# Audio and video Terminal Engine

# FAQs

# Contents



# FAQs
# Billing

Last updated：2025-03-17 15:16:43

## How Does the Audio and Video Terminal SDK (Tencent Cloud RT-Cube) Charge?

The Audio and Video Terminal SDK (Tencent Cloud Video Cube) unlocks the authorization of corresponding feature modules in the SDK by purchasing corresponding cloud service resource packages. For details, see Billing Information.

## Does the Audio and Video Terminal SDK (Tencent Cloud RT-Cube) Incur Any Other Charges Besides the License Fee During Use?

Audio and video terminal SDK (Tencent Cloud RT-Cube). Depending on the version features used in your business, services of other cloud products may be involved. During use, if resources related to CSS, VOD, TRTC, Chat <Product Name> (IM), etc. are used, corresponding fees will be charged. For details, see Billing Overview.

## Why Can'T I Use the User Generated Short Video SDK After Purchasing the On-Demand Resource Pack?

You need to purchase an on-demand resource pack of **10 TB or more** to obtain the corresponding License and access to the UGSV SDK.

## Packages to Purchase For Using the User Generated Short Video SDK

You can purchase a 10TB Video on Demand traffic package and get a short video SDK Lite version License for free. For 50TB, 200TB, and 1PB packages, you can get a short video SDK Basic version License for free.

## Are Refunds Supported For Live Stream/On-Demand Resource Packages?

If your live stream / on-demand video resource package has not been used within five days and the License has not been bound, it can be refunded within 5 days. For more details, see CSS Refunds and VOD Refunds.

## Can I Get a Free Basic Version License of Short Video SDK When Purchasing a 10TB Cloud Video On Demand Resource Package?

For a 10 T VOD resource pack, only the UGSV SDK Lite License is provided. To obtain the right to use the UGSV SDK Basic License, you need to purchase a traffic package of 50 TB / 200 TB / 1 PB.

## What Are the Purchased Live Stream/Video-On-Demand Resource Packages For Unlocking Tencent Cloud RT-Cube License Used For?

Live stream / on-demand video resource packages can be used to deduct the traffic / bandwidth usage generated by live stream / on-demand video viewing.

## Tencent Cloud Account in Arrears Will Affect My Audio and Video Terminal SDK (Tencent Cloud RT-Cube)?

The valid period of 1 year for the Tencent Cloud RT-Cube License can be obtained by purchasing relevant cloud service resource packages. It takes effect immediately upon purchase, and arrears will not affect the use of the License. If your Tencent Cloud account is in arrears, it will cause relevant cloud products to stop services. To avoid the business of cloud products related to the audio and video terminal SDK (Tencent Cloud RT-Cube) being affected, please top up your account as soon as possible.

# License

Last updated: 2025-03-17 15:16:52

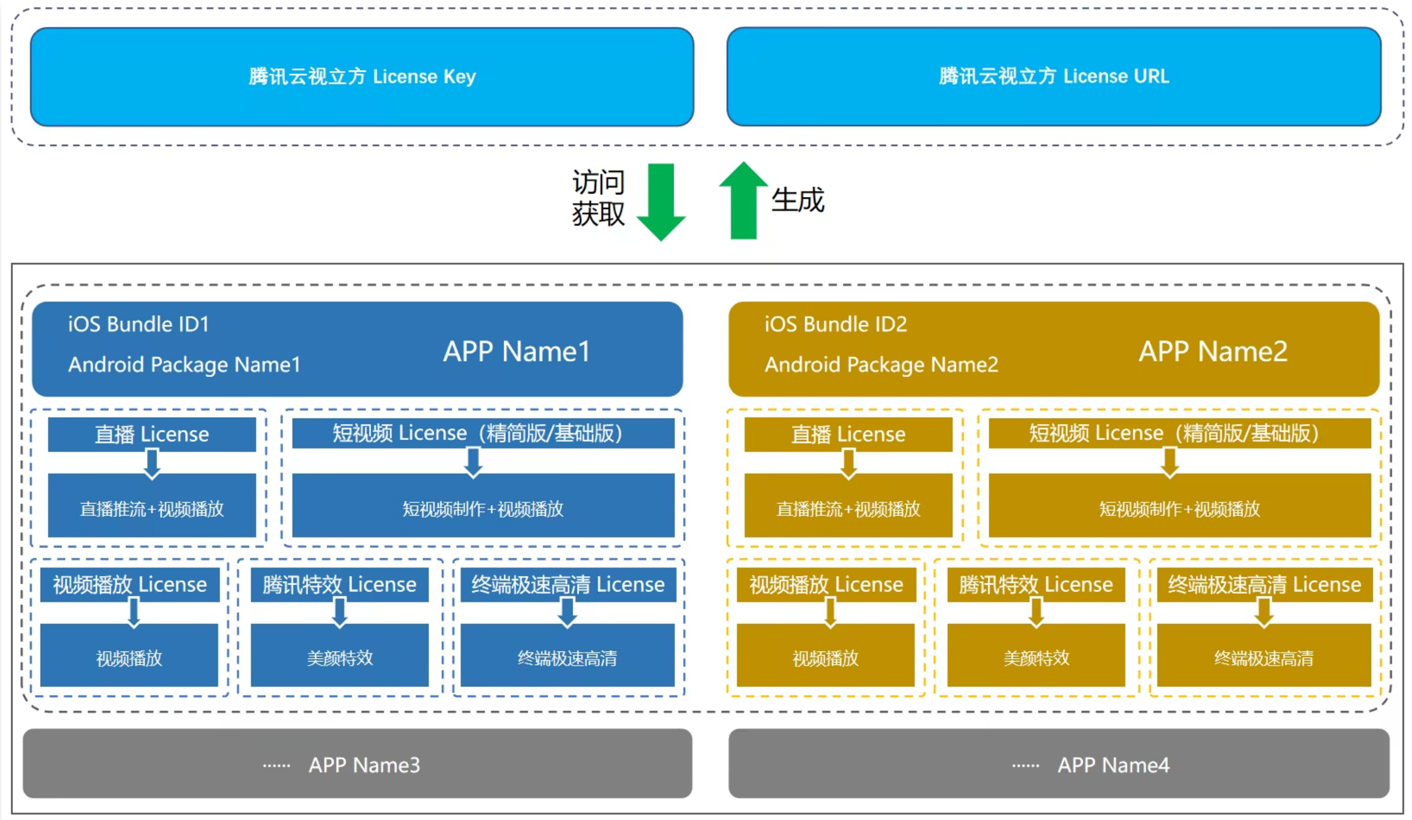
## Is the Tencent Cloud RT-Cube License Mandatory to Purchase?

If the Tencent Cloud RT-Cube feature modules you downloaded include live streaming (anchor broadcasting and anchor-audience mic connection / anchor cross-room PK), short video (video recording and editing / video uploading and publishing), terminal Top Speed Codec (TSC) and Special Effect SDK feature modules, you should unlock them for free by obtaining a License through purchasing the corresponding cloud service resource package. For details about unlocking feature modules, see SDK Download.

## Is There a Separate Purchase Entrance For the Tencent Cloud RT-Cube License?

- Unlock the functional modules of **Live Streaming Push**, **Short Video Production** and **Video Playback**. You can **get a 1-year validity period (counting from the purchase date) of live streaming License, short video License and player License for free by purchasing the corresponding resource package of cloud services**, or **purchase independent** live streaming License, short video License and player License. For details of relevant License billing, see Tencent Cloud Video Cube Price Overview. Click Buy Now.
- **Terminal Top Speed Codec Transcoding** module is in the trial period and there is no official version License available. No purchase is required. If needed, you can use Trial License.
- Unlocking the **Tencent Effect** feature module requires purchasing the Tencent Effect SDK package to obtain the corresponding Tencent Effect License usage authorization (valid from the purchase date until 00:00:00 the day after expiration). Click to go to the Tencent Effect SDK purchase page to buy.

## What Is the Difference Between the Tencent Cloud RT-Cube License and the Feature Module License?




| Type | Description |
| --- | --- |
| Tencent Cloud RT-Cube License | Obtain and verify the authorization of feature modules under an application through a set of License URLs and Keys, and manage the unlocking and use of live stream publishing license, short video license (Lite and Basic Edition) and player License feature modules under this application. |
| Feature module License | Including live stream publishing license, short video license and player License. It is to obtain License authorization for free use within a valid period of 1 year and unlock feature module authorization by purchasing the corresponding cloud service resource package, or purchase independent License authorization. |
| Live stream publishing License | (RTMP streaming + RTC streaming) can be used to enable the live stream publishing (anchor starts live streaming and mic connection between anchor and audience/anchor cross-room PK) feature module. The short video License (simplified edition/basic version) can be used to enable the short video (video recording and editing/video upload and publish) feature module. |

## What Is the Difference Between the Short Video Simplified Edition License and the Short Video Basic Version License?

The short video License includes the simplified edition License and the basic version License.

- The simplified edition License supports multiple features such as video generation, upload, processing, distribution and playback.
- The basic version License adds capabilities such as filters, special effects and transitions on the basis of the simplified edition, quickly and easily implementing mobile-based short video applications.

> **Note:**
> For more detailed descriptions of additional features, see Short Video License Feature Details.

## Can Multiple Tencent Cloud RT-Cube Licenses Be Created Under One Account?

A Tencent Cloud RT-Cube project is managed as a whole as a License for application management. One Tencent Cloud RT-Cube License corresponds to one Bundle ID and Package Name, managing live streaming, short video production (Lite/Basic version), video playback, terminal ultra-fast HD, and Special Effect features under this application.
There is no limit on the quantity of Tencent Cloud RT-Cube Licenses created under the same account, and multiple application projects can be managed. For ease of user management, it is recommended to extend the validity time of Tencent Cloud RT-Cube Licenses with the same package name by renewing them.

> **Note:**
> **Package Name** is the package name for Android, and **Bundle Id** is the package name for iOS.

## Can Multiple Tencent Cloud RT-Cube Licenses Be Created With the Same Package Name?

It will not affect the use to fill in the same package name for multiple Tencent Cloud RT-Cube Licenses. Generally, it is not recommended to create multiple Licenses with the same package name.

## Can the License of Tencent Cloud RT-Cube Modify the Bundle ID and Package Name?

The package name information of the official version License of Tencent Cloud RT-Cube cannot be modified. The package name of the trail version License of Tencent Cloud RT-Cube can be edited and changed. Please check whether the package name is occupied in the app store before adding the official version License of Tencent Cloud RT-Cube. It cannot be modified or replaced after submission.

## When Does Tencent Cloud RT-Cube License Expire?

- **About Live Stream License, Short Video License and Player License**:
  - **Official Version License:** The expiration time depends on the valid period of its bound feature module License. The latest valid period among any bound feature module License is the valid period of the official version License.
    - If the bound License is a feature module License given away with the purchase of a resource package, **calculated from the date of resource package purchase**, the authorization is valid until 00:00:00 the day after 1 year expires.
    - If it is an independent License, the activation validity period of the independent Tencent Cloud RT-Cube License binding package name. **Calculated from the binding package name**, the authorization validity period expires at 00:00:00 the day after 1 year.
  - **Trail Version License:** Each functional module can only apply for a trail License once, and each functional module can be renewed for free once (the Terminal TSC does not support self-renewal. If you need to modify the validity period, please contact the business or submit a ticket). For long-term use, it is recommended to purchase an official License. If you apply for a trail renewal during the trial period, the expiration time of the renewal will be based on the time of applying for the trail; if you apply for a trail renewal after the trial period ends, the expiration time of the renewal will be based on the time of applying for the trail renewal.

> **Example:**
> - When the application test start time is `2021-08-12 10:28:41`, the expiration time after 14 days is `2021-08-26 10:28:41`.
> - When you can renew for free once, if you apply for renewal within 14 days of the trial period, the expiration time is `2021-09-09 10:28:41`; if you apply for renewal after the 14-day trial period ends, and the application time is `2021-08-30 22:26:20`, then the expiration time of the renewal is `2021-09-13 22:26:20`.

- **About Special Effect License of Tencent** (**The Special Effect feature module of Tencent needs to pass the review before authorization can be issued**):
  - **Official Version License:** The authorization expiration time is calculated from the approval time and expires at 00:00:00 the day after 1 year.
  - **Trail version License:** The authorization expiration time is based on the approval time; if a test renewal is applied for after the trial period ends, the renewal expiration time is

based on the application time for the test renewal.

> **Example:**
> - **Official Version License**: After creating a License and submitting the review information for the official version of Special Effect Module, it enters **under review status**. The review usually takes 1 – 2 working days. If the submission time is `2022-05-24 12:34:11` and the approval time is `2022-05-25 17:56:24`, then the start time is `2022-05-25 17:56:24`, and the expiration time after 1 year is `2023-05-26 00:00:00`.
> - **Trail Version License** :
>   - When submitting the review information for the trail version of Special Effect Module, it enters **under review status**. The review usually takes 1 – 2 working days. If the submission time is `2022-05-24 12:47:33` and the approval time is `2022-05-24 15:23:46`, then the start time is `2022-05-24 15:23:46`, and the expiration time 14 days later is `2022-06-09 00:00:00`.
>   - When you can renew for free once, if you apply for renewal within the first 14 days of the trial period, the expiration time will be `2022-06-23 00:00:00`; if you apply for renewal after the first 14 days of the trial period, for example, at `2022-08-06 22:26:20`, the expiration time of the renewal will be `2022-08-22 00:00:00`.

## Created Multiple Tencent Cloud RT-Cube Licenses, Are the License URL and License Key the Same?

The Tencent Cloud RT-Cube Licenses under the same account have the same default License URL and License Key for easy maintenance and management.

## Is the Resource Package Associated With the License Only Available For This License?

The daily traffic settlement postpaid consumption generated by the live streaming playback domain under this account can be deducted. The association of the resource package is only used to synchronize the validity period. The traffic in it is not limited to License usage (the usage of the License will not be affected even if the traffic is used up).

**For example:**

User A is billed by daily traffic settlement postpaid. He purchased a 10TB live streaming playback domain traffic resource package and a 50TB live streaming playback domain traffic resource package, and created License A and License B respectively:

- License A corresponds to an App using the domain name `abc.com` for playback, generating 20 TB of playback traffic.

- License B corresponds to an App using the domain name `def.com` for playback, generating 30 TB of playback traffic.

As long as `abc.com` and `def.com` are the playback domains of live streaming under user A's Cloud Streaming Services account, the purchased 10TB + 50TB resource package can be deducted. After the deduction, there will be 10TB of traffic left in user A's live streaming traffic resource package.

## Can Purchasing a Live Stream License Be Used For Mini Program Live Streams?

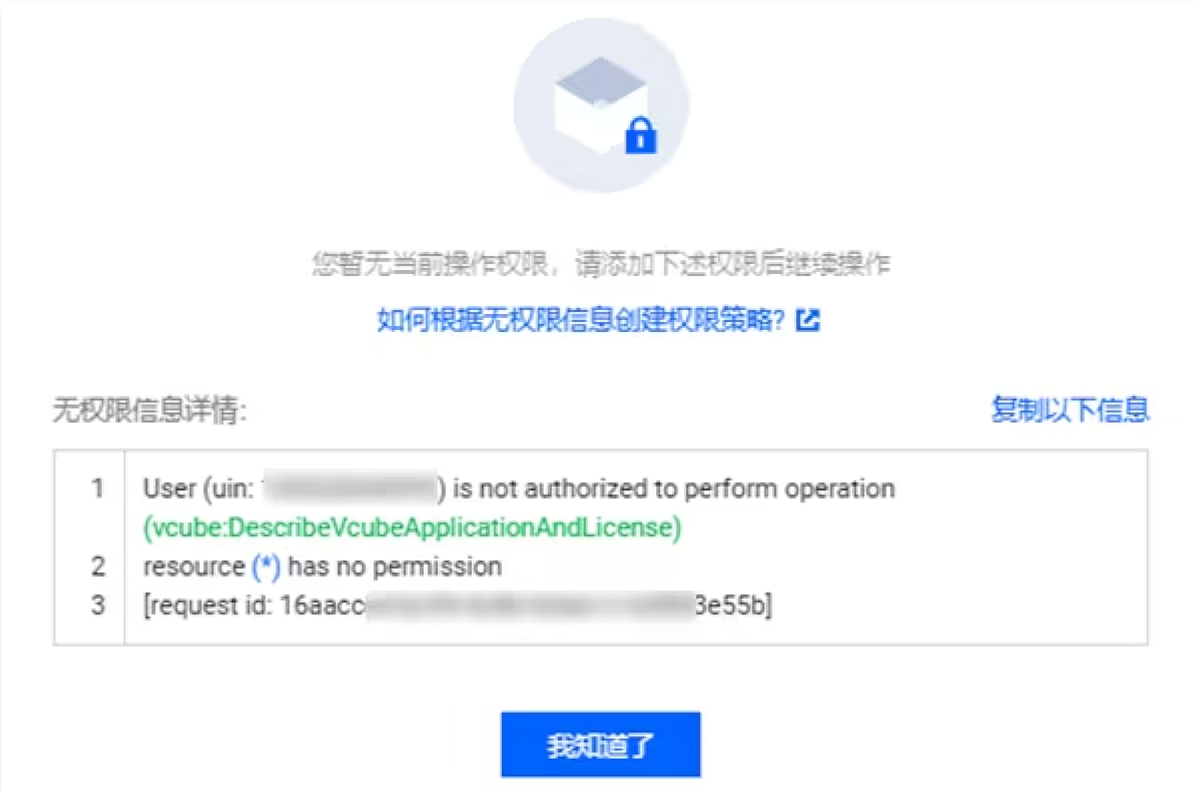
Not supported. The live stream License is only applicable to Apps on iOS and Android platforms when using the live stream Streaming (host starting a live stream and host-audience co - anchoring/host cross - room PK) functional modules. For mini programs to access the live stream function, corresponding service categories must be in place first. For details, see Scheme selection.

## Why Does an Available Resource Pack Increase and a License Decrease After Upgrading the New Version License?

The new version of the License adds duplicate package name verification logic. When multiple similar Licenses are bound to the same set of package names (Bundle ID and Package Name), in fact, only one License will take effect. We will unbind the invalid and shorter-validity License binding relationships, release the resource packages of invalid bindings for you, and the released resource packages can be bound to new Licenses for use.
**Example:**
User A previously purchased two 10 TB live streaming traffic resource packages (with different expiration dates), received two live streaming Licenses, and bound the same set of package names twice (in fact, only one License was valid). After upgrading to Tencent Cloud RT-Cube Licenses, only the binding of the License with the longer validity period is retained; the other License with the shorter validity period is unbound, and the 10TB resource package is released. User A can bind a new License again.

## Why Does a Subaccount Get an Unauthorized Prompt When Opening the Audio and Video Terminal SDK (Tencent Cloud RT-Cube) Console?

- **Screenshot of the issue:**




- **Problem analysis:**
  The Tencent Cloud Video Cube Console requires the root account to independently authorize policies for sub-accounts before accessing the console page.
  - If you only need to provide sub-accounts with read-only permission to access the Audio and Video Terminal SDK Console, please authorize the QcloudVCUBEReadOnlyAccess policy.
  - If you need to provide sub-accounts with all operation permissions to access the Audio and Video Terminal SDK, please authorize the QcloudVCUBEFullAccess policy.
    For association guidelines on associating policies for users/user groups to authorize relevant operation permissions, see Policy Authorization Management.
- **Related issues:**
  - Why can't subaccounts access the License-related interface of the live streaming console?
  - Why can't subaccounts access the License-related interface of the VOD console?

## Why Can'T Messages Related to Audio and Video Terminal SDK Products and Licenses Be Received?

You can subscribe to the audio and video terminal SDK in Message Subscription and configure message reception channels such as **Internal Messages/Email/Short Message Service/WeChat/WeCom** to receive expiry reminders for the official License. The official

License will send you an expiration reminder once at 30 days, 15 days, 7 days, and 1 day before the expiration time, reminding you to renew in time to avoid affecting normal business operations.

## Why Can a Live Stream License Unlock the Video Playback Feature Module?

After version 10.1 (launched at the end of May 2022), live streaming License (formerly live push stream License), Short Video License and player License **can all** authorize to unlock the **video playback** functional module of the new version SDK. You only need to purchase **any one** of these Licenses to use the live stream and on-demand playback features in the new version SDK normally. The details of authorization unlocking for each License are as follows:

| License Type | Unlocked Feature Module Authorization | | | License Acquisition Method |
|---|---|---|---|---|
| | Live stream pushing | Short video production | Video playback | |
| Live stream License | ✓ | - | ✓ | • Purchase 10TB, 50TB, 200TB, or 1PB CSS traffic resource packages and get one-year live stream publishing License usage authorization as a giveaway.<br>• Purchase one - year usage authorization of standalone live streaming License |
| Short video License | - | ✓ | ✓ | • Purchase 10TB, 50TB, 200TB, or 1PB Video on Demand (VOD) traffic resource packages and get one-year usage authorization for the short video lite/basic version License as a giveaway.<br>• Purchase one-year usage authorization for the independent short video License |
| Player License | - | - | ✓ | • Purchase 100GB, 500GB, 1TB, or 5TB live/video traffic packages and get one-year usage authorization of the basic version player for mobile terminals for free.<br>• Purchase one-year usage authorization for the standalone player License |

# Functionality Related to User Generated Short Video SDK

Last updated：2025-03-17 15:17:32

## What System Versions Does the Short Video SDK Support?

- **iOS:** Mobile phone system 8.0 or above, Xcode 9 or higher version, OS X 10.10 or higher version.
- **Android:** Mobile phone system 4.0.3 (API 15) or above.

> **Note:**
> Only systems with (Android 4.3) API 18 or higher can enable hardware encoding.

## Does the Short Video SDK Support X86?

- **iOS:** Supported.
- **Android:** Not supported.

## Does the Short Video SDK Recording Support Anti-Shake?

It is not supported.

## Is a Video License Required For Integrating the Short Video SDK?

Not required.

## Can Customers Use the BGM in the Demo and Short Videos?

The BGM in the Demo and short videos are only used for feature demonstration. If you use it in a commercial App, there will be legal risks.

## Does Video Recording and Editing Support Converting to GIF?

It is not supported. You can obtain the video sample image list through our SDK API (TXVideoEditer > TXVideoInfoReader > getSampleImages) and then generate a GIF by yourself.

## Does the Short Video SDK Support the Photo-Taking Function?

Supported. You can call the snapshot API in the recording API to take a photo (TXUGCRecord > snapshot).

## Can the Video Generated By the Audio and Video Terminal SDK (Tencent Cloud RT-Cube) Be Directly Uploaded to Non-Tencent Cloud Platforms (Such As WeChat Official Accounts)?

