudioAvailable	用户是否开启音频上行。
onFirstVideoFrame	开始渲染本地或远程用户的首帧画面。
onFirstAudioFrame	开始播放远程用户的首帧音频（本地声音暂不通知）。
onSendFirstLocalVideoFrame	首帧本地视频数据已经被送出。
onSendFirstLocalAudioFrame	首帧本地音频数据已经被送出。

统计和质量回调

API	描述
onNetworkQuality	网络质量：该回调每2秒触发一次，统计当前网络的上行和下行质量。
onStatistics	技术指标统计回调。

服务器事件回调

API	描述
onConnectionLost	SDK 跟服务器的连接断开。
onTryToReconnect	SDK 尝试重新连接到服务器。
onConnectionRecovery	SDK 跟服务器的连接恢复。
onSpeedTest	服务器测速的回调，SDK 对多个服务器 IP 做测速，每个 IP 的测速结果通过这

	个回调通知。

硬件设备事件回调

API	描述
onCameraDidReady	摄像头准备就绪。
onMicDidReady	麦克风准备就绪。
onUserVoiceVolume	用于提示音量大小的回调,包括每个 userId 的音量和远端总音量。
onDeviceChange	本地设备通断回调。
onTestMicVolume	麦克风测试音量回调。
onTestSpeakerVolume	扬声器测试音量回调。
onAudioDeviceCaptureVolumeChanged	当前音频采集设备音量变化通知。
onAudioDevicePlayoutVolumeChanged	当前音频播放设备音量变化通知。

自定义消息的接收回调

API	描述
onRecvCustomCmdMsg	收到自定义消息回调。
onMissCustomCmdMsg	自定义消息丢失回调。
onRecvSEIMsg	收到 SEI 消息的回调。

CDN 旁路转推回调

API	描述
onStartPublishing	开始向腾讯云的直播 CDN 推流的回调，对应于 TRTCCloud 中的 startPublishing() 接口。
onStopPublishing	停止向腾讯云的直播 CDN 推流的回调，对应于 TRTCCloud 中的 stopPublishing() 接口。
onStartPublishCDNStream	启动旁路推流到 CDN 完成的回调。

onStopPublishCDNStream	停止旁路推流到 CDN 完成的回调。
onSetMixTranscodingConfig	设置云端的混流转码参数的回调，对应于 TRTCCloud 中的 setMixTranscodingConfig() 接口。

屏幕分享回调

API	描述
onScreenCaptureCovered	当屏幕分享窗口被遮挡无法正常捕获时，SDK 会通过此回调通知，可在此回调里通知用户移开遮挡窗口。
onScreenCaptureStarted	当屏幕分享开始时，SDK 会通过此回调通知。
onScreenCapturePaused	当屏幕分享暂停时，SDK 会通过此回调通知。
onScreenCaptureResumed	当屏幕分享恢复时，SDK 会通过此回调通知。
onScreenCaptureStoped	当屏幕分享停止时，SDK 会通过此回调通知。

弃用接口回调

API	描述
onUserEnter	废弃接口：有主播加入当前房间。
onUserExit	废弃接口：有用户（主播）离开当前房间。
onAudioEffectFinished	废弃接口：播放音效结束回调。
onPlayBGMBegin	废弃接口：开始播放背景音乐。
onPlayBGMProgress	废弃接口：播放背景音乐的进度。
onPlayBGMComplete	废弃接口：播放背景音乐结束。

视频数据帧的自定义处理回调

API	描述
onRenderVideoFrame	自定义视频渲染回调。

版权所有：腾讯云计算（北京）有限责任公司

声音数据帧的自定义处理回调（只读）

回调函数是在 SDK 内部线程同步抛出来的，请不要做耗时操作。

> **说明**
> 请按需定义相关函数实现，减少不必要的性能损耗。

API	描述
onCapturedAudioFrame	本地麦克风采集到的音频数据回调。
onPlayAudioFrame	混音前的每一路远程用户的音频数据（例如您要对某一路的语音进行文字转换，必须要使用这里的原始数据，而不是混音之后的数据）。
onMixedPlayAudioFrame	各路音频数据混合后送入喇叭播放的音频数据。

日志相关回调

API	描述
onLog	有日志打印时的回调。

关键类型定义

类名	描述
TRTCImageBuffer	图缓存。
TRTCScreenCaptureSourceInfo	屏幕采集信息。
SIZE	记录 buffer 的长宽。
ITRTCScreenCaptureSourceList	屏幕窗口列表。
ITRTCDeviceCollection	设备列表。
ITRTCDeviceInfo	设备 Item 信息。
TRTCParams	进房相关参数。
TRTCVideoEncParam	视频编码参数。
TRTCNetworkQosParam	网络流控相关参数。

版权所有：腾讯云计算（北京）有限责任公司

TRTCQualityInfo	视频质量。
TRTCVolumeInfo	音量大小。
TRTCVideoFrame	视频帧数据。
TRTCAudioFrame	音频帧数据。
TRTCSpeedTestResult	网络测速结果。
RECT	记录矩形的四个点坐标。
TRTCMixUser	云端混流中每一路子画面的位置信息。
TRTCTranscodingConfig	云端混流（转码）配置。
TRTCPublishCDNParam	CDN 旁路推流参数。
TRTCAudioRecordingParams	录音参数。
TRTCAudioEffectParam	音效播放。
TRTCLocalStatistics	自己本地的音视频统计信息。
TRTCRemoteStatistics	远端成员的音视频统计信息。
TRTCStatistics	统计数据。

版权所有：腾讯云计算（北京）有限责任公司

错误码

最近更新时间：2023-09-19 14:58:13

错误码表

基础错误码

符号	值	含义
ERR_NULL	0	无错误

进房相关错误码

`TRTCCloud.enterRoom()` 在进房失败时会触发此类错误码，您可以通过回调函数 `TRTCCloudDelegate.onEnterRoom()` 和 `TRTCCloudDelegate.OnError()` 捕获相关通知。

符号	值	含义
ERR_ROOM_ENTER_FAIL	-3301	进入房间失败
ERR_ENTER_ROOM_PARAM_NULL	-3316	进房参数为空，请检查 `TRTCCloud.enterRoom()` 接口调用是否传入有效的 param
ERR_SDK_APPID_INVALID	-3317	进房参数 sdkAppId 错误
ERR_ROOM_ID_INVALID	-3318	进房参数 roomId 错误
ERR_USER_ID_INVALID	-3319	进房参数 userID 不正确
ERR_USER_SIG_INVALID	-3320	进房参数 userSig 不正确
ERR_ROOM_REQUEST_ENTER_ROOM_TIMEOUT	-3308	请求进房超时，请检查网络
ERR_SERVER_INFO_PRIVILEGE_FLAG_ERROR	-100006	权限位校验失败，请检查 privateMapKey 是否正确
ERR_SERVER_INFO_SERVICE_SUSPENDED	-100013	服务不可用。请检查：套餐包剩余分钟数是否大于0，腾讯云账号是否欠费
ERR_SERVER_INFO_ECDH_GET_TINYID	-100018	userSig 校验失败，请检查 userSig 是否正确

退房相关错误码

`TRTCCloud.exitRoom()` 在退房失败时会触发此类错误码，您可以通过回调函数 `TRTCCloudDelegate.OnError()` 捕获相关通知。

符号	值	含义
ERR_ROOM_REQUEST_QUIT_ROOM_TIMEOUT	-3325	请求退房超时

设备（摄像头、麦克风、扬声器）相关错误码

您可以通过回调函数 `TRTCCloudDelegate.OnError()` 捕获相关通知。

符号	值	含义
ERR_CAMERA_START_FAIL	-1301	打开摄像头失败，例如在 Windows 或 Mac 设备，摄像头的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_CAMERA_NOT_AUTHORIZED	-1314	摄像头设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_CAMERA_SET_PARAM_FAIL	-1315	摄像头参数设置出错（参数不支持或其它）
ERR_CAMERA_OCCUPY	-1316	摄像头正在被占用中，可尝试打开其他摄像头
ERR_MIC_START_FAIL	-1302	打开麦克风失败，例如在 Windows 或 Mac 设备，麦克风的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_MIC_NOT_AUTHORIZED	-1317	麦克风设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_MIC_SET_PARAM_FAIL	-1318	麦克风设置参数失败
ERR_MIC_OCCUPY	-1319	麦克风正在被占用中，例如移动设备正在通话时，打开麦克风会失败
ERR_MIC_STOP_FAIL	-1320	停止麦克风失败
ERR_SPEAKER_START_FAIL	-1321	打开扬声器失败，例如在 Windows 或 Mac 设备，扬声器的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_SPEAKER_SET_PARAM_FAIL	-1322	扬声器设置参数失败

ERR_SPEAKER_STOP_FAIL	-1323	停止扬声器失败

屏幕分享相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_SCREEN_CAPTURE_START_FAIL	-1308	开始录屏失败，如果在移动设备出现，可能是权限被用户拒绝了，如果在 Windows 或 Mac 系统的设备出现，请检查录屏接口的参数是否符合要求
ERR_SCREEN_CAPTURE_UNSURPORT	-1309	录屏失败，在 Android 平台，需要5.0以上的系统，在 iOS 平台，需要11.0以上的系统
ERR_SERVER_CENTER_NO_PRIVILEDGE_PUSH_SUB_VIDEO	-102015	没有权限上行辅路
ERR_SERVER_CENTER_ANOTHER_USER_PUSH_SUB_VIDEO	-102016	其他用户正在上行辅路
ERR_SCREEN_CAPTURE_STOPPED	-7001	录屏被系统中止

编解码相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_VIDEO_ENCODE_FAIL	-1303	视频帧编码失败，例如 iOS 设备切换到其他应用时，硬编码器可能被系统释放，再切换回来时，硬编码器重启前，可能会抛出
ERR_UNSUPPORTED_RESOLUTION	-1305	不支持的视频分辨率
ERR_AUDIO_ENCODE_FAIL	-1304	音频帧编码失败，例如传入自定义音频数据，SDK 无法处理
ERR_UNSUPPORTED_SAMPLERATE	-1306	不支持的音频采样率

自定义采集相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_PIXEL_FORMAT_UNSUPPORTED	-1327	设置的 pixel format 不支持
ERR_BUFFER_TYPE_UNSUPPORTED	-1328	设置的 buffer type 不支持

CDN 绑定和混流相关错误码

您可以通过回调函数 TRTCCloudDelegate.onStartPublishing() 和 TRTCCloudDelegate.onSetMixTranscodingConfig() 捕获相关通知。

符号	值	含义
ERR_PUBLISH_CDN_STREAM_REQUEST_TIME_OUT	-3321	旁路转推请求超时
ERR_CLOUD_MIX_TRANSCODING_REQUEST_TIME_OUT	-3322	云端混流请求超时
ERR_PUBLISH_CDN_STREAM_SERVER_FAILED	-3323	旁路转推回包异常
ERR_CLOUD_MIX_TRANSCODING_SERVER_FAILED	-3324	云端混流回包异常
ERR_ROOM_REQUEST_START_PUBLISHING_TIMEOUT	-3333	开始向腾讯云的直播 CDN 推流信令超时
ERR_ROOM_REQUEST_START_PUBLISHING_ERROR	-3334	开始向腾讯云的直播 CDN 推流信令异常
ERR_ROOM_REQUEST_STOP_PUBLISHING_TIMEOUT	-3335	停止向腾讯云的直播 CDN 推流信令超时
ERR_ROOM_REQUEST_STOP_PUBLISHING_ERROR	-3336	停止向腾讯云的直播 CDN 推流信令异常

跨房连麦相关错误码

TRTCCloud.ConnectOtherRoom() 在跨房失败时会触发此类错误码，您可以通过回调函数 TRTCCloudDelegate.onConnectOtherRoom() 捕获相关通知。

符号	值	含义

ERR_ROOM_REQUEST_CONN_ROOM_TIMEOUT	-3326	请求连麦超时
ERR_ROOM_REQUEST_DISCONN_ROOM_TIMEOUT	-3327	请求退出连麦超时
ERR_ROOM_REQUEST_CONN_ROOM_INVALID_PARAM	-3328	无效参数
ERR_CONNECT_OTHER_ROOM_AS_AUDIENCE	-3330	当前是观众角色，不能请求或断开跨房连麦，需要先 switchRole() 到主播
ERR_SERVER_CENTER_CONN_ROOM_NOT_SUPPORT	-102031	不支持跨房间连麦
ERR_SERVER_CENTER_CONN_ROOM_REACH_MAX_NUM	-102032	达到跨房间连麦上限
ERR_SERVER_CENTER_CONN_ROOM_REACH_MAX_RETRY_TIMES	-102033	跨房间连麦重试次数耗尽
ERR_SERVER_CENTER_CONN_ROOM_REQ_TIMEOUT	-102034	跨房间连麦请求超时
ERR_SERVER_CENTER_CONN_ROOM_REQ	-102035	跨房间连麦请求格式错误
ERR_SERVER_CENTER_CONN_ROOM_NO_SIG	-102036	跨房间连麦无签名
ERR_SERVER_CENTER_CONN_ROOM_DECRYPT_SIG	-102037	跨房间连麦签名解密失败
ERR_SERVER_CENTER_CONN_ROOM_NO_KEY	-102038	未找到跨房间连麦签名解密密钥
ERR_SERVER_CENTER_CONN_ROOM_PARSE_SIG	-102039	跨房间连麦签名解析错误
ERR_SERVER_CENTER_CONN_ROOM_INVALID_SIG_TIME	-102040	跨房间连麦签名时间戳错误
ERR_SERVER_CENTER_CONN_ROOM_SIG_GROUPID	-102041	跨房间连麦签名内房间号不匹配
ERR_SERVER_CENTER_CONN_ROOM_N	-102	跨房间连麦签名内用户名不匹配

OT_CONNED	042	
ERR_SERVER_CENTER_CONN_ROOM_USER_NOT_CONNED	-102043	本用户未发起连麦
ERR_SERVER_CENTER_CONN_ROOM_FAILED	-102044	跨房间连麦失败
ERR_SERVER_CENTER_CONN_ROOM_CANCEL_FAILED	-102045	取消跨房间连麦失败
ERR_SERVER_CENTER_CONN_ROOM_CONNED_ROOM_NOT_EXIST	-102046	被连麦房间不存在
ERR_SERVER_CENTER_CONN_ROOM_CONNED_REACH_MAX_ROOM	-102047	被连麦房间达到连麦上限
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_NOT_EXIST	-102048	被连麦用户不存在
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_DELETED	-102049	被连麦用户已被删除
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_FULL	-102050	被连麦用户达到资源上限
ERR_SERVER_CENTER_CONN_ROOM_INVALID_SEQ	-102051	连麦请求序号错乱

警告码表

警告码无需特别关注，您可以按需选择是否对当前用户进行相应的提示。

符号	值	含义
WARNING_HW_ENCODER_START_FAIL	1103	硬编码启动出现问题，自动切换到软编码
WARNING_VIDEO_ENCODER_SW_TO_HW	1107	当前 CPU 使用率太高，无法满足软件编码需求，自动切换到硬件编码
WARNING_INSUFFICIENT_CAPTURE_FPS	1108	摄像头采集帧率不足，部分自带美颜算法的 Android 手机上会出现
WARNING_SW_ENCODER_START_FAIL	1109	软编码启动失败
WARNING_REDUCE_CAPTURE_R	111	摄像头采集分辨率被降低，以满足当前帧率和性能

ESOLUTION	0	最优解。
WARNING_CAMERA_DEVICE_EMPTY	1111	没有检测到可用的摄像头设备
WARNING_CAMERA_NOT_AUTHORIZED	1112	用户未授权当前应用使用摄像头
WARNING_MICROPHONE_DEVICE_EMPTY	1201	没有检测到可用的麦克风设备
WARNING_SPEAKER_DEVICE_EMPTY	1202	没有检测到可用的扬声器设备
WARNING_MICROPHONE_NOT_AUTHORIZED	1203	用户未授权当前应用使用麦克风
WARNING_MICROPHONE_DEVICE_ABNORMAL	1204	音频采集设备不可用（例如被占用）
WARNING_SPEAKER_DEVICE_ABNORMAL	1205	音频播放设备不可用（例如被占用）
WARNING_VIDEO_FRAME_DECODE_FAIL	2101	当前视频帧解码失败
WARNING_AUDIO_FRAME_DECODE_FAIL	2102	当前音频帧解码失败
WARNING_VIDEO_PLAY_LAG	2105	当前视频播放出现卡顿
WARNING_HW_DECODER_START_FAIL	2106	硬解启动失败，采用软解码
WARNING_VIDEO_DECODER_HW_TO_SW	2108	当前流硬解第一个 I 帧失败，SDK 自动切软解
WARNING_SW_DECODER_START_FAIL	2109	软解码器启动失败
WARNING_VIDEO_RENDER_FAIL	2110	视频渲染失败
WARNING_START_CAPTURE_IGNORED	4000	已经在采集，启动采集被忽略
WARNING_AUDIO_RECORDING_W	700	音频录制写入文件失败

RITE_FAIL	1	
WARNING_ROOM_DISCONNECT	5101	网络断开连接
WARNING_IGNORE_UPSTREAM_FOR_AUDIENCE	6001	当前是观众角色，忽略上行音视频数据
WARNING_NET_BUSY	1101	网络状况不佳：上行带宽太小，上传数据受阻
WARNING_RTMP_SERVER_RECONNECT	1102	直播推流，网络断连, 已启动自动重连（自动重连连续失败超过三次会放弃）
WARNING_LIVE_STREAM_SERVER_RECONNECT	2103	直播拉流，网络断连, 已启动自动重连（自动重连连续失败超过三次会放弃）
WARNING_RECV_DATA_LAG	2104	网络来包不稳：可能是下行带宽不足，或由于主播端出流不均匀
WARNING_RTMP_DNS_FAIL	3001	直播，DNS 解析失败
WARNING_RTMP_SEVER_CONN_FAIL	3002	直播，服务器连接失败
WARNING_RTMP_SHAKE_FAIL	3003	直播，与 RTMP 服务器握手失败
WARNING_RTMP_SERVER_BREAK_CONNECT	3004	直播，服务器主动断开
WARNING_RTMP_READ_WRITE_FAIL	3005	直播，RTMP 读/写失败，将会断开连接
WARNING_RTMP_WRITE_FAIL	3006	直播，RTMP 写失败（SDK 内部错误码，不会对外抛出）
WARNING_RTMP_READ_FAIL	3007	直播，RTMP 读失败（SDK 内部错误码，不会对外抛出）
WARNING_RTMP_NO_DATA	3008	直播，超过30s 没有数据发送，主动断开连接
WARNING_PLAY_LIVE_STREAM_INFO_CONNECT_FAIL	3009	直播，connect 服务器调用失败（SDK 内部错误码，不会对外抛出）
WARNING_NO_STEAM_SOURCE_	301	直播，连接失败，该流地址无视频（SDK 内部错

FAIL	0	误码，不会对外抛出）
WARNING_ROOM_RECONNECT	5102	网络断连，已启动自动重连
WARNING_ROOM_NET_BUSY	5103	网络状况不佳：上行带宽太小，上传数据受阻

Windows（ActiveX）API 概览

最近更新时间：2023-03-10 16:15:41

ActiveX SDK 的接口

创建实例

CLASSID	描述
CLSID:99DD15EF-B353-4E47-9BE7-7DB4BC13613C	通过给 HTML 中的 OBJECT 对象设置 CLASSID 属性来加载 ActiveX 控件

房间相关接口函数

API	描述
enterRoom([in] INT sdkappid, [in] BSTR usersig, [in] BSTR room_id, [in] BSTR user_id, [in] enum AxTRTCRoleType role, [in] enum AxTRTCAppScene scene, [in] BSTR extend_info)	进入房间，其扩展参数 extend_info 参见 **主调接口的 JSON 格式**
exitRoom()	离开房间

视频相关接口函数

API	描述
startLocalPreview()	开启本地视频的预览画面
stopLocalPreview()	停止本地视频采集及预览
muteLocalVideo([in] enum AxTRTCVideoStreamType stream_type, [in] BOOL mute)	暂停/恢复推送本地的视频数据
setVideoMuteBlackFrame()	当调用了 `muteLocalVideo(true)` 暂停本地画面时，可以通过调用本接口推送黑帧数据到对方
startRemoteView([in] BSTR user_id, [in] enum AxTRTCVideoStreamType stream_type)	开始拉取并显示指定用户的远端画面

stopRemoteView([in] BSTR user_id, [in] enum AxTRTCVideoStreamType stream_type)	停止显示远端视频画面，同时不再拉取远端用户的视频数据流
muteRemoteVideoStream([in] BSTR user_id, [in] enum AxTRTCVideoStreamType stream_type, [in] BOOL mute)	暂停/恢复接收指定的远端视频流
setLocalRenderParams([in] enum AxTRTCVideoRotation rotation, [in] enum AxTRTCVideoFillMode fill_mode, [in] enum AxTRTCVideoMirrorType mirror_type)	设置本地图像（主流）的渲染参数
setRemoteRenderParams([in] BSTR user_id, [in] enum AxTRTCVideoStreamType stream_type, [in] enum AxTRTCVideoRotation rotation, [in] enum AxTRTCVideoFillMode fill_mode, [in] enum AxTRTCVideoMirrorType mirror_type)	设置远端图像的渲染模式
setVideoEncoderRotation([in] enum AxTRTCVideoRotation rotation)	设置视频编码输出的画面方向，即设置远端用户观看到的和服务器录制的画面方向
setVideoEncoderMirror([in] BOOL mirror)	设置编码器输出的画面镜像模式
setVideoEncoderParam([in] enum AxTRTCVideoResolution video_resolution, [in] enum AxTRTCVideoResolutionMode resolution_mode, [in, defaultvalue(15)] UINT video_fps, [in, defaultvalue(550)] UINT video_bitrate, [in, defaultvalue(0)] UINT min_video_bitrate)	设置视频编码器的编码参数，该设置能够决定远端用户看到的画面质量（第1、2个参数为必填项，分别代表分辨率和视频宽高比模式，第3、4、5个参数为选填项，分别代表帧率、目前码率和最小码率，默认值分别为15fps，550kbps，0kbps）
setVideoMuteImage([in] INT width, [in] INT height, [in] BSTR rgba_buffer, [in] INT length)	设置本地画面被暂停期间的替代图片
snapshotVideo([in] BSTR user_id, [in] enum AxTRTCVideoStreamType stream_type)	视频画面截图