The video generated by the audio and video terminal SDK (Tencent Cloud RT-Cube) can be uploaded to Tencent Cloud's VOD server. There is also corresponding source code for reference in the Demo. If you want to upload it to other platforms, please check the upload requirements of other platforms yourself.

## Does the Short Video SDK Support Mini Programs?

It is not supported.

## Does the Short Video SDK Support Enlarging Eyes and Slimming Face?

Not supported.

## How Does the Short Video SDK Obtain Video Information (Such As Width and Height)?

TXVideoEditer > TXVideoInfoReader > getVideoInfo.

## Does Video Editing Support Inserting Images At Any Position in the Video?

It is not supported.

## What Is the Decryption Key For Decompressing the Short Video SDK?

The short video SDK does not have a decryption key.

## Does the Demo Experience Only Have the Basic Beauty Effect Function?

The basic beauty effect is supported in the Demo experience only. If you want to experience other special effects such as big eyes, you need to purchase additionally.

## Are There Any Restrictions On the Number of Viewers For Audio and Video Terminal SDK (Tencent Cloud RT-Cube)?

There is no limit on the number of viewers.

## Does the Short Video SDK Support Third-Party Beauty Features?

The short video SDK does not support integrating third-party beauty features.

## Does the Short Video SDK Support the Background Music Function?

Supports selecting self - provided audio files or local MP3 files on the user's mobile phone as background music. Also supports cropping of background music and setting the volume.

## Why Is Setting Background Music Unavailable?

- The short video SDK simplified edition does not support the function of setting background music files. It is recommended to check the current integration version.
- If your current UGSV SDK is upgraded based on the simplified edition of UGSV SDK, please check whether the current UGSV SDK package is still the simplified edition.
- Check whether the current License is the same as the package name of the short video SDK simplified edition License. If yes, please submit a ticket and contact staff for handling.

## Using the Video Editing Feature to Insert Music, What Sources of Music Can Be Used?

For the music in short videos, you need to add it in the code. Currently, there is no music library available for you to use here. You can choose the music path by yourself. For more details, see Add Background Music.

## Can the Short Video SDK Use the Photo-Taking Function On the Basis of the Simplified Edition?

The photo - taking component that supports integration and use within the feature module of Tencent Cloud Video Cube UGSV SDK can be integrated and used in the UGSV SDK.

## Does the Short Video SDK Support Custom Animated Stickers? How to Implement It?

Supported. Use through the SDK source code. For details, see Stickers and Subtitles (iOS).

## Does the Short Video SDK Support Filters?

The short video SDK supports filter effects and video editing features. For details, see TikTok-like special effects, SDK Integration (XCode).

## Can the Editing Function of the Short Video SDK Be Used in WeChat Mini Program?

Integration in WeChat mini program is not supported.

## Does the Short Video SDK Support a Background Wall?

Not supported.

## Support Integration of Advanced Beauty Effect in Experience Demo?

The Demo is for experience only. If you need to integrate the advanced beauty effect on this basis, you need to enable this feature first.

## Does the Short Video SDK Support H5 Integration?

It is not supported.

## Does the Short Video SDK Support Flutter Version Integration?

It is not supported.

## Does the Short Video SDK Have a Prop Editor?

It is not supported.

## Does the Short Video SDK Upload in a Single Thread or Multi-Thread?

The short video SDK supports **multi-thread upload**.

# TRTC SDK Related

Last updated：2026-04-02 10:10:29

## What Is the RoomID of Tencent Real - Time Communication SDK (TRTC SDK)? What Is the Value Range?

RoomID is the room number, used to uniquely identify a room. The value range of the room number is from 1 to 4294967295, which is maintained and assigned by developers themselves.

## What Is the UserID For Entering a Room in the Real-Time Audio and Video SDK (TRTC SDK)? What Is the Value Range?

UserID, which is the user ID, is used to uniquely identify a user in a real-time audio and video application. It is recommended that the length of the value range should not exceed 32 bytes. Please use English letters, digits or underscores, case sensitive.

## What Is the Lifecycle of a Real-Time Audio and Video SDK (TRTC SDK) Room?

- The first user to join the room is the current room owner, but this user cannot actively dissolve the room.
- **In call mode**: The backend immediately dissolves the room when all users actively exit the room.
- **In live streaming mode:** When the last user to exit the room has the role of a streamer, the backend immediately dissolves the room; when the last user to exit the room has the role of an audience, the backend waits for 10 minutes before dissolving the room.
- If a single user in the room has an unexpected disconnection, the server will remove the user from the current room after 90 seconds. If all users in the room have an unexpected disconnection, the server will automatically dissolve the current room after 90 seconds. **The waiting time for a user's unexpected disconnection will be included in the billing duration statistics.**
- When the room that a user wants to join does not exist, the backend will automatically create a room.

## Does TRTC SDK Support Not Subscribing to Audio and Video Stream?

To achieve the "instant loading" effect, by default, streams are automatically subscribed when entering the room. You can switch to manual subscription mode through the setDefaultStreamRecvMode API.

## Does TRTC SDK Support Customizing the Stream ID For Bypass Stream Pushing?

Supported. You can specify streamId through the TRTCParams parameter of enterRoom, or call the startPublishing API and pass the streamId parameter.

## What Roles Does TRTC (Tencent Real – Time Communication) SDK Support For Live Streams? What Are the Differences?

In live-streaming scenarios (TRTCAppSceneLIVE and TRTCAppSceneVoiceChatRoom), there are two roles: TRTCRoleAnchor (streamer) and TRTCRoleAudience (audience). The difference is that the streamer role can upload and download audio and video data at the same time, while the audience role only supports downloading and playing other people's data. You can switch roles by calling switchRole().

## What Is the Role of the Real – Time Communication SDK (TRTC SDK)?

In the live stream scenario, you can set the roles of streamer and audience. The streamer role TRTCRoleAnchor has permissions for uplink and downlink audio and video, supporting up to 50 concurrent users. The audience role TRTCRoleAudience only has permissions for downlink audio and video, supporting up to 100,000 concurrent users.

## TRTC SDK Room Supports Which Use Cases?

Supported in the following scenarios:

- TRTCAppSceneVideoCall: Video call scenario, suitable for one-to-one video calls, 300-person video conferencing, online consultations, video chats, remote interviews, etc.
- TRTCAppSceneLIVE: Video interactive live streaming, suitable for low-latency video live streaming, interactive classrooms with one hundred thousand participants, video live PK, video blind date rooms, interactive classrooms, remote training, and ultra-large conferences.
- TRTCAppSceneAudioCall: Voice call scenario, suitable for 1 - on - 1 voice calls, 300 - person voice conferences, voice chats, audio conferencing, online Werewolf, etc.
- TRTCAppSceneVoiceChatRoom: Interactive live audio streaming, suitable for low-latency audio streaming, live audio co-anchoring, voice chat rooms, karaoke rooms, FM radio stations.

## What Platforms Does the Real – Time Audio and Video SDK (TRTC SDK) Support?

Supported platforms include iOS, Android, Windows (C++), Windows (C#), Mac, Web, Electron, and WeChat mini program. For more details, see Platform Support.

## How Many Rooms Can Be Created Simultaneously At Most in the Real-Time Interactive Audio and Video SDK (TRTC SDK)?

Supports up to 4294967294 rooms existing concurrently, with no limit on the total quantity of rooms.

## How to Create a Room?

Rooms are automatically created by the Tencent Cloud backend when clients enter a room. You do not need to manually create a room; simply call the relevant client API "Enter Room":

- iOS & Mac > enterRoom
- Android > enterRoom
- Windows (C++) > enterRoom
- Windows (C#) > enterRoom
- Electron > enterRoom
- Web > join
- Mini program > enterRoom

## What Is the Maximum Bandwidth Supported By the Video Server of the Real-Time Interactive Video SDK (TRTC SDK)?

There are no restrictions.

## Does TRTC SDK Support Private Deployment?

Real-time audio and video private deployment is not fully open. If you need to consult or use a private service, please fill out Private Service Questionnaire. We will complete the assessment and provide feedback within 2 to 3 business days.

## Does the Domain Name Need to Be Registered For Use When Enabling Relayed Live Streaming For TRTC (Tencent Real-Time Communication) SDK?

To enable relayed live streaming, according to the requirements of relevant national departments, the playback domain needs to be registered before use. For more details, see CDN live streaming viewing.

## What Is the Approximate Delay of the Real - Time Audio and Video SDK (TRTC SDK)?

Globally, end-to-end average latency is less than 300 ms.

## Does TRTC SDK Support the Active Calling Feature?

It needs to be resolved in combination with the signaling channel. For example, to make a call using the custom message of IM service, you can refer to the scenario-based Demo example in the SDK source code.

## Does TRTC (Tencent Real-Time Communication) SDK Support Bluetooth Headphones in One-To-One Video Calls?

It is supported.

## Does TRTC (Tencent Real - Time Communication) SDK Support Use Overseas?

It is supported.

## Does the Real-Time Audio and Video SDK (TRTC SDK) Support Screen Sharing When Accessing the PC?

It is supported. You can refer to the following documents:

- Screen sharing (Windows)
- Screen sharing (Mac)
- Screen sharing (Web)

For details about the screen sharing API, see Windows (C++) API or Windows (C#) API. Additionally, you can use Electron API.

## Does TRTC SDK Support Use in WeChat Official Accounts?

Due to wechat official account limitations, it is recommended to use mini program SDK in WeChat for a better experience.

## Can Local Video Files Be Shared to the Real-Time Communication SDK (TRTC SDK)?

Supported, it can be achieved through the custom collection feature.

## Can TRTC (Tencent Real - Time Communication) SDK Record Live Video and Store It On the Phone Locally?

Not supported to store directly on phone locally. After recording, video files are stored on the Video on Demand platform by default. You can download and save them to your phone. For more details, see On-Cloud Recording and Playback.

## Does TRTC SDK Support Pure Real-Time Audio?

Support audio-only.

## Can Multiple Screen Sharing Be Performed in One Room At the Same Time?

Currently, a room can only have one secondary stream of screen sharing.

## Specified Window Sharing (SourceTypeWindow) – Will the Resolution of the Video Stream Change When the Window Size Changes?

By default, the SDK will automatically adjust the encoding parameters according to the size of the shared window.
To fix the resolution, you need to call the setSubStreamEncoderParam API to set the encoding parameters for screen sharing, or specify the corresponding encoding parameters when calling startScreenCapture.

## Does the Tencent Real-Time Communication SDK (TRTC SDK) Support 1080P?

Supported. You can set the resolution through the video encoding parameter setVideoEncoderParam of the SDK.

## Can TRTC SDK Customize Acquisition Data?

Supported on some platforms. For detailed information, see custom capture and rendering.

## Can TRTC SDK Communicate With MLVB SDK?

No.

## Can TRTC SDK Communicate With MLVB SDK?

The solution architecture of audio and video call TRTC and MLVB SDK backend is different, so direct communication is not supported. It can only bypass stream pushing from the TRTC backend to CDN.

## What Are the Differences in the AppScene of the Room – Entry Mode of the Real – Time Interactive Audio and Video SDK (TRTC SDK)?

The TRTC SDK supports four different room entry modes. Among them, VideoCall and VoiceCall are collectively referred to as call modes, and Live and VoiceChatRoom are collectively referred to as live streaming modes.

- In calling mode, TRTC supports up to 300 people online in a single room simultaneously and up to 50 people speaking at the same time. It is suitable for application scenarios such as 1 – to – 1 video calls, 300 – person video conferences, online consultations, remote interviews, video customer services, online Werewolf games.

- Under the live streaming mode, TRTC supports up to 100,000 people being online simultaneously in a single room, with a mic on/off switching latency of less than 300 ms and a viewing latency of less than 1,000 ms, as well as smooth mic on/off switching technology. It is suitable for application scenarios such as low-latency interactive live streaming, 100,000-person interactive classrooms, video dating, online education, remote training, ultra-large conferences, etc.

## Does the Tencent Real-Time Communication SDK (TRTC SDK) Support Hands-Free Mode For Audio and Video Calls?

Supported. The hands-free mode is achieved by setting the audio routing. The Native SDK switches through the setAudioRoute API, and the mini program sets it through the sound-mode attribute of the <live-player> tag.

## Does TRTC SDK Support Volume Level Prompts?

Supported. Enabled through the enableAudioVolumeEvaluation API.

## Does TRTC SDK Support Setting a Mirrored Image?

Supported. Set the mirror mode of the local camera preview screen through the setLocalViewMirror API, or set the mirror mode of the encoder output screen through the setVideoEncoderMirror API.

## Does TRTC (Tencent Real – Time Communication) SDK Support Recording Audio During a Call to a Local File?

Supported. All audio (including local audio, remote audio, BGM, etc.) during a call can be recorded into a file through the startAudioRecording API. Currently supported audio formats include PCM, WAV, and AAC.

## Does TRTC SDK Support Video Recording Into Files During Audio and Video Interoperability?

Supports self-owned server recording (i.e. audio or video recording). If you need to use it, please submit a ticket to contact us for the SDK and relevant instructions. You can also use On-Cloud Recording and Playback to record videos.

## Does TRTC (Tencent Real-Time Communication) SDK Support Functions Similar to WeChat Video Call, Such As Floating Window and Switch Between Big and Small Windows?

This type of feature belongs to UI layout logic, and the SDK does not restrict UI display processing. In the official Demo, example codes for picture front-back stacking and nine-grid layout modes are provided, and support for floating window, switch between big and small windows, and picture dragging. For more details, see Official Demo.

## How Does TRTC SDK Implement Pure Audio Call?

The TRTC SDK does not distinguish between audio and video channels. When only startLocalAudio is called without calling startLocalPreview, it is a pure audio call mode.

## How Does TRTC SDK Implement Bypass Stream Pushing and Recording in Pure Audio Call?

- 6.9 previous versions: When entering the room, it is necessary to construct `json{\"Str_uc_params\":{\"pure_audio_push_mod\":1}}` and pass it into TRTCParams.businessInfo. 1 indicates bypass stream pushing, and 2 indicates bypass stream pushing + recording.
- TRTC SDK 6.9 and later versions: When entering the room, select the scene parameter as TRTCAppSceneAudioCall or TRTCAppSceneVoiceChatRoom.

## Does the TRTC (Tencent Real-Time Communication) SDK Room Support Removing Users, Prohibiting Speech, and Muting?

It is supported.

- For simple signaling operations, you can use TRTC's custom signaling interface sendCustomCmdMsg. Developers can define their own control signaling. The call recipient who receives the control signaling can perform the corresponding operation. For example, to remove a user, define a signal for removing a user. The user who receives this signal can exit the room on their own.
- If more comprehensive operation logic is required, it is recommended that developers use Chat to implement the relevant logic, map the TRTC room to the IM Group, and send and receive custom messages in the IM Group to achieve the corresponding operations.

## Does the Tencent Real-Time Communication SDK (TRTC SDK) Support Live Playback of RTMP/FLV Streams?

Supported. Currently, TXLivePlayer is packaged in TRTC SDK. If more player features are required, LiteAVSDK_Professional version can be directly used, which includes all features.

## How Many People Can the Real - Time Audio and Video SDK (TRTC SDK) Support in a Call At Most?

- In call mode, a single room supports up to 300 people online simultaneously, and up to 50 people can turn on their cameras or microphones simultaneously.
- In live streaming mode, a single room supports up to 100,000 people watching online as the audience, and up to 50 people starting their cameras or microphones as streamers.

## How Does the Tencent Real-Time Communication SDK (TRTC SDK) Implement Live - Streaming Scenario Applications?

TRTC has launched a low-latency interactive live streaming solution for online live streaming scenarios that can ensure the minimum delay between the anchor and the mic-connecting anchor to 200 ms, with the delay for general audiences within 1 s, and its strong resistance to weak networks adapts to the complex mobile terminal network environment. For specific operation instructions, see Run Live Streaming Mode.

## Can the Custom Message Sending API of the Real - Time Audio and Video SDK (TRTC SDK) Be Used to Implement Functions Such As Chat Rooms and Bullet Screens?

No. Sending custom messages via TRTC SDK is suitable for simple and low-frequency signaling transmission scenarios. For specific limitations, please refer to Usage Limits.

## Does the Playback of Background Audio in the Real-Time Audio and Video SDK (TRTC SDK) Support Loop Playback? Does It Support Adjusting the Playback Progress of the Background Audio?

Supported. Loop playback can be achieved by re-calling the playback API within the completion callback. Playback progress can be set via TXAudioEffectManager.seekMusicToPosInMS.

> **Note:**
> setBGMPosition() is deprecated in v7.3. Use TXAudioEffectManager.seekMusicToPosInMS instead.

## Does TRTC SDK Have a Listening Callback For Room Members Entering or Leaving the Room? Can onUserEnter/onUserExit Be Used?

Yes. TRTC uses onRemoteUserEnterRoom/onRemoteUserLeaveRoom to monitor room members' entry and exit (triggered only for users with upstream audio and video permissions).

> **Note:**
> onUserEnter/onUserExit was deprecated in version 6.8 and replaced by onRemoteUserEnterRoom/onRemoteUserLeaveRoom.

## How Does TRTC SDK Monitor Network Disconnection and Reconnection?

Listen via the following listening callbacks:

- onConnectionLost: The SDK's connection to the server is lost.
- onTryToReconnect: The SDK attempts to reconnect to the server.
- onConnectionRecovery: The SDK's connection to the server is restored.

## Does TRTC SDK Have a First Frame Rendering Callback? Can It Monitor the Start of Picture Rendering and Sound Playback?

Supported. It can be monitored through onFirstVideoFrame/onFirstAudioFrame.

## Does the Tencent Real-Time Communication SDK (TRTC SDK) Support the Screenshot Function of Video Footage?

Currently, calling snapshotVideo() on iOS/Android supports screenshot of local and remote video screens.

## Is There an Exception When the TRTC SDK (Tencent Real - Time Communication SDK) Is Connected to Peripherals Such As Bluetooth Headsets?

At present, TRTC is compatible with mainstream Bluetooth headphones and peripherals, but there are still compatibility issues on some devices. It is recommended to use the official Demo and WeChat/QQ audio and video calls to test and compare whether they are all normal.

## How to Obtain Information Such As Uplink and Downlink Bitrate, Resolution, Packet Loss Rate, and Audio Sample Rate During TRTC (Tencent Real - Time Communication) SDK Audio and Video Process?

These statistical information can be obtained through the SDK API onStatistics().

## Does the playBGM() Interface of the Real-Time Audio and Video SDK (TRTC SDK) Support Online Music?

Currently, only local music is supported. It can be downloaded to the local device first and then played by calling playBGM().

## Does TRTC (Tencent Real-Time Communication) SDK Support Setting Local Audio Capturing Volume? Does It Support Setting Playback Volume For Each Remote User?

Supported. The audio capturing volume of the SDK can be set through the setAudioCaptureVolume() API, and the playback volume of a remote user can be set through the setRemoteAudioVolume() API.

## Difference Between stopLocalPreview and muteLocalVideo?

- stopLocalPreview stops local video capture. After calling this API, both the local and remote screens will go black.
- muteLocalVideo is to set whether to send one's own video footage to the backend. After calling this API, the footage watched by other users will turn into a black screen, but one can still see the footage in the local preview.

## Difference Between stopLocalAudio and muteLocalAudio?

- stopLocalAudio stops the collection and uplink of local audio.
- muteLocalAudio does not stop sending audio and video data; instead, it continues to send muted data packets with extremely low bitrate.

## What Resolutions Does the Real – Time Audio and Video SDK (TRTC SDK) Support?

It is recommended to refer to Set Picture Quality to configure the resolution for a more appropriate picture quality.

## How to Set the Upstream Video Bitrate, Resolution, and Frame Rate For the Real-Time Audio and Video SDK (TRTC SDK)?

The videoResolution (resolution), videoFps (frame rate), and videoBitrate (bitrate) in the TRTCVideoEncParam parameters can be set through the setVideoEncoderParam() API of TRTCCloud.

## How Is the Control of Screen Angle and Orientation in TRTC SDK Achieved?

For details, see Video Image Rotation and Zooming.

## How to Achieve Landscape Video Call?

For details, see Implementing Landscape Video Call and Video Image Rotation and Zooming.

## How to Adjust When the Local and Remote Screen Orientations of TRTC SDK Are Inconsistent?

For details, see Video Image Rotation and Zooming.

## Are There Any Recommended Parameter Configurations Related to Picture Quality (Bitrate, Resolution, Frame Rate) For the Tencent Real-Time Communication SDK (TRTC SDK)?

For details, see Set Picture Quality.

## Does TRTC (Tencent Real-Time Communication) SDK Support Network Speed Measurement? How to Operate?

For details, see Pre-call Network Test.

## Does TRTC (Tencent Real-Time Communication) SDK Support Permission Validation For Rooms, Such As Scenarios Where Only Members Can Enter?

Supported. For details, see Enable Advanced Permission Control.

## Does the Audio and Video Stream of TRTC SDK Support Watching By Pulling Stream Through CDN?

Supported. For details, see Implement CDN Live Streaming Viewing.

## What Formats Does TRTC SDK'S Custom Rendering Support?

- iOS supports i420, NV12 and BGRA.
- Android supports I420 and texture2d.

## What Is Tencent Real – Time Communication SDK (TRTC SDK)?

Tencent Real-Time Communication SDK (TRTC SDK) is one of the sub-products of Audio/Video Terminal SDK (Tencent Video Cube), including a feature module of **Audio/Video Call**. It uses the same underlying basic module as Tencent Real-Time Communication for video products. TRTC SDK focuses on cross-platform interoperability multi-person audio/video call and low-latency interactive live streaming solutions, aiming to help developers quickly build low-cost, low-latency, and high-quality audio/video interactive solutions.

## How to Experience the TRTC SDK Demo?

For details, see Demo Trial.

## How to Quickly Get Started With TRTC SDK?

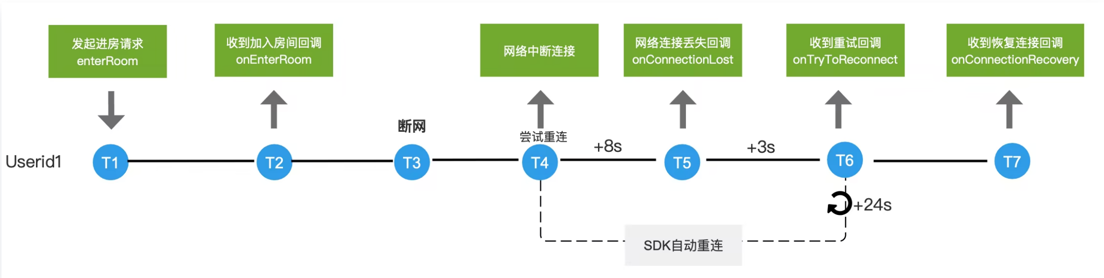
Tencent Real-Time Communication SDK (TRTC SDK) provides you with Demo source code for various platforms. You can quickly build your own small application in a very short time. For details, see Getting Started.