音频相关接口函数

API	描述
startLocalAudio([in] enum AxTRTCAudioQuality quality)	开启本地音频的采集和上行
stopLocalAudio()	关闭本地音频的采集和上行

muteLocalAudio([in] BOOL mute)	静音/取消静音本地的音频
muteRemoteAudio([in] BSTR user_id, [in] BOOL mute)	静音/取消静音指定的远端用户的声音
setAudioCaptureVolume([in] INT volume)	设置 SDK 采集音量
getAudioCaptureVolume([out, retval] INT* volume)	获取 SDK 采集音量，通过 volume 参数返回
setAudioPlayoutVolume([in] INT volume)	设置 SDK 播放音量
getAudioPlayoutVolume([out, retval] INT* volume)	获取 SDK 播放音量，通过 volume 参数返回
enableAudioVolumeEvaluation([in] INT interval)	启用或关闭音量大小提示
setRemoteAudioVolume([in] BSTR user_id, [in] INT volume)	设定某一个远端用户的声音播放音量

设备相关接口函数

API	描述
getDevicesList([in] enum AxTRTCMediaDeviceType device_type, [out, retval] BSTR* device_list)	获取设备列表，其 device_list 以JSON串格式返回，参见 **主调接口的 JSON 格式**
setCurrentDevice([in] enum AxTRTCMediaDeviceType device_type, [in] BSTR device_id)	设置当前使用的设备
getCurrentDevice([in] enum AxTRTCMediaDeviceType device_type, [out, retval] BSTR* device_info)	获取当前使用的设备，其 device_info 以 JSON 串格式返回，参见 **主调接口的 JSON 格式**
setCurrentDeviceVolume([in] enum AxTRTCMediaDeviceType device_type, [in] UINT volume)	设置当前麦克风或扬声器的音量
getCurrentDeviceVolume([in] enum AxTRTCMediaDeviceType device_type, [out, retval] UINT* volume)	获取当前麦克风或扬声器的音量，通过 volume 参数返回
startCameraDeviceTest()	开始测试摄像头

stopCameraDeviceTest()	停止测试摄像头
startMicDeviceTest([in] UINT interval)	开始测试麦克风
stopMicDeviceTest()	停止测试麦克风
startSpeakerDeviceTest([in] BSTR file_path)	开始测试扬声器
stopSpeakerDeviceTest()	停止测试扬声器

美颜特效和图像水印

API	描述
setBeautyStyle([in] enum AxTRTCBeautyStyle style, [in] UINT beauty_level, [in] UINT whiteness_level, [in] UINT ruddiness_level)	设置美颜、美白、红润效果级别

屏幕分享相关接口函数

API	描述
startScreenCapture([in] enum AxTRTCVideoStreamType stream_type, [in] enum AxTRTCVideoResolution video_resolution, [in] UINT video_fps, [in, defaultvalue(FALSE)] BOOL local_render)	启动屏幕分享
stopScreenCapture()	停止屏幕采集
getScreenCaptureSources([in] UINT thumbnail_width, [in] UINT thumbnail_height, [in] UINT icon_width, [in] UINT icon_height, [out, retval] BSTR* screen_source_list)	枚举可分享的屏幕和窗口，其 screen_source_list 以 JSON 串格式返回，参见 **主调接口的 JSON 格式**
selectScreenCaptureTarget([in] UINT source_id, [in] UINT capture_rect_left, [in] UINT capture_rect_top, [in] UINT capture_rect_width, [in] UINT capture_rect_height)	选取要分享的屏幕或窗口

其他相关接口函数

API	描述
getVersion([out, retval] BSTR* version)	获取 SDK 版本信息

callExperimentalAPI([in] BSTR json)	调用实验性接口

ActiveX 的回调监听

腾讯云 ActiveX SDK 视频通话功能的回调监听。

错误事件和警告事件

API	描述
onError([in] INT error_code,[in] BSTR error_message)	错误回调，SDK 不可恢复的错误，一定要监听，并分情况给用户适当的界面提示
onWarning([in] INT warning_code, [in] BSTR warning_message)	警告回调：用于告知您一些非严重性问题，例如出现了卡顿或者可恢复的解码失败

房间事件回调

API	描述
onEnterRoom([in] INT result)	已加入房间的回调
onExitRoom([in] INT reason)	离开房间的事件回调

成员事件回调

API	描述
onRemoteUserEnterRoom([in] BSTR user_id)	有用户加入当前房间
onRemoteUserLeaveRoom([in] BSTR user_id,[in] INT reason)	有用户离开当前房间
onUserVideoAvailable([in] BSTR user_id, BOOL available)	用户是否开启摄像头视频
onUserSubStreamAvailable([in] BSTR user_id, BOOL available)	用户是否开启屏幕分享
onUserAudioAvailable([in] BSTR user_id, BOOL available)	用户是否开启音频上行
onFirstVideoFrame([in] BSTR user_id, [in] INT stream_type, [in] INT width, [in] INT height)	开始渲染本地或远程用户的首帧画面

硬件设备事件回调

API	描述
onUserVoiceVolume([in] BSTR user_volumes, [in] INT user_volumes_count, [in] INT total_volume)	用于提示音量大小的回调,包括每个 userId 的音量和远端总音量， user_volumes 为 JSON 串格式，参见 **回调监听接口返回的 JSON 格式**
onTestMicVolume([in] INT volume)	麦克风测试音量回调
onTestSpeakerVolume([in] INT volume)	扬声器测试音量回调
onAudioDeviceCaptureVolumeChanged([in] INT volume, BOOL muted)	当前音频采集设备音量变化通知
onAudioDevicePlayoutVolumeChanged([in] INT volume, BOOL muted)	当前音频播放设备音量变化通知

屏幕分享回调

API	描述
onScreenCaptureStarted()	当屏幕分享开始时，SDK 会通过此回调通知
onScreenCaptureStoped([in] INT reason)	当屏幕分享停止时，SDK 会通过此回调通知

相关枚举值

AxTRTCVideoResolution 的枚举值

AxTRTCVideoResolution 的枚举值	描述
AxTRTCVideoResolution_120_120 = 1	宽高比 1:1；分辨率 120x120；建议码率（VideoCall）80kbps; 建议码率（LIVE）120kbps
AxTRTCVideoResolution_160_160 = 3	宽高比 1:1 分辨率 160x160；建议码率（VideoCall）100kbps; 建议码率（LIVE）150kbps
AxTRTCVideoResolution_270_270 = 5	宽高比 1:1；分辨率 270x270；建议码率（VideoCall）200kbps; 建议码率（LIVE）300kbps
AxTRTCVideoResolution_480_480 = 7	宽高比 1:1；分辨率 480x480；建议码率（VideoCall）350kbps; 建议码率（LIVE）500kbps
AxTRTCVideoResolution_160_120 = 50	宽高比4:3；分辨率 160x120；建议码率（VideoCall）100kbps; 建议码率（LIVE）150kbps

AxTRTCVideoResolution_240_180 = 52	宽高比 4:3；分辨率 240x180；建议码率（VideoCall）150kbps；建议码率（LIVE）250kbps
AxTRTCVideoResolution_280_210 = 54	宽高比 4:3；分辨率 280x210；建议码率（VideoCall）200kbps；建议码率（LIVE）300kbps
AxTRTCVideoResolution_320_240 = 56	宽高比 4:3；分辨率 320x240；建议码率（VideoCall）250kbps；建议码率（LIVE）375kbps
AxTRTCVideoResolution_400_300 = 58	宽高比 4:3；分辨率 400x300；建议码率（VideoCall）300kbps；建议码率（LIVE）450kbps
AxTRTCVideoResolution_480_360 = 60	宽高比 4:3；分辨率 480x360；建议码率（VideoCall）400kbps；建议码率（LIVE）600kbps
AxTRTCVideoResolution_640_480 = 62	宽高比 4:3；分辨率 640x480；建议码率（VideoCall）600kbps；建议码率（LIVE）900kbps
AxTRTCVideoResolution_960_720 = 64	宽高比 4:3；分辨率 960x720；建议码率（VideoCall）1000kbps；建议码率（LIVE）1500kbps
AxTRTCVideoResolution_160_90 = 100	宽高比 16:9；分辨率 160x90；建议码率（VideoCall）150kbps；建议码率（LIVE）250kbps
AxTRTCVideoResolution_256_144 = 102	宽高比 16:9；分辨率 256x144；建议码率（VideoCall）200kbps；建议码率（LIVE）300kbps
AxTRTCVideoResolution_320_180 = 104	宽高比 16:9；分辨率 320x180；建议码率（VideoCall）250kbps；建议码率（LIVE）400kbps
AxTRTCVideoResolution_480_270 = 106	宽高比 16:9；分辨率 480x270；建议码率（VideoCall）350kbps；建议码率（LIVE）550kbps
AxTRTCVideoResolution_640_360 = 108	宽高比 16:9；分辨率 640x360；建议码率（VideoCall）500kbps；建议码率（LIVE）900kbps
AxTRTCVideoResolution_960_540 = 110	宽高比 16:9；分辨率 960x540；建议码率（VideoCall）850kbps；建议码率（LIVE）1300kbps
AxTRTCVideoResolution_1280_720 = 112	宽高比 16:9；分辨率 1280x720；建议码率（VideoCall）1200kbps；建议码率（LIVE）1800kbps
AxTRTCVideoResolution_1920_1080 = 114	宽高比 16:9；分辨率 1920x1080；建议码率（VideoCall）2000kbps；建议码率（LIVE）3000kbps

AxTRTCVideoStreamType 的枚举值

枚举值	描述
AxTRTCVideoStreamTypeBig = 0	高清大画面，一般用来传输摄像头的视频数据
AxTRTCVideoStreamTypeSmall = 1	低清小画面：小画面和大画面的内容相互，但是分辨率和码率都比大画面低，因此清晰度也更低
AxTRTCVideoStreamTypeSub = 2	辅流画面：一般用于屏幕分享，同一时间在同一个房间中只允许一个用户发布辅流视频，其他用户必须要等该用户关闭之后才能发布自己的辅流

AxTRTCVideoFillMode 的枚举值

枚举值	描述
AxTRTCVideoFillMode_Fill = 0	**填充模式：**即将画面内容居中等比缩放以充满整个显示区域，超出显示区域的部分将会被裁剪掉，此模式下画面可能不完整
AxTRTCVideoFillMode_Fit = 1	**适应模式：**即按画面长边进行缩放以适应显示区域，短边部分会被填充为黑色，此模式下图像完整但可能留有黑边

AxTRTCVideoMirrorType 的枚举值

枚举值	描述
AxTRTCVideoMirrorType_Enable = 1	强制开启镜像
AxTRTCVideoMirrorType_Disable = 2	强制关闭镜像

AxTRTCVideoRotation 的枚举值

枚举值	描述
AxTRTCVideoRotation0 = 0	不旋转
AxTRTCVideoRotation90 = 1	顺时针旋转90度
AxTRTCVideoRotation180 = 2	顺时针旋转180度
AxTRTCVideoRotation270 = 3	顺时针旋转270度

版权所有：腾讯云计算（北京）有限责任公司

AxTRTCBeautyStyle 的枚举值

枚举值	描述
AxTRTCBeautyStyleSmooth = 0	光滑，算法比较激进，磨皮效果比较明显，适用于秀场直播
AxTRTCBeautyStyleNature = 1	自然，算法更多地保留了面部细节，磨皮效果更加自然，适用于绝大多数直播场景

AxTRTCAppScene 的枚举值

枚举值	描述
AxTRTCAppSceneVideoCall = 0	视频通话场景，支持720P、1080P高清画质，单个房间最多支持300人同时在线，最高支持50人同时发言。适用于**1对1视频通话、300人视频会议、在线问诊、教育小班课、远程面试**等业务场景
AxTRTCAppSceneLIVE = 1	视频互动直播，支持平滑上下麦，切换过程无需等待，主播延时小于300ms；支持十万级别观众同时播放，播放延时低至1000ms。 适用于**低延时互动直播、大班课、主播 PK、视频相亲、在线互动课堂、远程培训、超大型会议**等业务场景。此场景下，您必须通过 TRTCParams 中的 role 字段指定当前用户的角色
AxTRTCAppSceneAudioCall = 2	语音通话场景，默认采用 SPEECH 音质，单个房间最多支持300人同时在线，最高支持50人同时发言。 适用于**1对1语音通话、300人语音会议、语音聊天、语音会议、在线狼人杀**等业务场景
AxTRTCAppSceneVoiceChatRoom = 3	语音互动直播，支持平滑上下麦，切换过程无需等待，主播延时小于300ms；支持十万级别观众同时播放，播放延时低至1000ms。适用于**语音俱乐部、在线 K 歌房、音乐直播间、FM 电台**等业务场景。此场景下，您必须通过 TRTCParams 中的 role 字段指定当前用户的角色

AxTRTCRoleType 的枚举值

枚举值	描述
AxTRTCRoleAnchor = 20	**主播**：可以随时发布自己的音视频流，但人数有限制，同一个房间中最多只允许 50 个主播同时发布自己的音视频流
AxTRTCRoleAudience = 21	**观众**：只能观看其他用户的音视频流，要发布音视频流，需要先通过AxTRTCRoleAnchor角色进房，同一个房间中最多能容纳10万观众

AxTRTCVideoQosPreference 的枚举值

AxTRTCVideoQosPrefere	描述

nce 的枚举值	
AxTRTCVideoQosPreferenceSmooth = 1	**流畅优先**：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的流畅性，代价就是画面会比较模糊且伴随有较多的马赛克
AxTRTCVideoQosPreferenceClear = 2	**清晰优先（默认值）**：即当前网络不足以传输既清晰又流畅的画面时，优先保证画面的清晰度，代价就是画面会比较卡顿

AxTRTCQuality 的枚举值

AxTRTCQuality 的枚举值	描述
AxTRTCQuality_Unknown = 0	未定义
AxTRTCQuality_Excellent = 1	当前网络非常好
AxTRTCQuality_Good = 2	当前网络比较好
AxTRTCQuality_Poor = 3	当前网络一般
AxTRTCQuality_Bad = 4	当前网络较差
AxTRTCQuality_Vbad = 5	当前网络很差
AxTRTCQuality_Down = 6	当前网络不满足 TRTC 的最低要求

AxTRTCAudioQuality 的枚举值

枚举值	描述
AxTRTCAudioQualitySpeech = 1	**人声模式**，具备几个模式中最强的网络抗性，适合语音通话为主的场景，比如在线会议，语音通话等 • 采样率：16k • 单声道 • 编码码率：16kbps
AxTRTCAudioQualityDefault = 2	**默认模式**，介于 Speech 和 Music 之间的档位，SDK 默认档位，推荐选择 • 采样率：48k • 单声道 • 编码码率：50kbps
AxTRTCAudioQualityMusic = 3	**音乐模式**，适合需要高保真传输音乐的场景，比如在线 K 歌、音乐直播等

	• 采样率：48k • 全频带立体声 • 编码码率：128kbps

AxTRTCScreenCaptureSourceType 的枚举值

枚举值	描述
AxTRTCScreenCaptureSourceTypeUnknown = -1	未定义
AxTRTCScreenCaptureSourceTypeWindow = 0	该分享目标是某一个应用的窗口
AxTRTCScreenCaptureSourceTypeScreen = 1	该分享目标是某一台显示器的屏幕
AxTRTCScreenCaptureSourceTypeCustom = 2	该分享目标是用户自定义的数据源

AxTRTCMediaDeviceType 的枚举值

枚举值	描述
AxTRTCMediaDeviceTypeUnknown = -1	未定义的设备类型
AxTRTCMediaDeviceTypeMic = 0	麦克风类型设备
AxTRTCMediaDeviceTypeSpeaker = 1	扬声器类型设备
AxTRTCMediaDeviceTypeCamera = 2	摄像头类型设

AxTRTCVideoResolutionMode 的枚举值

枚举值	描述
AxTRTCVideoResolutionModeLandscape = 0	横屏分辨率，例如：AxTRTCVideoResolution_640_360 + AxTRTCVideoResolutionModeLandscape = 640 × 360
AxTRTCVideoResolutionModePortrait = 1	竖屏分辨率，例如：AxTRTCVideoResolution_640_360 + AxTRTCVideoResolutionModePortrait = 360 × 640

主调接口的 JSON 格式

enterRoom 接口的入参

进房接口 enterRoom([in] INT sdkappid, [in] BSTR usersig, [in] BSTR room_id, [in] BSTR user_id, [in] enum AxTRTCRoleType role, [in] enum AxTRTCAppScene scene, [in] BSTR extend_info) 的入参 extend_info 参数以 JSON 形式传入，如下：

```
{
 "stream_id":"",
 "user_define_record_id":"",
 "private_map_key":"",
 "business_info":""
}
```

getScreenCaptureSources 接口的出参

枚举可分享的屏幕窗口接口 getScreenCaptureSources([in] UINT thumbnail_width, [in] UINT thumbnail_height, [in] UINT icon_width, [in] UINT icon_height, [out, retval] BSTR* screen_source_list) 的出参 `screen_source_list` 以 JSON 形式返回给用户，其格式如下：

```
[
 {
  "sourceId":1,
  "sourceName":"Screen1",
  "iconBGRA":{
     "buffer":"",
     "height":0,
     "length":0,
     "width":0
  },
  "thumbBGRA":{
     "buffer":"iVBORw0KGgoAAAANSU...base64",
     "height":0,
     "length":0,
     "width":0
  }
 },
...
]
```

getDevicesList 接口的出参

设备列表的获取接口

`getDevicesList([in] enum AxTRTCMediaDeviceType device_type, [out, retval] BSTR* device_list)` **的出参** `device_list` **以 JSON 形式返回给用户，其格式如下：**

```
[
 {
   "name":"Integrated Camera",
   "device_id":"\\\\?\\usb#vid_04f2&pid_b67c&mi_00#6&325e987&0&0000#
{65e8773d-8f56-11d0-a3b9-00a0c9223196}\\global"
 },
 ...
]
```

getCurrentDevice 接口的出参

获取当前设备信息接口

`getCurrentDevice([in] enum AxTRTCMediaDeviceType device_type, [out, retval] BSTR* device_info)` **的出参** `device_info` **以 JSON 形式返回给用户，其格式如下：**

```
{
  "name":"Integrated Camera",
  "device_id":"\\\\?\\usb#vid_04f2&pid_b67c&mi_00#6&325e987&0&0000#
{65e8773d-8f56-11d0-a3b9-00a0c9223196}\\global"
}
```

回调监听接口返回的 JSON 格式

onUserVoiceVolume 回调监听

用户音量回调

`onUserVoiceVolume([in] BSTR user_volumes, [in] INT user_volumes_count, [in] INT total_volume)` **中的** `user_volumes` **以 JSON 数组的形式返回给用户，其格式如下：**

```
[
 {
   "userId":"",    //说话者的 userId，如果 userId 为空则代表是当前用户自己。
   "volume": 0     //说话者的音量大小，取值范围[0 - 100]。
 },
 ...
]
```

错误码

最近更新时间：2022-10-10 11:06:34

错误码表

基础错误码

符号	值	含义
ERR_NULL	0	无错误

进房相关错误码

TRTCCloud.enterRoom() 在进房失败时会触发此类错误码，您可以通过回调函数 TRTCCloudDelegate.onEnterRoom() 和 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_ROOM_ENTER_FAIL	-3301	进入房间失败
ERR_ENTER_ROOM_PARAM_NULL	-3316	进房参数为空，请检查 TRTCCloud.enterRoom(): 接口调用是否传入有效的 param
ERR_SDK_APPID_INVALID	-3317	进房参数 sdkAppId 错误
ERR_ROOM_ID_INVALID	-3318	进房参数 roomId 错误
ERR_USER_ID_INVALID	-3319	进房参数 userID 不正确
ERR_USER_SIG_INVALID	-3320	进房参数 userSig 不正确
ERR_ROOM_REQUEST_ENTER_ROOM_TIMEOUT	-3308	请求进房超时，请检查网络
ERR_SERVER_INFO_PRIVILEGE_FLAG_ERROR	-100006	权限位校验失败，请检查 privateMapKey 是否正确
ERR_SERVER_INFO_SERVICE_SUSPENDED	-100013	服务不可用。请检查：套餐包剩余分钟数是否大于0，腾讯云账号是否欠费

ERR_SERVER_INFO_ECDH_GET_TINYID	-100018	userSig 校验失败，请检查 userSig 是否正确

退房相关错误码

TRTCCloud.exitRoom() 在退房失败时会触发此类错误码，您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_ROOM_REQUEST_QUIT_ROOM_TIMEOUT	-3325	请求退房超时

设备（摄像头、麦克风、扬声器）相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_CAMERA_START_FAIL	-1301	打开摄像头失败，例如在 Windows 或 Mac 设备，摄像头的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_CAMERA_NOT_AUTHORIZED	-1314	摄像头设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_CAMERA_SET_PARAM_FAIL	-1315	摄像头参数设置出错（参数不支持或其它）
ERR_CAMERA_OCCUPY	-1316	摄像头正在被占用中，可尝试打开其他摄像头
ERR_MIC_START_FAIL	-1302	打开麦克风失败，例如在 Windows 或 Mac 设备，麦克风的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_MIC_NOT_AUTHORIZED	-1317	麦克风设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_MIC_SET_PARAM_FAIL	-1318	麦克风设置参数失败
ERR_MIC_OCCUPY	-1319	麦克风正在被占用中，例如移动设备正在通话时，打开麦克风会失败
ERR_MIC_STOP_FAIL	-1320	停止麦克风失败

ERR_SPEAKER_START_FAIL	-1321	打开扬声器失败，例如在 Windows 或 Mac 设备，扬声器的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_SPEAKER_SET_PARAM_FAIL	-1322	扬声器设置参数失败
ERR_SPEAKER_STOP_FAIL	-1323	停止扬声器失败

屏幕分享相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_SCREEN_CAPTURE_START_FAIL	-1308	开始录屏失败，如果在移动设备出现，可能是权限被用户拒绝了，如果在 Windows 或 Mac 系统的设备出现，请检查录屏接口的参数是否符合要求
ERR_SCREEN_CAPTURE_UNSURPORT	-1309	录屏失败，在 Android 平台，需要5.0以上的系统，在 iOS 平台，需要11.0以上的系统
ERR_SERVER_CENTER_NO_PRIVILEDGE_PUSH_SUB_VIDEO	-102015	没有权限上行辅路
ERR_SERVER_CENTER_ANOTHER_USER_PUSH_SUB_VIDEO	-102016	其他用户正在上行辅路
ERR_SCREEN_CAPTURE_STOPPED	-70001	录屏被系统中止

编解码相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_VIDEO_ENCODE_FAIL	-13	视频帧编码失败，例如 iOS 设备切换到其他应用

	03	时，硬编码器可能被系统释放，再切换回来时，硬编码器重启前，可能会抛出
ERR_UNSUPPORTED_RESOLUTION	-1305	不支持的视频分辨率
ERR_AUDIO_ENCODE_FAIL	-1304	音频帧编码失败，例如传入自定义音频数据，SDK 无法处理
ERR_UNSUPPORTED_SAMPLERATE	-1306	不支持的音频采样率

自定义采集相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_PIXEL_FORMAT_UNSUPPORTED	-1327	设置的 pixel format 不支持
ERR_BUFFER_TYPE_UNSUPPORTED	-1328	设置的 buffer type 不支持

CDN 绑定和混流相关错误码

您可以通过回调函数 TRTCCloudDelegate.onStartPublishing() 和 TRTCCloudDelegate.onSetMixTranscodingConfig() 捕获相关通知。

符号	值	含义
ERR_PUBLISH_CDN_STREAM_REQUEST_TIME_OUT	-3321	旁路转推请求超时
ERR_CLOUD_MIX_TRANSCODING_REQUEST_TIME_OUT	-3322	云端混流请求超时
ERR_PUBLISH_CDN_STREAM_SERVER_FAILED	-3323	旁路转推回包异常
ERR_CLOUD_MIX_TRANSCODING_SERVER_FAILED	-3324	云端混流回包异常
ERR_ROOM_REQUEST_START_PUBLISHING_TIMEOUT	-3333	开始向腾讯云的直播 CDN 推流信令超时
ERR_ROOM_REQUEST_START_PUB	-33	开始向腾讯云的直播 CDN 推流信令异常

LISHING_ERROR	34	
ERR_ROOM_REQUEST_STOP_PUBLISHING_TIMEOUT	-3335	停止向腾讯云的直播 CDN 推流信令超时
ERR_ROOM_REQUEST_STOP_PUBLISHING_ERROR	-3336	停止向腾讯云的直播 CDN 推流信令异常

跨房连麦相关错误码

TRTCCloud.ConnectOtherRoom() 在跨房失败时会触发此类错误码，您可以通过回调函数 TRTCCloudDelegate.onConnectOtherRoom() 捕获相关通知。

符号	值	含义
ERR_ROOM_REQUEST_CONN_ROOM_TIMEOUT	-3326	请求连麦超时
ERR_ROOM_REQUEST_DISCONN_ROOM_TIMEOUT	-3327	请求退出连麦超时
ERR_ROOM_REQUEST_CONN_ROOM_INVALID_PARAM	-3328	无效参数
ERR_CONNECT_OTHER_ROOM_AS_AUDIENCE	-3330	当前是观众角色，不能请求或断开跨房连麦，需要先 switchRole() 到主播
ERR_SERVER_CENTER_CONN_ROOM_NOT_SUPPORT	-102031	不支持跨房间连麦
ERR_SERVER_CENTER_CONN_ROOM_REACH_MAX_NUM	-102032	达到跨房间连麦上限
ERR_SERVER_CENTER_CONN_ROOM_REACH_MAX_RETRY_TIMES	-102033	跨房间连麦重试次数耗尽
ERR_SERVER_CENTER_CONN_ROOM_REQ_TIMEOUT	-102034	跨房间连麦请求超时
ERR_SERVER_CENTER_CONN_ROOM_REQ	-102035	跨房间连麦请求格式错误
ERR_SERVER_CENTER_CONN_ROOM_NO_SIG	-102036	跨房间连麦无签名
ERR_SERVER_CENTER_CONN_ROOM_DECRYPT_SIG	-102037	跨房间连麦签名解密失败

ERR_SERVER_CENTER_CONN_ROOM_NO_KEY	-102038	未找到跨房间连麦签名解密密钥
ERR_SERVER_CENTER_CONN_ROOM_PARSE_SIG	-102039	跨房间连麦签名解析错误
ERR_SERVER_CENTER_CONN_ROOM_INVALID_SIG_TIME	-102040	跨房间连麦签名时间戳错误
ERR_SERVER_CENTER_CONN_ROOM_SIG_GROUPID	-102041	跨房间连麦签名内房间号不匹配
ERR_SERVER_CENTER_CONN_ROOM_NOT_CONNED	-102042	跨房间连麦签名内用户名不匹配
ERR_SERVER_CENTER_CONN_ROOM_USER_NOT_CONNED	-102043	本用户未发起连麦
ERR_SERVER_CENTER_CONN_ROOM_FAILED	-102044	跨房间连麦失败
ERR_SERVER_CENTER_CONN_ROOM_CANCEL_FAILED	-102045	取消跨房间连麦失败
ERR_SERVER_CENTER_CONN_ROOM_CONNED_ROOM_NOT_EXIST	-102046	被连麦房间不存在
ERR_SERVER_CENTER_CONN_ROOM_CONNED_REACH_MAX_ROOM	-102047	被连麦房间达到连麦上限
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_NOT_EXIST	-102048	被连麦用户不存在
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_DELETED	-102049	被连麦用户已被删除
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_FULL	-102050	被连麦用户达到资源上限
ERR_SERVER_CENTER_CONN_ROOM_INVALID_SEQ	-102051	连麦请求序号错乱

警告码表

警告码无需特别关注，您可以按需选择是否对当前用户进行相应的提示。

符号	值	含义

WARNING_HW_ENCODER_START_FAIL	1103	硬编码启动出现问题，自动切换到软编码
WARNING_VIDEO_ENCODER_SW_TO_HW	1107	当前 CPU 使用率太高，无法满足软件编码需求，自动切换到硬件编码
WARNING_INSUFFICIENT_CAPTURE_FPS	1108	摄像头采集帧率不足，部分自带美颜算法的 Android 手机上会出现
WARNING_SW_ENCODER_START_FAIL	1109	软编码启动失败
WARNING_REDUCE_CAPTURE_RESOLUTION	1110	摄像头采集分辨率被降低，以满足当前帧率和性能最优解。
WARNING_CAMERA_DEVICE_EMPTY	1111	没有检测到可用的摄像头设备
WARNING_CAMERA_NOT_AUTHORIZED	1112	用户未授权当前应用使用摄像头
WARNING_MICROPHONE_DEVICE_EMPTY	1201	没有检测到可用的麦克风设备
WARNING_SPEAKER_DEVICE_EMPTY	1202	没有检测到可用的扬声器设备
WARNING_MICROPHONE_NOT_AUTHORIZED	12	用户未授权当前应用使用麦克风

	03	
WARNING_MICROPHONE_DEVICE_ABNORMAL	1204	音频采集设备不可用（例如被占用）
WARNING_SPEAKER_DEVICE_ABNORMAL	1205	音频播放设备不可用（例如被占用）
WARNING_VIDEO_FRAME_DECODE_FAIL	2101	当前视频帧解码失败
WARNING_AUDIO_FRAME_DECODE_FAIL	2102	当前音频帧解码失败
WARNING_VIDEO_PLAY_LAG	2105	当前视频播放出现卡顿
WARNING_HW_DECODER_START_FAIL	2106	硬解启动失败，采用软解码
WARNING_VIDEO_DECODER_HW_TO_SW	2108	当前流硬解第一个 I 帧失败，SDK 自动切软解
WARNING_SW_DECODER_START_FAIL	2109	软解码器启动失败
WARNING_VIDEO_RENDER_FAIL	2110	视频渲染失败

WARNING_START_CAPTURE_IGNORED	4000	已经在采集，启动采集被忽略
WARNING_AUDIO_RECORDING_WRITE_FAIL	7001	音频录制写入文件失败
WARNING_ROOM_DISCONNECT	5101	网络断开连接
WARNING_IGNORE_UPSTREAM_FOR_AUDIENCE	6001	当前是观众角色，忽略上行音视频数据
WARNING_NET_BUSY	1101	网络状况不佳：上行带宽太小，上传数据受阻
WARNING_RTMP_SERVER_RECONNECT	1102	直播推流，网络断连，已启动自动重连（自动重连连续失败超过三次会放弃）
WARNING_LIVE_STREAM_SERVER_RECONNECT	2103	直播拉流，网络断连，已启动自动重连（自动重连连续失败超过三次会放弃）
WARNING_RECV_DATA_LAG	2104	网络来包不稳：可能是下行带宽不足，或由于主播端出流不均匀
WARNING_RTMP_DNS_FAIL	3001	直播，DNS 解析失败
WARNING_RTMP_SEVER_CONN_FAIL	30	直播，服务器连接失败

	02	
WARNING_RTMP_SHAKE_FAIL	30003	直播，与 RTMP 服务器握手失败
WARNING_RTMP_SERVER_BREAK_CONNECT	30004	直播，服务器主动断开
WARNING_RTMP_READ_WRITE_FAIL	30005	直播，RTMP 读/写失败，将会断开连接
WARNING_RTMP_WRITE_FAIL	30006	直播，RTMP 写失败（SDK 内部错误码，不会对外抛出）
WARNING_RTMP_READ_FAIL	30007	直播，RTMP 读失败（SDK 内部错误码，不会对外抛出）
WARNING_RTMP_NO_DATA	30008	直播，超过30s 没有数据发送，主动断开连接
WARNING_PLAY_LIVE_STREAM_INFO_CONNECT_FAIL	30009	直播，connect 服务器调用失败（SDK 内部错误码，不会对外抛出）
WARNING_NO_STEAM_SOURCE_FAIL	30100	直播，连接失败，该流地址无视频（SDK 内部错误码，不会对外抛出）
WARNING_ROOM_RECONNECT	51002	网络断连，已启动自动重连

WARNING_ROOM_NET_BUSY	5103	网络状况不佳：上行带宽太小，上传数据受阻

Web
API 概览

最近更新时间：2024-01-17 10:47:12

API 介绍

TRTC

> **注意**
> 本文适用于 5.x 版本的 TRTC Web SDK。若您使用 4.x 版本，可以参考 此处文档 。

TRTC 对象通过调用 TRTC.create() 方法创建，提供实时音视频的核心能力：

- 进入一个音视频房间，调用 trtc.enterRoom() 方法。
- 退出当前音视频房间，调用 trtc.exitRoom() 方法。
- 预览/发布本地视频，调用 trtc.startLocalVideo() 方法。
- 采集/发布本地音频，调用 trtc.startLocalAudio() 方法。
- 取消预览/发布本地视频，调用 trtc.stopLocalVideo() 方法。
- 取消采集/发布本地音频，调用 trtc.stopLocalAudio() 方法。
- 观看远端视频，调用 trtc.startRemoteVideo() 方法。
- 取消观看远端视频，调用 trtc.stopRemoteVideo() 方法。
- 静默/取消静默远端音频，调用 trtc.muteRemoteAudio() 方法。

静态方法

API	描述
create	创建一个 TRTC 对象，用于实现进房、预览、推流、拉流等功能。
setLogLevel	设置日志输出等级建议在开发测试阶段设置为 DEBUG 等级，该日志等级包含详细的提示信息。默认输出 INFO 日志等级，该日志等级包含 SDK 主要功能的日志信息。
isSupported	检测 TRTC Web SDK 是否支持当前浏览器
getCameraList	返回摄像头设备列表
getMicrophoneList	返回麦克风设备列表
getSpeakerList	返回扬声器设备列表，出于安全的考虑，在用户未授权摄像头或麦克风访问权限

	前，label 及 deviceId 字段可能都是空的。因此建议在用户授权访问后再调用该接口获取设备详情。
setCurrentSpeaker	设置当前音频播放的扬声器

成员方法

创建实例后调用。

API	描述
enterRoom	进入一个音视频通话房间（以下简称"进房"）
exitRoom	退出当前音视频通话房间
switchRole	切换用户角色，仅在 TRTC.TYPE.SCENE_LIVE 互动直播模式下生效
destroy	销毁 TRTC 实例
startLocalAudio	开启本地麦克风采集，并发布到当前的房间中
updateLocalAudio	更新本地麦克风配置
stopLocalAudio	停止本地麦克风的采集及发布
startLocalVideo	开启本地摄像头采集，在您指定的 HTMLElement 标签下播放摄像头画面，并将摄像头画面发布到当前所在房间中
updateLocalVideo	更新本地摄像头配置
stopLocalVideo	停止本地摄像头的采集、预览及发布
startScreenShare	开启屏幕分享
updateScreenShare	更新屏幕分享配置
stopScreenShare	停止屏幕分享
startRemoteVideo	播放远端视频
updateRemoteVideo	更新远端视频播放配置
stopRemoteVideo	用于停止远端视频播放
muteRemoteAudio	静音某个远端用户，并且不再拉取该用户的音频数据。仅对当前用户有效，房间内的其他用户依然可以听到被静音用户的声音。

版权所有：腾讯云计算（北京）有限责任公司

setRemoteAudioVolume	用于控制远端音频的播放音量
enableAudioVolumeEvaluation	开启或关闭音量大小回调
on	监听 TRTC 事件，详细事件列表请参见：TRTC.EVENT
off	取消事件监听
getVideoSnapshot	获取视频截图
getVideoTrack	获取视频轨道
getAudioTrack	获取音频轨道

教程

您可以参考如下教程实现各种功能：

功能	Sample Code 指引
基础音视频通话	指引链接
实现互动直播连麦	指引链接
切换摄像头和麦克风	指引链接
设置本地视频属性	指引链接
开关摄像头、麦克风	指引链接
屏幕分享	指引链接
音量大小检测	指引链接
自定义采集与自定义播放渲染	指引链接
房间内上行用户个数限制	指引链接
通话前环境与设备检测	指引链接
通话前的网络质量检测	指引链接
处理设备插拔	指引链接
云端混流和转推 CDN	指引链接

版权所有：腾讯云计算（北京）有限责任公司

开启水印	指引链接
开启虚拟背景	指引链接
开启 AI 降噪	指引链接
实现跨房连麦	指引链接
实时语音识别	指引链接

最佳实践

最佳实践	指引链接
多人视频通话	指引链接
自动播放处理建议	指引链接
应对防火墙受限	指引链接

> **说明：**
> 常见问题参见 Web 端相关 。

支持的平台

TRTC Web SDK 基于 WebRTC 实现，目前支持桌面端和移动端的主流浏览器，详细支持度参见下方表格。如果您的应用场景不在支持的表格里（例如 WebView），可以使用 TRTC.isSupported 接口 或者 打开 TRTC Web SDK 能力检测页面 检测当前环境是否支持 WebRTC 所有能力。

操作系统	浏览器类型	浏览器最低版本要求	SDK 版本要求	接收（播放）	发送（上麦）	屏幕分享
Windows	桌面版 Chrome 浏览器	56+	无要求	支持	支持	支持 Chrome72+ 版本
	桌面版 QQ 浏览器（极速内核）	10.4+	无要求	支持	支持	不支持
	桌面版 Firefox 浏览器	56+	v4.7.0+	支持	支持	支持 Firefox66+ 版本
	桌面版 Edge 浏览器	80+	v4.7.0+	支持	支持	支持

	桌面版搜狗浏览器（高速模式）	11+	v4.7.0+	支持	支持	支持
	桌面版搜狗浏览器（兼容模式）	不支持	不支持	不支持	不支持	不支持
	桌面版 Opera 浏览器	46+	v4.7.0+	支持	支持	支持 Opera60+ 版本
	桌面版 360 安全浏览器（极速模式）	13+	v4.7.0+	支持	支持	支持
	桌面版 360 安全浏览器（兼容模式）	不支持	不支持	不支持	不支持	不支持
	桌面版微信内嵌浏览器	无明确信息	无要求	支持	不支持	不支持
	桌面版企业微信内嵌浏览器	无明确信息	无要求	支持	不支持	不支持
Mac OS	桌面版 Safari 浏览器	11+	无要求	支持	支持	支持 Safari13+ 版本
	桌面版 Chrome 浏览器	56+	无要求	支持	支持	支持 Chrome72+ 版本
	桌面版 Firefox 浏览器	56+	v4.7.0+	支持	支持	支持 Firefox66+ 版本（注意[3]）
	桌面版 Edge 浏览器	80+	v4.7.0+	支持	支持	支持
	桌面版 Opera 浏览器	46+	v4.7.0+	支持	支持	支持 Opera60+ 版本
	桌面版微信内嵌浏览器	无明确信息	无要求	支持	不支持	不支持
	桌面版企业微信内嵌浏览器	无明确信息	无要求	支持	不支持	不支持
Android	微信内嵌浏览器（TBS 内核）	无明确信息	无要求	支持	支持	不支持
	微信内嵌浏览器	无明确	无要	支持	支持	不支持

	（XWEB 内核）	信息	求			
	企业微信内嵌浏览器	无明确信息	无要求	支持	支持	不支持
	移动版 Chrome 浏览器	无明确信息	无要求	支持	支持	不支持
	移动版 QQ 浏览器	13+	v4.1 5.0+	支持	支持	不支持
	移动版 UC 浏览器	不支持	不支持	不支持	不支持	不支持
	夸克浏览器	不支持	不支持	不支持	不支持	不支持
iOS 12.1.4 +	微信内嵌浏览器	无明确信息	无要求	支持	不支持	不支持
iOS 14.3+	微信内嵌浏览器	6.5+（微信版本）	无要求	支持	支持	不支持
iOS	企业微信内嵌浏览器	无明确信息	无要求	支持	不支持	不支持
iOS 11.1.2 +	移动版 Safari 浏览器	11+	无要求	支持	支持	不支持
iOS 12.1.4 +	移动版 Chrome 浏览器	无明确信息	无要求	支持	不支持	不支持
iOS 14.3+	移动版 Chrome 浏览器	无明确信息	无要求	支持	支持	不支持

注意

1. 由于 H.264 版权限制，华为 Chrome 88 以下版本以及部分厂商的 Android 设备，不具备 H264 编码能力（即无法推流）。如果您希望在这些设备的浏览器中，使用 TRTC Web SDK 推流，请提交 Web SDK 用户能力支持申请 开通 VP8 编解码能力。
2. 建议您及时将 TRTC Web SDK 更新至最新版本，以便获得更好的产品稳定性及在线支持。版本升级注意事项请参见：升级指引 。

URL 域名协议限制

由于浏览器安全策略的限制，使用 WebRTC 能力对页面的访问协议有严格的要求，请参照以下表格进行开发和部署应用。

应用场景	协议	接收（播放）	发送（上麦）	屏幕分享	备注
生产环境	HTTPS 协议	支持	支持	支持	**推荐**
生产环境	HTTP 协议	支持	不支持	不支持	-
本地开发环境	http://localhost	支持	支持	支持	**推荐**
本地开发环境	http://127.0.0.1	支持	支持	支持	-
本地开发环境	http://[本机IP]	支持	不支持	不支持	-
本地开发环境	file:///	支持	支持	支持	-

联系我们

若您遇到任何问题，请随时 联系我们 。

错误码

最近更新时间：2023-08-16 16:41:03

> **注意：**
> 本文适用于 5.x.x 版本的 TRTC Web SDK，详情参考：ERROR_CODE 。
> 若您使用 4.x.x 版本，请参考 此处文档 。

TRTC Web SDK 定义了 8 种错误码类型，通过 RtcError 对象来获取 errorCode 并做相应的处理。

错误码定义

Key	code	描述
INVALID_PARAMETER	5000	说明：调用接口时传入了不满足 API 要求的参数。 处理建议：请检查传入参数是否符合 API 的规范，例如参数类型是否正确。
INVALID_OPERATION	5100	说明：调用接口时，不满足 API 的前提要求。 处理建议：请根据对应 API 文档检查调用逻辑是否符合 API 的前提要求。 例如：1.未进房成功就进行切换角色2.播放的远端用户和流不存在。
ENV_NOT_SUPPORTED	5200	说明：当前环境不支持该功能，表明当前浏览器不支持调用对应 API 处理建议：通常使用 TRTC.isSupported 可感知当前浏览器支持哪些能力。如果浏览器不支持，需要引导用户使用支持该能力的浏览器，参考：检测浏览器支持性 。
DEVICE_ERROR	5300	说明：获取设备或者采集音视频出现异常 以下接口出现异常时会抛出该错误码：startLocalVideo、updateLocalVideo、startLocalAudio、updateLocalAudio、startScreenShare、updateScreenShare 处理建议：引导用户检查设备是否有摄像头及麦克风、系统是否给浏览器授权以及浏览器是否给页面授权。建议增加进房前的设备检测流程，确认麦克风和摄像头是否存在，并且能正常采集，再进行下一步通话操作。通常经过设备检查后都能避免该异常。 实现方式请参考：通话前环境与设备检测 。 如果需要区分更详细的异常类别，可以参考 DEVICE_ERROR 。
SERVER_ERROR	5400	说明：收到服务端返回的异常数据时抛出该错误码 以下接口出现异常时会抛出该错误码：enterRoom、startLocalVideo、startLocalAudio、startScreenShare、startRemoteVideo、switchRole 处理建议：服务端异常通常在开发阶段处理，常见的异常有：传入的

		userSig 过期，腾讯云账号欠费，未开通TRTC服务等，服务端返回异常数据有以下原因。
OPERATION_FAILED	5500	说明：在满足 API 调用要求的情况下，SDK 经过多次重试仍然无法解决的异常，通常是由于浏览器、网络的问题造成。 以下接口出现异常时会抛出该错误码：enterRoom、startLocalVideo、startLocalAudio、startScreenShare、startRemoteVideo、switchRole 处理建议： • 确认通信必需的域名和端口是否满足您的网络环境要求，参考文档 应对防火墙限制及设置代理 。 • 其他问题需要联系工程师处理，请 联系我们 。
OPERATION_ABORT	5998	说明：中止 API 执行时抛出该错误码。在不满足 API 生命周期的调用或重复调用时 API 会中止执行，避免无意义的操作。 例如：连续调用 enterRoom，startLocalXxx等接口，在没有进房就调用退房。 以下接口出现异常时会抛出该错误码：enterRoom、startLocalVideo、startLocalAudio、startScreenShare、startRemoteVideo、switchRole 处理建议：捕获并识别该错误码，然后在业务逻辑规避不必要的调用，或者也可以不做任何处理，因为 SDK 做了无副作用处理，您只需在 catch 时识别该错误码并忽略。
UNKNOWN_ERROR	5999	说明：未知错误或者未被定义的错误。 处理建议：联系我们 。

Electron API 概览

最近更新时间：2024-03-08 17:58:22

TRTCCloud @ TXLiteAVSDK

腾讯云视频通话功能的主要接口类。

- **主要文档地址：** TRTC Electron SDK
- **示例代码地址：** TRTC Electron Demo

创建 TRTC 对象

```
const TRTCCloud = require('trtc-electron-sdk').default;
// import TRTCCloud from 'trtc-electron-sdk';
const rtcCloud = TRTCCloud.getTRTCShareInstance();
```