## How Does TRTC SDK Achieve Cloud Recording and Replay?

For details, see Cloud Recording and Replay.

# Does TRTC (Tencent Real-Time Communication) SDK Support Disconnect and Reconnect?

The SDK supports an infinite reconnection mechanism for users in case of disconnection. The specific connection status and processing logic during the connection process are as follows. The following figure shows the listening callback events received from when user Userid1 joins the channel, to when the connection is interrupted, and then to when rejoining the room:




**Detailed Description:**

- T1: The user-side initiates an API call to the `enterRoom` interface to make a room entry request.
- T2: Receive the `onEnterRoom` callback.
- T3: If the client is disconnected due to a network issue, the SDK will try to re-enter the room.
- T4: If no connection to the Server-side is established for 8 consecutive seconds, receive the `onConnectionLost` onTryToReconnect callback.
- T5: Then, if there is no connection to the server for 3 consecutive seconds, receive the `onTryToReconnect` retry callback.
- T6: Then, every 24 seconds, receive the `onTryToReconnect` retry callback.
- T7: Reconnect successfully at any time during the disconnection period and receive the `onConnectionRecovery` recovery callback.

# Live Stream Related
# Live Stream Basics

Last updated：2025-03-17 15:18:00

## What Are Streaming, Live Stream and Video-On-Demand Respectively?

- **Streaming:** The anchor pushes local video and audio sources to Tencent Video Cloud servers, which is also referred to as "RTMP publish" in some scenarios.
- **Live stream:** The video source of a live stream is generated in real time. A live stream makes sense only when someone is streaming. Once the anchor suspends the stream, the live stream URL becomes invalid. Moreover, since it is a live broadcast, there is no progress bar when the player plays the live stream video.
- **On-demand:** The video source for on-demand is a file in the cloud. As long as the file is not deleted by the provider, it can be played at any time (similar to Tencent Video). And since the entire video is on the server, there is a progress bar when playing.

## What Are the Requirements For Cloud Streaming Services Playback Domain?

Before submitting a domain name for management on the console, you need to file for record the domain name. The character limit for the domain name is 45 characters. Uppercase domain names are not supported currently. Please enter a lowercase domain address not exceeding 45 characters. For details, see Domain Name Management.

## Can the Live Stream Domain Name Be the Same As the Playback Domain Name and the Streaming Domain? Can a Second-Level Domain Be Used?

The playback domain and the streaming domain must be two different domain names, but they can be distinguished by second-level domains.
For example: `123.abc.com` is used for the streaming domain, and `456.abc.com` is used for the playback domain.

## What Streaming Protocols Are Supported?

Although RTMP is not particularly popular in the live stream field, it dominates in the stream push service, that is, from the "streamer" to the "server". Currently, domestic video cloud services all use RTMP as the main stream publishing protocol (since the first feature module of the Mobile Live Video Broadcasting SDK is streamer push, it is also called RTMP SDK).

## What Playback Protocols Are Supported?

Common live streaming protocols currently include: RTMP, FLV, HLS and WebRTC.

- **RTMP**: The RTMP Protocol is quite versatile; it can be used for both pushing and live streaming. Its core concept involves splitting large video and audio frames into smaller data packets for transmission over the Internet. It also supports encryption, thus offering relatively ideal privacy. However, the process of packetizing and depacketizing is complex, which can lead to some unexpected stability issues during high-concurrency scenarios.
- **FLV**: The FLV Protocol is mainly promoted by Adobe. Its format is extremely simple, adding only some header information to large video frames and audio and video headers. Due to this simplicity, it is very mature in terms of delay performance and large-scale concurrency. The only drawback is its very limited support on mobile browsers. However, it is extremely suitable as a live streaming protocol for mobile apps.
- **HLS**: A solution introduced by Apple that divides videos into small fragments of 5 to 10 seconds each and then manages them with an m3u8 index table. Since the video downloaded by the client is complete data of 5 to 10 seconds, the fluency of the video is very good, but it also introduces significant delay (the typical delay of HLS is around 10 to 30 seconds). Compared to FLV, HLS has very strong support on iPhones and most Android mobile browsers, so it is commonly used for URL sharing in QQ and WeChat Moments.
- **WebRTC**: The name is derived from the abbreviation of Web Real-Time Communication, which is an API that enables web browsers to conduct real-time voice or video conversations. It was open-sourced on June 1, 2011, and was incorporated into the W3C recommendation standard of the World Wide Web Consortium with the support of Google, Mozilla, and Opera. LeLive uses the WebRTC protocol, which is an extension of standard live streaming in ultra-low-latency playback scenarios, with lower latency than traditional live streaming protocols, providing viewers with millisecond-level live viewing experiences. It can meet the needs of specific scenarios with higher latency performance requirements, such as online education, sports event live streaming, online quiz taking, etc.

| Live Streaming Protocol | Strengths | Disadvantages | Playback Latency |
| --- | --- | --- | --- |
| FLV | Mature, suited for high-concurrency scenes | Requires integration of SDK for playback | 2s - 3s |
| RTMP | Relatively low delay | Poor performance in high-concurrency scenes | 1s - 3s |
| HLS(m3u8) | High support on mobile | Very high delay | 10s - 30s |

| | | | |
|---|---|---|---|
| | browsers | | |
| WebRTC | Lowest delay | Requires integration of SDK for playback | < 1s |

## What Is the Playback Address Composed Of?

The playback address of Tencent Cloud mainly consists of playback prefix, playback domain (domain), application name (AppName), stream name (StreamName), playback protocol suffix, authentication parameter and other custom parameters, as follows:

```
rtmp://domain/AppName/StreamName?
txSecret=Md5(key+StreamName+hex(time))&txTime=hex(time)
http://domain/AppName/StreamName.m3u8?
txSecret=Md5(key+StreamName+hex(time))&txTime=hex(time)
http://domain/AppName/StreamName.flv?
txSecret=Md5(key+StreamName+hex(time))&txTime=hex(time)
https://domain/AppName/StreamName.m3u8?
txSecret=Md5(key+StreamName+hex(time))&txTime=hex(time)
https://domain/AppName/StreamName.flv?
txSecret=Md5(key+StreamName+hex(time))&txTime=hex(time)
webrtc://domain/AppName/StreamName?
txSecret=Md5(key+StreamName+hex(time))&txTime=hex(time)
```

- **Playback prefix**
  RTMP playback protocol: **rtmp://**.
  HTTP-FLV playback protocol: **http://** or **https://**.
  HLS playback protocol: **http://** or **https://**.
  WebRTC playback protocol: **webrtc://**.
- **Application Name (AppName)**
  The application name refers to the storage path for live stream media files. By default, CSS allocates a path: **live**.
- Stream name (StreamName)
  Stream name (StreamName) refers to the unique identifier of each live stream.
- **Authentication parameters and other custom parameters**
  Authentication parameters: `txSecret=Md5(key+StreamName+hex(time))&txTime=hex(time)`.

## Common Streaming Methods?

- **On mobile terminals (Android/iOS), use the camera**: Use third-party software, Tencent Cloud RT-Cube MLVB SDK or Tencent Cloud RT-Cube to capture camera video and push

the video stream to the push URL.
- **Desktop or notebook, using camera or screen recording**: Use third-party software to capture camera video or desktop images and stream the video or desktop content to the live streaming push URL. Third-party streaming software includes: OBS (recommended), XSplit, FMLE, etc.
- **Video capture devices**: If HD camera devices have HDMI or SDI output interfaces, they can be connected to an encoder and push live content to the live streaming service in the form of RTMP streaming. You need to configure the live streaming push URL to the RTMP publishing address of the encoder.
  For webcam devices, if they support RTMP streaming, the live streaming push URL can be configured to the RTMP publishing address of the webcam.
- **Video file to video stream**: Read a video file and output it as an RTMP stream as the video source for video release to the RTMP push address of the live streaming service. You can use the `ffmpeg` command to achieve this (applicable to Windows, Linux, and Mac).

## Difference Between Stream Disconnection and Broadcast Prohibition?

- **Stream Disconnection Feature**: If the stream in a live stream is disconnected, the current streaming will be interrupted and the audience end will not be able to watch the live stream. After the disconnection, the streamer side can initiate streaming again to continue the live activity.
- **Block Stream Feature**: If a stream in a live stream is blocked, the current streaming will be interrupted and the audience end will not be able to watch the live stream. After the stream is blocked, the streamer side cannot initiate streaming again during the block period. The block stream feature can be configured through the flow management page on the CSS console. The disabled live stream will be displayed on the list of disabled streams. It can be re-enabled after clicking **Enable**.

# Streaming and Playback Related

Last updated：2025-03-17 15:18:08

## Is There a Capacity Limit On the Number of Online Users in a Live Stream?

By default, Tencent Cloud Streaming Services (CSS) does not limit the number of online users watching live streams. Anyone can watch the live stream as long as network and other conditions permit. If the user configures a bandwidth limit, when there are too many viewers and the bandwidth limit is exceeded, new users cannot watch the live stream. In this case, the number of online users is limited.

## How to Use Playback Transcoding?

Considering different network factors to meet your needs of different bitrates and resolutions, you can go to Transcode Configuration to set transcode templates with different bitrates and resolutions. For more information about transcoding, see Live Stream Remuxing and Transcoding.

### Raw, High-Definition, Standard Definition Scenarios

In business playback scenarios, three bitrates are generally used: raw, high-definition, and standard definition.

- The original stream has the same bitrate and resolution as the streaming.
- For HD stream, it is recommended to use a bitrate of 2000 kbps and a resolution of 1080p.
- For SD stream, it is recommended to use a bitrate of 1000 kbps and a resolution of 720p.

## How Can I Use Time Shift For Replay?

If you want to review wonderful content from a past period, you can use the time shift feature, which currently only supports the HLS protocol. For detailed introductions and activation methods of time shift, see Live Streaming Time Shift.

## How to Use HTTPS For Playback?

If your playback domain needs to support HTTPS, you need to prepare valid certificate content and private key content and go to Domain Name Management, select **Playback Domain Management** > **Advanced Configuration** > **HTTPS Configuration** to add the configuration. After successful addition, there will be an effective time (2 hours). After it takes effect, your live stream can support HTTPS protocol playback.

## How to Use Overseas Acceleration Nodes For Playback?

Tencent Cloud CDN nodes are not only spread across mainland China, but also cover all continents worldwide, providing extensive and stable coverage. If your users are located in Hong Kong (China), Macao (China), Taiwan (China) or other regions overseas, you can select Global Acceleration or Hong Kong/Macao/Taiwan (China Region) and other regions when configuring the domain name in Domain Management to obtain support from overseas node coverage.

> **Note:**
> Global acceleration for CSS currently supports only the HTTP-FLV and HLS protocols.

## How Can I Enable Hotlink Protection?

To prevent illegal users from stealing your playback URL and playing it elsewhere, causing traffic loss, it is highly recommended that you add hotlink protection to the playback address to prevent unnecessary losses caused by hotlinking. The playback hotlink protection of CSS is mainly controlled by four parameter values: txTime, key (hash key), txSecret, and validity time.

| Hotlink Protection Parameter | Description | Supplementary Notes |
| --- | --- | --- |
| txTime | Validity time of the playback URL | In the format of hexadecimal UNIX time.<br>If the current value of txTime is greater than the current request time, playback can proceed normally; otherwise, playback will be rejected by the backend. |
| key | Key for MD5 calculation method | It can be customized, and two keys for primary and secondary can be set.<br>When your primary key is accidentally leaked, you can use the secondary key to splice the playback URL and change the value of the primary key at the same time. |
| txSecret | Encryption parameter in the playback URL | The value is obtained by performing MD5 encryption algorithm on the string spliced in sequence by key, StreamName and txTime.<br>txSecret = MD5 (key + StreamName + txTime). |
| Effective Time | Validity time of address | The effective time setting must be greater than 0.<br>If txTime is set to the current time and the effective time is set to 300s, the expiration time of the playback URL is the current time + 300s. |

### Hotlink Protection Calculation

Hotlink protection calculation requires three parameters: key (a random string), StreamName (stream name), and txTime (in hexadecimal format).
Assume that the key you set is **somestring**, the stream name (StreamName) is **test**, and txTime is **5c2acacc** (January 1, 2019, 10:05:00). The high-definition bitrate is: **900kbps**, and the transcoding template name is: **900**.
Original stream playback address:

```
txSecret = MD5(somestringtest5c2acacc) =
b77e812107e1d8b8f247885a46e1bd34
http://domain/live/test.flv?
txTime=5c2acacc&txSecret=b77e812107e1d8b8f247885a46e1bd34
http://domain/live/test.m3u8?
txTime=5c2acacc&txSecret=b77e812107e1d8b8f247885a46e1bd34
```

HD stream playback address

```
txSecret = MD5(somestringtest_9005c2acacc) =
4beae959b16c77da6a65c7edda1dfefe
http://domain/live/test_900.flv?
txTime=5c2acacc&txSecret=4beae959b16c77da6a65c7edda1dfefe
http://domain/live/test_900.m3u8?
txTime=5c2acacc&txSecret=4beae959b16c77da6a65c7edda1dfefe
```

### Enable Playback Hotlink Protection

1. Log in to domain name management.
2. Select a playback domain or click **Manage** in the corresponding row to enter the domain details page.
3. **Select <Access Control>**, click **Edit**.
4. Set **Playback Authentication** to enabled and click **Save**.

**Note:**
- Playback authentication settings take effect 30 minutes after successful configuration.
- HTTP-FLV: The URL being played can still play normally after the txTime expires. Re-requesting playback after the txTime expires will be rejected.

- HLS: Since HLS is a short link, it will continuously request m3u8 to obtain the latest ts segment. Suppose you set the txTime value to the current time + 10 minutes, then the HLS playback URL request will be rejected after 10 minutes. To address this issue, you can dynamically update the HLS request address on the business side or set a longer expiration time for the HLS playback address.

## Are There Any Requirements For the Format of the Master Key in the Playback Authentication Configuration? Is There a Limit On the Validity Time?

In the authentication configuration, the primary Key value only supports uppercase letters, lowercase letters and digits, with a maximum length of 256 bits. A random combination of letters and digits is acceptable.
It is recommended that the effective time duration be set to the length of a live stream.

## How to Get the Recording File After Live Stream Recording?

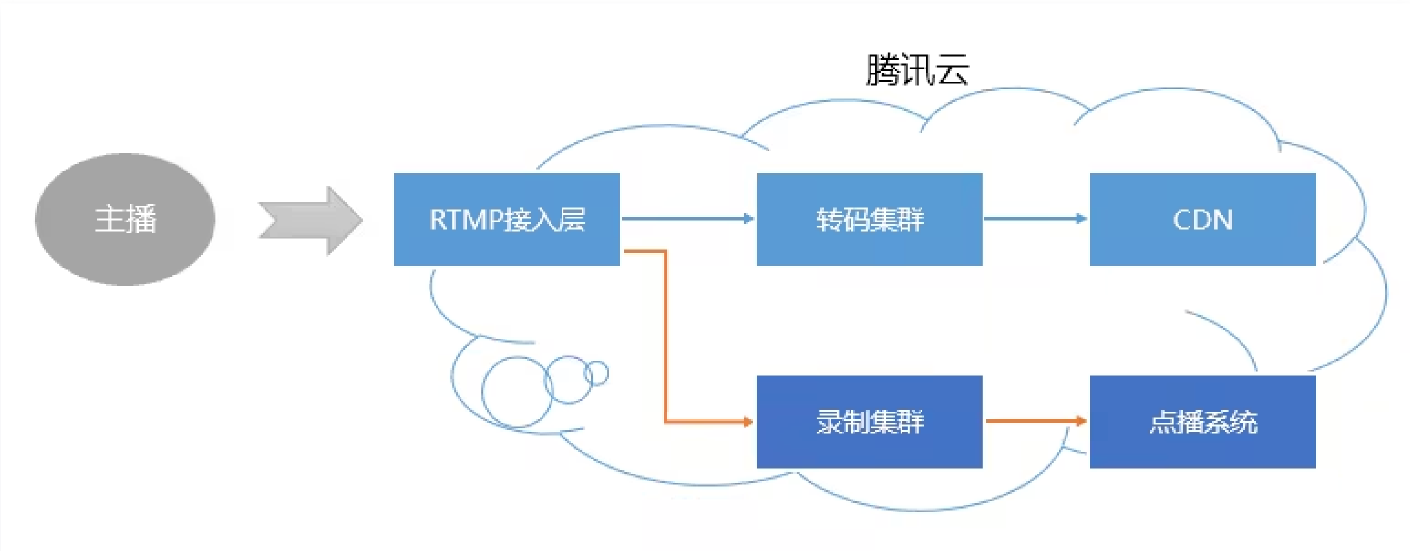
Recording files are automatically stored in the Video on Demand system after being generated. There are the following ways to obtain recording files:

- VOD console
- Recording Event Notification
- Video-on-demand API query

# Live Stream Recording Related

Last updated：2025-03-17 15:18:14

## What Is the Principle of Live Stream Recording?




For a live streaming, once the recording is started, audio and video data will be bypassed to the recording system. Every frame of data pushed up by the anchor's mobile phone will be appended to the recording file by the recording system.
Once the live streaming is interrupted, the connection layer will immediately notify the recording server to finalize the file being written, transfer it to the VOD system, and generate an index for it. Thus, you will see this newly generated recording file in the VOD system. Meanwhile, if you have configured recording event notifications, the recording system will notify your previously configured server of information such as the **Index ID** and **online playback URL** of this file.
However, if a file is too large, it is easy to encounter errors during the transfer and processing on the cloud. Therefore, to ensure the success rate, our single recording file will not exceed 120 minutes at most. You can specify shorter fragments through the RecordInterval parameter.

## Why Can'T Video Recording Be Done During Live Stream?

The live recording and playback feature relies on Tencent Cloud's **Video on Demand Service**. If you want to use the recording feature, you first need to enable the Video on Demand Service in Tencent Cloud's management console. For more information about live recording and playback operations, see Recording and Playback.

## How Long Does It Take to See the Recording Files After the Live Stream Ends?

It is expected that the recording files can be obtained about 5 minutes after the live stream ends. There will be an event callback after the recording is completed. The details are subject to the received callback time. For more information, see Callback Configuration.

## How to Get the Recording Files After Live Stream Recording?

Recording files are automatically stored in the Video on Demand System after being generated. Customers need to enable the on-demand service for successful storage. Recording files can be obtained in the following ways:

- VOD Console
- Recording Event Notification
- Video-on-demand API query

## Can Live Video Be Migrated?

Currently, you need to obtain the video download address and migrate it yourself.

## How to Set Video Storage Duration?

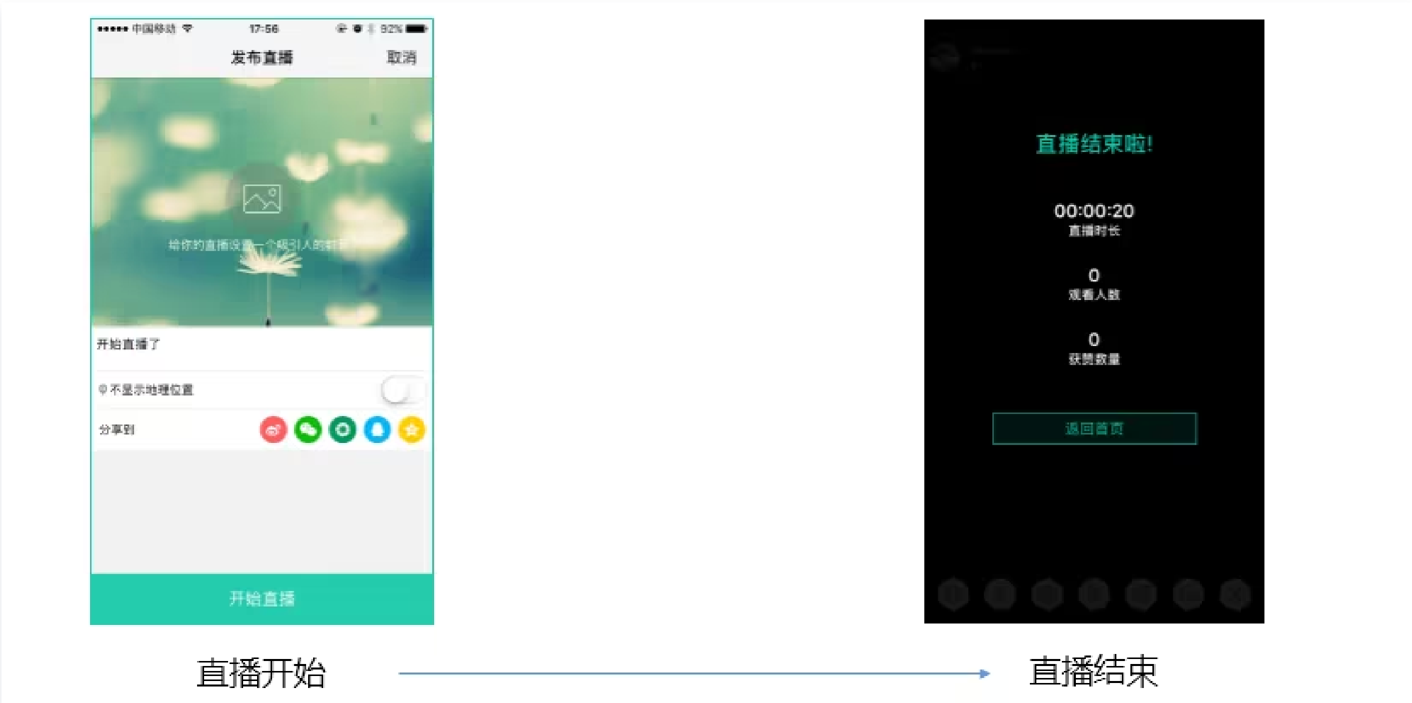
There is currently no time limit on the video storage of Cloud Streaming Services (CSS). You can manage video files through the console and the REST API.