从v7.9.348起，TRTC Electron SDK 增加了 trtc.d.ts 文件，方便使用 typescript 的开发者。

```
import TRTCCloud from 'trtc-electron-sdk';

const rtcCloud: TRTCCloud = TRTCCloud.getTRTCShareInstance();
// 获取 SDK 版本号
const version: string = rtcCloud.getSDKVersion();
```

注册事件监听函数

```
rtcCloud.on('onError', (errcode, errmsg) => {
  console.info('trtc_demo: onError :' + errcode + " msg" + errmsg);
});
rtcCloud.on('onEnterRoom', (elapsed) => {
  console.info('trtc_demo: onEnterRoom elapsed:' + elapsed);
});
rtcCloud.on('onExitRoom', (reason) => {
  console.info('onExitRoom: userenter reason:' + reason);
});
```

创建与销毁 TRTCCloud 单例

API	描述

getTRTCShareInstance	创建 TRTCCloud 主实例（单例对象）。
destroyTRTCShareInstance	释放 TRTCCloud 主实例（单例对象）并清理资源。
createSubCloud	创建子实例。注意：只有主实例才能创建子实例，子实例不能创建子实例。

房间相关接口函数

API	描述
enterRoom	进入房间，若房间不存在，系统将自动创建一个新房间。
exitRoom	退出房间。
switchRoom	切换房间。
switchRole	切换角色，仅适用于直播场景（TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom）。
connectOtherRoom	请求跨房连麦（主播跨房 PK）。
disconnectOtherRoom	关闭跨房连麦（主播跨房 PK）。
setDefaultStreamRecvMode	设置音视频数据接收模式（需要在进房前设置才能生效）。

CDN 相关接口函数

API	描述
startPublishing	开始向腾讯云的直播 CDN 推流。
stopPublishing	停止向腾讯云的直播 CDN 推流。
startPublishCDNStream	开始向非腾讯云的直播 CDN 转推。
stopPublishCDNStream	停止向非腾讯云的直播 CDN 推流。
setMixTranscodingConfig	设置云端的混流转码参数。

视频相关接口函数

版权所有：腾讯云计算（北京）有限责任公司

API	描述
startLocalPreview	启动本地摄像头采集和预览。
stopLocalPreview	停止本地摄像头采集和预览。
updateLocalView	修改本地摄像头预览的 HTML 元素。
setCameraCaptureParams	设置摄像头采集偏好。
muteLocalVideo	是否屏蔽自己的视频画面。
setVideoMuteImage	设置本地画面被暂停期间的替代图片。
startRemoteView	开始显示远端视频画面。
stopRemoteView	停止显示远端视频画面，同时不再拉取该远端用户的视频数据流。
updateRemoteView	修改远端视频渲染的 HTML 元素。
stopAllRemoteView	停止显示所有远端视频画面，同时不再拉取远端用户的视频数据流。
muteRemoteVideoStream	暂停接收指定的远端视频流。
muteAllRemoteVideoStreams	停止接收所有远端视频流。
setVideoEncoderParam	设置视频编码器相关参数。
setNetworkQosParam	设置网络流控相关参数。
setLocalRenderParams	设置本地图像（主流）的渲染参数。
setLocalViewFillMode	废弃接口：设置本地图像的渲染模式。
setRemoteRenderParams	设置远端图像的渲染参数。
setRemoteViewFillMode	废弃接口：设置远端图像的渲染模式。
setLocalViewRotation	废弃接口：设置本地图像的顺时针旋转角度。
setRemoteViewRotation	废弃接口：设置远端图像的顺时针旋转角度。

版权所有：腾讯云计算（北京）有限责任公司

setVideoEncoderRotation	设置视频编码输出的（即远端用户观看到的以及服务器录制下来的）画面方向。
setVideoEncoderMirror	设置编码器输出的画面镜像模式。
enableSmallVideoStream	开启大小画面双路编码模式。
setRemoteVideoStreamType	选定观看指定 userId 的大画面或小画面。
snapshotVideo	视频画面截图。
startLocalRecording	开启本地媒体录制。
stopLocalRecording	停止本地媒体录制。

音频相关接口函数

API	描述
startLocalAudio	开启本地音频的采集和上行。
stopLocalAudio	关闭本地音频的采集和上行。
muteLocalAudio	静音本地的音频。
muteRemoteAudio	静音某一个用户的声音，同时不再拉取该远端用户的音频数据流。
muteAllRemoteAudio	静音所有用户的声音，同时不再拉取远端用户的音频数据流。
setAudioCaptureVolume	设置 SDK 采集音量。
getAudioCaptureVolume	获取 SDK 采集音量。
setAudioPlayoutVolume	设置 SDK 播放音量。
getAudioPlayoutVolume	获取 SDK 播放音量。
enableAudioVolumeEvaluation	启用或关闭音量大小提示。
startAudioRecording	开始录音。

stopAudioRecording	停止录音。
setAudioQuality	废弃接口：设置音频质量。
setRemoteAudioVolume	设置远程用户播放音量。
setRemoteAudioParallelParams	设置远端音频流智能并发播放策略。

摄像头相关接口函数

API	描述
getCameraDevicesList	获取摄像头设备列表。
setCurrentCameraDevice	设置要使用的摄像头。
getCurrentCameraDevice	获取当前使用的摄像头。

音频设备相关接口函数

API	描述
getMicDevicesList	获取麦克风设备列表。
getCurrentMicDevice	获取当前选择的麦克风。
setCurrentMicDevice	设置要使用的麦克风。
getCurrentMicDeviceVolume	获取系统当前麦克风设备音量。
setCurrentMicDeviceVolume	设置系统当前麦克风设备的音量。
setCurrentMicDeviceMute	设置系统当前麦克风设备的静音状态。
getCurrentMicDeviceMute	获取系统当前麦克风设备是否静音。
getSpeakerDevicesList	获取扬声器设备列表。

getCurrentSpeakerDevice	获取当前的扬声器设备。
setCurrentSpeakerDevice	设置要使用的扬声器。
getCurrentSpeakerVolume	获取系统当前扬声器设备音量。
setCurrentSpeakerVolume	设置系统当前扬声器设备音量。
setCurrentSpeakerDeviceMute	设置系统当前扬声器设备的静音状态。
getCurrentSpeakerDeviceMute	获取系统当前扬声器设备是否静音。
enableFollowingDefaultAudioDevice	设置 SDK 使用的音频设备自动跟随系统默认设备。

美颜相关接口函数

API	描述
setBeautyStyle	设置美颜、美白、红润效果级别。
setWaterMark	设置水印。

辅流相关接口函数

API	描述
startRemoteSubStreamView	废弃接口：开始渲染远端用户的辅流（屏幕分享）画面。
stopRemoteSubStreamView	废弃接口：停止渲染远端用户的辅流（屏幕分享）画面。
setRemoteSubStreamViewFillMode	废弃接口：设置辅流（屏幕分享）画面的渲染模式。
setRemoteSubStreamViewRotation	废弃接口：设置辅流（屏幕分享）画面的顺时针旋转角度。
getScreenCaptureSources	枚举可共享的窗口列表。

selectScreenCaptureTarget	设置屏幕共享参数，该方法在屏幕共享过程中也可以调用。
startScreenCapture	启动屏幕分享。
pauseScreenCapture	暂停屏幕分享。
resumeScreenCapture	恢复屏幕分享。
stopScreenCapture	停止屏幕分享。
setSubStreamEncoderParam	设置辅流（屏幕分享）的编码器参数。
setSubStreamMixVolume	设置辅流（屏幕分享）的混音音量大小。
addExcludedShareWindow	将指定窗口加入屏幕分享的排除列表中，加入排除列表中的窗口不会被分享出去。
removeExcludedShareWindow	将指定窗口从屏幕分享的排除列表中移除。
removeAllExcludedShareWindow	将所有窗口从屏幕分享的排除列表中移除。
addIncludedShareWindow	将指定窗口加入屏幕分享的包含列表中。
removeIncludedShareWindow	将指定窗口从屏幕分享的包含列表中移除。
removeAllIncludedShareWindow	将全部窗口从屏幕分享的包含列表中移除。

自定义采集

API	描述
enableCustomAudioCapture	启用音频自定义采集模式。
sendCustomAudioData	向 SDK 投送自己采集的音频数据。
enableMixExternalAudioFrame	启用/关闭自定义音轨。
mixExternalAudioFram	向 SDK 混入自定义音轨。

版权所有：腾讯云计算（北京）有限责任公司

e	
setMixExternalAudioVolume	设置推流时混入外部音频的推流音量和播放音量。
generateCustomPTS	生成自定义采集时的时间戳。

自定义消息发送

API	描述
sendCustomCmdMsg	发送自定义消息给房间内所有用户。
sendSEIMsg	将小数据量的自定义数据嵌入视频帧中。

背景混音相关接口函数

API	描述
playBGM	废弃接口：启动播放背景音乐。
stopBGM	废弃接口：停止播放背景音乐。
pauseBGM	废弃接口：暂停播放背景音乐。
resumeBGM	废弃接口：继续播放背景音乐。
getBGMDuration	废弃接口：获取背景音乐文件总时长，单位毫秒。
setBGMPosition	废弃接口：设置背景音乐播放进度。
setBGMVolume	废弃接口：设置背景音乐播放音量的大小。
setBGMPlayoutVolume	废弃接口：设置背景音乐本地播放音量的大小。
setBGMPublishVolume	废弃接口：设置背景音乐远端播放音量的大小。
startSystemAudioLoopback	打开系统声音采集。
stopSystemAudioLoopback	关闭系统声音采集。
setSystemAudioLoopbackVolume	设置系统声音采集的音量。
setMusicObserver	设置背景音乐的事件回调监听。

版权所有：腾讯云计算（北京）有限责任公司

startPlayMusic	启动播放背景音乐。
stopPlayMusic	停止播放背景音乐。
pausePlayMusic	暂停播放背景音乐。
resumePlayMusic	恢复播放背景音乐。
getMusicDurationInMS	获取背景音乐文件总时长，单位毫秒。
seekMusicToPosInTime	设置背景音乐播放进度。
setAllMusicVolume	设置背景音乐的音量大小，播放背景音乐混音时使用，用来控制背景音音量大小。
setMusicPlayoutVolume	设置背景音乐本地播放音量的大小。
setMusicPublishVolume	设置背景音乐远端播放音量的大小。
setVoiceReverbType	设置人声混响效果。

音效相关接口函数

API	描述
playAudioEffect	废弃接口：播放音效。
setAudioEffectVolume	废弃接口：设置音效音量。
stopAudioEffect	废弃接口：停止音效。
stopAllAudioEffects	废弃接口：停止所有音效。
setAllAudioEffectsVolume	废弃接口：设置所有音效的音量。
pauseAudioEffect	废弃接口：暂停音效。
resumeAudioEffect	废弃接口：恢复音效。

设备和网络测试

API	描述
startSpeedTest	开始进行网络测速（视频通话期间请勿测试，以免影响通话质量）。

stopSpeedTest	停止网络测速。
startCameraDeviceTest	开始进行摄像头测试。
stopCameraDeviceTest	停止摄像头测试。
startMicDeviceTest	开始进行麦克风测试。
stopMicDeviceTest	停止麦克风测试。
startSpeakerDeviceTest	开始进行扬声器测试。
stopSpeakerDeviceTest	停止扬声器测试。

LOG 相关接口函数

API	描述
getSDKVersion	获取 SDK 版本信息。
setLogLevel	设置 Log 输出级别。
setConsoleEnabled	启用或禁用控制台日志打印。
setLogCompressEnabled	启用或禁用 Log 的本地压缩。
setLogDirPath	设置日志保存路径。
setLogCallback	设置日志回调。
callExperimentalAPI	调用实验性 API 接口。

弃用接口函数

API	描述
setMicVolumeOnMixing	从 v6.9 版本开始废弃。

TRTCCallback @ TXLiteAVSDK

腾讯云视频通话功能的回调接口类。

错误事件和警告事件

API	描述
onError	错误回调：SDK 不可恢复的错误，必须对此事件注册回调函数，并分情况给用户适当的界面提示。
onWarning	警告回调：用于告知您一些非严重性问题，例如出现了卡顿或可恢复的解码失败。

房间事件回调

API	描述
onEnterRoom	已加入房间的回调。
onExitRoom	退出房间的事件回调。
onSwitchRole	切换角色的事件回调。
onConnectOtherRoom	请求跨房连麦（主播跨房 PK）的结果回调。
onDisconnectOtherRoom	关闭跨房连麦（主播跨房 PK）的结果回调。
onSwitchRoom	切换房间。

成员事件回调

API	描述
onRemoteUserEnterRoom	有用户加入当前房间。
onRemoteUserLeaveRoom	有用户离开当前房间。
onUserVideoAvailable	用户是否开启摄像头视频。
onUserSubStreamAvailable	用户是否开启屏幕分享。
onUserAudioAvailable	用户是否开启音频上行。
onFirstVideoFrame	开始渲染本地或远程用户的首帧画面。

版权所有：腾讯云计算（北京）有限责任公司

onFirstAudioFrame	开始播放远程用户的首帧音频（本地声音暂不通知）。
onSendFirstLocalVideoFrame	首帧本地视频数据已经被送出。
onSendFirstLocalAudioFrame	首帧本地音频数据已经被送出。
onUserEnter	废弃接口：有主播加入当前房间。
onUserExit	废弃接口： 有主播离开当前房间。

统计和质量回调

API	描述
onNetworkQuality	网络质量：该回调每2秒触发一次，统计当前网络的上行和下行质量。
onStatistics	技术指标统计回调。

服务器事件回调

API	描述
onConnectionLost	SDK 与服务器的连接断开。
onTryToReconnect	SDK 尝试重新连接到服务器。
onConnectionRecovery	SDK 与服务器的连接恢复。
onSpeedTest	废弃接口：服务器测速的回调，SDK 对多个服务器 IP 进行测速，每个 IP 的测速结果通过这个回调通知。
onSpeedTestResult	网速测试的结果回调。

硬件设备事件回调

API	描述
onCameraDidReady	摄像头准备就绪。
onMicDidReady	麦克风准备就绪。
onUserVoiceVolume	用于提示音量大小的回调，包括每个 userId 的音量和远端总音量；本地用户 userid 为"。

onDeviceChange	本地设备通断回调。
onTestMicVolume	麦克风测试音量回调。
onTestSpeakerVolume	扬声器测试音量回调。
onAudioDeviceCaptureVolumeChanged	当前音频采集设备音量变化回调。
onAudioDevicePlayoutVolumeChanged	当前音频播放设备音量变化回调。

自定义消息的接收回调

API	描述
onRecvCustomCmdMsg	收到自定义消息回调。
onMissCustomCmdMsg	自定义消息丢失回调。
onRecvSEIMsg	收到 SEI 消息的回调。

CDN 旁路转推回调

API	描述
onStartPublishing	开始向腾讯云的直播 CDN 推流的回调，对应于 TRTCCloud 中的 startPublishing() 接口。
onStopPublishing	停止向腾讯云的直播 CDN 推流的回调，对应于 TRTCCloud 中的 stopPublishing() 接口。
onStartPublishCDNStream	启动旁路推流到 CDN 完成的回调。
onStopPublishCDNStream	停止旁路推流到 CDN 完成的回调。
onSetMixTranscodingConfig	设置云端的混流转码参数的回调，对应于 TRTCCloud 中的 setMixTranscodingConfig() 接口。

系统音量采集回调

API	描述

onSystemAudioLoopbackError	系统音量采集状态的回调（仅在 Mac 上有效）。

音效回调

API	描述
onAudioEffectFinished	废弃接口：播放音效结束回调。

屏幕分享回调

API	描述
onScreenCaptureCovered	当屏幕分享窗口被遮挡无法正常捕获时，SDK 会通过此回调通知，可在此回调里通知用户移开遮挡窗口。
onScreenCaptureStarted	当屏幕分享开始时，SDK 会通过此回调通知。
onScreenCapturePaused	当屏幕分享暂停时，SDK 会通过此回调通知。
onScreenCaptureResumed	当屏幕分享恢复时，SDK 会通过此回调通知。
onScreenCaptureStopped	当屏幕分享停止时，SDK 会通过此回调通知。

截图回调

API	描述
onSnapshotComplete	截图完成时，SDK 会通过此回调通知。

背景混音事件回调

API	描述
onPlayBGMBegin	废弃接口：开始播放背景音乐。
onPlayBGMProgress	废弃接口：播放背景音乐的进度。
onPlayBGMComplete	废弃接口：播放背景音乐结束。

关键类型定义

关键类型

类名	描述
TRTCParams	进房相关参数。
TRTCVideoEncParam	视频编码参数。
TRTCNetworkQosParam	网络流控相关参数。
TRTCQualityInfo	视频质量。
TRTCVolumeInfo	音量大小。
TRTCSpeedTestResult	网络测速结果。
TRTCMixUser	云端混流中每一路子画面的位置信息。
TRTCTranscodingConfig	云端混流（转码）配置。
TRTCPublishCDNParam	CDN 旁路推流参数。
TRTCAudioRecordingParams	录音参数。
TRTCLocalStatistics	自己本地的音视频统计信息。
TRTCRemoteStatistics	远端成员的音视频统计信息。
TRTCStatistics	统计数据。

枚举值

枚举	描述
TRTCVideoResolution	视频分辨率。
TRTCVideoResolutionMode	视频分辨率模式。
TRTCVideoStreamType	视频流类型。
TRTCQuality	画质级别。

版权所有：腾讯云计算（北京）有限责任公司

TRTCVideoFillMode	视频画面填充模式。
TRTCBeautyStyle	美颜（磨皮）算法。
TRTCAppScene	应用场景。
TRTCRoleType	角色，仅适用于直播场景（TRTCAppSceneLIVE）。
TRTCQosControlMode	流控模式。
TRTCVideoQosPreference	画质偏好。
TRTCDeviceState	设备操作。
TRTCDeviceType	设备类型。
TRTCWaterMarkSrcType	水印图片的源类型。
TRTCTranscodingConfigMode	混流参数配置模式。

版权所有：腾讯云计算（北京）有限责任公司

错误码

最近更新时间：2023-09-19 14:58:13

错误码表

基础错误码

符号	值	含义
ERR_NULL	0	无错误

进房相关错误码

`TRTCCloud.enterRoom()` 在进房失败时会触发此类错误码，您可以通过回调函数 `TRTCCloudDelegate.onEnterRoom()` 和 `TRTCCloudDelegate.OnError()` 捕获相关通知。

符号	值	含义
ERR_ROOM_ENTER_FAIL	-3301	进入房间失败
ERR_ENTER_ROOM_PARAM_NULL	-3316	进房参数为空，请检查 TRTCCloud.enterRoom(): 接口调用是否传入有效的 param
ERR_SDK_APPID_INVALID	-3317	进房参数 sdkAppId 错误
ERR_ROOM_ID_INVALID	-3318	进房参数 roomId 错误
ERR_USER_ID_INVALID	-3319	进房参数 userID 不正确
ERR_USER_SIG_INVALID	-3320	进房参数 userSig 不正确
ERR_ROOM_REQUEST_ENTER_ROOM_TIMEOUT	-3308	请求进房超时，请检查网络
ERR_SERVER_INFO_PRIVILEGE_FLAG_ERROR	-100006	权限位校验失败，请检查 privateMapKey 是否正确
ERR_SERVER_INFO_SERVICE_SUSPENDED	-100013	服务不可用。请检查：套餐包剩余分钟数是否大于0，腾讯云账号是否欠费

版权所有：腾讯云计算（北京）有限责任公司

ERR_SERVER_INFO_ECDH_GET_TINYID	-100018	userSig 校验失败，请检查 userSig 是否正确

退房相关错误码

TRTCCloud.exitRoom() 在退房失败时会触发此类错误码，您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_ROOM_REQUEST_QUIT_ROOM_TIMEOUT	-3325	请求退房超时

设备（摄像头、麦克风、扬声器）相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_CAMERA_START_FAIL	-1301	打开摄像头失败，例如在 Windows 或 Mac 设备，摄像头的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_CAMERA_NOT_AUTHORIZED	-1314	摄像头设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_CAMERA_SET_PARAM_FAIL	-1315	摄像头参数设置出错（参数不支持或其它）
ERR_CAMERA_OCCUPY	-1316	摄像头正在被占用中，可尝试打开其他摄像头
ERR_MIC_START_FAIL	-1302	打开麦克风失败，例如在 Windows 或 Mac 设备，麦克风的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_MIC_NOT_AUTHORIZED	-1317	麦克风设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_MIC_SET_PARAM_FAIL	-1318	麦克风设置参数失败
ERR_MIC_OCCUPY	-1319	麦克风正在被占用中，例如移动设备正在通话时，打开麦克风会失败
ERR_MIC_STOP_FAIL	-1320	停止麦克风失败
ERR_SPEAKER_START_FAIL	-1321	打开扬声器失败，例如在 Windows 或 Mac 设

		备，扬声器的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_SPEAKER_SET_PARAM_FAIL	-1322	扬声器设置参数失败
ERR_SPEAKER_STOP_FAIL	-1323	停止扬声器失败

屏幕分享相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_SCREEN_CAPTURE_START_FAIL	-1308	开始录屏失败，如果在移动设备出现，可能是权限被用户拒绝了，如果在 Windows 或 Mac 系统的设备出现，请检查录屏接口的参数是否符合要求
ERR_SCREEN_CAPTURE_UNSURPORT	-1309	录屏失败，在 Android 平台，需要5.0以上的系统，在 iOS 平台，需要11.0以上的系统
ERR_SERVER_CENTER_NO_PRIVILEDGE_PUSH_SUB_VIDEO	-102015	没有权限上行辅路
ERR_SERVER_CENTER_ANOTHER_USER_PUSH_SUB_VIDEO	-102016	其他用户正在上行辅路
ERR_SCREEN_CAPTURE_STOPPED	-7001	录屏被系统中止

编解码相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_VIDEO_ENCODE_FAIL	-1303	视频帧编码失败，例如 iOS 设备切换到其他应用时，硬编码器可能被系统释放，再切换回来时，硬编码器重启前，可能会抛出
ERR_UNSUPPORTED_RESOLUTION	-1305	不支持的视频分辨率
ERR_AUDIO_ENCODE_FAIL	-130	音频帧编码失败，例如传入自定义音频数据，SDK

	4	无法处理
ERR_UNSUPPORTED_SAMPLERATE	-1306	不支持的音频采样率

自定义采集相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_PIXEL_FORMAT_UNSUPPORTED	-1327	设置的 pixel format 不支持
ERR_BUFFER_TYPE_UNSUPPORTED	-1328	设置的 buffer type 不支持

CDN 绑定和混流相关错误码

您可以通过回调函数 TRTCCloudDelegate.onStartPublishing() 和 TRTCCloudDelegate.onSetMixTranscodingConfig() 捕获相关通知。