## How Many Recording Files Are Generated in One Live Recording?

- **Record in MP4, FLV, or AAC format**: The duration limit for a single file is 1 to 120 minutes. You can specify shorter segments through the RecordInterval parameter in the Create Recording Template API.
  - If a live stream is very short and the recording module does not start before the streaming ends, the system will fail to generate recorded files.
  - If a live stream does not last long (less than RecordInterval) and there is no interruption during the process, usually there is only one file.
  - If a live stream lasts a long time (exceeding RecordInterval), it will be fragmented according to the time length specified by RecordInterval. The reason for fragmentation is to avoid the uncertainty of the circulation time of overly long files in distributed systems.
  - If the streaming is interrupted during a live stream (the SDK will try to resume the stream later), each interruption will generate a new fragment.
- **Record in HLS format**: There is no limit on the maximum duration of a single file. If it exceeds the continuous recording timeout period, a new file will be created to continue recording. The continuous recording timeout period can be set from 0s to 1800s.

## How to Know Which Files Belong to a Certain Live Stream?

To be precise, as a PAAS, Tencent Cloud does not know how you define a live stream. If your live stream lasts for 20 minutes, but there is a stream disconnection due to network switching in the middle, as well as a manual stop and restart, then is this one live stream or three?
For ordinary mobile live streaming scenarios, we generally define the time between the following interfaces as one live stream:




So the time information from the App client is very important. If you want all the recording files within this period to be considered part of this live stream, you only need to retrieve the received recording notifications using the live stream code and time information (each recording notification event will carry information such as **Stream ID**, **Start Time**, and **End Time**).

## How to Concatenate Fragments?

Currently, Tencent Cloud supports splicing video segments using cloud-based APIs. For detailed API usage, see Video Stitching.

## Only One Recording Template Is Set, but There Are Two Streams in Live Streaming Recording. How to Troubleshoot?

Under normal circumstances, there may be two concurrent recording tasks under the current streaming domain name. It is recommended to troubleshoot in the following order:

1. Check the console recording configuration information to confirm whether only one format is selected for the recording file type.
   - If the console is the **New Console**, go to Domain Name Management, click **Manage** on the right side of the streaming domain, enter to view **Template Configuration** in

**Recording Configuration**, and check the "Recording Format" information of the associated template.
   - If the console is the **Legacy Console**, go to **Live Stream Code Access** > **Access Configuration** to check the live recording configuration information.

2. Create a recording task and Create a recording template are two ways to initiate recording. In actual use, choose one as needed. If a recording task is created while configuring a recording template for the same live streaming, it will cause duplicate recording. Check whether a recording task has been initiated in the console while calling the CreateRecordTask API of API 3.0 or the Live_Tape_Start API of API 2.0.

> **Note:**
> - If your live streaming recording was enabled in the legacy console and you need to disable it in the new console, you can submit a ticket and ask relevant personnel for assistance.
> - If the above methods cannot solve your problem, submit a ticket for resolution, and a dedicated person will assist you.

# Short Video Related
# Upload Related

Last updated：2025-06-27 16:32:06

## What Is Client Video Upload

Client video upload refers to App users directly uploading local videos to VOD.

## Video Upload Feature TXUGCPublish Not Found

The video upload module has been separated from the SDK as a standalone component and open-sourced in the Demo. Users are advised to integrate short video upload themselves, as follows:

1. Download the Demo.
2. Copy the uploaded jar package from `app\libs\upload` to your project directory `..\app\libs\upload`.
3. Copy the short video upload source code directory `Demo\app\src\main\java\com\tencent\liteav\demo\videoupload` to your own project directory and modify the package name in the source code.
4. Add the code to reference the jar package in build.gradle under the project directory.

```
dependencies {
  compile fileTree(include: ['*.jar'], dir: 'libs/upload')
}
```

5. Configure App permissions in `AndroidManifest.xml`.

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

## Upload Failed with CLS Internal Error 1000?

Please check whether the VOD service is enabled.

## Short Video Upload Parameter Error

Please check whether the file URL and image address are correct and find the corresponding file in the path.

## Short Video Upload Signature Error

Before initiating upload, the client must request an upload signature from the application server. If the application server grants client upload permission, it will generate an upload signature for the client as per signature rules. The client must carry this signature for VOD to verify whether the upload is granted.
The steps for client upload signature are as follows:

1. Obtain API key.
2. Concatenate plaintext string.
3. Convert the plaintext string to the final signature.
4. After the service setup is complete, developers can verify signature correctness through the tools provided by VOD:
   - Signature generation tool: Generate signature quickly based on parameters and key.
   - Signature verification tool: Parse the signature to get the parameters used by signature generation.

For more information, see client upload signature.

## Maximum Allowed Video Upload Duration and Sizes Is There a Limit

The UGSV SDK has no limit on upload video duration and sizes.

## Image Upload

**Image upload is not currently supported**, but you can attach a cover image when uploading videos. For related notes, please refer to Video Upload.

# Android Related

Last updated：2025-03-17 15:19:01

## What to Do If Integration Encounters an Exception?

```
java.lang.UnsatisfiedLinkError: No implementation found for void com.tencent.liteav.basic.log.TXCLog.nativeLogInit() (tried
om_tencent_liteav_basic_log_TXCLog_nativeLogInit__)
    at com.tencent.liteav.basic.log.TXCLog.nativeLogInit(Native Method)
    at com.tencent.liteav.basic.log.TXCLog.init(TXCLog.java:31)
    at com.tencent.liteav.basic.log.TXCLog.setConsoleEnabled(TXCLog.java:52)
    at com.tencent.rtmp.TXLiveBase.setConsoleEnabled(TXLiveBase.java:73)
```

You can use the armeabi and armeabi-v7a architectures.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion rootProject.ext.compileSdkVersion
    buildToolsVersion rootProject.ext.buildToolsVersion
    defaultConfig {
        applicationId "com.tencent.liteav.demo"
        minSdkVersion rootProject.ext.minSdkVersion
        targetSdkVersion rootProject.ext.targetSdkVersion
        versionCode 1
        versionName "2.0"
        multiDexEnabled true

        ndk {
            abiFilters "armeabi"
        }
        testInstrumentationRunner "android.support.test.ru
    }
```

As shown above, specify abiFilters as "armeabi" in the `app` 's build.gradle.

## Function Module Upgraded, Can'T Use the Features of User Generated Short Video SDK?

1. If you are using Android Studio, after replacing with a new aar, please modify the aar reference in the build.gradle of `app` in `/libs` directory of your project to check if it is consistent with the **file name of the aar**. Then clean and rebuild your project.
2. Confirm the functional module type. The short video SDK functional modules are divided into two authorization types: basic version and simplified edition. The included features are different. The difference lies in the presence or absence of special effects.

## Currently, Can the Resolution Generated By User Generated Short Video SDK Recording Be Customized? What Are the Customizable Outputs?

Customizable parameters for short video SDK recording include FPS (frames per second), GOP (key I-frame interval in seconds), video bitrate (amount of audio and video data generated per second by the encoder), maximum/minimum recording duration, and recording resolution. Four resolution options are provided as constants for you to choose from: 360×640, 540×960, 720×1280, 1080×1920.

The reason why recording is provided in constant sizes instead of user-customized sizes is as follows:

- The above four are the mainstream recording resolutions.
- Compatibility issue of Android mobile phones. Some non-mainstream resolutions are not supported, which may cause some screen glitches, green screens, and mosaics.

By calling the startCameraCustomPreview interface of the TXUGCRecord class and passing in the custom recording parameters, the code is as follows:

```
// Custom configuration
TXRecordCommon.TXUGCCustomConfig customConfig = new
TXRecordCommon.TXUGCCustomConfig();
customConfig.videoResolution =  TXRecordCommon.VIDEO_RESOLUTION_540_960;
customConfig.minDuration = mMinDuration;  // Minimum duration
customConfig.maxDuration = mMaxDuration;  // Maximum duration
customConfig.videoBitrate = mBiteRate;    // Video bitrate
customConfig.videoGop = mGop;             // GOP size
customConfig.videoFps = mFps;             // FPS
customConfig.isFront = mFront;            // Whether it is a front
camera
mTXCameraRecord.startCameraCustomPreview(customConfig, mVideoView);
```

## Why Is There No onRecordComplete Callback Received After Android Short Video Recording Ends?

Before starting to record a short video, please first set the recording callback listener by calling the setVideoRecordListener() API of the TXUGCRecord class. When it ends, you need to call the stopRecord() API of the TXUGCRecord class to end the recording.

```
// Before recording
mTXCameraRecord = TXUGCRecord.getInstance(this.getApplicationContext());
mTXCameraRecord.setVideoRecordListener(this);
```

```
...

// End recording
mTXCameraRecord.stopRecord();
```

## Exiting Short Video Recording and Starting the Second Recording, How to Continue Recording From the Previous Content?

After the onRecordComplete callback in the Demo, mTXCameraRecord.getPartsManager().deleteAllParts() is called to delete all shard files, because stopRecord has already completed the synthesis of shard files.
If the recording exits and continues from the last time, there is no need to delete the fragments. Do not call mTXCameraRecord.getPartsManager().deleteAllParts().

```
@Override
public void onRecordComplete(TXRecordCommon.TXRecordResult result) {
    TXCLog.i(TAG, "onRecordComplete, result retCode = " + result.retCode
+ ", descMsg = " + result.descMsg + ", videoPath + " + result.videoPath
+ ", coverPath = " + result.coverPath);
    if (mTXRecordResult.retCode < 0) {

Toast.makeText(TCVideoRecordActivity.this.getApplicationContext(),
"Recording failed. Reason: " + mTXRecordResult.descMsg,
Toast.LENGTH_SHORT).show();
    } else {
        mDuration = mTXCameraRecord.getPartsManager().getDuration(); //
Total recording duration
        if (mTXCameraRecord != null) {
            mTXCameraRecord.getPartsManager().deleteAllParts();  //
Delete the recorded shard files multiple times
        }
        startPreview(); // Enter the preview interface
    }
```

## Why Does the Background Audio Setting in Short Video Recording Not Take Effect?

The background music must be set before starting the recording (the startRecord API of TXUGCRecord) to take effect. Refer to the following example for the code call sequence:.

```
TXRecordCommon.TXUGCSimpleConfig simpleConfig = new
TXRecordCommon.TXUGCSimpleConfig();
simpleConfig.videoQuality = TXRecordCommon.VIDEO_QUALITY_MEDIUM;
simpleConfig.minDuration = mMinDuration;
simpleConfig.maxDuration = mMaxDuration;
// 1. First, start the preview
mTXCameraRecord.startCameraSimplePreview(simpleConfig, mVideoView);
// 2. Then set the path of the background music and play it
mBGMDuration = mTXCameraRecord.setBGM(mBGMPath);
mTXCameraRecord.playBGMFromTime(0, mBGMDuration);
// 3. Start recording (customVideoPath: video path after recording,
customPartFolder: folder for recorded videos, customCoverPath: cover
path of the recorded video)
int result = mTXCameraRecord.startRecord(customVideoPath,
customPartFolder, customCoverPath);
```

## Does Recording Have a Photo-Taking Feature?

The short video SDK includes a photo - capturing function. Call the snapshot API of the TXUGCRecord class, and the photo taken will be returned asynchronously through the TXRecordCommon.ITXSnapshotListener callback. The code example is as follows:

```
private void snapshot() {
    if (mTXCameraRecord != null) {
        mTXCameraRecord.snapshot(new
TXRecordCommon.ITXSnapshotListener() {
            @Override
            public void onSnapshot(Bitmap bmp) {
                // Image taken by photo
                saveBitmap(bmp);
            }
        });
    }
}
```

## Variable Speed Recording: What Is the Multiple of the Speed?

Adjustable-speed shoot does not support custom speed.

| Definition | Corresponding Constant in TXRecordCommon | Multiple |
| --- | --- | --- |
| Extremely slow speed | RECORD_SPEED_SLOWEST | 0.5x |
| Slow | RECORD_SPEED_SLOW | 0.8x |
| Standard | RECORD_SPEED_NORMAL | 1 |
| Fast | RECORD_SPEED_FAST | 1.25 times |
| Ultra - fast | RECORD_SPEED_FASTEST | 1.5x |

Adjustable-speed shoot can be achieved by calling setRecordSpeed(record) of TXUGCRecord to set different recording speeds:

```
mTXCameraRecord.setRecordSpeed(TXRecordCommon.RECORD_SPEED_FAST);
```

## What Are the Format Requirements For Importing Videos? Does It Support Importing Videos With a Resolution Higher Than 720P (For Example, 2K, 4K)? Are There Specific Size Limits For Imported Files?

Currently, on the Android side, only MP4 is supported for video import. There is no limitation on resolution, and there is no file size limit for imported files.

- There is no limit on the resolution of imported videos. No matter how large the original video is, it will be up to 720P after pre-processing.
- To quickly import videos, preprocessing (some features are restricted, such as reverse playback, single-frame preview, etc.) can be skipped after SDK 4.7. For videos with a resolution greater than 720P, it is recommended to add preprocessing because preview involves decoding each frame. Some mobile phones have poor performance, resulting in too long a time to decode a frame and render it on the interface, causing lag.

## Which Format of Background Music Does the Current Short Video UGSV SDK Editing Support?

Currently, only MP3 and M4A types are supported.

## What Are the Customizable Outputs of the Current Short Video UGSV SDK Editing?

The short video UGSV SDK for editing allows you to customize video bitrate (available in SDK 4.5 and later), audio bitrate (available in SDK 4.7 and later), and resolution. Several resolutions are provided as constants for you to choose from: 360 × 640, 480 × 640, 540 × 960, 720 × 1280, 1080 × 1920.

| Resolution | Corresponding Constant in TXVideoEditConstants |
| --- | --- |
| 360x640 | VIDEO_COMPRESSED_360P |
| 480x640 | VIDEO_COMPRESSED_480P |
| 540x960 | VIDEO_COMPRESSED_540P |
| 720x1280 | VIDEO_COMPRESSED_720P |
| 1080x1920 | VIDEO_COMPRESSED_1080P |

```
// Set output video bitrate
mTXVideoEditer.setVideoBitrate(3600);
// Set output resolution
mTXVideoEditer.generateVideo(TXVideoEditConstants.VIDEO_COMPRESSED_720P,
mVideoOutputPath);
```

## Can the Audio Be Removed From the Video Recorded By the Recording Feature of the Short Video UGSV SDK?

Currently, the short video UGSV SDK does not support simultaneous recording of BGM and voice. Therefore, after entering the editing mode, you can reset the BGM and set the original video sound volume to 0 to achieve the purpose of replacing the BGM. The code is as follows:

```
// Set the volume of the original video sound (set to 0 to remove the
recorded BGM)
mTXVideoEditer.setVideoVolume(0.0f);
// Set the local path of the background music
String bgmPath = getBGMPath();
mTXVideoEditer.setBGM(bgmPath);
// Set the volume of the background music, ranging from 0.0f to 1.0f
mTXVideoEditer.setBGMVolume(1.0f);
```

## How to Switch the Video Preview in the Same Activity Window and Full-Screen Mode?

Dynamically modify the size of the parent layout of the video preview View passed into the SDK. The SDK will dynamically adjust the size of the video according to the size of the parent layout and the video width and height. The calling sequence of the SDK API:

1. Perform stopPlay.
2. Modify the width and height of the FrameLayout passed to the SDK.
3. Call initWithPreview(parm) and pass in the layout of the new FrameLayout that hosts the playback component.
4. Start play again.

```
// Stop playback
mTXVideoEditer.stopPlay();
if (isFullScreen) {
    // If it is in full-screen mode, switch to window mode below
    FrameLayout.LayoutParams params = new
FrameLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, 1500);
    mVideoPlayerLayout.setLayoutParams(params);
    initPlayerLayout(false);
    isFullScreen = false;
} else {
    // If it is in window mode, switch to full-screen mode below
    FrameLayout.LayoutParams params = new
FrameLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.MATCH_PARENT);
    mVideoPlayerLayout.setLayoutParams(params);
    initPlayerLayout(false);
    isFullScreen = true;
}
// Start playback
mTXVideoEditer.startPlayFromTime(startTime, endTime);

Reset the preview View
private void initPlayerLayout(boolean isFullScreen) {
    TXVideoEditConstants.TXPreviewParam param = new
TXVideoEditConstants.TXPreviewParam();
    param.videoView = mVideoPlayerLayout;
    if (isFullScreen) {
        param.renderMode =
TXVideoEditConstants.PREVIEW_RENDER_MODE_FILL_SCREEN;
    } else {
```

```
        param.renderMode =
TXVideoEditConstants.PREVIEW_RENDER_MODE_FILL_EDGE;
    }
    mTXVideoEditer.initWithPreview(param);
}
```

## How to Separate the "Edit" and "Filter" Features Into Two Pages When Editing With the Short Video UGSV SDK?

The short video Demo of Tencent Cloud RT-Cube combines features such as "edit" and "filter" on one page. To separate the "edit" and "filter" features into two pages, you can first perform cropping (setCutTimeFrom) + pre-processing (processVideo) simultaneously. As a result, a cropped and pre-processed video is generated. Then, perform various editing operations, set the cropping to the entire duration (setCutTimeFrom), and finally call generateVideo to generate a video to prevent the video quality from being reduced due to double compression.

> **Note**
> After cropping in pre-processing, when generating the preprocessed video, before the final generation, be sure to set the cropping duration to the entire video duration, otherwise cropping will be performed again.

```
// Cropping page
mTXVideoEditer = new TXVideoEditer(mContext);
mTXVideoEditer.setCutFromTime(mTCVideoEditView.getSegmentFrom(),
mTCVideoEditView.getSegmentTo());
mTXVideoEditer.processVideo();

// Set cropping to the entire duration (setCutTimeFrom)
mTXVideoEditer.setCutFromTime(0, mVideoDuration);
// Jump to the Special Effects Page to generate
mTXVideoEditer.generateVideo(TXVideoEditConstants.VIDEO_COMPRESSED_720P,
mVideoOutputPath);
```

# For iOS

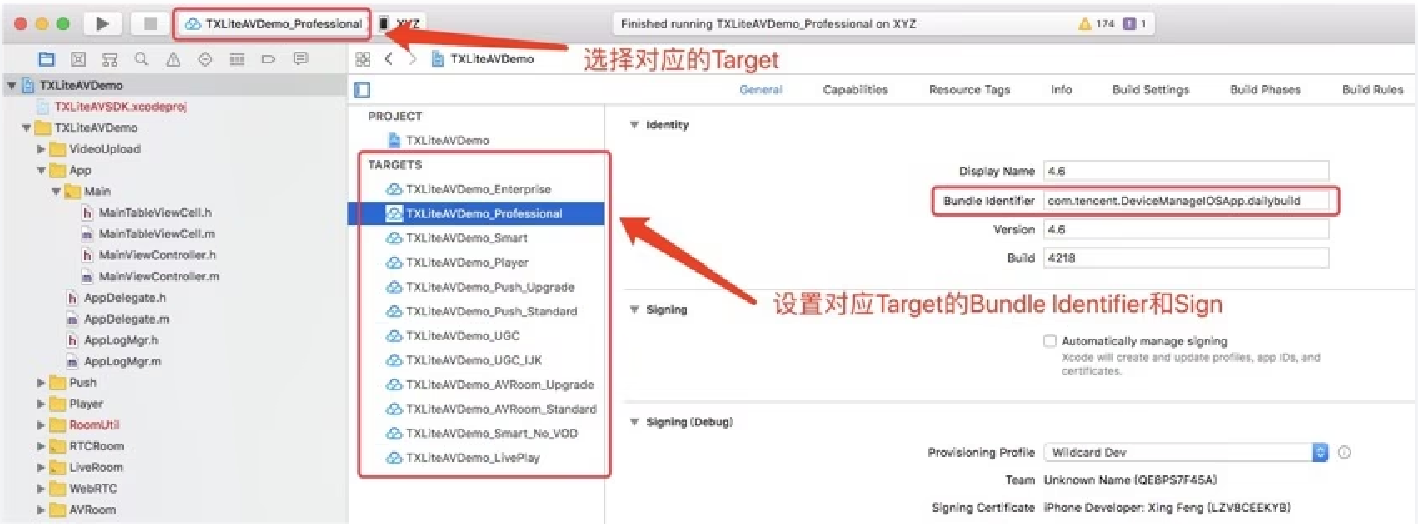
Last updated：2025-03-17 15:19:10

## Issues About TXUGCPublish.H?

Starting from version 4.5, the classes related to `TXUGCPublish` have been moved from the SDK to the Demo layer. Developers who need to use them can simply drag the entire `VideoUpload` directory into their own projects.

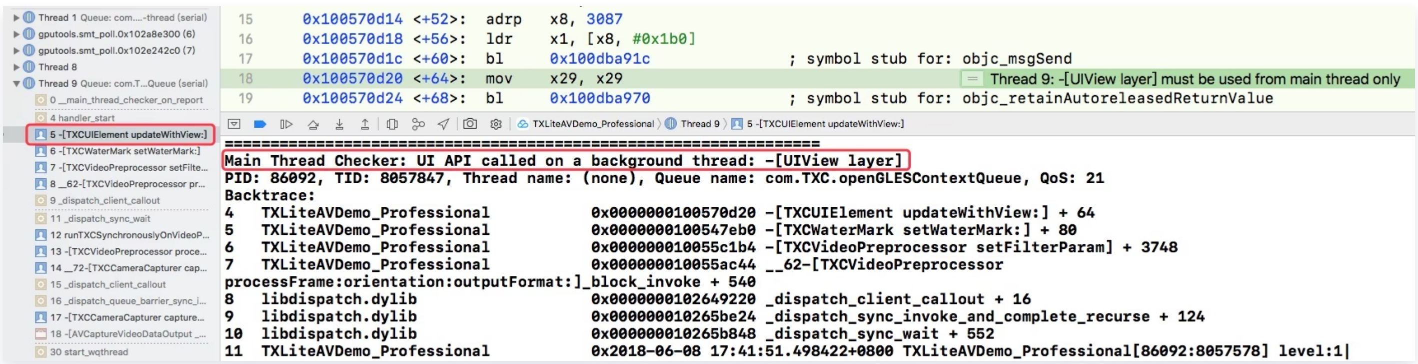
## Error When Directly Running Demo in Xcode?

You need to select the corresponding Target, as shown below:




## Connecting Xcode For Debugging, Error in Short Video Recording?

Connect to Xcode for debugging. An error occurred during short video recording. Error message: `Main Thread Checker: UI API called on a background thread`




The reason is that some APIs (generally UI-related) need to be called on the main thread. If they are called on a non-main thread and `Main Thread Checker` is enabled, an error will occur.

Troubleshooting: `Product` `Scheme` `Edit Scheme` `Run` `Diagnostics`, uncheck `Main Thread Checker`.

> **Note:**
> This issue has been fixed in version 4.9.

## Cannot Find the Header File When Using the Short Video SDK Functional Module?

- Add the header file search path in `Build Settings` > `Search Paths` > `Header Search Paths`.
- Use the `"TXLiteAVSDK_UGC/XXX.h"` method to reference the SDK's header file.
- Use the `@import TXLiteAVSDK_UGC;` method to reference the SDK (version 5.0 and later).

Choose one of the above methods.

## Issue With Missing Category Methods or crash When Running the Project?

The short video UGSV SDK uses some class methods. To load class methods, add `-ObjC` in project configuration: `Build Settings` > `Linking` > `Other Linker Flags`.

## What Should I Do If the Background Music Set For Short Video Recording Is Unavailable?

1. Determine whether there is a file under the transmitted BGM path and whether it can be played normally.
2. Determine the call sequence of the API: `startCameraSimple:preview:` > `setBGM:` > `startRecord`.

> **Note:**
> Many API calls have time sequence requirements; otherwise, they will be invalid.
> Usually, there will be notes in the comments.
> For example, for short video recording, APIs such as `setVideoResolution:`, `setVideoBitrate:`, `setAspectRatio:`, etc., need to be set before `startRecord` to be effective.

## Recording Settings: BGM Cannot Loop Playback?

Currently, the logic does not support loop playback.

## Recording Settings: BGM, Callback Not Completed At `endTime`?

If the set `endTime` is less than the total duration of the music file, a complete callback will be triggered at `endTime`.

## Why Is It Slow to Open the Camera For the First Time When Recording?

When the camera of an iPhone is opened for the first time (cold startup), the duration is relatively long. The same is true when opening the camera through the system API.
Since opening the camera is not suitable for the child thread, tests have shown that the duration of opening the camera in the child thread is longer. Moreover, when the camera is continuously opened/closed in the main thread, the response delay of the child thread will be higher, resulting in a poor experience.

## How to Implement Returning to Continue Recording?

When the first recording is completed, do not call `stopRecord` and `stopCameraPreview` (after calling, you cannot continue recording, but can only start a new recording). You can call `pauseRecord`, and then obtain the recorded video clips through `TXUGCPartsManager.getVideoPathList`. Synthesize the final video through `TXVideoJoiner.joinVideo` (for versions before 4.5). You can also directly call `TXUGCPartsManager.joinAllParts` to synthesize the final video. This method has a faster synthesis speed (supported in versions after 4.5). In this way, when you return to continue recording, all recording states are retained, and you can continue recording.

## Why Can'T the Complete Callback Be Received When the Short Video Recording Is Completed?

- Determine whether stopRecord has been called. Only after calling stopRecord will the complete callback be received.
- Determine whether all function calls are on the main thread.

## Can'T Record Sound When Returning to Continue Recording While Playing Video With Another Player During Recording?

The AudioSession in iOS is shared by all audio and video applications. When other players are playing, the AudioSession will be occupied. If the AudioSession is not released or not released in time after playback ends, the AudioSession of the recording module will become invalid.
The SDK provides two APIs: `-(void) pauseAudioSession` and `-(void) resumeAudioSession`. Call `pauseAudioSession` before going to other player previews, and call `resumeAudioSession` before returning to continue recording.

## Why Is the Recorded Video Unclear?

If the bitrate and resolution do not match, the recorded video will be unclear. You can improve the video quality by appropriately increasing the bitrate and enabling B-frames.

## Video Generation Failure When Switching to Background and Then Back to Frontend During Video Editing?

By default, video generation uses hardcoding (high coding efficiency and good image quality). The hard encoder will stop working when the program enters the background, resulting in video generation failure. The UGSV SDK provides two APIs: `pauseGenerate` and `resumeGenerate`. When the App enters the background, you can call `pauseGenerate` to suspend video generation. When the App returns to the foreground, you can call `resumeGenerate` to continue video generation.

> **Note:**
> Call `resumeGenerate`. The SDK will restart the hardware encoder. There is a certain probability that the restart will fail, or the encoding of the first few frames of data will fail after the restart. At this time, the SDK will throw an error event in `TXVideoGenerateListener`. After receiving the error event, it is necessary to regenerate the video.

## File Upload Failure?

Status code for file upload:

```
typedef NS_ENUM(NSInteger, TXPublishResultCode)
{
    PUBLISH_RESULT_OK                           = 0,        //发布成功
    PUBLISH_RESULT_UPLOAD_REQUEST_FAILED        = 1001,     //step1: "文件上传请求"发送失败
    PUBLISH_RESULT_UPLOAD_RESPONSE_ERROR        = 1002,     //step1: "文件上传请求"收到错误响应
    PUBLISH_RESULT_UPLOAD_VIDEO_FAILED          = 1003,     //step2: "视频文件"上传失败
    PUBLISH_RESULT_UPLOAD_COVER_FAILED          = 1004,     //step2: "封面文件"上传失败
    PUBLISH_RESULT_PUBLISH_REQUEST_FAILED       = 1005,     //step3: "短视频发布请求"发送失败
    PUBLISH_RESULT_PUBLISH_RESPONSE_ERROR       = 1006,     //step3: "短视频发布请求"收到错误响应
};
```

1. Determine whether the uploaded file is in the local sandbox. If uploading a file to the media library, it needs to be copied to the local sandbox first.
2. Return error code 1002: Signature problem, timestamp expired, vod service problem (not activated or service suspended).
3. Return error code 1003: Request parameter problem, unsupported file upload format.

## Does the Short Video UGSV SDK Recording Have a Photo - Taking Feature?

The short video SDK supports photo capturing. You can call the snapshot API of the `TXUGCRecord` class to get an image after starting the preview.

## Error "Use of undeclared identifier 'TXVideoInfo'" Keeps Occurring During Integration?

This error occurs because the compiler did not detect the TXVideoInfo class. It is recommended to check whether the SDK (framework) is correct. You can re-import the project according to SDK Integration (XCode).

## What to Do If an Error "-1, Failed to enable encoder" Occurs When Calling Video Synthesis?

1. Please confirm whether the issue is reproducible. It is recommended to conduct a test with a different model.
2. You can download the latest version of Demo to reproduce the problem. If the problem occurs consistently, please provide complete log information and submit a ticket for resolution.

# Video Playback Related

Last updated：2025-03-17 15:19:17

## Browser Hijacking Video Playback

Currently, in most cases, videos are played on web pages through browsers. Browsers have the highest processing permissions for video playback. The built-in player of the browser can replace the original video control, and it is prohibited to modify it through JS or CSS. Hijacking playback usually occurs in mobile terminal browsers. It is manifested that the style of the player is not the expected style, extra UI and advertisements appear during video playback, or forced full-screen playback occurs.
The following are issues caused by browser hijack and corresponding solutions:

### Forced Full-Screen After Video Activation and Playback

- **Problem manifestation:** After clicking to activate and play the video, it directly enters full-screen playback, usually occurring in browsers such as WeChat, Mobile QQ, and QQ Browser on Android and iOS.
- **Solution:** To achieve in-page (non-fullscreen) playback, you need to add the playsinline and webkit-playsinline attributes to the video tag. Tencent Cloud Player will automatically add the playsinline and webkit-playsinline attributes to the video tag by default. iOS 10+ recognizes the playsinline attribute, while systems with versions lower than 10 recognize the webkit-playsinline attribute. After testing, in-page (inline) playback can be achieved in iOS Safari. Android recognizes webkit-playsinline, but due to the openness of Android, there are many customized browsers, and these attributes may not work. For example, in TBS kernel browsers (including but not limited to WeChat, Mobile QQ, QQ Browser), you may need to use the same-layer player attribute ( Connect to Documentation ) to avoid the system forcing fullscreen video.

If forced fullscreen still occurs after configuring the above-mentioned attributes, the universal solution is ineffective and browser vendors need to provide a solution.

### Video Cannot Be Overlaid By Other Elements

- **Problem manifestation:** Other elements cannot be overlaid on the video area, and the player control is the browser's built-in control.
- **Solution:** Browser vendors are required to provide a method to unpin the video. There is no universal solution at present.

### Extra Icons Appear On the Player

- **Problem manifestation:** When the video is initializing, an image not belonging to Tencent Cloud player appears in the video area.
- **Solution:** You can try hiding the video tag and displaying it again when the event of video starting playback is detected.

### The Player Displays Ads, Downloads, Video Recommendations, Etc

- **Problem manifestation:** When the video is playing, paused, or ended, advertising content or download buttons appear in the video area.
- **Solution:** Browser vendors need to provide a closing method. There is no general solution at present.

### Android End Video Playback Does Not Scroll With Page

- **Problem manifestation:** In some Android system browsers, the player area does not scroll with the page or there is a delay in scrolling when the page is slid.
- **Solution:** After testing, it has been found that this kind of problem cannot be effectively solved through frontend methods. After the browser hijacks video playback, the experience is not optimized well. You can try directly using the video tag to play (without going through the player generation) or try using Canvas to draw the video. If it still cannot be solved, it can only be solved by upgrading the browser.

## Player Display Size

### Black Border Appears On the Player

- **Problem manifestation:** When playing video, a black border appears in the player area.
- **Solution:** Set the aspect ratio of the player to match the actual aspect ratio of the video. For example, if the video resolution is 1280 x 720, the player size can be set to 640 x 360 or 1280 x 720, etc. As long as the 16:9 (1280:720) aspect ratio is met, the video can be fully displayed without black bars on the player. If the video has its own black bars, it is necessary to cut off the black bar content of the video during transcoding and change the resolution of the video.

### Streaming End Switches Portrait/Landscape Orientation While Playback Side Does Not Switch

- **Problem manifestation:** During the push process, the streaming end device switches between portrait and landscape orientations, while the playback side maintains the original aspect ratio.
- **Solution:** The Web player currently cannot detect that the streaming end has switched between portrait and landscape orientations and can only be handled through other

means. For example, when the stream starts in portrait mode with an upstream video aspect ratio of 9:16, and the Web playback side also plays at 9:16. At this time, if the streaming device does not disconnect the stream (whether the stream is disconnected needs to be supported by the streaming SDK) and switches to landscape mode, and the upstream video aspect ratio changes to 16:9. If the downstream video also changes to 16:9, the Web player needs to be reconnected to play the video with the switched aspect ratio. This operation requires an external API to notify the Web player. If the downstream video remains at 9:16, the video will continue to play at 9:16.

## Full-Screen Related Issues

This mainly introduces full-screen related issues. First, you need to understand two concepts: screen full-screen (system fullscreen) and webpage full-screen (page fullscreen, pseudo-fullscreen).

- **Screen Full-Screen**: It refers to being fullscreen within the screen range. After entering fullscreen, only the video picture content is visible, and the browser's address bar and other interfaces cannot be seen. This kind of fullscreen requires API support from the browser. There are two APIs that support screen fullscreen. One is called Fullscreen API. After entering screen fullscreen through Fullscreen API, the player interface composed of HTML and CSS can still be seen. The other API is webkitEnterFullScreen, which can only be applied to the video tag. It is usually used when the mobile terminal does not support Fullscreen API. After entering fullscreen through this API, the player interface is the system's built-in interface.
- **Web Page Full-Screen**: It refers to being fullscreen within the web page display area. After entering fullscreen, the browser's address bar and other interfaces can still be seen. Usually, web page fullscreen is a way to achieve a similar fullscreen effect when the browser does not support system fullscreen, so it is also called pseudo-fullscreen. This fullscreen method is implemented by CSS.

The Web player of Video on Demand (VOD) adopts a full-screen scheme with screen Fullscreen as the main method and Web page Fullscreen as the auxiliary method. The priority of Fullscreen modes is Fullscreen API > webkitEnterFullScreen > Web page Fullscreen.
Due to the gradual restriction of Flash operation by browsers, the Web player of Video on Demand (VOD) has been developed using the HTML5 standard and the use of Flash has been reduced. On some older browsers, the fullscreen feature is limited. The old version of the VOD player 1.0 was developed using Flash and implemented screen fullscreen using the Flash plugin. If you need to enter screen fullscreen in a browser that does not support Fullscreen API, you can only use the old version of the VOD player 1.0.
Currently known fullscreen situations:

| Device Type | Fullscreen Situations |
| --- | --- |

| | |
|---|---|
| x5 kernel (including WeChat, Mobile QQ and QQ browser on Android end) | Does not support Fullscreen API, but supports webkitEnterFullScreen. After entering fullscreen, it enters the screen fullscreen mode of the x5 kernel. |
| Android Chrome | Supports Fullscreen API. After entering fullscreen, it enters the screen fullscreen mode with Tencent Cloud Player UI. |
| iOS (including WeChat, Mobile QQ, Safari) | Does not support Fullscreen API, but supports webkitEnterFullScreen. After entering fullscreen, it enters the screen fullscreen mode of the iOS system UI. |
| IE8/9/10 | Does not support Fullscreen API or webkitEnterFullScreen. Fullscreen is in web page fullscreen mode. |
| WeChat Browser on desktop | Does not support Fullscreen API or webkitEnterFullScreen. Fullscreen is in web page fullscreen mode (macOS WeChat Browser currently does not support any fullscreen mode). |
| Other modern browsers on desktop | Usually supports Fullscreen API. After entering fullscreen, it enters the screen fullscreen mode with Tencent Cloud Player UI. |

## Default Fullscreen Playback

Same as Forced Full-Screen After Video Activation and Playback, refer to its solution.

## Default Fullscreen Playback in the WebView of IOS Hybrid App

- **Problem manifestation:** Videos play in fullscreen by default in the App WebView.
- **Solution:** Configure the WebView parameter allowsInlineMediaPlayback = YES to allow inline video playback, i.e., prohibit WebView/UiWebView from forcing videos to play in fullscreen.

## Using a Player in an Iframe Cannot Be Full-Screen

- **Problem manifestation:** Clicking the fullscreen button is unavailable when embedding the player page in an iframe.
- **Solution:** Set the attribute allowfullscreen in the iframe tag. Example code:

```
<iframe allowfullscreen src="" frameborder="0" scrolling="no"
```

```
width="100%" height="270"></iframe>
```

## Unable to Enter Fullscreen in IE8, 9, 10 Browsers

- **Problem manifestation:** The player cannot enter fullscreen mode in IE8/9/10 browsers and can only fill the page area. Or when embedding the playback page using an iframe, even with the allowfullscreen attribute added to the iframe, fullscreen is not possible.
- **Solution:** In older browsers that do not support the Full Screen API, VOD player uses CSS to achieve webpage full screen. Combined with browser fullscreen (the shortcut key for browser fullscreen is usually "F11"), it can achieve the screen fullscreen effect. Here, the CSS of the page should not restrict the player's in-page fullscreen style. For example, do not set the player's parent container to `overflow:hidden` .
 If it is in an iframe, the player cannot modify the CSS style outside the iframe. An external page needs to provide script and style support. Usually, the external page needs to support cross-domain access to achieve webpage full screen. Therefore, it is not recommended to use the player in an iframe.

> **Note:**
> IE8/9/10 browsers do not support the Full Screen API, so fullscreen cannot be achieved through the Full Screen API.

## Drag and Drop Time Shift Playback Failure

- **Problem manifestation:** Unable to play when dragging to a certain time point, or jumping to the opening scene.
- **Solution:** Avoid using the original video for playback. Use the video transcoded by Tencent Cloud Transcoding for playback. Avoid using Flash for playback. Switch to HTML5 playback mode. If the video duration is too short, there is usually only one key frame, and drag-and-drop playback is not supported.

## Issues Related to Autoplay

### Autoplay Failure

- **Problem manifestation:** The autoplay attribute is set, but the video does not autoplay.
- **Solution:** In many browsers, autoplay of multimedia files is disabled, especially on mobile terminals. Some browsers allow autoplay of muted videos or video with no audio track. Therefore, you can try setting the player to mute. For browsers that do not support autoplay even in mute mode, there is currently no solution.

### Autoplay Failure in the WebView of a Hybrid App

- **Problem manifestation:** Autoplay fails in the App WebView.
- **Solution:** You need to set the attributes of WebView for automatic playback of multimedia:
  - iOS: mediaPlaybackRequiresUserAction = NO
  - Android: webView.getSettings().setMediaPlaybackRequiresUserGesture(false)

# Other Issues

## Video Footage Not Visible After Player Initialization

- **Problem manifestation:** After the player is initialized and before playback starts, the video footage is not visible and the player area shows a black screen.
- **Solution:** Whether the Web player displays the first frame image of the video depends on whether the browser supports it. Currently, not all browsers support the first frame image. The solution is to set the cover of the video.

## The Player Has No Adjustable-Speed Playback Button or the Adjustable-Speed Playback Feature Is Unavailable

- **Problem manifestation:** In some browsers, there is no adjustable-speed playback button or the adjustable-speed playback feature is unavailable when playing videos.
- **Solution:** Currently, only some modern browsers support the adjustable-speed playback feature in the HTML5 playback mode, and the Flash playback mode does not support adjustable-speed playback. Therefore, browsers that do not support HTML5 mode playback also do not support adjustable-speed playback. You can give priority to using the HTML5 mode for playback. If there is no adjustable-speed playback button, it means that the current playback mode does not support adjustable-speed playback; if there is an adjustable-speed playback button but switching has no effect, it means that the player detects that the current browser supports the adjustable-speed playback API setting, but there is no effect after actual setting. It is recommended to hide the adjustable-speed playback button in this browser.

# Real-Time Interaction Related
# Android and iOS Related

Last updated: 2025-03-17 15:21:13

## What System Volume Modes Are Supported On the Mobile Terminal (Android/iOS)?

Support two system volume types, namely call volume type and media volume type.

- **Call Volume:** A volume type specifically designed for call scenarios on mobile phones. It uses the built-in echo cancellation function of the mobile phone. The sound quality is relatively poor compared to the media volume type. The volume cannot be adjusted to zero using the volume keys, but it supports the microphone on Bluetooth headphones.
- **Media volume:** A volume type specifically designed for music on mobile phones. Compared with the call volume type, the sound quality is better. The volume can be turned down to zero using the volume keys. When using the media volume type, if you want to enable the echo cancellation (AEC) feature, the SDK will activate the built-in acoustic processing algorithm to process the sound a second time. In media volume mode, Bluetooth headphones cannot use their built-in microphones to collect sound and can only use the microphone on the mobile phone for sound collection.

## How to Set 1080p Resolution For Mobile Terminal SDK Streaming?

1080P is defined as 114 in TX_Enum_Type_VideoResolution. You can directly set the resolution by passing the enumeration value.

## Create a Room On the Mini Program and Can the Mobile Terminal Enter It?

Yes, Tencent Real-Time Communication (TRTC) supports interconnectivity across the platform.

## Audio and Video Call How to Achieve Screen Recording (Screen Sharing) On TRTC Mobile Terminal?

- **Android end**: Screen recording on mobile phones is supported in Version 7.2 and later. For specific practices, see Real-time screen sharing (Android).
- **iOS platform**: Screen recording within the App is supported in Version 7.2 and later; phone screen recording and screen recording within the App are supported in Version 7.6 and later. For specific practices, see Real-time screen sharing (iOS).

## Audio and Video Call Can TRTC Android Support 64-Bit arm64-v8a Architecture?

Since version 6.3 of audio and video call TRTC, ABI support for the arm64-v8a architecture has been provided.

## How to Achieve Dynamic Loading of so Libraries On Android?

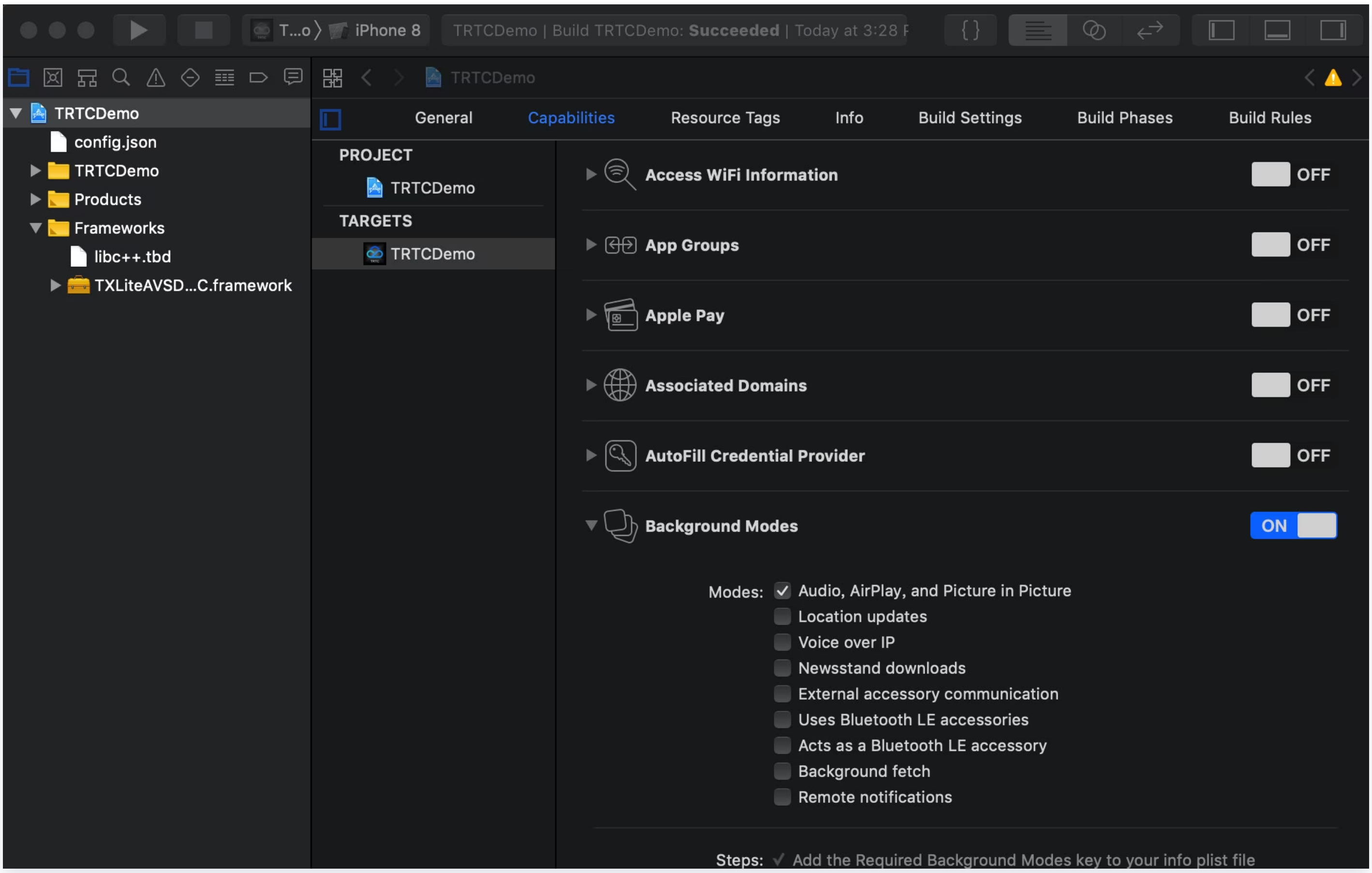
For specific operation steps, see Implementation of Dynamic Loading of so Libraries on Android.

## How to Solve the Conflict Error Problem Between iOS SDK and Other Third-Party Libraries?

For details, see Conflict Error Problem between iOS TXLiteAVSDK and Other Third-party Libraries.

## Can the TRTC SDK Support Backend Operation On iOS?

Yes, you can. To run the SDK in the background, select your current project, set Background Modes under Capabilities to ON, and check Audio, AirPlay, and Picture in Picture. For details, see the figure below:



## Can I Listen For Room Exit By Remote Users On iOS?

You can use onRemoteUserLeaveRoom to listen for user leaving room events. This API will only trigger a callback when all users in VideoCall or the streamer in LIVE mode leave the room. There will be no callback when an audience member leaves the room.

## How to Connect Audio and Video When the Mobile Phone Is in Screen Lock Status, the App Is in the Background or the App Is Closed?

Implement features such as offline answering. For details, see Implement Offline Answering.

## Is Android and Web Interconnectivity Supported?

Supported. Use the same SDKAppID and enter the same room for a call. For details, see the following document link to configure the Demo:

- Run through the Demo (Android)
- Run through the Demo (desktop browser)

## Streamer and Fan Mic Connections During Live Streams, Can Both Parties Initiate Mic Connections Actively?

Both parties can initiate proactively. The logic for audiences and streamers to initiate is the same. For specific operations, see Run Live Mode (Android).

## Multi-Person Audio-Video Room: Can Mobile Terminals and Web Access the Same Room?

Yes. Ensure that the SDKAppID and room number are consistent, and the user IDs are different.

## Create N TRTC Objects On the Same Page, Log in to N Rooms Through N UserIDs Respectively?

Yes. Starting from Version 7.6, a user can enter multiple rooms.

# Web Related

Last updated：2025-03-17 15:21:21

## Basic Environment Problem

### What Browsers Does the Web SDK Support?

For detailed browser support information of TRTC Web SDK, see TRTC Web SDK Browser Support.
For environments not listed above, you can open TRTC Capability Testing in the current browser to test whether WebRTC features are fully supported.

### Pre-Call Audio and Video Device Test?

You can view Pre-call Environment and Device Detection.

### How to Detect the Current Network Situation in Real Time?

For details, see Network Quality Detection Before Calls.

### Why Can the TRTC Web SDK Work Normally in Local Development Test but Not When Deployed Online?

Due to considerations of user security and privacy, browsers restrict web pages from accessing microphones and cameras only in secure environments (such as `https`, `localhost`, `file://` protocols). The HTTP protocol is insecure, and browsers will prohibit media device access under the HTTP protocol.
If your local development and testing work properly, but after the page is deployed, it cannot collect the camera and microphone normally. Then check whether your web page is deployed on the HTTP protocol. If so, deploy your web page using HTTPS and ensure that you have a valid HTTPS security certificate.
For more detailed information, see URL Domain Name and Protocol Restriction Description.

### Support For Stream Mixing, Bypass Streaming, Large and Small Flows, Beauty Effect, and Watermark?

You can refer to stream mixing, bypass streaming, large and small flows, beauty effect, watermark documents to implement advanced features.

### What Are the Known Issues of WebRTC?

For details, see Known Issues and Workarounds of WebRTC.

# 2. Push and Pull Stream Problems

## Meanings of NotFoundError, NotAllowedError, NotReadableError, OverConstrainedError and AbortError in Web SDK Logs

| Error Name | Description | Handling Suggestions |
| --- | --- | --- |
| NotFoundError | Cannot find a media type (including audio, video, screen sharing) that meets the request parameters. For example: If there is no camera on the PC, but the browser is requested to obtain a video stream, this error will be reported. | It is recommended to guide users to check the devices required for calls, such as cameras or microphones, before starting the call. If there is no camera and a voice call is needed, you can specify only capturing the microphone in TRTC.createStream({ audio: true, video: false }). |
| NotAllowedError | The user has rejected the current browser instance's requests for access to audio, video, and screen sharing. | Note that users will not be able to make audio and video calls if they do not authorize camera/microphone access. |
| NotReadableError | The user has authorized the use of the corresponding device, but the device cannot be accessed due to an error occurring at the hardware, browser, or web page level on the operating system. | Handle according to the browser's error information, and prompt the user "Temporarily unable to access the camera/microphone. Make sure no other applications are currently requesting access to the camera/microphone and try again". |
| OverConstrainedError | The value of the cameraId/microphoneId parameter is invalid. | Ensure that the values passed for cameraId/microphoneId are correct and valid. |
| AbortError | The device cannot be used due to some unknown reasons. | – |

For more detailed information, see initialize.

## Some Browsers On Mobile Phones Cannot Run TRTC Normally For Live Streaming, Publishing and Playback?

For detailed browser support information of TRTC Web SDK, see TRTC Web SDK Browser Support.

For environments not listed above, you can open TRTC Capability Testing in the current browser to test whether WebRTC features are fully supported.

## Is the Resolution Set By Width and Height For Web – End Streaming Applicable to All Browsers?

Due to device and browser limitations, the video resolution may not match exactly. In this case, the browser will automatically adjust the resolution to make it close to the resolution corresponding to the Profile. For more details, see setVideoProfile.

## Can the Style of Web – End Screen Sharing Be Modified?

The style of screen sharing is controlled by the browser and cannot be modified at present.

## Does the Web End Support Stream Mixing?

The Web end supports initiating stream mixing. For details, refer to How to Call the Mixed-Stream Transcoding API.

## Clearing Data in the Camera List When Unplugging the Camera During the Use of Web SDK

You can try calling the TRTC.getCameras method to see if you can obtain a new device list. If the information of the unplugged camera is still present, it indicates that the browser layer has not refreshed this list either, and the Web end SDK cannot obtain new device list information either.

## WeChat Embedded Browser On iOS Can'T Stream Normally?

Refer to Browser Support Status to check the support status of push/pull stream on the WeChat embedded browser on iOS.

# Playback Issue

## Audio and Video Interconnection Issue Where There Is Picture but No Sound?

- Due to browser autoplay policy restrictions, audio playback may encounter a PLAY_NOT_ALLOWED exception. At this time, the business layer needs to guide users to manually operate Stream.resume() to resume audio playback. For details, see Handling Suggestions for Restricted Autoplay.
- Caused by unknown exceptions. Please query the audioLevel & audioEnergy at both the sending and receiving ends through the monitoring dashboard.

## Web Call Visual Not Displaying?

Check whether data has been obtained on the Web page. When confirming that data transmission and reception are normal, you can check whether the srcObject property of the `<video>` element has been assigned the correct mediaStream object. If the assignment is incorrect, it definitely won't display.

### Echo, Noise, Noise, Low Volume During Web Call?

When the devices of both parties are too close to each other, it is normal. Please keep A greater distance from each other during testing. When the other end hears echo, noise, or other sounds in the Web, it indicates that the 3A processing on the Web has not taken effect. If you use the native getUserMedia API of the browser for custom capture, you need to manually set the 3A parameters:

- echoCancellation: Echo Cancellation Switch
- noiseSuppression: Noise Suppression Switch
- autoGainControl: Automatic Gain Control Switch. For detailed settings, see media tracking constraints.

If you use the TRTC.createStream API for collection, there is no need to manually set the 3A parameters. The SDK enables 3A by default.

## Other

### How to Handle It When the SDK of Version 2.x and 3.x Cannot Make Normal Calls in Chrome Version 96+?

The latest version of Chrome 96 has deprecated Plan-B, which will cause the older versions (2.x, 3.x) of TRTC Tencent Real-Time Communication Web SDK to be unable to make calls. Please upgrade your Web SDK to our latest version (4.x) as soon as possible. The API of the 4.x version SDK is not compatible with the older versions (2.x, 3.x). Please refer to Quick Connection (Web) to upgrade and access the 4.x version SDK.

### Handling Errors When Running Web SDK: "RtcError: no valid ice candidate found"

This error indicates that the STUN penetration failed in the TRTC desktop browser SDK. Please check the firewall configuration. The TRTC desktop browser SDK relies on the following ports for data transmission. Add them to the firewall allowlist. After the configuration is completed, you can check whether the configuration takes effect by accessing and experiencing Official Website Demo.
For details, please refer to Firewall Limitation Response.

### Handling Client Errors: "RtcError: ICE/DTLS Transport connection failed" or "RtcError: DTLS Transport connection timeout"

This error indicates that the TRTC desktop browser SDK failed to establish a media transmission channel. Please check the firewall configuration. The TRTC desktop browser SDK relies on the following ports for data transmission. Add them to the firewall allowlist. After the configuration is completed, you can check whether the configuration takes effect by accessing and experiencing Official Website Demo.
For details, please refer to Firewall Limitation Response.

## Can the Web SDK Get the Current Volume?

You can obtain the current volume through getAudioLevel. For details, please refer to Switching Camera and Microphone.

## What Situations Will Trigger Client.on('Client-banned')?

This event will be triggered when a user is kicked out, for example: logging in with the same name user at the same time, or kicking the user out of the room by calling the backend RESTAPI Removing User.

> **Note**
> Simultaneous log-in by same-name users is not allowed, which may cause call issues for both parties. The business layer should avoid simultaneous log-in by same-name users.

For more specific details, please refer to CLIENT_BANNED Event.

## Can the Web End Listen to the Remote Leaving Room?

Supports listening to remote user departure events. It is recommended to use the client.on('peer-leave') event in client events to implement remote user departure notifications.

## Is TRTC'S Web, Mini Program Side and PC Interconnect?

Yes, TRTC (Tencent Real-Time Communication) supports interconnectivity across the platform.

## How to Implement the Screenshot Feature of TRTC Web

For details, please refer to Stream.getVideoFrame() API.

## How Does the Web SDK Record Audio-Only Streaming? Why Does Automatic By-Pass and Automatic Recording Not Succeed When Enabled in the Console?

You need to set the pureAudioPushMode parameter of createClient.

## How to Handle the Client.on('error') Issue?

This indicates that the SDK has encountered an unrecoverable error. The business layer can either refresh the page and retry or call Client.leave to exit the room and then call Client.join to retry.

## Do Mini Programs and the Web Support Custom Stream IDs?

Custom Stream ID is supported in Web versions 4.3.8 and above. You can update the SDK version. Mini programs do not support it for now.

## How to Capture System Audio During Screen Sharing On the Web End?

For specific operations, see Screen sharing system sound capturing.
Currently, system sound capturing only supports Chrome M74+ . On Windows and Chrome OS, it can capture the audio of the entire system. On Linux and Mac, it can only capture the audio of tabs. Other Chrome versions, other systems, and other browsers are not supported.

## How to Switch the Camera and Microphone On the Web End?

You can first obtain the system's camera and microphone devices, and then call switchDevice to switch. For specific operations, see Switch Camera and Microphone.

## Permission denied Error When Using TRTC Web SDK in iframe?

Using WebRTC in an iframe requires adding attributes to the iframe tag to enable relevant permissions. For details, see below.
Microphone, camera, and screen sharing permissions:

```
<iframe allow="microphone; camera; display-capture;">
```

## Can'T Collect Camera and Microphone in WeChat H5?

In this case, it may be that your WeChat has cleared data or it is the first installation of WeChat. In this situation, the browser kernel that comes with WeChat during installation does not support device capturing.
WeChat will automatically download and install the XWeb kernel that supports device capturing when connected to Wi-Fi. You can try again after being connected to Wi-Fi for a while (for example, half an hour).

# Mini Program Side Related

Last updated：2025-03-17 15:21:29

## Environmental Issues

### What Are the Environmental Requirements For a Mini Program?

- Minimum version requirement of WeChat App for iOS: 7.0.9
- Minimum version requirement of WeChat App for Android: 7.0.8
- Minimum version requirement of mini program basic library: 2.10.0
- Since the test account of the mini program does not have the permission to use <live-pusher> and <live-player>, please use the enterprise mini program account to apply for relevant permissions for development.
- Since the WeChat Developer Tools do not support native components (i.e., <live-pusher> and <live-player> tags), it is necessary to run and experience on a real device.
- The uniapp development environment is not supported. Please use the native mini program development environment.

For more detailed information, see Quick Integration (Mini Program).

### What Should I Do If I Need to Launch or Deploy to the Official Environment On the Mini Program Side?

1. Apply for a domain name and complete the ICP Filing Registration.
2. Please deploy the server-side code on the applied server.
3. Configure the streaming domain and IM trusted domain in the request legal domains of the mini program console:
   - `https://official.opensso.tencent-cloud.com`
   - `https://yun.tim.qq.com`
   - `https://cloud.tencent.com`
   - `https://webim.tim.qq.com`
   - `https://query.tencent-cloud.com`
   - `wss://wss.im.qcloud.com`
   - `wss://wss.tim.qq.com`
   - `https://web.sdk.qcloud.com`

### Does the Mini Program Support Development Environments Such As uniapp and taro?

TUICalling, TUIVoiceRoom and other native components are not supported for now.

### Can Gravity Sensing Be Disabled On the Mini Program Side?

The gravity sensor API for mini programs is not available yet.

### Can the Mini Program Support Being Minimized Into a Floating Window?

Currently, when there is a live-pusher with mode='RTC' and at least one live-player on the page, the mini program can normally collect and play audio when running in the background. Otherwise, the mini program will terminate the audio and video call when switching to the background.

## Integration Issues

### Does the Mini Program Side Distinguish Check-Out Event Types, Such As Voluntary Check-Out, Being Kicked, and Dissolving Room?

EVENT.KICKED_OUT indicates that the server removes a user or the room is dissolved and exited. LOCAL_LEAVE indicates local room exit.

### Why Is the Picture Cropped When Horizontal Streaming On the Mini Program Side?

On iOS, you can enable portrait orientation lock. On Android, there is currently no way to avoid this issue. It requires modification at the mini program's underlying level.

### Which Value Is Used to Judge the Network Fluctuation of the Mini Program?

Determine by netQualityLevel:

- Undefined
- Best
- 2: Good
- 3: Normal
- 4: Poorly
- 5: Very Poor
- 6: Unavailable

### Does the Mini Program Side and Web Support Custom Stream ID?

- The Mini Program supports custom stream IDs starting from WeChat version 7.0.12. When constructing rtcConfig, fill in the custom stream ID in the field. For details, see Demo implementation.

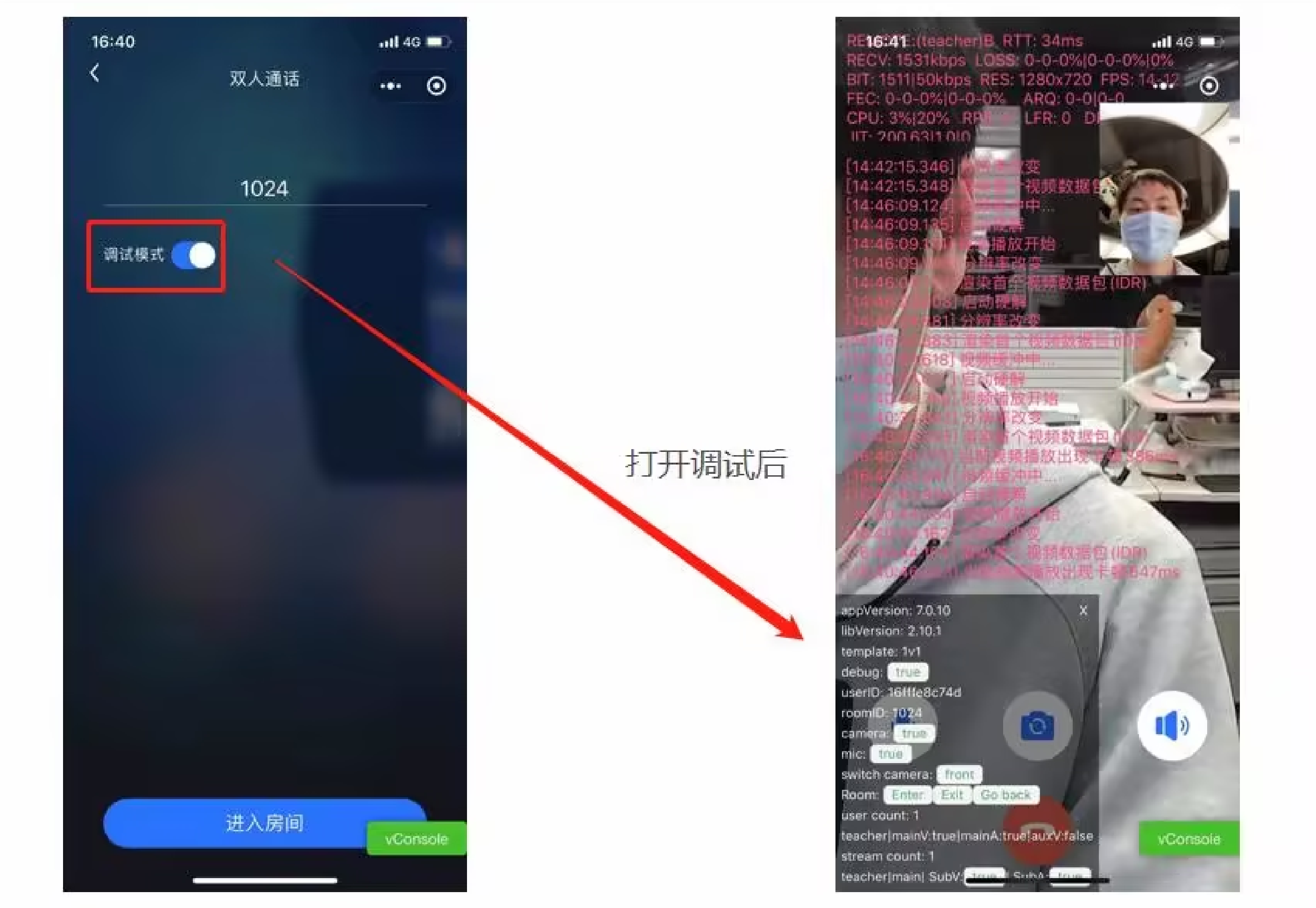
- Web version 4.3.8 and above supports custom stream ID. Customize this field when creating a client.

## Why Does One Need to Enable Debug Mode When Debugging On the Mini Program Side?

After enabling debugging, you can skip adding "request legal domain" to the allowlist of the mini program to avoid login failure and call connection issues.

## Why Does a Black Screen/Screen Freeze Occur On the Mini Program Side?

You can check the control panel in the lower left corner of the mini program Demo. Turn on **Debug Mode** to see detailed push and pull stream information on the interface. If there is no push and pull stream information, it means that room entry or live-pusher/live-player creation has failed.




## Preparations Needed Before Integrating Tencent Real-Time Communication (TRTC) SDK On the Mini Program Side

1. Create a Tencent Cloud TRTC application, purchase the corresponding package, and obtain the SDKAppID and key information.
2. Mini program server domain name configuration.
3. Enable permissions for mini program categories and live streaming, publishing and playback tags. (If not enabled, normal use will be unavailable.)
   For policy and compliance reasons, WeChat has not yet enabled support for all mini programs to use real-time audio and video features (i.e., <live-pusher> and <live-player> tags):
   - Mini program push/pull stream Tags do not support personal mini programs, but only support enterprise mini programs.
   - Mini program push/pull stream Tag usage permissions are currently only open to a limited number of categories.
   - Mini programs that meet the category requirements need to self-enable this component permission in WeChat public platform > **Develop** > **API Settings**, as shown below:


For more detailed information, see Run through Demo (Mini Program) and Quick Integration (Mini Program).

## How to Handle No Visual On the Mini Program Side When Entering Multi-Person Audio and Video?

1. Please run on a real mobile phone. The emulator inside WeChat Developer Tools currently does not support direct running.

2. Please query the mini program base library version through wx.getSystemInfo. The minimum version requirement of the mini program base library is 2.10.0.
3. Please confirm the category to which the mini program belongs. Due to regulatory requirements, the permission to use the Push and Pull Streams Tags of the mini program is currently only open to a limited number of categories.

If you have more requirements or wish to cooperate further, you can submit a ticket or call 4009100100 to contact us.

## Troubleshooting When a Running Error Occurs On the Mini Program Side

1. Please check if the opened mini program category is correct, and whether the <live-pusher> and <live-player> tags have been enabled.
2. Allowlist of mini program domains has been added to the valid domain of mini program requests, or debugging mode has been enabled.
3. Please decompress the Mini Program end Demo and run it directly. If it runs normally, it is recommended to refer to Quick Integration (Mini Program) to reintegrate the SDK.
4. If the problem persists, you can log in to WeChat Mini Program Developer Community to find relevant information, or submit a ticket or call 4009100100 to contact us.

## <live-pusher> and <live-player> Tag Usage and Error Code?

- live-pusher error code
- live-player error code
- livePusherContext
- livePlayerContext

## Can the Mini Program Be Monitored When It Is Minimized to the Background?

Yes. You can listen to the onHide method of the mini program to check whether the user has minimized it to the background.

## Why Can'T I Call or Am I Forced to Log Out?

The component does not currently support login multiple instances and does not support the **Offline Push Signal** feature. Please confirm the uniqueness of account login.

- Multi-instance: A userID logging in repeatedly or logging in on different ends will cause signaling chaos.
- Offline push: An instance can receive messages only when it is online. Signaling received when an instance is offline will not be pushed again after it goes online. That is, a mini

program cannot receive incoming call reminders or call reminders when it is in the background or offline.

## How to Distinguish Between Streamers and Audiences?

On the access side, there is no need to set the streamer/audience identity. The SDK itself distinguishes based on whether there is an upstream flow. When `enableCamera || enableMic` in the pusherAttributes properties is true, it is a streamer. When there is no audio or video upstream, it is an audience.

## How Does the Mini Program Receive SEI Messages?

You can refer to WeChat mini program API LivePusherContext.sendMessage for SEI sending. Receive it by listening to the [live-player] bindstatechange 2012 event. The code reference is as follows:

```
playerStateChange(e) {
  const { code, message } = e.detail
  console.log('playerStateChange', code, e)
  switch (code) {
    case 2012:
    {
      wx.showToast({ title: `收到SEI消息:${message}`, icon: 'none' })
      console.log('收到SEI消息: ', code, message)
      break
    }
  }
},
```

## How Does the Mini Program Use a String Room Number to Enter Room?

Use strRoomID when entering room. The priority of this parameter is higher than that of roomID.

## How to Play Background Music?

The playBGM method is provided in pusherInstance. For details, see pusherInstance. If you use WeChat's native tags for playback, it may be incompatible with some models, resulting in abnormal volume modes or abnormal playback on the receiver speaker.

## How to Suspend Streaming On the Mini Program Side?

On the mini program side, audio stream and video stream can be set to pause streaming separately. The relevant methods in pusherInstance are as follows:

- Pause video stream: The pause method is provided in <pusherInstance>. Calling this API will display a black screen frame on the remote end.
- Pause the audio stream: The setMICVolume method is provided in pusherInstance. Calling this API can set the locally collected volume to 0, and the remote end will not receive any sound.

For example, in scenarios where it is necessary to suspend two streams simultaneously, you can call pause to suspend the video stream and at the same time, call setMICVolume to set the volume to 0 to suspend the audio stream.
Both of the above methods will retain the occupation of the camera and microphone devices and are the recommended ways to pause streaming.
pusherAttributes also provides attributes to directly disable devices: enableMic, enableCamera. When these two attributes change, the remote end will receive a status change event. For specific phenomena, see the table below:

| State Changes Invocation | Enable Mic Current Value | Enable Camera Current Value | Mini Program Phenomenon | Web Phenomenon |
| --- | --- | --- | --- | --- |
| enableMic: false | true | true | REMOTE_AUDIO_REMOVE event | Receive the mute audio event in PLAYER_STATE_CHANGED |
| | true | false | REMOTE_AUDIO_REMOVE event | Receive the stream-removed event |
| enableCamera: false | true | true | REMOTE_VIDEO_REMOVE event | Receive the mute video event in PLAYER_STATE_CHANGED |
| | false | true | REMOTE_VIDEO_REMOVE event | Receive stream-removed event |

## Are the Mini Program Side, Desktop Browser and PC of TRTC (Tencent Real - Time Communication) Interconnect?

Yes, TRTC supports interconnectivity across the platform.

# Flutter Related

Last updated：2025-03-17 15:21:40

## Why Can'T Two Mobile Phones Running the Demo Simultaneously See Each Other'S Visuals?

Ensure that two phones use different UserIDs when running the Demo. TRTC does not support the same UserID (unless the SDKAppID is different) being used on two terminals simultaneously.


## What Limitations Does the Firewall Have?

Since the SDK uses the UDP protocol for audio and video transmission, it cannot be used in office networks with UDP interception. If you encounter similar problems, see Responding to Company Firewall Restrictions to troubleshoot and solve them.

## iOS Package Running Crash?

Please check if it is an issue with the debug mode in iOS 14 or later. For details, see Official Description.

## What Should I Do If Videos Do Not Show On iOS but Do On Android?

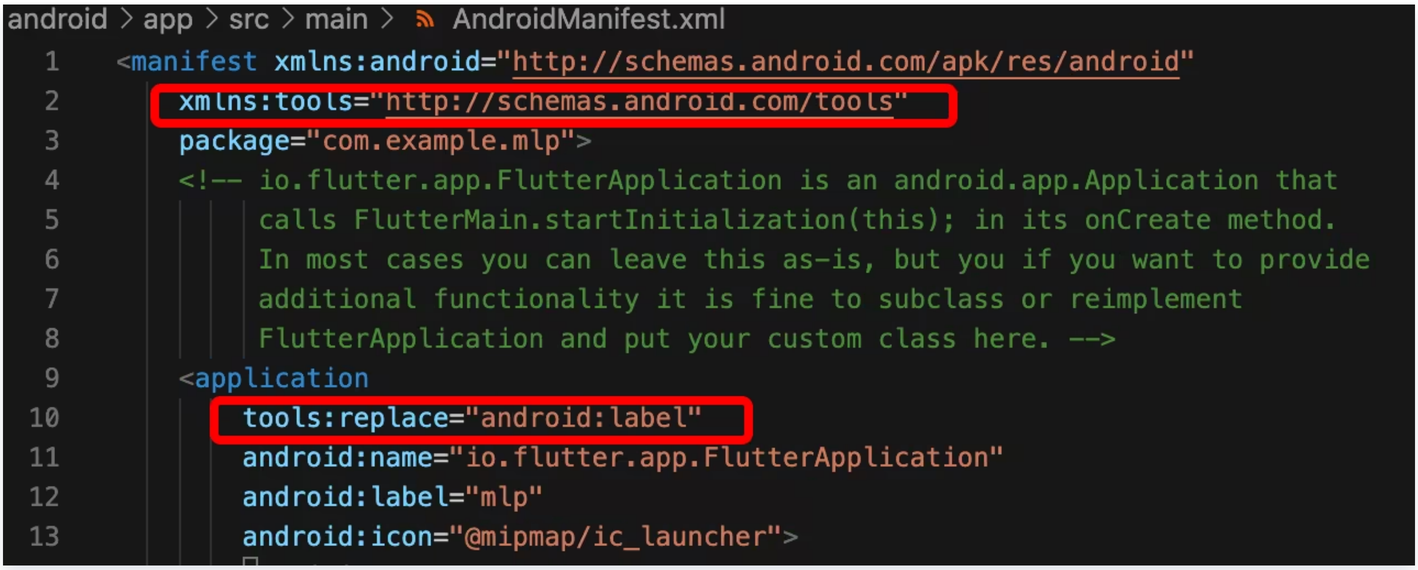
Please confirm that the value of `io.flutter.embedded_views_preview` in your project's `info.plist` is YES.

## Error When iOS CocoaPods Runs After Updating SDK Version?

1. Delete the `Podfile.lock` file in the iOS directory.
2. Run `pod repo update`.
3. Run `pod install`.
4. Rerun.

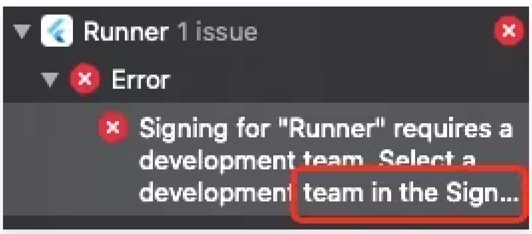
## Android Manifest merge failed Compilation Failure?

1. Open `/example/android/app/src/main/AndroidManifest.xml`.
2. Add `xmlns:tools="http://schemas.android.com/tools"` to the manifest.
3. Add `tools:replace="android:label"` to the application.




## Error in Real Device Debugging Due to No Signature?

If the error message is as shown below:



1. Purchase an Apple certificate, and you can debug on a real device after configuration and signing.
2. After the certificate is successfully purchased, configure it in `target > signing & capabilities`.

## After Addition and Deletion of swift Files in the Plug-In, Corresponding Files Cannot Be Found During build?

In the main project directory, run `pod install` under the `/ios` file path.

## Run error "Info.plit, error: No value at that key path or invalid key path: NSBonjourServices"?

After executing `flutter clean`, rerun it.

## Pod install Error?

If the error information is as shown below:

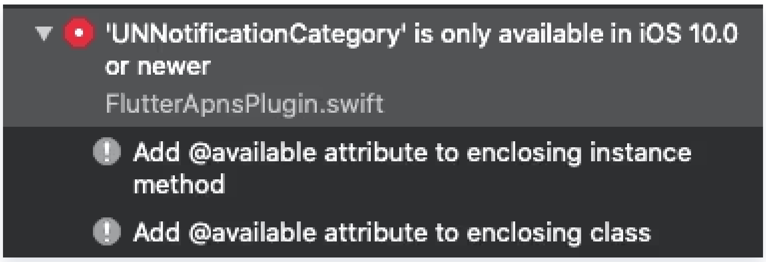
```
linzhi@MacBook-Pro ios % pod install

[!] Invalid `Podfile` file: /Users/linzhi/Desktop/source/trtc-flutter-plugin-demo/trtc-flutter-plugin/example/ios/Flutter/Generated.xcconfig must exist. If you're running pod install manually, make sure flutter pub get is executed first.
```

The error information indicates that there is no `generated.xconfig` file during pod install, so a running error occurs. According to the note , **you need to execute flutter pub get to solve it.**

> **Note:**
> This issue occurs after Flutter compilation. It does not exist in new projects or after executing `flutter clean`.

## iOS Version Dependency Error When Running?

If the error message is as shown below:




The error may occur because the pods target version fails to meet the requirements of the plugin being depended on. You need to change the target in the pods in question to the specified version.

## Does Flutter Support Custom Collection and Rendering?

Currently not supported. For details on custom capture and rendering platform support, please refer to Supported Platforms.

# Issues Related to Electron

Last updated：2025-03-17 15:24:19

## Installation Related

### Is Trtc-Electron-Sdk Compatible With the Official Electron V12.0.1 Version?

Compatible. The trtc-electron-sdk does not have special dependencies on electron's own sdk, so there are no related version dependencies.

### Slow Download or Even Getting Stuck When Downloading Electron?

When starting to download the `tmp-3320-1-SHASUMS256.txt-6.1.9` file or other files, the download may be extremely slow. Even after waiting for a long time, you may encounter an npm Timeout error:

```
Downloading tmp-3320-1-SHASUMS256.txt-6.1.9
[=>                                     ] 1.0% of 5.56 kB (0 B/s)
```

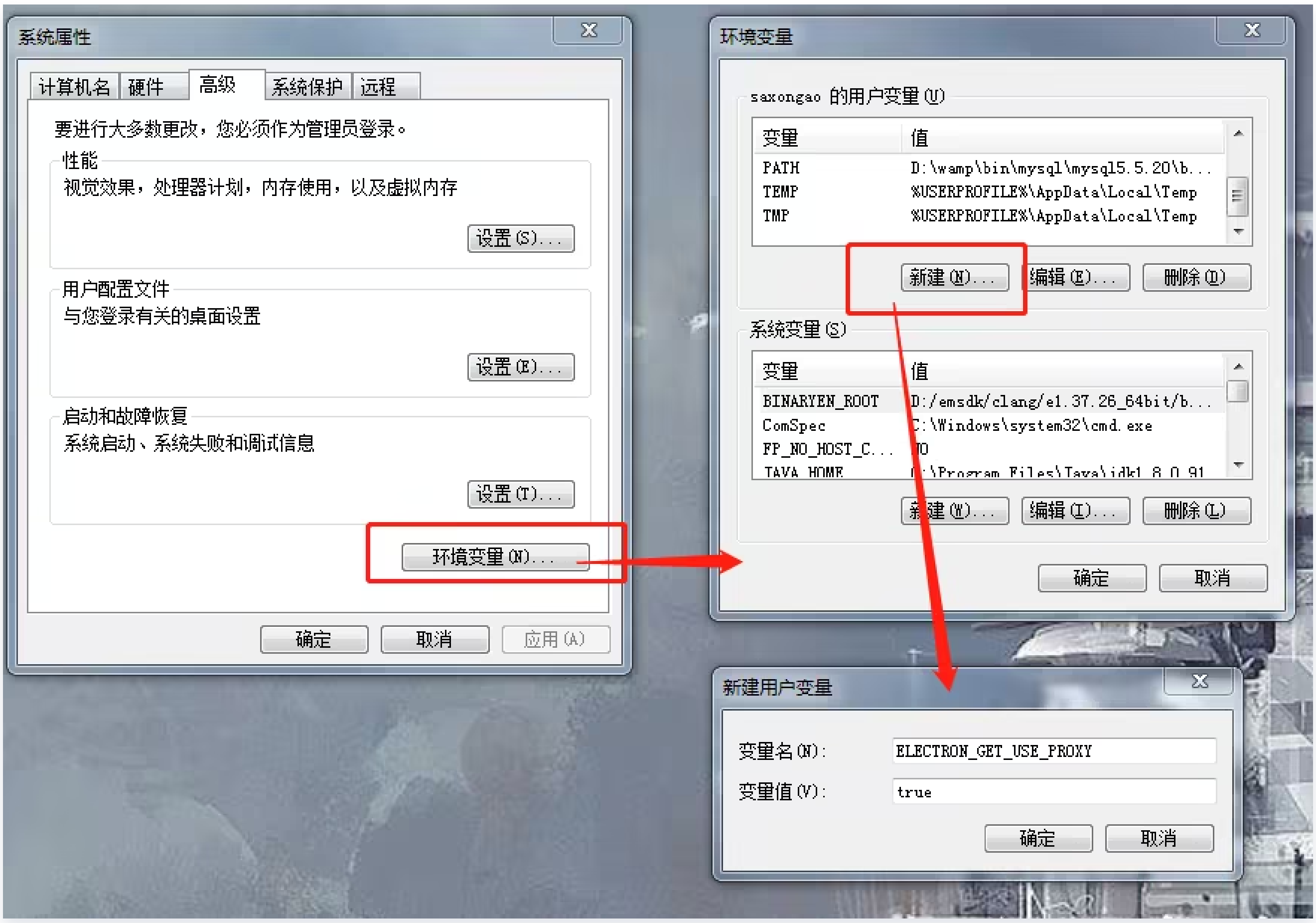
- <Solution A: >If you are working from home, you can switch to a Chinese mainland npm mirror.

  ```
  # Specify the domestic mirror of npm
  npm config set registry http://mirrors.tencent.com/npm/

  # Specify the domestic mirror address of Electron
  npm config set
  ELECTRON_MIRROR=https://registry.npmmirror.com/mirrors/electron/
  npm install
  ```

- **Solution B:** If you are working in a company, your company's network administrator may have already set up a proxy. You need to confirm whether npm's proxy configuration points to the company's proxy server and whether the environment variable `ELECTRON_GET_USE_PROXY` is configured. If none of them are configured, follow the steps below.
  1.1 Set npm proxy : `npm config set all_proxy=[your proxy address]` .
  1.2 Configure the `ELECTRON_GET_USE_PROXY` environment variable so that the Electron installation script downloads via the npm proxy.
- **Solution C:** If you are in a Mac environment.

```
export ELECTRON_GET_USE_PROXY=true
```

- **Solution D:** If you are in a Windows environment.
  1.1 Right-click **Computer** > **Properties** > **Advanced system settings** > **Environment variable.**
  1.2 Follow the steps below to set the environment variable ELECTRON_GET_USE_PROXY, and then execute `npm install` or `npm install --proxy=[your proxy address]`:




## 404 Error When Downloading Electron?

```
(node:23166) UnhandledPromiseRejectionWarning: HTTPError: Response code 404 (Not Found) for https://npm.[redacted].org/mirrors/electron/v8.1.1/electron-v8.1.1-darwin-x64.zip
    at EventEmitter.emitter.on.response (/Users/[redacted]/.nvm/versions/node/v10.16.3/lib/node_modules/electron/node_modules/got/source/as-stream.js:35:24)
    at EventEmitter.emit (events.js:198:13)
    at module.exports (/Users/[redacted]/.nvm/versions/node/v10.16.3/lib/node_modules/electron/node_modules/got/source/get-response.js:22:10)
    at ClientRequest.handleResponse (/Users/[redacted]/.nvm/versions/node/v10.16.3/lib/node_modules/electron/node_modules/got/source/request-as-event-emitter.js:155:5)
    at Object.onceWrapper (events.js:286:20)
    at ClientRequest.emit (events.js:203:15)
    at ClientRequest.origin.emit.args [as emit] (/Users/[redacted]/.nvm/versions/node/v10.16.3/lib/node_modules/electron/node_modules/@szmarczak/http-timer/source/index.js:37:11)
    at HTTPParser.parserOnIncomingClient [as onIncoming] (_http_client.js:556:21)
    at HTTPParser.parserOnHeadersComplete (_http_common.js:109:17)
    at TLSSocket.socketOnData (_http_client.js:442:20)
(node:23166) UnhandledPromiseRejectionWarning: Unhandled promise rejection. This error originated either by throwing inside of an async function without a catch block, or by rejecting a promise which
was not handled with .catch(). (rejection id: 1)
(node:23166) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections that are not handled will terminate the Node.js process with a non-zero exit c
ode.
```

Enter the following command in the terminal:

```
npm config set electron_custom_dir 8.1.1 # Determine based on the
```

```
version number
```

# Operation Related

## Windows 32-Bit System Runtime Error `Error: Resource\Trtc_electron_sdk.Node Is Not a Valid Win32 Application`, Note That a 32-Bit Trtc_electron_sdk.Node Is Required?

```
▸Error: Cannot open resources\trtc_electron_sdk.node: Error:                chunk-vendors.be39b84f.js:57
resources\trtc_electron_sdk.node is not a valid Win32 application.
resources\trtc_electron_sdk.node
    at Object.<anonymous> (chunk-5974bb06.287d70ae.js:12)
    at Object.6a75 (chunk-5974bb06.287d70ae.js:12)
    at c (app.2e27c2f9.js:1)
    at Object.<anonymous> (chunk-5974bb06.287d70ae.js:1)
    at Object.0542 (chunk-5974bb06.287d70ae.js:1)
    at c (app.2e27c2f9.js:1)
    at Module.97ec (chunk-5974bb06.287d70ae.js:24)
    at c (app.2e27c2f9.js:1)
▸Uncaught (in promise) Error: Cannot open resources\trtc_electron_sdk.node:  chunk-vendors.be39b84f.js:57
Error: resources\trtc_electron_sdk.node is not a valid Win32 application.
resources\trtc_electron_sdk.node
    at Object.<anonymous> (chunk-5974bb06.287d70ae.js:12)
    at Object.6a75 (chunk-5974bb06.287d70ae.js:12)
    at c (app.2e27c2f9.js:1)
    at Object.<anonymous> (chunk-5974bb06.287d70ae.js:1)
    at Object.0542 (chunk-5974bb06.287d70ae.js:1)
    at c (app.2e27c2f9.js:1)
    at Module.97ec (chunk-5974bb06.287d70ae.js:24)
    at c (app.2e27c2f9.js:1)
```

**Solution:**

1. Enter the library directory of trtc-electron-sdk under the project directory (xxx/node_modules/trtc-electron-sdk). Execute:

```
npm run install -- arch=ia32
```

2. After downloading the 32-bit `trtc_electron_sdk.node`, repackage the project.

## Vscode Terminal Launching Electron Demo, White Screen After Entering the Room?

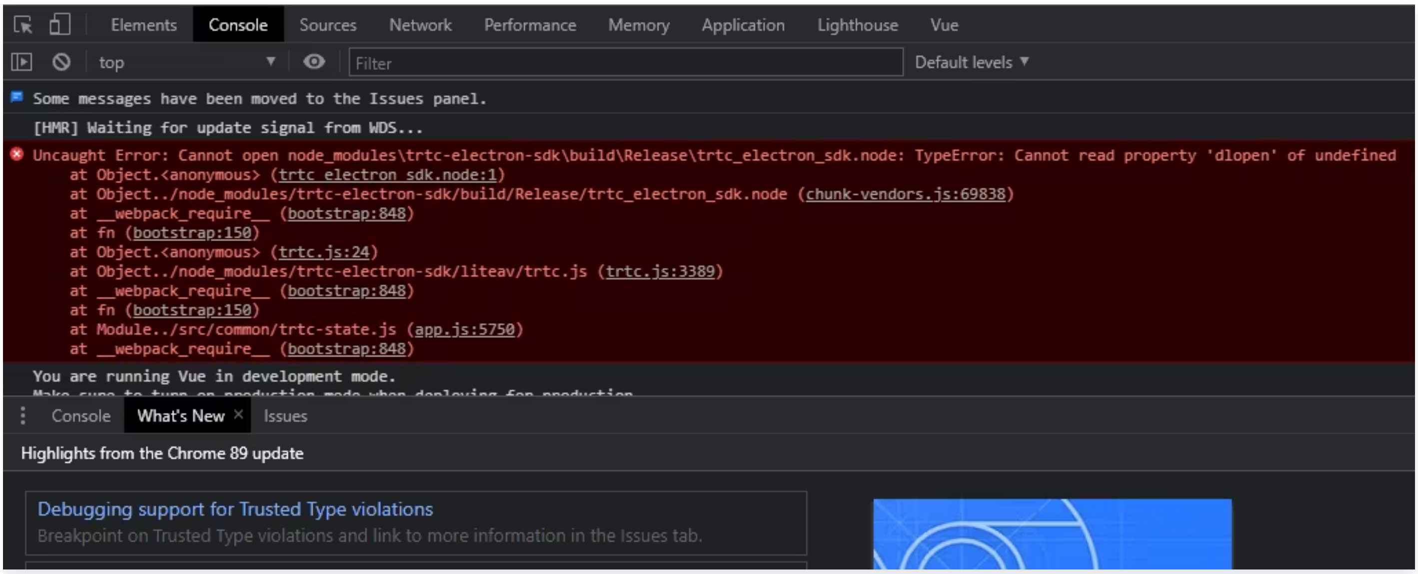
vscode needs camera permission, which can be added in the following way.

```
cd ~/Library/Application\ Support/com.apple.TCC/
cp TCC.db TCC.db.bak
sqlite3 TCC.db    # sqlite> prompt appears.

# for Mojave, Catalina
```

```
INSERT into access
VALUES('kTCCServiceCamera',"com.microsoft.VSCode",0,1,1,NULL,NULL,NULL,'
UNUSED',NULL,0,1541440109);

# for BigSur
INSERT into access
VALUES('kTCCServiceCamera',"com.microsoft.VSCode",0,1,1,1,NULL,NULL,NULL
,'UNUSED',NULL,0,1541440109);
```

## Running Demo Throws a Null Pointer Undefined Error: `"Cannot Read Property 'Dlopen' of Undefined"`?




**Solution:**

In Electron version 12, context isolation is enabled by default. You can set contextIsolation to false.