符号	值	含义
ERR_PUBLISH_CDN_STREAM_REQUEST_TIME_OUT	-3321	旁路转推请求超时
ERR_CLOUD_MIX_TRANSCODING_REQUEST_TIME_OUT	-3322	云端混流请求超时
ERR_PUBLISH_CDN_STREAM_SERVER_FAILED	-3323	旁路转推回包异常
ERR_CLOUD_MIX_TRANSCODING_SERVER_FAILED	-3324	云端混流回包异常
ERR_ROOM_REQUEST_START_PUBLISHING_TIMEOUT	-3333	开始向腾讯云的直播 CDN 推流信令超时
ERR_ROOM_REQUEST_START_PUBLISHING_ERROR	-3334	开始向腾讯云的直播 CDN 推流信令异常
ERR_ROOM_REQUEST_STOP_PUBLISHING_TIMEOUT	-3335	停止向腾讯云的直播 CDN 推流信令超时
ERR_ROOM_REQUEST_STOP_PUBLISHING_ERROR	-3336	停止向腾讯云的直播 CDN 推流信令异常

跨房连麦相关错误码

`TRTCCloud.ConnectOtherRoom()` 在跨房失败时会触发此类错误码，您可以通过回调函数 `TRTCCloudDelegate.onConnectOtherRoom()` 捕获相关通知。

符号	值	含义
ERR_ROOM_REQUEST_CONN_ROOM_TIMEOUT	-3326	请求连麦超时
ERR_ROOM_REQUEST_DISCONN_ROOM_TIMEOUT	-3327	请求退出连麦超时
ERR_ROOM_REQUEST_CONN_ROOM_INVALID_PARAM	-3328	无效参数
ERR_CONNECT_OTHER_ROOM_AS_AUDIENCE	-3330	当前是观众角色，不能请求或断开跨房连麦，需要先 switchRole() 到主播
ERR_SERVER_CENTER_CONN_ROOM_NOT_SUPPORT	-102031	不支持跨房间连麦
ERR_SERVER_CENTER_CONN_ROOM_REACH_MAX_NUM	-102032	达到跨房间连麦上限
ERR_SERVER_CENTER_CONN_ROOM_REACH_MAX_RETRY_TIMES	-102033	跨房间连麦重试次数耗尽
ERR_SERVER_CENTER_CONN_ROOM_REQ_TIMEOUT	-102034	跨房间连麦请求超时
ERR_SERVER_CENTER_CONN_ROOM_REQ	-102035	跨房间连麦请求格式错误
ERR_SERVER_CENTER_CONN_ROOM_NO_SIG	-102036	跨房间连麦无签名
ERR_SERVER_CENTER_CONN_ROOM_DECRYPT_SIG	-102037	跨房间连麦签名解密失败
ERR_SERVER_CENTER_CONN_ROOM_NO_KEY	-102038	未找到跨房间连麦签名解密密钥
ERR_SERVER_CENTER_CONN_ROOM_PARSE_SIG	-102039	跨房间连麦签名解析错误
ERR_SERVER_CENTER_CONN_ROOM_IN	-102	跨房间连麦签名时间戳错误

VALID_SIG_TIME	040	
ERR_SERVER_CENTER_CONN_ROOM_SIG_GROUPID	-102041	跨房间连麦签名内房间号不匹配
ERR_SERVER_CENTER_CONN_ROOM_NOT_CONNED	-102042	跨房间连麦签名内用户名不匹配
ERR_SERVER_CENTER_CONN_ROOM_USER_NOT_CONNED	-102043	本用户未发起连麦
ERR_SERVER_CENTER_CONN_ROOM_FAILED	-102044	跨房间连麦失败
ERR_SERVER_CENTER_CONN_ROOM_CANCEL_FAILED	-102045	取消跨房间连麦失败
ERR_SERVER_CENTER_CONN_ROOM_CONNED_ROOM_NOT_EXIST	-102046	被连麦房间不存在
ERR_SERVER_CENTER_CONN_ROOM_CONNED_REACH_MAX_ROOM	-102047	被连麦房间达到连麦上限
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_NOT_EXIST	-102048	被连麦用户不存在
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_DELETED	-102049	被连麦用户已被删除
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_FULL	-102050	被连麦用户达到资源上限
ERR_SERVER_CENTER_CONN_ROOM_INVALID_SEQ	-102051	连麦请求序号错乱

警告码表

警告码无需特别关注，您可以按需选择是否对当前用户进行相应的提示。

符号	值	含义
WARNING_HW_ENCODER_START_FAIL	1103	硬编码启动出现问题，自动切换到软编码
WARNING_VIDEO_ENCODER_SW_TO_HW	1107	当前 CPU 使用率太高，无法满足软件编码需求，自动切换到硬件编码

WARNING_INSUFFICIENT_CAPTURE_FPS	1108	摄像头采集帧率不足，部分自带美颜算法的 Android 手机上会出现
WARNING_SW_ENCODER_START_FAIL	1109	软编码启动失败
WARNING_REDUCE_CAPTURE_RESOLUTION	1110	摄像头采集分辨率被降低，以满足当前帧率和性能最优解。
WARNING_CAMERA_DEVICE_EMPTY	1111	没有检测到可用的摄像头设备
WARNING_CAMERA_NOT_AUTHORIZED	1112	用户未授权当前应用使用摄像头
WARNING_MICROPHONE_DEVICE_EMPTY	1201	没有检测到可用的麦克风设备
WARNING_SPEAKER_DEVICE_EMPTY	1202	没有检测到可用的扬声器设备
WARNING_MICROPHONE_NOT_AUTHORIZED	1203	用户未授权当前应用使用麦克风
WARNING_MICROPHONE_DEVICE_ABNORMAL	1204	音频采集设备不可用（例如被占用）
WARNING_SPEAKER_DEVICE_ABNORMAL	1205	音频播放设备不可用（例如被占用）
WARNING_VIDEO_FRAME_DECODE_FAIL	2101	当前视频帧解码失败
WARNING_AUDIO_FRAME_DECODE_FAIL	2102	当前音频帧解码失败
WARNING_VIDEO_PLAY_LAG	21	当前视频播放出现卡顿

	05	
WARNING_HW_DECODER_START_FAIL	2106	硬解启动失败，采用软解码
WARNING_VIDEO_DECODER_HW_TO_SW	2108	当前流硬解第一个 I 帧失败，SDK 自动切软解
WARNING_SW_DECODER_START_FAIL	2109	软解码器启动失败
WARNING_VIDEO_RENDER_FAIL	2110	视频渲染失败
WARNING_START_CAPTURE_IGNORED	40000	已经在采集，启动采集被忽略
WARNING_AUDIO_RECORDING_WRITE_FAIL	70001	音频录制写入文件失败
WARNING_ROOM_DISCONNECT	51001	网络断开连接
WARNING_IGNORE_UPSTREAM_FOR_AUDIENCE	60001	当前是观众角色，忽略上行音视频数据
WARNING_NET_BUSY	11001	网络状况不佳：上行带宽太小，上传数据受阻
WARNING_RTMP_SERVER_RECONNECT	11002	直播推流，网络断连，已启动自动重连（自动重连连续失败超过三次会放弃）
WARNING_LIVE_STREAM_SERVE	21	直播拉流，网络断连，已启动自动重连（自动重连连

R_RECONNECT	03	续失败超过三次会放弃）
WARNING_RECV_DATA_LAG	2104	网络来包不稳：可能是下行带宽不足，或由于主播端出流不均匀
WARNING_RTMP_DNS_FAIL	30001	直播，DNS 解析失败
WARNING_RTMP_SEVER_CONN_FAIL	30002	直播，服务器连接失败
WARNING_RTMP_SHAKE_FAIL	30003	直播，与 RTMP 服务器握手失败
WARNING_RTMP_SERVER_BREAK_CONNECT	30004	直播，服务器主动断开
WARNING_RTMP_READ_WRITE_FAIL	30005	直播，RTMP 读/写失败，将会断开连接
WARNING_RTMP_WRITE_FAIL	30006	直播，RTMP 写失败（SDK 内部错误码，不会对外抛出）
WARNING_RTMP_READ_FAIL	30007	直播，RTMP 读失败（SDK 内部错误码，不会对外抛出）
WARNING_RTMP_NO_DATA	30008	直播，超过30s 没有数据发送，主动断开连接

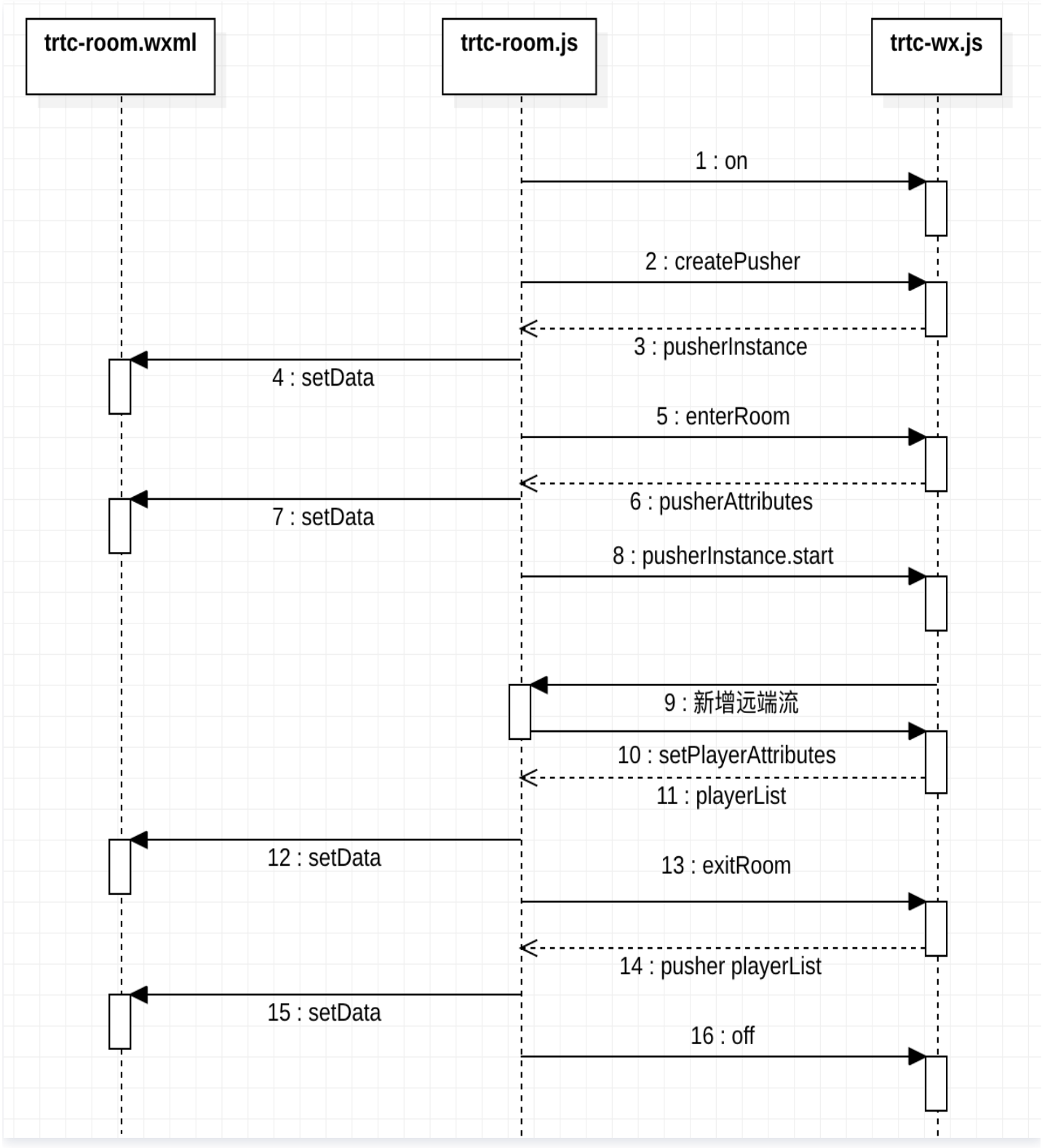
WARNING_PLAY_LIVE_STREAM_INFO_CONNECT_FAIL	3009	直播，connect 服务器调用失败（SDK 内部错误码，不会对外抛出）
WARNING_NO_STEAM_SOURCE_FAIL	3010	直播，连接失败，该流地址无视频（SDK 内部错误码，不会对外抛出）
WARNING_ROOM_RECONNECT	5102	网络断连，已启动自动重连
WARNING_ROOM_NET_BUSY	5103	网络状况不佳：上行带宽太小，上传数据受阻

小程序
trtc-wx

最近更新时间：2024-01-24 10:23:11

小程序的实时音视频是基于微信原生组件标签 <live-pusher> 和 <live-player> 实现的，您可以通过 live-pusher 文档 和 live-player 文档 对这两个标签有一个简单认识。本篇介绍的 trtc-wx 包是一个专门为您管理 TRTC 状态的一个类，作为一个纯 js 模块，您可以根据业务场景，自主编写页面的元素布局。trtc-wx 可以帮助您管理所有与实时音视频相关的状态，以及调用挂载在 <live-pusher> 和 <live-player> 上的方法。整体的调用逻辑如下图所示， `trtc-room.wxml` 是您自主编写的 wxml 文件，其中包括 <live-pusher> 和 <live-player> 节点。 `trtc-room.js` 是您的业务层代码，您需要在这个文件中引用我们的 `trtc-wx.js` 。
本文档包含 trtc-wx 模块 API 接口和所有事件说明，您通过查阅此文档能够获得更多 TRTC 的使用帮助。本文将在 API 概览部分中为您介绍 trtc-wx 提供的所有接口及其含义，在 API 使用指引的部分为您介绍，这些接口详细的使用方式，及注意事项。在更多高级特性中，会向您展示 trtc-wx 中是如何管理页面上的 live-pusher 和 live-player 实例的，最后，我们会为您介绍所有的事件通知，以及事件通知中的各种数据。

版权所有：腾讯云计算（北京）有限责任公司



上图是一个简单的使用示例，您可以通过 trtc-wx 提供的 API 方法对您的推拉流状态进行设置，之后通过 setData 的方式，在微信页面上的原生标签上更新生效，您通过事件订阅的接口可以捕获服务端的一些状态通知，trtc-wx 也会在这些事件通知中，返回更新部分的状态，您可以及时的将这些属性更新到页面当中。

版权所有：腾讯云计算（北京）有限责任公司

环境要求

- 微信 App iOS 最低版本要求：7.0.9。
- 微信 App Android 最低版本要求：7.0.8。
- 小程序基础库最低版本要求：2.10.0。
- 由于小程序测试号不具备 <live-pusher> 和 <live-player> 的使用权限，请使用企业小程序账号申请相关权限进行开发。
- 由于微信开发者工具不支持原生组件（即 <live-pusher> 和 <live-player> 标签），需要在真机上进行运行体验。

前提条件

1. 您已 注册腾讯云 账号，并完成 实名认证 。
2. **开通小程序类目与推拉流标签权限（如不开通则无法正常使用）。**
   出于政策和合规的考虑，微信暂未放开所有小程序对实时音视频功能（即 <live-pusher> 和 <live-player> 标签）的支持：

- 小程序推拉流标签不支持个人小程序，只支持企业类小程序。
- 小程序推拉流标签使用权限暂时只开放给有限 类目 。
- 符合类目要求的小程序，需要在 微信公众平台 > **开发** > **开发管理** > **接口设置** 中自助开通该组件权限，如下图所示：



API 概览

以下会为您介绍每个 API 的含义，基础方法是创建 TRTC 房间，使用 TRTC 服务的必用的 API，如果您需要主动改变推流的一些状态属性，您则需要调用 pusher 和 player 的属性变更的 API，如果您需要更多的功能，您也可以直接获取相关的实例，调用我们 SDK 开放的能力。

trtc-wx 与<live-pusher>/<live-player>的关系

trtc-wx SDK 意在帮助用户处理原生标签复杂的状态码以及各种属性状态变更。用户通过对<live-pusher>/<live-player>标签绑定回调，从而与 trtc-wx.js 建立联系，通过 SDK 内部处理，抛出 回调事件 。

```
<live-pusher
    url="{{pusher.url}}"
    bindstatechange="_pusherStateChangeHandler"
    bindnetstatus="_pusherNetStatusHandler"
    binderror="_pusherErrorHandler"
    bindbgmstart="_pusherBGMStartHandler"
    bindbgmprogress="_pusherBGMProgressHandler"
    bindbgmcomplete="_pusherBGMCompleteHandler"
    bindaudiovolumenotify="_pusherAudioVolumeNotify"
/>

// 请保持跟 wxml 中绑定的事件名称一致
_pusherStateChangeHandler(event) {
    this.TRTC.pusherEventHandler(event)
},
_pusherNetStatusHandler(event) {
    this.TRTC.pusherNetStatusHandler(event)
},
_pusherErrorHandler(event) {
    this.TRTC.pusherErrorHandler(event)
},
_pusherBGMStartHandler(event) {
    this.TRTC.pusherBGMStartHandler(event)
},
_pusherBGMProgressHandler(event) {
    this.TRTC.pusherBGMProgressHandler(event)
},
_pusherBGMCompleteHandler(event) {
    this.TRTC.pusherBGMCompleteHandler(event)
},
_pusherAudioVolumeNotify(event) {
    this.TRTC.pusherAudioVolumeNotify(event)
}
```

版权所有：腾讯云计算（北京）有限责任公司

> **说明**
> 单击下载 Demo 代码 。

基础方法

您可以通过这些方法完成事件监听，并创建 TRTC 房间，获取远端的拉流信息等操作。

API	描述
on(EventCode, handler, context)	用于监听组件派发的事件，详细事件请参见 事件表
off(EventCode, handler)	取消事件监听
createPusher(pusherAttributes)	创建推流实例
enterRoom(params)	进入房间，若房间不存在，系统将自动创建一个新房间
exitRoom()	停止推流和取消订阅所有远端音视频，并退出房间
getPlayerList()	获取当前远端 player 的列表

pusher & player 状态变更方法

您可以通过这些 API 主动改变推拉流的状态。

API	描述
setPusherAttributes(config)	设置推流的参数
setPlayerAttributes(id, config)	改变拉流 player 的参数

获取实例

获取相应的 pusher 和 player 的实例，可以调用更多的 SDK 开放能力。

API	描述
getPusherInstance()	获取推流实例，可以调用 pusherContext 的相关方法
getPlayerInstance(id)	获取某个 id 的 player 实例，可以调用 playerContext 相关的方法

API 使用指引

这部分会为您介绍如何进行使用上述 API 方法。

初始化 TRTC 实例

> **注意**
> 初始化 TRTC 实例时传入的上下文应该是当前 page 的上下文，否则在调用实例方法的时候会出现无法获取的情况。

```
import TRTC from 'trtc-wx.js'

page({
  onLoad(){
    this.TRTC = new TRTC(this)
    this.EVENT = this.TRTC.EVENT
  }
})
```

on(EventCode, handler, context)

用于监听组件派发的事件，详细事件请参考 事件表 。

> **注意**
> 请在调用 enterRoom 前监听事件，避免漏掉组件派发的事件。

参数：

参数名	类型	默认值	说明
EventCode	String	-	事件码
handler	Function	-	监听函数
context	Object	-	当前执行上下文

返回值：

无

示例代码：

```
this.TRTC.on(this.EVENT.LOCAL_JOIN, onLocalJoin, this)
```

off(EventCode, handler)

取消事件监听。

参数：

参数名	类型	默认值	说明
EventCode	String	-	事件码
handler	Function	-	需要取消的具名监听函数

返回值：

无

示例代码：

```
function onLocalJoin(event) {
 // 本地进房成功
}
trtcRoomContext.off(this.EVENT.LOCAL_JOIN, onLocalJoin)
```

createPusher(pusherAttributes)

创建推流实例，可以传入初始化的参数，以下的参数都是可选参数。

参数：

参数名	类型	默认值	说明
pusherAttributes	Object	default	可选，关于初始化参数请参见 pusherAttributes

返回值：

pusherInstance

示例代码：

```
this.TRTC.createPusher({'frontCamera': 'back'})
```

enterRoom(params)

进入 TRTC 房间。

参数：

参数名	类型	默认值	说明
sdkAppID	Number	–	必填，您的腾讯云账号的 sdkAppID
userID	String	–	必填，您进房的 userD，建议限制长度为32字节，只允许包含大小写英文字母(a-zA-Z)、数字(0-9)及下划线和连词符
userSig	String	–	必填，当前 userID 对应的验证签名，相当于使用云服务的登录密码
roomID	Number	–	必填，如已填写 strRoomID，roomID 可不填写。roomID 为您要进入的房间号，如该房间不存在，系统会为您自动创建，取值范围：1 - 4294967294
strRoomID	String	–	选填，您要进入的字符串房间号，如填写该参数，将优先进入字符串房间，限制长度为64字节，roomID可不填写。以下为支持的字符集范围（共 89 个字符）： • 大小写英文字母（a-zA-Z） • 数字（0-9） • 空格、!、#、$、%、&、(、)、+、-、:、;、<、=、.、>、?、@、[、]、^、_、{、}、。
privateMapKey	String	–	选填，给房间中加入进房限制或者上麦限制，参考 高级权限控制 。
userDefineRecordId	String	–	选填，设置云端录制完成后的回调消息中的“userdefinerecordid” 字段内容，便于您更方便的识别录制回调。 • **推荐取值：**限制长度为64字节，只允许包含大小写英文字母（a-zA-Z）、数字（0-9）及下划线和连词符。 • **参考文档：**云端录制
scene	String	'rtc'	选填，必填参数（与TRTCAppScene为同一字段），使用场景： • **rtc**：实时通话，采用优质线路，同一房间中的人数不应超过300人。 • **live**：直播模式，采用混合线路，支持单一房间十万人在线（同时上麦的人数应控制在50人以内）

recvMode	Number	1	选填，设置live-pusher在房间内音视频流的接收模式 • 1：自动接收音视频 • 2: 仅自动接收音频 • 3：仅自动接收视频 • 4：音视频都不自动接收

注意

- **视频通话&语音通话（scene = "rtc"）**
  - 视频通话场景，支持 720P、1080P 高清画质。
  - 视频通话场景，支持48kHz全频带，支持双声道。
  - 单个房间最多支持300人同时在线，最高支持50人同时发言。
  - 适用场景：1对1视频通话、300人视频会议、在线问诊、远程面试、视频客服、在线狼人杀等。
- **互动直播&语音聊天室（scene = "live"）**
  - 支持十万人级别观众同时播放，播放延时低至1000ms。
  - 支持平滑上下麦，切换过程无需等待，主播延时小于300ms。
  - 适用场景：视频低延时直播、十万人互动课堂、视频相亲、在线教育、远程培训、超大型会议等。

其他 pusherAttributes 需要设置也可以在这里进行设置。

返回值：

pusherAttributes

示例代码：

```
enterRoom(options) {
  this.setData({
    pusher: this.TRTC.enterRoom({
      sdkAppID: 1400xxxxx, // 您的腾讯云账号
      userID: 'trtc-user', //当前进房用户的userID
      userSig: 'xxxxxxx', // 您服务端生成的userSig
      roomID: 1234, // 您进房的房间号，
      enableMic: true, // 进房默认开启音频上行
      enableCamera: true, // 进房默认开启视频上行
    }),
  }, () => {
    this.TRTC.getPusherInstance().start() // 开始进行推流
  })
},
```

exitRoom()

退房，重置状态机的状态，并同步到页面中，防止下次进房发生状态的混乱。

参数：

无

返回值：

- Object。
- 状态机重置，会返回更新后的 pusher 和 playerList。

```
{
  pusher: {},
  playerList: {}
}
```

示例代码：

```
exitRoom() {
  const result = this.TRTC.exitRoom()
  // 状态机重置，会返回更新后的pusher和playerList
  this.setData({
    pusher: result.pusher,
    playerList: result.playerList,
  })
},
```

getPlayerList()

获取播放的 player 列表。

参数：

无

返回值：

Array 播放 player 的列表。

示例代码：

```
this.setData({
  playerList: this.TRTC.getPlayerList()
})
```

setPusherAttributes(config)

设置推流的参数。

参数：

参数名	类型	默认值	说明
config	Object	-	您可以设置 pusherAttributes 中的属性

返回值：

pusherAttributes，返回更新后的推流状态。

示例代码：

```
this.setData({
  pusher: this.TRTC.setPusherAttributes({
    enableMic: false, // 关闭音频上行
    enableCamera: false, // 关闭视频上行
  }),
})
```

setPlayerAttributes(id, config)

改变拉流 player 的参数。

参数：

参数名	类型	默认值	说明
id	String	-	您需要改变状态的 player 的 ID
config	Object	-	您可以设置 playerAttributes 中的属性

返回值：

playerList，返回更新后的拉流状态列表。

示例代码：

```
this.setData({
    playerList: this.TRTC.setPlayerAttributes(
      playerId,
      {
        muteVideo: true,  // 停止接收远程用户的视频流，视频画面会冻屏在 mute 前的最后一帧。
        orientation: 'horizontal',  // 将拉流画面设置为水平方向
        // 其余参数参考 playerAttributes 中的属性
      }
    )
})
```

getPusherInstance()

获取 pusher 的推流实例，调用挂载在实例上的相关方法，具体请参见 pusherInstance 。

参数：

无

返回值：

pusherInstance

示例代码：

```
this.TRTC.getPusherInstance().start() // 开始推流
```

getPlayerInstance(id)

获取对应的 player 实例，调用挂载在实例上的相关方法，具体请参见 playerInstance 。

参数：

参数名	类型	默认值	说明
id	String	-	您需要获取实例的 player 的 ID

返回值：

playerInstance

示例代码：

```
this.TRTC.getPlayerInstance().stop() // 停止这个player的播放
```

更多高级特性

<live-pusher> 和 <live-player> 不仅具有原生标签所通用的标签属性，还具有相应 context 的能力，在每个标签 context 中都会有很多的 SDK 开放能力提供，trtc-wx 已经为您创建了相关的 context 对象，您只需要获取相应的 Instance 实例，就可以调用这些方法，参数的传递也是兼容微信原生方法的。

pusherInstance

pusherInstance 是 trtc-wx 帮助您管理 <live-pusher> 的一个实例。这个对象是全局单例的，对于实例上的方法，大部分基于微信的原生 context 上的方法进行封装，所有参数均与官方文档一致，详细介绍如下表：

参数名	参数类型	是否必选	参数说明	API说明
pusherAttributes	-	-	pusherAttributes 说明	<live-pusher> 标签上的属性
stop(options)	Object	可选	stop 参数	停止推流，您调用这个接口后，远端会收到您退房的通知
pause(options)	Object	可选	pause 参数	暂停推流
resume(options)	Object	可选	resume参数	继续推流
snapshot()	-	-	-	该接口已经进行了封装，对推流画面进行截图，并保存到本地相册
playBGM(params)	parma.url	必填	playBGM 参数	播放对应 URL 资源的音乐，作为背景音乐
pauseBGM(options)	Object	可选	pauseBGM 参数	暂停播放 BGM
resumeBGM(options)	Object	可选	resumeBGM 参数	继续播放 BGM
stopBGM(options)	Object	可选	stopBGM 参数	停止播放 BGM
setBGMVolume(params)	params.volume	必填	setBGMVolume	设置BGM的播放音量，默认是 1.0

setMICVolume(params)	params.volume	必填	setMICVolume	设置麦克风的音量，默认是1.0
startPreview(options)	Object	可选	startPreview	开启预览
stopPreview()	Object	可选	stopPreview	停止预览
switchCamera(options)	Object	可选	switchCamera	前置或后置摄像头，可选值：front，back

pusherAttributes 说明

参数名	类型	默认值	说明
mode	String	RTC	SD（标清）、HD（高清）、FHD（超清）、RTC（实时通话）
enableCamera	Boolean	false	是否开启摄像头
enableMic	Boolean	false	是否开启麦克风
enableAgc	Boolean	false	是否开启音频自动增益，该特性可以补偿部分手机麦克风音量太小的问题，但也会放大噪音，建议配合 ANS 同时开启
enableAns	Boolean	false	是否开启音频噪声抑制，该特性会自动检测背景噪音并进行过滤，但也会误伤周围的音乐，只有会议、教学等场景才适合开启
minBitrate	Number	600	最小码率，不建议设置太低
maxBitrate	Number	900	最大码率，需要跟分辨率相匹配，请参见 分辨率码率参照表
frontCamera	String	front	前置或后置摄像头，可选值：front，back
enableZoom	Boolean	false	是否支持双手滑动调整摄像头焦距
videoWidth	Number	360	上推的视频流的分辨率宽度（若设置了视频宽高就会忽略 aspect）
videoHeight	Number	640	上推的视频流的分辨率高度（若设置了视频宽高就会忽略 aspect）

beautyLevel	Number	0	美颜。取值范围 0-9 ，0 表示关闭
whitenessLevel	Number	0	美白。取值范围 0-9 ，0 表示关闭
videoOrientation	String	vertical	推流方向。vertical：垂直方向，horizontal：水平方向
enableRemoteMirror	Boolean	false	设置推流画面是否镜像，产生的效果会表现在 live-player
localMirror	String	'auto'	设置主播本地摄像头预览画面的镜像效果，支持如下取值： • auto：前置摄像头镜像，后置摄像头不镜像（系统相机的表现 ） • enable：前置摄像头和后置摄像头都镜像 disable: 前置摄像头和后置摄像头都不镜像
audioQuality	String	'high'	高音质（48KHz）或低音质（16KHz），可选值：high、low
audioVolumeType	String	'voicecall'	声音类型。可选值： media: 媒体音量，voicecall: 通话音量
audioReverbType	Number	0	音频混响类型。0：关闭，1：KTV，2：小房间，3：大会堂，4：低沉，5：洪亮，6：金属声，7：磁性
beautyStyle	String	'smooth'	美颜类型。取值有：smooth：光滑，nature：自然
filter	String	''	滤镜类型。standard：标准，pink：粉嫩，nostalgia：怀旧，blues：蓝调，romantic：浪漫，cool：清凉，fresher：清新，solor：日系，aestheticism：唯美，whitening：美白，cerisered：樱红

playerInstance

playerInstance 是 trtc-wx 为您管理的 <live-player> 的实例。

参数名	参数类型	是否必填	参数说明	API说明
playerAttributes	–	–	playerAttributes 说明	<live-player> 标签上的属性
play(options)	Object	可选	play 参数	播放远端流，默认是自动模式，无需手动播放

stop(options)	Object	可选	stop 参数	停止播放
pause(options)	Object	可选	pause 参数	暂停播放
resume(options)	Object	可选	resume 参数	继续播放
snapshot()	-	-	Promise	对 player 画面进行截图，保存到本地相册，返回值
requestFullScreen(params)	parma.direction	必填	Promise	将这个 player 全屏播放
requestExitFullScreen()	-	-	Promise	将这个 player 退出全屏播放
mute(options)	Object	可选	mute 参数	对当前 player 进行静音操作

playerAttributes 说明

参数名	类型	默认值	说明
mode	String	RTC	• RTC：实时通话 • live：直播模式
autoplay	Boolean	true	有音频流下行后，是否自动播放
muteAudio	Boolean	true	默认通过 live-pusher 主动拉音频流，不需要您手动进行订阅
muteVideo	Boolean	true	默认不主动拉视频流，需要您手动进行订阅
orientation	String	vertical	player 方向： • vertical：垂直方向 • horizontal：水平方向
objectFit	String	fillCrop	填充模式，可选值有 contain，fillCrop • contain：图像长边填满屏幕，短边区域会被填充 • fillCrop：图像铺满屏幕，超出显示区域的部分将被截掉
minCache	Number	1	最小缓冲区，单位：s

maxCache	Number	2	最大缓冲区，单位：s
soundMode	String	speaker	声音输出方式： • speaker：扬声器 • ear：听筒（通话音量模式下听筒才有效，对应 RTC 模式）
enableRecvMessage	Boolean	false	是否接收 SEI 消息
autoPauseIfNavigate	Boolean	true	当跳转到其它小程序页面时，是否自动暂停本页面的实时音视频播放
autoPauseIfOpenNative	Boolean	true	当跳转到其它微信原生页面时，是否自动暂停本页面的实时音视频播放

事件表

通过组件实例的 EVENT 属性可以获取到事件常量字段，这些事件会通知您服务器推送的一些信息，以及 SDK 内部的一些参数提示。

```
this.TRTC = new TRTC(this)
const EVENT = this.TRTC.EVENT
```

事件表

CODE	说明
LOCAL_JOIN	成功进入房间
LOCAL_LEAVE	成功离开房间
KICKED_OUT	服务端踢人或房间被解散退房
REMOTE_USER_JOIN	远端用户进入房间时触发
REMOTE_USER_LEAVE	远端用户退出房间时触发
REMOTE_VIDEO_ADD	远端视频流添加事件，当远端用户发布视频流后会收到该通知
REMOTE_VIDEO_REMOVE	远端视频流移出事件，当远端用户取消发布视频流后会收到该通知
REMOTE_AUDIO_ADD	远端音频流添加事件，当远端用户发布音频流后会收到该通知
REMOTE_AUDIO_REMOVE	远端音频流移除事件，当远端用户取消发布音频流后会收到该通知

版权所有：腾讯云计算（北京）有限责任公司

REMOTE_STATE_UPDATE	远端用户播放状态变更通知
LOCAL_NET_STATE_UPDATE	本地推流的网络状态变更通知
REMOTE_NET_STATE_UPDATE	远端用户网络状态变更通知
REMOTE_AUDIO_VOLUME_UPDATE	远端用户音量变更通知
LOCAL_AUDIO_VOLUME_UPDATE	本地音量变更通知
VIDEO_FULLSCREEN_UPDATE	远端视图全屏状态变更通知
BGM_PLAY_PROGRESS	BGM 播放时间戳变更通知
BGM_PLAY_COMPLETE	BGM 播放结束通知
ERROR	本地推流出现错误、渲染错误事件等

LOCAL_JOIN

本地进房成功后的回调。

```
let onLocalJoin = function(){
  console.log('本地进房成功')
}
this.TRTC.on(EVENT.LOCAL_JOIN, onLocalJoin)
```

LOCAL_LEAVE

本地离开房间后的回调。

```
let onLocalLeave = function(){
  console.log('本地退房成功')
}
this.TRTC.on(EVENT.LOCAL_LEAVE, onLocalLeave)
```

KICKED_OUT

服务端踢人或房间被解散退房后的回调。

```
let onKickedout = function(){
  console.log('被服务端踢出或房间被解散')
}
this.TRTC.on(EVENT.KICKED_OUT, onKickedout)
```

REMOTE_USER_JOIN

远端用户加入此房间。

```
let onRemoteUserJoin = function(event){
  // userID 是加入的用户 ID
  const { userID, userList, playerList } = event.data
}
this.TRTC.on(EVENT.REMOTE_USER_JOIN, onRemoteUserJoin)
```

REMOTE_USER_LEAVE

远端的用户离开。

```
let onRemoteUserLeave = function(event){
  // userID 是离开的用户 ID
  const { userID, userList, playerList } = event.data
}
this.TRTC.on(EVENT.REMOTE_USER_LEAVE, onRemoteUserLeave)
```

REMOTE_VIDEO_ADD

远端的用户有新的视频上行。

```
let onRemoteVideoAdd = function(event){
  // id 是对应的player的id
  const { player, userList, playerList } = event.data
}
this.TRTC.on(EVENT.REMOTE_VIDEO_ADD, onRemoteVideoAdd)
```

REMOTE_VIDEO_REMOVE

版权所有：腾讯云计算（北京）有限责任公司

远端的用户有视频上行移除。

```
let onRemoteVideoRemove = function(event){
  // id 是对应的player的id
  const { player, userList, playerList } = event.data
}
this.TRTC.on(EVENT.REMOTE_VIDEO_REMOVE, onRemoteVideoRemove)
```

REMOTE_AUDIO_ADD

远端的用户有新的音频上行。

```
let onRemoteAudioAdd = function(event){
  // id 是对应的player的id
  const { player, userList, playerList } = event.data
}
this.TRTC.on(EVENT.REMOTE_AUDIO_ADD, onRemoteAudioAdd)
```

REMOTE_AUDIO_REMOVE

远端的用户有音频上行移除。

```
let onRemoteAudioRemove = function(event){
  // id 是对应的player的id
  const { id, userList, playerList } = event.data
}
this.TRTC.on(EVENT.REMOTE_AUDIO_REMOVE, onRemoteAudioRemove)
```

REMOTE_STATE_UPDATE

远端用户播放状态变更通知。

```
let onRemoteStateUpdate = function(event){
  // id 是对应触发的 player 的 id，目前 streamid 和 id 是相同的
  const id = event.data.currentTarget.dataset.streamid
  const data = event.data // 这里是微信原生组件抛出的关于player的信息，若有需要您可以自
主获取
}
this.TRTC.on(EVENT.REMOTE_STATE_UPDATE, onRemoteStateUpdate)
```

版权所有：腾讯云计算（北京）有限责任公司

LOCAL_NET_STATE_UPDATE

本地网络相关状态变更。

```
let onLocalNetStateUpdate = function(event){
  // 这里会返回更新后的 pusherAttributes，上面有个属性是 netStatus 对应网络状态的对象
  // 其中 netQualityLevel 对应网络状态的好坏，1 代表最好，数字越大代表网络越差
  const netStatus = event.data.pusher.netStatus
}
this.TRTC.on(EVENT.LOCAL_NET_STATE_UPDATE, onLocalNetStateUpdate)
```

REMOTE_NET_STATE_UPDATE

远端用户网络相关状态变更。

```
let onRemoteNetStateUpdate = function(event){
  // 这里会返回更新后的 playerList，上面有个属性是 netStatus 对应网络状态的对象
  // 其中 netQualityLevel 对应网络状态的好坏，1 代表最好，数字越大代表网络越差
  const { playerList } = event.data

}
this.TRTC.on(EVENT.REMOTE_NET_STATE_UPDATE, onRemoteNetStateUpdate)
```

REMOTE_AUDIO_VOLUME_UPDATE

远端用户音量状态变更。

```
let onRemoteAudioVolumeUpdate = function(event){
  // 这里会返回更新后的 playerList
  const { playerList } = event.data
}
this.TRTC.on(EVENT.REMOTE_AUDIO_VOLUME_UPDATE,
onRemoteAudioVolumeUpdate)
```

LOCAL_AUDIO_VOLUME_UPDATE

本地音量状态变更。

```
let onLocalAudioVolumeUpdate = function(event){
```

```
  // 这里会返回更新后的 pusher 状态
  const { pusher } = event.data
}
this.TRTC.on(EVENT.LOCAL_AUDIO_VOLUME_UPDATE, onLocalAudioVolumeUpdate)
```

VIDEO_FULLSCREEN_UPDATE

远端视图全屏状态变更。

```
let onVideoFullscreenUpdate = function(){
  // 您可以进行业务的操作，目前没有抛出data字段
}
this.TRTC.on(EVENT.VIDEO_FULLSCREEN_UPDATE, onVideoFullscreenUpdate)
```

BGM_PLAY_PROGRESS

BGM 播放时间戳变更通知。

```
let onBgmPlayProgress = function(event){
  // progress是已经播放的时长，duration是总时长，比值代表当前的进度
  const { progress, duration } = event.data

}
this.TRTC.on(EVENT.BGM_PLAY_PROGRESS, onBgmPlayProgress)
```

BGM_PLAY_COMPLETE

BGM 播放结束通知。

```
let onBgmPlayComplete = function(){
  // 您可以进行业务的操作，目前没有抛出data字段
}
this.TRTC.on(EVENT.BGM_PLAY_COMPLETE, onBgmPlayComplete)
```

ERROR

本地推流出现错误、渲染错误事件等。

```
let onError = function(event){
```

```
  // 您可以监听一些预期之外的错误信息
  const { code, message } = event.data
}
this.TRTC.on(EVENT.ERROR, onError)
```

技术咨询

了解更多详情您可 QQ 咨询：941036374（技术交流群）。

参考文档

- 快速集成(小程序)
- 语音聊天室(小程序)
- 多人音视频房间(小程序)
- 视频通话(小程序)
- 语音通话(小程序)

版权所有：腾讯云计算（北京）有限责任公司

错误码

最近更新时间：2023-05-19 17:34:54

小程序端 SDK 相关的错误码如下表所示。

CODE	说明
10001	用户禁止使用摄像头
10002	用户禁止使用录音
10003	背景音资源（BGM）加载失败
10004	等待画面资源（waiting-image）加载失败
-1301	打开摄像头失败
-1302	打开麦克风失败
-1303	视频编码失败
-1304	音频编码失败
-1307	推流出现网络断开，且经过多次重试无法恢复（可能是由于账号欠费，未购买 包月套餐 解锁小程序SDK能力导致）
-3301	进入房间失败
-100006	权限位校验失败，请检查 privateMapKey 是否正确
-100015	SDKAppID 异常，请检查 SDKAppID 是否正确
-100018	userSig 校验失败，请检查 userSig 是否正确

说明

小程序和 native 端的错误码是保持一致的，更多其他错误码查询请参见 iOS 错误码 和 Android 错误码 。

Flutter
API 概览

最近更新时间：2024-03-01 14:48:31

TRTCCloud

基础方法

API	描述
sharedInstance	创建 TRTCCloud 单例。
destroySharedInstance	销毁 TRTCCloud单例。
registerListener	设置事件监听
unRegisterListener	移除事件监听

房间相关接口函数

API	描述
enterRoom	进入房间，若房间不存在，系统将自动创建一个新房间。
exitRoom	离开房间。
switchRole	切换角色，仅适用于直播场景（TRTC_APP_SCENE_LIVE 和 TRTC_APP_SCENE_VOICE_CHATROOM）。
setDefaultStreamRecvMode	设置音视频数据接收模式，需要在进房前设置才能生效。
connectOtherRoom	请求跨房通话（主播 PK）。
disconnectOtherRoom	退出跨房通话。
switchRoom	切换房间。

CDN 相关接口函数

API	描述
startPublishing	开始向腾讯云的直播 CDN 推流。
stopPublishing	停止向腾讯云的直播 CDN 推流。
startPublishCDNStream	开始向友商云的直播 CDN 转推。
stopPublishCDNStream	停止向非腾讯云地址转推。
setMixTranscodingConfig	设置云端的混流转码参数。
startPublishMediaStream	开始发布媒体流。
updatePublishMediaStream	更新发布媒体流。
stopPublishMediaStream	停止发布媒体流。

视频相关接口函数

API	描述
startLocalPreview	开启本地视频的预览画面。
updateLocalView	更新本地视频预览画面。
updateRemoteView	更新远端用户的视频渲染控件。
stopLocalPreview	停止本地视频采集及预览。
muteLocalVideo	暂停/恢复推送本地的视频数据。
startRemoteView	开始显示远端视频画面。
stopRemoteView	停止显示远端视频画面，同时不再拉取该远端用户的视频数据流。
stopAllRemoteView	停止显示所有远端视频画面，同时不再拉取远端用户的视频数据流。
setVideoMuteImage	设置暂停推送本地视频时要推送的图片。

版权所有：腾讯云计算（北京）有限责任公司

muteRemoteVideoStream	暂停/恢复接收指定的远端视频流。
muteAllRemoteVideoStreams	暂停/恢复接收所有远端视频流。
setVideoEncoderParam	设置视频编码器相关参数。
setNetworkQosParam	设置网络流控相关参数。
setLocalRenderParams	设置本地图像的渲染模式。
setRemoteRenderParams	设置远端图像相关参数。
setVideoEncoderRotation	设置视频编码输出的画面方向，即设置远端用户观看到的和服务器录制的画面方向。
setVideoEncoderMirror	设置编码器输出的画面镜像模式。
setGSensorMode	设置重力感应的适应模式。
enableEncSmallVideoStream	开启大小画面双路编码模式。
setRemoteVideoStreamType	选定观看指定 uid 的大画面或小画面。
snapshotVideo	视频画面截图。
startLocalRecording	开始本地媒体录制，同时包含音频与视频数据。
stopLocalRecording	停止本地媒体录制。

音频相关接口函数

API	描述
startLocalAudio	开启本地音频的采集和上行。
stopLocalAudio	关闭本地音频的采集和上行。

版权所有：腾讯云计算（北京）有限责任公司

muteLocalAudio	静音/取消静音本地的音频。
setAudioRoute	设置音频路由。
muteRemoteAudio	静音/取消静音指定的远端用户的声音。
muteAllRemoteAudio	静音/取消静音所有用户的声音。
setRemoteAudioVolume	设定某一个远端用户的声音播放音量
setAudioCaptureVolume	设置 SDK 采集音量。
getAudioCaptureVolume	获取 SDK 采集音量。
setAudioPlayoutVolume	设置 SDK 播放音量。
getAudioPlayoutVolume	获取 SDK 播放音量。
enableAudioVolumeEvaluation	启用音量大小提示。
startAudioRecording	开始录音。
stopAudioRecording	停止录音。
setSystemVolumeType	设置通话时使用的系统音量类型。
startSystemAudioLoopback	启用系统音频采集。
stopSystemAudioLoopback	停止系统音频采集。
setSystemAudioLoopbackVolume	设置系统音频采集的音量。

设备管理接口

API	描述
getDeviceManager	获取设备管理模块。接口详情见 设备管理接口文档

美颜滤镜相关接口函数

API	描述
getBeautyManager	获取美颜管理对象。接口详情见 美颜管理文档
setWatermark	添加水印。

自定义采集和自定义渲染

API	描述
setLocalVideoRenderListener	设置本地视频的自定义渲染回调
setRemoteVideoRenderListener	设置远端视频自定义渲染回调
unregisterTexture	取消自定义的渲染回调
enableCustomVideoProcess	启用/禁用自定义视频处理
setAudioFrameListener	设置自定义音频数据回调

音乐特效和人声特效

API	描述
getAudioEffectManager	获取音效管理类 TXAudioEffectManager，用于管理BGM，短音效和人声特效。接口详情见 音效管理文档

辅流相关接口函数

API	描述
startScreenCapture	启动屏幕分享
stopScreenCapture	停止屏幕采集
pauseScreenCapture	暂停屏幕分享
resumeScreenCapture	恢复屏幕分享

自定义消息发送

API	描述
sendCustomCmdMsg	发送自定义消息给房间内所有用户。
sendSEIMsg	将小数据量的自定义数据嵌入视频帧中。

网络测试

API	描述
startSpeedTest	开始进行网络测速（视频通话期间请勿测试，以免影响通话质量）。
stopSpeedTest	停止服务器测速。

Log 相关接口函数

API	描述
getSDKVersion	获取 SDK 版本信息。
setLogLevel	设置 Log 输出级别。
setLogDirPath	修改日志保存路径。
setLogCompressEnabled	启用或禁用 Log 的本地压缩。
setConsoleEnabled	启用或禁用控制台日志打印。
showDebugView	显示调试信息浮层(可以展示音视频信息和事件信息)。
callExperimentalAPI	调用实验性 API。

TRTCCloudListener

腾讯云视频通话功能的事件回调接口。

错误事件和警告事件

API	描述
onError	错误回调，表示 SDK 不可恢复的错误，一定要监听并分情况给用户适当的界面提示。
onWarning	警告回调，用于告知您一些非严重性问题，例如出现卡顿或者可恢复的解码

	失败。

房间事件回调

API	描述
onEnterRoom	已加入房间的回调。
onExitRoom	离开房间的事件回调。
onSwitchRole	切换角色的事件回调。
onConnectOtherRoom	请求跨房通话（主播 PK）的结果回调。
onDisConnectOtherRoom	结束跨房通话（主播 PK）的结果回调。
onSwitchRoom	切换房间 (switchRoom) 的结果回调。

成员事件回调

API	描述
onRemoteUserEnterRoom	有用户加入当前房间。
onRemoteUserLeaveRoom	有用户离开当前房间。
onUserVideoAvailable	远端用户是否存在可播放的主路画面（一般用于摄像头）。
onUserSubStreamAvailable	远端用户是否存在可播放的辅路画面（一般用于屏幕分享）。
onUserAudioAvailable	远端用户是否存在可播放的音频数据。
onFirstVideoFrame	开始渲染本地或远程用户的首帧画面。
onFirstAudioFrame	开始播放远程用户的首帧音频（本地声音暂不通知）。
onSendFirstLocalVideoFrame	首帧本地视频数据已经被送出。
onSendFirstLocalAudioFrame	首帧本地音频数据已经被送出。

版权所有：腾讯云计算（北京）有限责任公司

录制任务回调

API	描述
onLocalRecordBegin	当本地录制任务已经开始时，SDK会通过这个回调通知。
onLocalRecording	当本地录制任务正在进行时，SDK会定期通过这个回调通知用户。
onLocalRecordFragment	当您启用分段录制时，SDK会在每次完成一个片段时通过这个回调通知您。
onLocalRecordComplete	当本地录制任务已经结束时，SDK会通过这个回调通知您。

播放背景音乐的回调接口

API	描述
onMusicObserverStart	音乐播放开始的回调通知。
onMusicObserverPlayProgress	音乐播放进度的回调通知。
onMusicObserverComplete	音乐播放结束的回调通知。

统计和质量回调

API	描述
onNetworkQuality	网络质量，该回调每2秒触发一次，统计当前网络的上行和下行质量。
onStatistics	技术指标统计回调。

服务器事件回调

API	描述
onConnectionLost	SDK 跟服务器的连接断开。
onTryToReconnect	SDK 尝试重新连接到服务器。
onConnectionRecovery	SDK 跟服务器的连接恢复。
onSpeedTest	服务器测速的回调，SDK 对多个服务器 IP 做测速，每个 IP 的测速结果通

	过这个回调通知。

硬件设备事件回调

API	描述
onCameraDidReady	摄像头准备就绪。
onMicDidReady	麦克风准备就绪。
onUserVoiceVolume	用于提示音量大小的回调，包括每个 userId 的音量和远端总音量。
onDeviceChange	本地设备连接和断开的回调
onTestMicVolume	麦克风测试音量的回调
onTestSpeakerVolume	扬声器测试音量的回调

自定义消息的接收回调

API	描述
onRecvCustomCmdMsg	收到自定义消息回调。
onMissCustomCmdMsg	自定义消息丢失回调。
onRecvSEIMsg	收到 SEI 消息的回调。

CDN 旁路转推回调

API	描述
onStartPublishing	开始向腾讯云的直播 CDN 推流的回调，对应于 TRTCCloud 中的 startPublishing() 接口。
onStopPublishing	停止向腾讯云的直播 CDN 推流的回调，对应于 TRTCCloud 中的 stopPublishing() 接口。
onStartPublishCDNStream	启动旁路推流到 CDN 完成的回调。
onStopPublishCDNStream	停止旁路推流到 CDN 完成的回调。

onSetMixTranscodingConfig	设置云端的混流转码参数的回调，对应于 TRTCCloud 中的 setMixTranscodingConfig() 接口。
onStartPublishMediaStream	设置云端的混流转码参数的回调，对应于 TRTCCloud 中的 startPublishMediaStream() 接口。
onUpdatePublishMediaStream	设置云端的混流转码参数的回调，对应于 TRTCCloud 中的 updatePublishMediaStream() 接口。
onStopPublishMediaStream	设置云端的混流转码参数的回调，对应于 TRTCCloud 中的 stopPublishMediaStream() 接口。

屏幕分享回调

API	描述
onScreenCaptureStarted	当屏幕分享开始时，SDK 会通过此回调通知
onScreenCapturePaused	当屏幕分享调用 pauseScreenCapture() 暂停时，SDK 会通过此回调通知。
onScreenCaptureResumed	当屏幕分享调用 resumeScreenCapture() 恢复时，SDK 会通过此回调通知。
onScreenCaptureStoped	当屏幕分享停止时，SDK 会通过此回调通知。

截图回调

API	描述
onSnapshotComplete	截图完成时回调。

TXAudioEffectManager

API	描述
enableVoiceEarMonitor	启用耳返，使本地用户可以听到自己的声音。
setVoiceEarMonitorVolume	设置耳返音量
setVoiceReverbType	设置语音混响效果（KTV、小房间、大会堂、低沉、洪亮等效果）
setVoiceChangerTyp	设置语音变声效果（萝莉、大叔、重金属、朋克等效果）

e	
setVoiceCaptureVolume	设置麦克风语音音量
startPlayMusic	开始播放背景音乐
stopPlayMusic	停止播放背景音乐
pausePlayMusic	暂停播放背景音乐
resumePlayMusic	恢复播放背景音乐
setMusicPublishVolume	设置背景音乐的远程音量。主播可以使用此 API 设置远程观众听到的背景音乐的音量。
setMusicPlayoutVolume	设置背景音乐的本地音量。主播可以使用此 API 设置本地背景音乐的音量。
setAllMusicVolume	设置全局背景音乐的本地和远程音量
setMusicPitch	调整背景音乐的音调
setMusicSpeedRate	调整背景音乐的速度
getMusicCurrentPosInMS	获取背景音乐的当前播放进度（毫秒）
seekMusicToPosInMS	设置背景音乐的播放进度（毫秒）
getMusicDurationInMS	获取背景音乐文件的总时长（毫秒）

TXBeautyManager

API	描述
setBeautyStyle	设置美颜滤镜类型
setFilter	指定素材滤镜效果
setFilterStrength	设置滤镜强度
setBeautyLevel	设置美颜滤镜强度
setWhitenessLevel	设置亮白滤镜强度
enableSharpnessEnh	启用清晰度增强

ancement	
setRuddyLevel	设置红润滤镜强度

TXDeviceManager

API	描述
isFrontCamera	设置是否使用前置摄像头
switchCamera	切换摄像头（前置/后置）
getCameraZoomMaxRatio	获取摄像头的最大缩放倍数
setCameraZoomRatio	设置摄像头的缩放因子（焦距）
enableCameraAutoFocus	设置是否启用面部位置自动识别
isAutoFocusEnabled	查询设备是否支持自动识别人脸位置
setCameraFocusPosition	设置摄像头对焦位置
enableCameraTorch	启用/禁用闪光灯，也就是手电筒模式
setSystemVolumeType	设置通话中使用的系统音量类型
setAudioRoute	设置音频路由，即使用听筒或者扬声器
getDevicesList	获取指定类型的设备列表
setCurrentDevice	设置当前要使用的设备
getCurrentDevice	获取当前正在使用的设备
setCurrentDeviceVolume	设置当前设备的音量
getCurrentDeviceVolume	获取当前设备的音量
setCurrentDeviceMute	设置当前设备的静音状态
getCurrentDeviceMut	查询当前设备的静音状态

版权所有：腾讯云计算（北京）有限责任公司

e	
startMicDeviceTest	开始麦克风测试
stopMicDeviceTest	停止麦克风测试
startSpeakerDeviceTest	开始扬声器测试
stopSpeakerDeviceTest	停止扬声器测试
setApplicationPlayVolume	设置 Windows 系统音量混合器中当前进程的音量
getApplicationPlayVolume	获取 Windows 系统音量混合器中当前进程的音量
setApplicationMuteState	设置 Windows 系统音量混合器中当前进程的静音状态
getApplicationMuteState	获取 Windows 系统音量混合器中当前进程的静音状态

关键类型定义

类名	描述
TRTCCloudDef	关键类型定义变量。
TRTCParams	进房参数。
TRTCSwitchRoomConfig	切换房间参数参数。
TRTCVideoEncParam	视频编码参数。
TRTCNetworkQosParam	网络流控相关参数。
TRTCRenderParams	远端图像参数。
TRTCMixUser	云端混流中每一路子画面的位置信息。
TRTCTranscodingConfig	云端混流（转码）配置。
TXVoiceChangerType	变声类型定义（萝莉、大叔、重金属、外国人...）。

版权所有：腾讯云计算（北京）有限责任公司

TXVoiceReverbType	变声类型定义（KTV、小房间、大会堂、低沉、洪亮...）。
AudioMusicParam	音乐和人声设置接口参数。
TRTCAudioRecordingParams	录音参数。
TRTCLocalRecordingParams	录制参数
TRTCPublishCDNParam	CDN转推参数。
CustomLocalRender	本地视频使用外部纹理渲染的参数
CustomRemoteRender	远程视频使用外部纹理渲染的参数
CustomRender	视频使用外部纹理渲染的参数
TRTCPublishMode	媒体流发布模式
TRTCPublishCdnUrl	配置将实时音视频（TRTC）流发布到腾讯云或第三方 CDN
TRTCUser	TRTC 用户的信息，主要包含用户ID和用户所在的房间号
TRTCPublishTarget	配置 TRTC 流的发布目标
TRTCStreamEncoderParam	发布流的相关编码设置，包含分辨率、帧率、关键帧间隔等
Rect	用于描述一些视图的坐标
TRTCVideoFillMode	TRTC 视频视图显示模式的枚举，包括填充模式和适应模式
TRTCVideoStreamType	TRTC 所提供的不同的视频流类型
TRTCVideoLayout	TRTC 混流转码时的视频布局属性配置，包括位置、大小、图层等
TRTCWatermark	TRTC 水印功能的属性配置
TRTCStreamMixingConfig	TRTC 混流转码的相关设置，包括背景颜色、背景图像、所有要混合的视频和音频流的信息以及水印设置等。
TRTCAudioFrame	音视频帧数据类，用于处理和传输音频数据。

视频渲染组件

API	描述
TRTCCloudVideoView	视频视图窗口，用于显示本地视频、远程视频或辅流。

错误码

最近更新时间：2023-08-24 17:55:21

错误码表

基础错误码

符号	值	含义
ERR_NULL	0	无错误

进房相关错误码

`TRTCCloud.enterRoom()` 在进房失败时会触发此类错误码，您可以通过回调函数 `TRTCCloudDelegate.onEnterRoom()` 和 `TRTCCloudDelegate.OnError()` 捕获相关通知。

符号	值	含义
ERR_ROOM_ENTER_FAIL	-3301	进入房间失败
ERR_ENTER_ROOM_PARAM_NULL	-3316	进房参数为空，请检查 TRTCCloud.enterRoom(): 接口调用是否传入有效的 param
ERR_SDK_APPID_INVALID	-3317	进房参数 sdkAppId 错误
ERR_ROOM_ID_INVALID	-3318	进房参数 roomId 错误
ERR_USER_ID_INVALID	-3319	进房参数 userID 不正确
ERR_USER_SIG_INVALID	-3320	进房参数 userSig 不正确
ERR_ROOM_REQUEST_ENTER_ROOM_TIMEOUT	-3308	请求进房超时，请检查网络
ERR_SERVER_INFO_PRIVILEGE_FLAG_ERROR	-100006	权限位校验失败，请检查 privateMapKey 是否正确
ERR_SERVER_INFO_SERVICE_SUSPENDED	-100013	服务不可用。请检查：套餐包剩余分钟数是否大于0，腾讯云账号是否欠费
ERR_SERVER_INFO_ECDH_GET_TINYID	-100018	userSig 校验失败，请检查 userSig 是否正确

退房相关错误码

`TRTCCloud.exitRoom()` 在退房失败时会触发此类错误码，您可以通过回调函数 `TRTCCloudDelegate.OnError()` 捕获相关通知。

符号	值	含义
ERR_ROOM_REQUEST_QUIT_ROOM_TIMEOUT	-3325	请求退房超时

设备（摄像头、麦克风、扬声器）相关错误码

您可以通过回调函数 `TRTCCloudDelegate.OnError()` 捕获相关通知。

符号	值	含义
ERR_CAMERA_START_FAIL	-1301	打开摄像头失败，例如在 Windows 或 Mac 设备，摄像头的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_CAMERA_NOT_AUTHORIZED	-1314	摄像头设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_CAMERA_SET_PARAM_FAIL	-1315	摄像头参数设置出错（参数不支持或其它）
ERR_CAMERA_OCCUPY	-1316	摄像头正在被占用中，可尝试打开其他摄像头
ERR_MIC_START_FAIL	-1302	打开麦克风失败，例如在 Windows 或 Mac 设备，麦克风的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_MIC_NOT_AUTHORIZED	-1317	麦克风设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_MIC_SET_PARAM_FAIL	-1318	麦克风设置参数失败
ERR_MIC_OCCUPY	-1319	麦克风正在被占用中，例如移动设备正在通话时，打开麦克风会失败
ERR_MIC_STOP_FAIL	-1320	停止麦克风失败
ERR_SPEAKER_START_FAIL	-1321	打开扬声器失败，例如在 Windows 或 Mac 设备，扬声器的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序

ERR_SPEAKER_SET_PARAM_FAIL	-1322	扬声器设置参数失败
ERR_SPEAKER_STOP_FAIL	-1323	停止扬声器失败

屏幕分享相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_SCREEN_CAPTURE_START_FAIL	-1308	开始录屏失败，如果在移动设备出现，可能是权限被用户拒绝了，如果在 Windows 或 Mac 系统的设备出现，请检查录屏接口的参数是否符合要求
ERR_SCREEN_CAPTURE_UNSURPORT	-1309	录屏失败，在 Android 平台，需要5.0以上的系统，在 iOS 平台，需要11.0以上的系统
ERR_SERVER_CENTER_NO_PRIVILEDGE_PUSH_SUB_VIDEO	-102015	没有权限上行辅路
ERR_SERVER_CENTER_ANOTHER_USER_PUSH_SUB_VIDEO	-102016	其他用户正在上行辅路
ERR_SCREEN_CAPTURE_STOPPED	-7001	录屏被系统中止

编解码相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_VIDEO_ENCODE_FAIL	-1303	视频帧编码失败，例如 iOS 设备切换到其他应用时，硬编码器可能被系统释放，再切换回来时，硬编码器重启前，可能会抛出
ERR_UNSUPPORTED_RESOLUTION	-1305	不支持的视频分辨率
ERR_AUDIO_ENCODE_FAIL	-1304	音频帧编码失败，例如传入自定义音频数据，SDK 无法处理
ERR_UNSUPPORTED_	-1306	不支持的音频采样率

SAMPLERATE	

自定义采集相关错误码

您可以通过回调函数 TRTCCloudDelegate.OnError() 捕获相关通知。

符号	值	含义
ERR_PIXEL_FORMAT_UNSUPPORTED	-1327	设置的 pixel format 不支持
ERR_BUFFER_TYPE_UNSUPPORTED	-1328	设置的 buffer type 不支持

CDN 绑定和混流相关错误码

您可以通过回调函数 TRTCCloudDelegate.onStartPublishing() 和 TRTCCloudDelegate.onSetMixTranscodingConfig() 捕获相关通知。

符号	值	含义
ERR_PUBLISH_CDN_STREAM_REQUEST_TIME_OUT	-3321	旁路转推请求超时
ERR_CLOUD_MIX_TRANSCODING_REQUEST_TIME_OUT	-3322	云端混流请求超时
ERR_PUBLISH_CDN_STREAM_SERVER_FAILED	-3323	旁路转推回包异常
ERR_CLOUD_MIX_TRANSCODING_SERVER_FAILED	-3324	云端混流回包异常
ERR_ROOM_REQUEST_START_PUBLISHING_TIMEOUT	-3333	开始向腾讯云的直播 CDN 推流信令超时
ERR_ROOM_REQUEST_START_PUBLISHING_ERROR	-3334	开始向腾讯云的直播 CDN 推流信令异常
ERR_ROOM_REQUEST_STOP_PUBLISHING_TIMEOUT	-3335	停止向腾讯云的直播 CDN 推流信令超时
ERR_ROOM_REQUEST_STOP_PUBLISHING_ERROR	-3336	停止向腾讯云的直播 CDN 推流信令异常

跨房连麦相关错误码

版权所有：腾讯云计算（北京）有限责任公司

TRTCCloud.ConnectOtherRoom() 在跨房失败时会触发此类错误码，您可以通过回调函数 TRTCCloudDelegate.onConnectOtherRoom() 捕获相关通知。

符号	值	含义
ERR_ROOM_REQUEST_CONN_ROOM_TIMEOUT	-3326	请求连麦超时
ERR_ROOM_REQUEST_DISCONN_ROOM_TIMEOUT	-3327	请求退出连麦超时
ERR_ROOM_REQUEST_CONN_ROOM_INVALID_PARAM	-3328	无效参数
ERR_CONNECT_OTHER_ROOM_AS_AUDIENCE	-3330	当前是观众角色，不能请求或断开跨房连麦，需要先 switchRole() 到主播
ERR_SERVER_CENTER_CONN_ROOM_NOT_SUPPORT	-102031	不支持跨房间连麦
ERR_SERVER_CENTER_CONN_ROOM_REACH_MAX_NUM	-102032	达到跨房间连麦上限
ERR_SERVER_CENTER_CONN_ROOM_REACH_MAX_RETRY_TIMES	-102033	跨房间连麦重试次数耗尽
ERR_SERVER_CENTER_CONN_ROOM_REQ_TIMEOUT	-102034	跨房间连麦请求超时
ERR_SERVER_CENTER_CONN_ROOM_REQ	-102035	跨房间连麦请求格式错误
ERR_SERVER_CENTER_CONN_ROOM_NO_SIG	-102036	跨房间连麦无签名
ERR_SERVER_CENTER_CONN_ROOM_DECRYPT_SIG	-102037	跨房间连麦签名解密失败
ERR_SERVER_CENTER_CONN_ROOM_NO_KEY	-10203	未找到跨房间连麦签名解密密钥

	8	
ERR_SERVER_CENTER_CONN_ROOM_PARSE_SIG	-10203 9	跨房间连麦签名解析错误
ERR_SERVER_CENTER_CONN_ROOM_INVALID_SIG_TIME	-10204 0	跨房间连麦签名时间戳错误
ERR_SERVER_CENTER_CONN_ROOM_SIG_GROUPID	-10204 1	跨房间连麦签名内房间号不匹配
ERR_SERVER_CENTER_CONN_ROOM_NOT_CONNED	-10204 2	跨房间连麦签名内用户名不匹配
ERR_SERVER_CENTER_CONN_ROOM_USER_NOT_CONNED	-10204 3	本用户未发起连麦
ERR_SERVER_CENTER_CONN_ROOM_FAILED	-10204 4	跨房间连麦失败
ERR_SERVER_CENTER_CONN_ROOM_CANCEL_FAILED	-10204 5	取消跨房间连麦失败
ERR_SERVER_CENTER_CONN_ROOM_CONNED_ROOM_NOT_EXIST	-10204 6	被连麦房间不存在
ERR_SERVER_CENTER_CONN_ROOM_CONNED_REACH_MAX_ROOM	-10204 7	被连麦房间达到连麦上限
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_NOT_EXIST	-10204 8	被连麦用户不存在
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_DELETED	-10204 9	被连麦用户已被删除
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_FULL	-10205	被连麦用户达到资源上限

版权所有：腾讯云计算（北京）有限责任公司

	0	
ERR_SERVER_CENTER_CONN_ROOM_INVALID_SEQ	-10205 1	连麦请求序号错乱

警告码表

警告码无需特别关注，您可以按需选择是否对当前用户进行相应的提示。

符号	值	含义
WARNING_HW_ENCODER_START_FAIL	1103	硬编码启动出现问题，自动切换到软编码
WARNING_VIDEO_ENCODER_SW_TO_HW	1107	当前 CPU 使用率太高，无法满足软件编码需求，自动切换到硬件编码
WARNING_INSUFFICIENT_CAPTURE_FPS	1108	摄像头采集帧率不足，部分自带美颜算法的 Android 手机上会出现
WARNING_SW_ENCODER_START_FAIL	1109	软编码启动失败
WARNING_REDUCE_CAPTURE_RESOLUTION	1110	摄像头采集分辨率被降低，以满足当前帧率和性能最优解。
WARNING_CAMERA_DEVICE_EMPTY	1111	没有检测到可用的摄像头设备
WARNING_CAMERA_NOT_AUTHORIZED	1112	用户未授权当前应用使用摄像头
WARNING_MICROPHONE_DEVICE_EMPTY	1201	没有检测到可用的麦克风设备
WARNING_SPEAKER_DEVICE_EMPTY	1202	没有检测到可用的扬声器设备
WARNING_MICROPHONE_NOT_AUTHORIZED	1203	用户未授权当前应用使用麦克风
WARNING_MICROPHONE_DEVICE_ABNORMAL	1204	音频采集设备不可用（例如被占用）
WARNING_SPEAKER_DEVICE_ABNORMAL	1205	音频播放设备不可用（例如被占用）

WARNING_VIDEO_FRAME_DECODE_FAIL	2101	当前视频帧解码失败
WARNING_AUDIO_FRAME_DECODE_FAIL	2102	当前音频帧解码失败
WARNING_VIDEO_PLAY_LAG	2105	当前视频播放出现卡顿
WARNING_HW_DECODER_START_FAIL	2106	硬解启动失败，采用软解码
WARNING_VIDEO_DECODER_HW_TO_SW	2108	当前流硬解第一个 I 帧失败，SDK 自动切软解
WARNING_SW_DECODER_START_FAIL	2109	软解码器启动失败
WARNING_VIDEO_RENDER_FAIL	2110	视频渲染失败
WARNING_START_CAPTURE_IGNORED	4000	已经在采集，启动采集被忽略
WARNING_AUDIO_RECORDING_WRITE_FAIL	7001	音频录制写入文件失败
WARNING_ROOM_DISCONNECT	5101	网络断开连接
WARNING_IGNORE_UPSTREAM_FOR_AUDIENCE	6001	当前是观众角色，忽略上行音视频数据
WARNING_NET_BUSY	1101	网络状况不佳：上行带宽太小，上传数据受阻
WARNING_RTMP_SERVER_RECONNECT	1102	直播推流，网络断连, 已启动自动重连（自动重连连续失败超过三次会放弃）
WARNING_LIVE_STREAM_SERVER_RECONNECT	2103	直播拉流，网络断连, 已启动自动重连（自动重连连续失败超过三次会放弃）
WARNING_RECV_DATA_LAG	2104	网络来包不稳：可能是下行带宽不足，或由于主播端出流不均匀
WARNING_RTMP_DNS_FAIL	3001	直播，DNS 解析失败

WARNING_RTMP_SEVER_CONN_FAIL	3002	直播，服务器连接失败
WARNING_RTMP_SHAKE_FAIL	3003	直播，与 RTMP 服务器握手失败
WARNING_RTMP_SERVER_BREAK_CONNECT	3004	直播，服务器主动断开
WARNING_RTMP_READ_WRITE_FAIL	3005	直播，RTMP 读/写失败，将会断开连接
WARNING_RTMP_WRITE_FAIL	3006	直播，RTMP 写失败（SDK 内部错误码，不会对外抛出）
WARNING_RTMP_READ_FAIL	3007	直播，RTMP 读失败（SDK 内部错误码，不会对外抛出）
WARNING_RTMP_NO_DATA	3008	直播，超过30s 没有数据发送，主动断开连接
WARNING_PLAY_LIVE_STREAM_INFO_CONNECT_FAIL	3009	直播，connect 服务器调用失败（SDK 内部错误码，不会对外抛出）
WARNING_NO_STEAM_SOURCE_FAIL	3010	直播，连接失败，该流地址无视频（SDK 内部错误码，不会对外抛出）
WARNING_ROOM_RECONNECT	5102	网络断连，已启动自动重连
WARNING_ROOM_NET_BUSY	5103	网络状况不佳：上行带宽太小，上传数据受阻

Uni-App API 概览

最近更新时间：2023-01-06 10:15:32

TRTCCloud

基础方法

API	描述
createInstance	创建 TrtcCloud 单例。
destroyInstance	销毁 TrtcCloud 单例。
on	设置 TrtcCloud 事件监听。
off	移除 TrtcCloud 事件监听。

房间相关接口函数

API	描述
enterRoom	进入房间。
exitRoom	离开房间。
switchRole	切换角色，仅适用于直播场景（TRTCAppSceneLIVE 和 TRTCAppSceneVoiceChatRoom）。

视频相关接口函数

API	描述
startLocalPreview	开启本地视频的预览画面。
stopLocalPreview	停止本地视频采集及预览。
muteLocalVideo	暂停/恢复推送本地的视频数据。
startRemoteView	开始显示远端视频画面。
stopRemoteView	停止显示远端视频画面，同时不再拉取该远端用户的视频数据流。

版权所有：腾讯云计算（北京）有限责任公司

setLocalRenderParams	设置本地图像的渲染模式。
setRemoteRenderParams	设置远端图像相关参数。
snapshotVideo	视频画面截图。

音频相关接口函数

API	描述
startLocalAudio	开启本地音频的采集和上行。
stopLocalAudio	关闭本地音频的采集和上行。
muteLocalAudio	静音/取消静音本地的音频。
setAudioRoute	设置音频路由。
muteRemoteAudio	静音/取消静音指定的远端用户的声音。
muteAllRemoteAudio	静音/取消静音所有用户的声音。
enableAudioVolumeEvaluation	启用音量大小提示。

美颜滤镜相关接口函数

API	描述
setBeautyLevel	设置美颜级别。
setBeautyStyle	设置美颜（磨皮）算法 TRTC 内置多种不同的磨皮算法，您可以选择最适合您产品定位的方案。

背景音效相关接口函数

API	描述
startPlayMusic	开始播放背景音乐 每个音乐都需要您指定具体的 ID，您可以通过该 ID 对音乐的开始、停止、音量等进行设置。。
stopPlayMusic	停止播放背景音乐。
pausePlayMusic	暂停播放背景音乐。

resumePlayMusic	恢复播放背景音乐。

辅流相关接口函数

API	描述
startScreenCapture	启动屏幕分享。
stopScreenCapture	停止屏幕采集。
pauseScreenCapture	暂停屏幕分享。
resumeScreenCapture	恢复屏幕分享。

TRTCCloudListener

腾讯云视频通话功能的事件回调接口。

错误事件和警告事件

API	描述
onError	错误回调，表示 SDK 不可恢复的错误，一定要监听并分情况给用户适当的界面提示。
onWarning	警告回调，用于告知您一些非严重性问题，例如出现卡顿或者可恢复的解码失败。

房间事件回调

API	描述
onEnterRoom	已加入房间的回调。
onExitRoom	离开房间的事件回调。
onSwitchRole	切换角色的事件回调。

成员事件回调

API	描述
onRemoteUserEnterRoom	有用户加入当前房间。
onRemoteUserLeaveRoom	有用户离开当前房间。

版权所有：腾讯云计算（北京）有限责任公司

onUserVideoAvailable	远端用户是否存在可播放的主路画面（一般用于摄像头）。
onUserSubStreamAvailable	远端用户是否存在可播放的辅路画面（一般用于屏幕分享）。
onUserAudioAvailable	远端用户是否存在可播放的音频数据。
onFirstVideoFrame	开始渲染本地或远程用户的首帧画面。
onFirstAudioFrame	开始播放远程用户的首帧音频（本地声音暂不通知）。
onSendFirstLocalVideoFrame	首帧本地视频数据已经被送出。
onSendFirstLocalAudioFrame	首帧本地音频数据已经被送出。

硬件设备事件回调

API	描述
onUserVoiceVolume	用于提示音量大小的回调，包括每个 userId 的音量和远端总音量。

屏幕分享回调

API	描述
onScreenCaptureStarted	当屏幕分享开始时，SDK 会通过此回调通知
onScreenCapturePaused	当屏幕分享调用 pauseScreenCapture() 暂停时，SDK 会通过此回调通知。
onScreenCaptureResumed	当屏幕分享调用 resumeScreenCapture() 恢复时，SDK 会通过此回调通知。
onScreenCaptureStopped	当屏幕分享停止时，SDK 会通过此回调通知。

截图回调

API	描述
onSnapshotComplete	截图完成时回调。

背景音效回调

API	描述
onStart	背景音乐开始播放。

onPlayProgress	背景音乐的播放进度。
onComplete	背景音乐已经播放完毕。

关键类型定义

类名	描述
TRTCParams	进房参数。
TRTCVideoEncParam	视频编码参数。
TRTCRenderParams	远端图像参数。

错误码

最近更新时间：2023-09-18 18:08:02

错误码表

基础错误码

符号	值	含义
ERR_NULL	0	无错误

进房相关错误码

`TrtcCloud.enterRoom()` 在进房失败时会触发此类错误码，您可以通过回调函数 `TrtcCloud.onEnterRoom()` 和 `TrtcCloud.OnError()` 捕获相关通知。

符号	值	含义
ERR_ROOM_ENTER_FAIL	-3301	进入房间失败
ERR_ENTER_ROOM_PARAM_NULL	-3316	进房参数为空，请检查 `TRTCCloud.enterRoom()` 接口调用是否传入有效的 param
ERR_SDK_APPID_INVALID	-3317	进房参数 sdkAppId 错误
ERR_ROOM_ID_INVALID	-3318	进房参数 roomId 错误
ERR_USER_ID_INVALID	-3319	进房参数 userID 不正确
ERR_USER_SIG_INVALID	-3320	进房参数 userSig 不正确
ERR_ROOM_REQUEST_ENTER_ROOM_TIMEOUT	-3308	请求进房超时，请检查网络
ERR_SERVER_INFO_PRIVILEGE_FLAG_ERROR	-100006	权限位校验失败，请检查 privateMapKey 是否正确
ERR_SERVER_INFO_SERVICE_SUSPENDED	-10001	服务不可用。请检查：套餐包剩余分钟数是否大于0，腾讯云账号是否欠费

	3	
ERR_SERVER_INFO_ECDH_GET_TINYID	-10001018	userSig 校验失败，请检查 userSig 是否正确

退房相关错误码

`TrtcCloud.exitRoom()` 在退房失败时会触发此类错误码，您可以通过回调函数 `TrtcCloud.OnError()` 捕获相关通知。

符号	值	含义
ERR_ROOM_REQUEST_QUIT_ROOM_TIMEOUT	-3325	请求退房超时

设备（摄像头、麦克风、扬声器）相关错误码

您可以通过回调函数 TrtcCloud.OnError() 捕获相关通知。

符号	值	含义
ERR_CAMERA_START_FAIL	-1301	打开摄像头失败，例如在 Windows 或 Mac 设备，摄像头的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_CAMERA_NOT_AUTHORIZED	-1314	摄像头设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_CAMERA_SET_PARAM_FAIL	-1315	摄像头参数设置出错（参数不支持或其它）
ERR_CAMERA_OCCUPY	-1316	摄像头正在被占用中，可尝试打开其他摄像头
ERR_MIC_START_FAIL	-1302	打开麦克风失败，例如在 Windows 或 Mac 设备，麦克风的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_MIC_NOT_AUTHORIZED	-1317	麦克风设备未授权，通常在移动设备出现，可能是权限被用户拒绝了
ERR_MIC_SET_PARAM_FAIL	-1318	麦克风设置参数失败
ERR_MIC_OCCUPY	-1319	麦克风正在被占用中，例如移动设备正在通话时，打开麦克风会失败

ERR_MIC_STOP_FAIL	-1320	停止麦克风失败
ERR_SPEAKER_START_FAIL	-1321	打开扬声器失败，例如在 Windows 或 Mac 设备，扬声器的配置程序（驱动程序）异常，禁用后重新启用设备，或者重启机器，或者更新配置程序
ERR_SPEAKER_SET_PARAM_FAIL	-1322	扬声器设置参数失败
ERR_SPEAKER_STOP_FAIL	-1323	停止扬声器失败

屏幕分享相关错误码

您可以通过回调函数 `TrtcCloud.OnError()` 捕获相关通知。

符号	值	含义
ERR_SCREEN_CAPTURE_START_FAIL	-1308	开始录屏失败，如果在移动设备出现，可能是权限被用户拒绝了，如果在 Windows 或 Mac 系统的设备出现，请检查录屏接口的参数是否符合要求
ERR_SCREEN_CAPTURE_UNSURPORT	-1309	录屏失败，在 Android 平台，需要5.0以上的系统，在 iOS 平台，需要11.0以上的系统
ERR_SERVER_CENTER_NO_PRIVILEDGE_PUSH_SUB_VIDEO	-102015	没有权限上行辅路
ERR_SERVER_CENTER_ANOTHER_USER_PUSH_SUB_VIDEO	-102016	其他用户正在上行辅路
ERR_SCREEN_CAPTURE_STOPPED	-7001	录屏被系统中止

编解码相关错误码

您可以通过回调函数 `TrtcCloud.OnError()` 捕获相关通知。

符号	值	含义
ERR_VIDEO_ENCODE_FAIL	-1303	视频帧编码失败，例如 iOS 设备切换到其他应用时，硬编码器可能被系统释放，再切换回来时，硬编码器重启前，可能会抛出

ERR_UNSUPPORTED_RESOLUTION	-1305	不支持的视频分辨率
ERR_AUDIO_ENCODE_FAIL	-1304	音频帧编码失败，例如传入自定义音频数据，SDK 无法处理
ERR_UNSUPPORTED_SAMPLERATE	-1306	不支持的音频采样率

自定义采集相关错误码

您可以通过回调函数 TrtcCloud.OnError() 捕获相关通知。

符号	值	含义
ERR_PIXEL_FORMAT_UNSUPPORTED	-1327	设置的 pixel format 不支持
ERR_BUFFER_TYPE_UNSUPPORTED	-1328	设置的 buffer type 不支持

CDN 绑定和混流相关错误码

您可以通过回调函数 TrtcCloud.onStartPublishing() 和 TrtcCloud.onSetMixTranscodingConfig() 捕获相关通知。

符号	值	含义
ERR_PUBLISH_CDN_STREAM_REQUEST_TIME_OUT	-3321	旁路转推请求超时
ERR_CLOUD_MIX_TRANSCODING_REQUEST_TIME_OUT	-3322	云端混流请求超时
ERR_PUBLISH_CDN_STREAM_SERVER_FAILED	-3323	旁路转推回包异常
ERR_CLOUD_MIX_TRANSCODING_SERVER_FAILED	-3324	云端混流回包异常
ERR_ROOM_REQUEST_START_PUBLISHING_TIMEOUT	-3333	开始向腾讯云的直播 CDN 推流信令超时
ERR_ROOM_REQUEST_START_PUBLISHING_ERROR	-3334	开始向腾讯云的直播 CDN 推流信令异常
ERR_ROOM_REQUEST_STOP_PUBLISHI	-333	停止向腾讯云的直播 CDN 推流信令超

版权所有：腾讯云计算（北京）有限责任公司

NG_TIMEOUT	5	时
ERR_ROOM_REQUEST_STOP_PUBLISHING_ERROR	-3336	停止向腾讯云的直播 CDN 推流信令异常

跨房连麦相关错误码

`TrtcCloud.ConnectOtherRoom()` 在跨房失败时会触发此类错误码，您可以通过回调函数 `TrtcCloud.onConnectOtherRoom()` 捕获相关通知。

符号	值	含义
ERR_ROOM_REQUEST_CONN_ROOM_TIMEOUT	-3326	请求连麦超时
ERR_ROOM_REQUEST_DISCONN_ROOM_TIMEOUT	-3327	请求退出连麦超时
ERR_ROOM_REQUEST_CONN_ROOM_INVALID_PARAM	-3328	无效参数
ERR_CONNECT_OTHER_ROOM_AS_AUDIENCE	-3330	当前是观众角色，不能请求或断开跨房连麦，需要先 switchRole() 到主播
ERR_SERVER_CENTER_CONN_ROOM_NOT_SUPPORT	-102031	不支持跨房间连麦
ERR_SERVER_CENTER_CONN_ROOM_REACH_MAX_NUM	-102032	达到跨房间连麦上限
ERR_SERVER_CENTER_CONN_ROOM_REACH_MAX_RETRY_TIMES	-102033	跨房间连麦重试次数耗尽
ERR_SERVER_CENTER_CONN_ROOM_REQ_TIMEOUT	-102034	跨房间连麦请求超时
ERR_SERVER_CENTER_CONN_ROOM_REQ	-102035	跨房间连麦请求格式错误
ERR_SERVER_CENTER_CONN_ROOM_NO_SIG	-102036	跨房间连麦无签名
ERR_SERVER_CENTER_CONN_ROOM_DECRYPT_SIG	-102037	跨房间连麦签名解密失败
ERR_SERVER_CENTER_CONN_ROOM_NO_KEY	-102038	未找到跨房间连麦签名解密密钥

ERR_SERVER_CENTER_CONN_ROOM_PARSE_SIG	-102039	跨房间连麦签名解析错误
ERR_SERVER_CENTER_CONN_ROOM_INVALID_SIG_TIME	-102040	跨房间连麦签名时间戳错误
ERR_SERVER_CENTER_CONN_ROOM_SIG_GROUPID	-102041	跨房间连麦签名内房间号不匹配
ERR_SERVER_CENTER_CONN_ROOM_NOT_CONNED	-102042	跨房间连麦签名内用户名不匹配
ERR_SERVER_CENTER_CONN_ROOM_USER_NOT_CONNED	-102043	本用户未发起连麦
ERR_SERVER_CENTER_CONN_ROOM_FAILED	-102044	跨房间连麦失败
ERR_SERVER_CENTER_CONN_ROOM_CANCEL_FAILED	-102045	取消跨房间连麦失败
ERR_SERVER_CENTER_CONN_ROOM_CONNED_ROOM_NOT_EXIST	-102046	被连麦房间不存在
ERR_SERVER_CENTER_CONN_ROOM_CONNED_REACH_MAX_ROOM	-102047	被连麦房间达到连麦上限
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_NOT_EXIST	-102048	被连麦用户不存在
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_DELETED	-102049	被连麦用户已被删除
ERR_SERVER_CENTER_CONN_ROOM_CONNED_USER_FULL	-102050	被连麦用户达到资源上限
ERR_SERVER_CENTER_CONN_ROOM_INVALID_SEQ	-102051	连麦请求序号错乱

警告码表

警告码无需特别关注，您可以按需选择是否对当前用户进行相应的提示。

符号	值	含义
WARNING_HW_ENCODER_START_FAIL	1103	硬编码启动出现问题，自动切换到软编码

WARNING_VIDEO_ENCODER_SW_TO_HW	1107	当前 CPU 使用率太高，无法满足软件编码需求，自动切换到硬件编码
WARNING_INSUFFICIENT_CAPTURE_FPS	1108	摄像头采集帧率不足，部分自带美颜算法的 Android 手机上会出现
WARNING_SW_ENCODER_START_FAIL	1109	软编码启动失败
WARNING_REDUCE_CAPTURE_RESOLUTION	1110	摄像头采集分辨率被降低，以满足当前帧率和性能最优解。
WARNING_CAMERA_DEVICE_EMPTY	1111	没有检测到可用的摄像头设备
WARNING_CAMERA_NOT_AUTHORIZED	1112	用户未授权当前应用使用摄像头
WARNING_MICROPHONE_DEVICE_EMPTY	1201	没有检测到可用的麦克风设备
WARNING_SPEAKER_DEVICE_EMPTY	1202	没有检测到可用的扬声器设备
WARNING_MICROPHONE_NOT_AUTHORIZED	1203	用户未授权当前应用使用麦克风
WARNING_MICROPHONE_DEVICE_ABNORMAL	1204	音频采集设备不可用（例如被占用）
WARNING_SPEAKER_DEVICE_ABNORMAL	1205	音频播放设备不可用（例如被占用）
WARNING_VIDEO_FRAME_DECODE_FAIL	2101	当前视频帧解码失败
WARNING_AUDIO_FRAME_DECODE_FAIL	2102	当前音频帧解码失败
WARNING_VIDEO_PLAY_LAG	2105	当前视频播放出现卡顿
WARNING_HW_DECODER_START_FAIL	2106	硬解启动失败，采用软解码
WARNING_VIDEO_DECODER_HW_TO_SW	2108	当前流硬解第一个 I 帧失败，SDK 自动切软解

WARNING_SW_DECODER_START_FAIL	2109	软解码器启动失败
WARNING_VIDEO_RENDER_FAIL	2110	视频渲染失败
WARNING_START_CAPTURE_IGNORED	4000	已经在采集，启动采集被忽略
WARNING_AUDIO_RECORDING_WRITE_FAIL	7001	音频录制写入文件失败
WARNING_ROOM_DISCONNECT	5101	网络断开连接
WARNING_IGNORE_UPSTREAM_FOR_AUDIENCE	6001	当前是观众角色，忽略上行音视频数据
WARNING_NET_BUSY	1101	网络状况不佳：上行带宽太小，上传数据受阻
WARNING_RTMP_SERVER_RECONNECT	1102	直播推流，网络断连, 已启动自动重连（自动重连连续失败超过三次会放弃）
WARNING_LIVE_STREAM_SERVER_RECONNECT	2103	直播拉流，网络断连, 已启动自动重连（自动重连连续失败超过三次会放弃）
WARNING_RECV_DATA_LAG	2104	网络来包不稳：可能是下行带宽不足，或由于主播端出流不均匀
WARNING_RTMP_DNS_FAIL	3001	直播，DNS 解析失败
WARNING_RTMP_SEVER_CONN_FAIL	3002	直播，服务器连接失败
WARNING_RTMP_SHAKE_FAIL	3003	直播，与 RTMP 服务器握手失败
WARNING_RTMP_SERVER_BREAK_CONNECT	3004	直播，服务器主动断开
WARNING_RTMP_READ_WRITE_FAIL	3005	直播，RTMP 读/写失败，将会断开连接
WARNING_RTMP_WRITE_FAIL	3006	直播，RTMP 写失败（SDK 内部错误码，不会对外抛出）

WARNING_RTMP_READ_FAIL	3007	直播，RTMP 读失败（SDK 内部错误码，不会对外抛出）
WARNING_RTMP_NO_DATA	3008	直播，超过30s 没有数据发送，主动断开连接
WARNING_PLAY_LIVE_STREAM_INFO_CONNECT_FAIL	3009	直播，connect 服务器调用失败（SDK 内部错误码，不会对外抛出）
WARNING_NO_STEAM_SOURCE_FAIL	3010	直播，连接失败，该流地址无视频（SDK 内部错误码，不会对外抛出）
WARNING_ROOM_RECONNECT	5102	网络断连，已启动自动重连
WARNING_ROOM_NET_BUSY	5103	网络状况不佳：上行带宽太小，上传数据受阻

版权所有：腾讯云计算（北京）有限责任公司