```
let win = new BrowserWindow({
    width: 1366,
    height: 1024,
    minWidth: 800,
    minHeight: 600,
    webPreferences: {
    nodeIntegration: true,
    contextIsolation: false
        },
});
```

## Electron Frequently Re-Enters Room Issue?

Specific cases need to be analyzed. The approximate reasons are as follows:

- The client's network status is poor (network disconnection will trigger re-entering the room).
- Sending the room entry signaling message twice in a row may also cause reentry into the room.
- It may be that the device load is too high, resulting in decoding failure and reentering the room.
- Re-entering the room caused by mutual kick due to multi-terminal log-in with the same UID.

## Terminal Shows the Prompt "Electron Failed to Install Correctly"?

When it seems that the installation is complete and the project is running, the following error appears on the terminal:

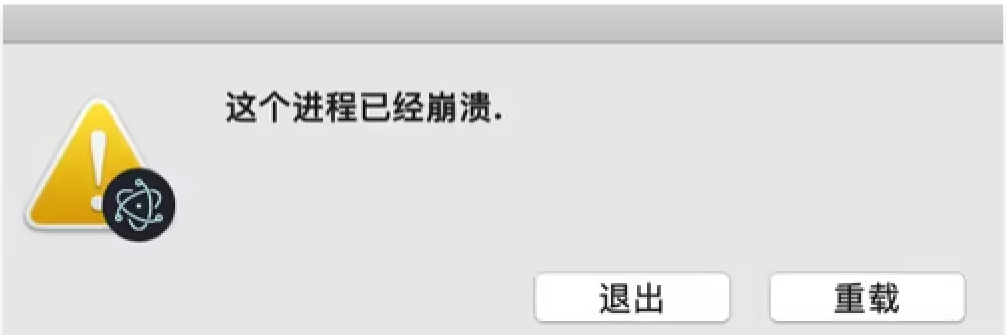
```
Error: Electron failed to install correctly, please delete
node_modules/electron and try installing again
```

Follow these three steps for **manual download**:

1. Perform `npm config get cache` to view the cache directory.
2. Manually download Electron and place it in the cache directory.
3. Re-run `npm install`.

## Crash When Calling the Camera or Microphone?

When using the vscode terminal to start the project, when trtc-electron-sdk starts the camera and microphone, the program crashes directly:




- **Solution A**: Run the project using an authorized terminal.
- **Solution B**: Authorize vscode: Allow vscode authorization in **System Preferences > Security and Privacy**.

- **Solution C**: Disable the protection mechanism by following the steps below:
  1.1 Restart the system. **Hold down** the command + r keys **until** the system enters safe mode.
  1.2 Open the terminal and enter `csrutil disable` to disable the protection mechanism.
  1.3 Restart and enter the system normally. At this time, you can use the vscode terminal to start the project.
  1.4 If you need to restart the protection mechanism, just execute `csrutil enable` in step two.

## Electron Reporting "Xx Is Not Defined" Error in the Console?

When running the project, Electron prompts `xx is not defined` in the console, where `xx` refers to a node module. For example:

```
Uncaught ReferenceError: require is not defined
```

In the `main.js` file of Electron, change the `nodeIntegration` configuration item to true:

```
let win = new BrowserWindow({
    width: 1366,
    height: 1024,
    webPreferences: {
        nodeIntegration: true,  // Set this item to true
    },
});
```

## White Screen and Crash Issues When Running After Packaging and Installation On Mac

On Mac OS 10.4 and later versions, when running the installation package, if the permissions of camera, microphone, and screen recording cannot be obtained, the program will directly experience a white screen or crash after entering the TRTC room due to lack of access permissions to these hardware devices. The solutions are as follows:

1. Add the entitlements.mac.plist file. The file content is as follows. For the specific meaning of relevant configurations, see Apple Developer Website.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
```

```
<plist version="1.0">
  <dict>
 <key>com.apple.security.cs.allow-unsigned-executable-memory</key>
 <true></true>
 <key>com.apple.security.cs.allow-jit</key>
 <true></true>
 <key>com.apple.security.cs.allow-dyld-environment-variables</key>
 <true></true>
 <key>com.apple.security.device.audio-input</key>
 <true></true>
 <key>com.apple.security.device.camera</key>
 <true></true>
  </dict>
</plist>
```

2. When packaging with electron-builder, you need to configure the file path of entitlements.mac.plist into the electron-builder packaging configuration. The reference code is as follows. Note `"entitlements"` and `"entitlementsInherit"` two configuration items. `"hardenedRuntime"` needs to be configured as true. For the meaning of the configuration items, see electron-builder official website.

```
{
  "build": {
    "mac": {
      "extraFiles": [
        {
          "from": "node_modules/trtc-electron-
sdk/build/Release/trtc_electron_sdk.node",
          "to": "./Resources"
        }
      ],
      "type": "distribution",
      "hardenedRuntime": true,
      "entitlements": "assets/entitlements.mac.plist",
      "entitlementsInherit": "assets/entitlements.mac.plist",
      "gatekeeperAssess": false,
      "target": [
        "dmg"
      ]
```

```
    },
  }
}
```

3. For Mac OS 12.1 and later versions, the above configuration alone is no longer sufficient to apply for microphone and camera permissions. It is necessary to use the Electron API's systemPreferences.askForMediaAccess() API to proactively request camera and microphone permissions in the main process. The reference code is as follows. The systemPreferences.getMediaAccessStatus() API can detect microphone, camera, and screen recording permissions, but the systemPreferences.askForMediaAccess() API can only request camera and microphone permissions.

```
async checkAndApplyDeviceAccessPrivilege() {
  const cameraPrivilege =
systemPreferences.getMediaAccessStatus('camera');
  console.log(
    `checkAndApplyDeviceAccessPrivilege before apply cameraPrivilege:
${cameraPrivilege}`
  );
  if (cameraPrivilege !== 'granted') {
    await systemPreferences.askForMediaAccess('camera');
  }

      const micPrivilege =
systemPreferences.getMediaAccessStatus('microphone');
  console.log(
    `checkAndApplyDeviceAccessPrivilege before apply micPrivilege:
${micPrivilege}`
  );
  if (micPrivilege !== 'granted') {
    await systemPreferences.askForMediaAccess('microphone');
  }

      const screenPrivilege =
systemPreferences.getMediaAccessStatus('screen');
  console.log(
    `checkAndApplyDeviceAccessPrivilege before apply screenPrivilege:
${screenPrivilege}`
  );
```

```
}
```

4. For more crash issue handling methods, please refer to Troubleshooting and Problem-Solving for Electron Application Crash Issues.

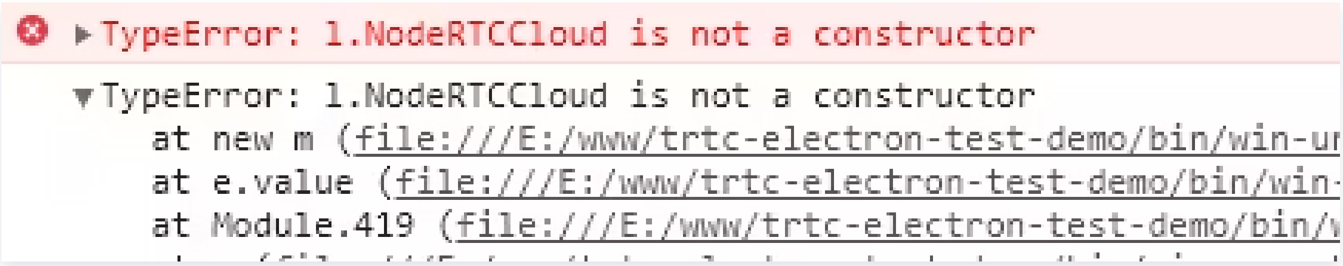
# Packaging Related

## Loading Issue of Node Module

### Error Message

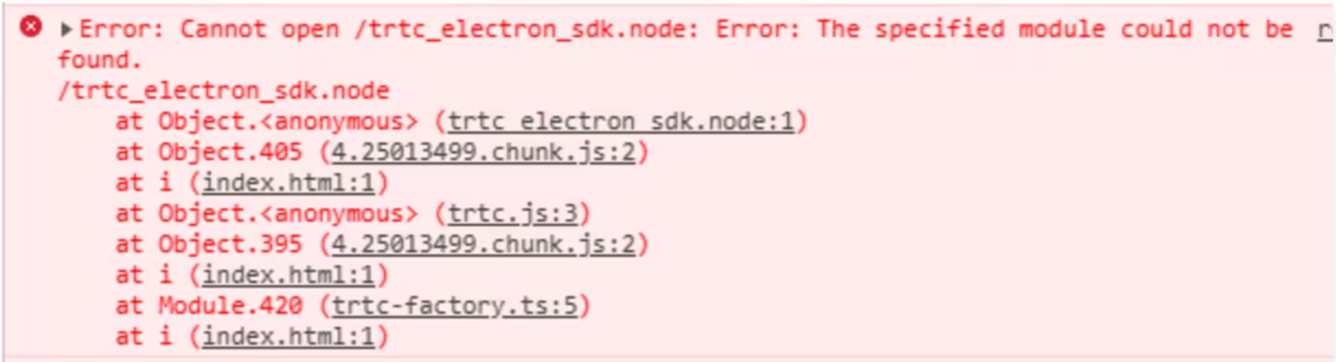
The packaged and compiled program shows similar error information in the console at runtime:

- `NodeRTCCloud is not a constructor`



```
TypeError: l.NodeRTCCloud is not a constructor
TypeError: l.NodeRTCCloud is not a constructor
    at new m (file:///E:/www/trtc-electron-test-demo/bin/win-ur
    at e.value (file:///E:/www/trtc-electron-test-demo/bin/win-
    at Module.419 (file:///E:/www/trtc-electron-test-demo/bin/w
```

- `Cannot open xxx/trtc_electron_sdk.node` or `The specified module could not be found`



```
Error: Cannot open /trtc_electron_sdk.node: Error: The specified module could not be
found.
/trtc_electron_sdk.node
    at Object.<anonymous> (trtc_electron_sdk.node:1)
    at Object.405 (4.25013499.chunk.js:2)
    at i (index.html:1)
    at Object.<anonymous> (trtc.js:3)
    at Object.395 (4.25013499.chunk.js:2)
    at i (index.html:1)
    at Module.420 (trtc-factory.ts:5)
    at i (index.html:1)
```

- `dlopen(xxx/trtc_electron_sdk.node, 1): image not found`

```
Error: Can not open nodeFile: Error: dlopen(/build/trtc_electron_sdk.node, 1): image not found
  at Object.<anonymous> (4.d24a6187.chunk.js:2)
  at Object.405 (4.d24a6187.chunk.js:2)
  at a (index.html:1)
  at Object.<anonymous> (4.d24a6187.chunk.js:2)
  at Object.395 (4.d24a6187.chunk.js:2)
  at a (index.html:1)
  at Module.420 (6.b83695ae.chunk.js:1)
```

### Solution

If similar information as above appears, it indicates that the trtc_electron_sdk.node module has not been correctly packaged into the program. You can follow the steps below to handle it.

1. `Install native-ext-loader`.

```
npm i native-ext-loader -D
```

2. Modify the webpack configuration.

2.1

```
const os = require('os');
// If the target_platform parameter is not passed, the program
will be packaged according to the current platform type by
default.
const targetPlatform = (function(){
 let target = os.platform();
 for (let i=0; i<process.argv.length; i++) {
    if (process.argv[i].includes('--target_platform=')) {
            target = process.argv[i].replace('--target_platform=',
'');
            break;
        }
    }
  // win32 uniformly represents the Windows platform, including
both 32-bit and 64-bit. darwin represents the Mac platform.
  if (!['win32', 'darwin'].includes) target = os.platform();
  return target;
})();
```

2.2 Add the following rules configuration:

```
module: {
 rules: [
 {
     test: /\.node$/,
     loader: 'native-ext-loader',
     options: {
         rewritePath: targetPlatform === 'win32' ? './resources' :
'../Resources'
     }
 },
```

```
  ]
}
```

> ⚠ **Note:**
> - For projects created with `vue-cli`, the webpack configuration is placed in the `configureWebpack` option of the `vue.config.js` file.
> - For projects created with `create-react-app`, the webpack configuration file is `[project directory]/node_modules/react-scripts/config/webpack.config.js`.

3. Configure the packages.json file, add packaging configuration and building scripts.

   3.1 Add `electron-builder` packaging configuration (note the case):

```
"build": {
 "Omission": "..."
 "win": {
 "extraFiles": [
     {
     "from": "node_modules/trtc-electron-sdk/build/Release/",
     "to": "./resources",
     "filter": ["**/*"]
     }
 ]
 },
 "mac": {
 "extraFiles": [
     {
     "from": "node_modules/trtc-electron-
sdk/build/Release/trtc_electron_sdk.node",
     "to": "./Resources"
     }
 ]
 },
 "directories": {
 "output": "./bin"
 }
},
```

3.2 Add scripts for building and packaging the `create-react-app` project. Refer to the following configuration:

```
"scripts": {
  "build:mac": "react-scripts build --target_platform=darwin",
  "build:win": "react-scripts build --target_platform=win32",
  "compile:mac": "node_modules/.bin/electron-builder --mac",
  "compile:win64": "node_modules/.bin/electron-builder --win --x64",
  "pack:mac": "npm run build:mac && npm run compile:mac",
  "pack:win64": "npm run build:win && npm run compile:win64"
}
```

3.3 For `vue-cli` projects, refer to the following configurations:

```
"scripts": {
  "build:mac": "vue-cli-service build --target_platform=darwin",
  "build:win": "vue-cli-service build --target_platform=win32",
  "compile:mac": "node_modules/.bin/electron-builder --mac",
  "compile:win64": "node_modules/.bin/electron-builder --win --x64",
  "pack:mac": "npm run build:mac && npm run compile:mac",
  "pack:win64": "npm run build:win && npm run compile:win64"
}
```

## Can'T Find the Entry File?

For projects created using `create-react-app`, this issue may occur when packaging with `electron-builder`:

```
$ node_modules\.bin\electron-builder.cmd
  • electron-builder  version=22.6.0 os=6.1.7601
  • loaded configuration  file=package.json ("build" field)
  • public/electron.js not found. Please see https://medium.com/@kitze/%EF%B8%8F-from-react-to-an-electron-app-ready-for-production-a0468ecb1da3
  • loaded parent configuration  preset=react-cra
```

`public/electron.js not found` means being unable to find the entry file.

### Solution

1. Move and rename the entry file:

```
cd [project directory]
mv main.electron.js ./public/electron.js
```

2. Modify the package.json file:

```
{
 "main": "public/electron.js",
 "Omission": "..."
}
```

## Syntax Error of Fs-Extra Module When Executing Packaging?

```
[Project directory]\node_modules\electron-builder\node_modules\fs-
extra\lib\empty\index.js:33
    } catch {
            ^


SyntaxError: Unexpected token {
    at new Script (vm.js:51:7)
```

You can upgrade to the latest version of node. For details, see Node.js Official Website.

## How to Build a Dual-Architecture Package For X64 and ARM64 On Mac?

Starting from version 10.6.403, TRTC Electron SDK supports building dual-architecture packages for X64 and ARM64 on Mac by default. You only need to modify the electron-builder and build tools such as Webpack or Vite configurations. For detailed configurations, see TRTC Electron SDK: Building Dual-Architecture Packages on Mac.
After supporting dual-architecture packaging, if you do not modify the configuration, a single-architecture application package will be built by default, and the architecture type is the same as the CPU type of the build machine.

> **Note:**
> For more Electron-related questions, please follow FAQ on Electron and FAQ on Electron II. We will continue to update.

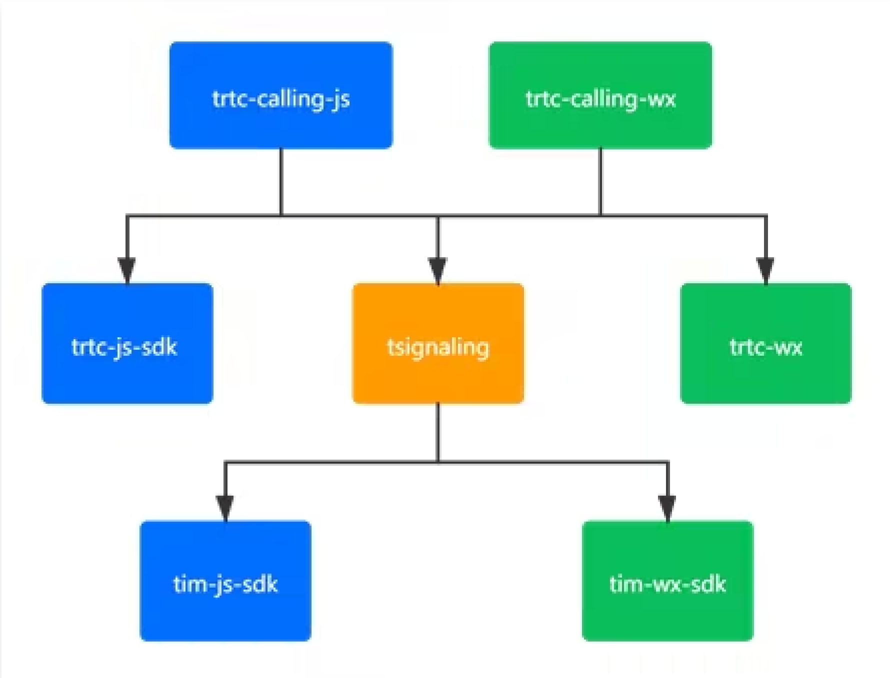
# TRTCCalling Web Related

Last updated：2025-03-17 15:26:49

## Basic Questions

### What Is TRTCCalling?

TRTCCalling is a quick integration audio and video solution born on the basis of TRTC and TIM. It supports 1v1 and multiple users' video/voice calls.




### Does TRTCCalling Support Accepting roomID As a String?

roomID can be a string, but only numeric strings are allowed.

## Environmental Problems

### Web SDK Supports Which Browsers?

For detailed browser support of TRTC Web SDK, see TRTC Web SDK Browser Support.
For environments not listed above, you can open TRTC Capability Testing in the current browser to test whether WebRTC features are fully supported.

### How to Detect the Current Network Situation in Real Time?

For specific operations, see Network Quality Detection Before Call.

## IM H5 Demo Project Runs Normally Locally but Fails in Video/Voice Call When Accessed Via IP On Server

- **Background**: After the H5 Demo of IM runs successfully locally, using localhost can normally implement text message sending, and video/voice call features. After putting the project on a server and accessing it via IP, text message sending and receiving, and console request responses are normal, and there are no errors in the console, but voice/video calls cannot run properly, and video images cannot be obtained.
- **Reason**: In IM, speech/video call uses the TRTCCalling SDK. When users access via IP, the HTTP protocol is used.
- **Solution**: The TRTCCalling SDK needs to run in an HTTPS or localhost environment.

# Integration Issues

## calling Online Demo Cannot Enter NO_RESP?

- **Reason**: NO_RESP. Trigger conditions: 1 – Inviter timeout. 2 – Invitee offline.
- **Solution**: Please perform event handling based on trigger conditions.

## calling in iPhone WeChat Browser Fails to Hear the Other Party'S Voice?

- Reason: Autoplay is restricted.
- Solution: The calling was processed in version 1.0.0. It is recommended that you upgrade calling to version 1.0.0 or later.

## TRTCCalling handup() Error: "uncaught (in promise) TypeError: cannot read property 'stop' of null"?

- **Reason**: The user calls handup() multiple times in the event listener, causing hangup to be triggered again before it is completed.
- **Solution**: handup() only needs to be executed once. For subsequent operations of the event listener, TRTCCalling has been processed internally. There is no need to execute the hangup() method again. Just do your own business-related operations.

## Latest Version 90 of Google Chrome Browser, trtccalling.js Prompts: "Unsupported, TRTCClinet.Your Browser Is Incompatible With This Application"?

- **Reason**: The IM version is too low and the detection mechanism is missing.
- **Solution**: It is recommended to upgrade the IM version.

## Reporting "TypeError: Cannot read property 'getVideoTracks' of null" During the Connection Process?

- **Reason**: This is caused by the user not having obtained permission to use the video and microphone when accepting.
- **Solution**: When using device methods such as startRemoteView and startLocalView, it is recommended to use asynchronous methods. Or upgrade the TRTCCaling version to 1.0.0.

## sdkAppid Reports "TSignaling._onMessageReceived unknown bussinessID=undefined" When Introduced By script Method?

- **Details**: For the same sdkAppid introduced via the script method, communication is possible with those introduced via script, but not with those introduced via npm or Android/iOS, and a warning message will be returned: `TSignaling._onMessageReceived unknown bussinessID=undefined`.
- **Reason**: `bussinessId=undefined` indicates that the tsignaling version is an old version, and there are problems with the signaling in the old version.
- **Solution**: Upgrade the tsignaling version. And during the introduction process, note that **the file name of the new version of tsignaling is** `tsignaling-js`.

## Reminder: "Uncaught (in promise) Error: the createCustomMessage API Can Only Be Called After the SDK Is in the ready Status."?

- **Reason**: Initialization was not completed according to the correct steps.
- **Solution**: Upgrade the TRTCCalling version to 1.0.0 and listen for the SDK_READY event for subsequent operations.

## Reminder: "Uncaught (in promise) RTCError: duplicated play() call observed, please stop() firstly <INVALID_OPERATION 0x1001>"?

- **Reason**: Calling the startRemoteView API during speech transmission.
- **Solution**: Cancel the startRemoteView operation during a voice call.

## Reminder: "Uncaught (in promise) Error: inviteID is invalid or invitation has been processed"?

- **Details**: Web – end trtccalling and native – end interconnectivity. After web calls native, native answers but the camera on the web end has not been turned on yet, and the local preview has no picture when hanging up. Native is still on the call page. Return error information: `Uncaught ( in promise ) Error: inviteID is invalid or invitation has been processed`.

- **Reason**: When obtaining user devices, if the user does not authorize audio and video devices, they can enter the audio and video call room, but native cannot receive the hangup signal when hanging up.
- **Solution**: In version 1.0.0 of calling, when pre – fetching is carried out and fails, users are not allowed to enter the call. It is recommended that you upgrade calling to version 1.0.0 or later.

## Caller Successfully Called, Callee Printed Logs (Should Have Received the Call), but Did Not Trigger the Callback handleNewInvitationReceived?

- **Reason**: The versions of TRTCCalling <= 0.6.0 and Tsignaling <= 0.3.0 are too low.
- **Solution**: Upgrade TRTCCalling and Tsignaling to the latest versions.

## Why Can'T I CALL Again After Actively Rejecting in TRTCCalling After CALL?

- **Reason**: Caused by the failure to reset the calling status after actively rejecting following a call.
- **Solution**: Upgrade TRTCCalling to version ⩾ 1.0.3.

## Error: TRTCCalling.call – Failed to Obtain User Device Permissions?

- **Reason**: TRTCCalling has no device permissions or access to the device.
- Solution
    - Use TRTC Device Detection to perform a check.
    - Access **Chrome's website settings** (chrome://settings/content) to check whether the websites using TRTCCalling have camera/mic permissions enabled.

## TRTCCalling web Support Receiving Offline Messages?

- Offline messages are not supported for reception.
- Supports offline message push. You can add the messages to be pushed through offlinePushInfo in call / groupCall.

# Stream Mixing and Recording Related

Last updated：2025-03-17 15:26:59

## Mixstream Transcoding Issues

### How Does TRTC Know Whether It Is Using the New mcu Mixed Streaming or the Old Cloud Mixed Streaming?

If the following conditions are met and the client log prints mcumix = 1, the new mcu stream mixing is used.

- Applications created on or after January 9, 2020.
- TRTC SDK version is after 6.9.

### Troubleshooting When TRTC Fails to Call the Mixed Streaming API and There Is No Effect

1. Ensure that **automatic bypass streaming** has been enabled in Tencent Real-Time Communication Console.
2. Listen to the onSetMixTranscodingConfig() API and modify according to the returned error information.
3. If the bypass stream ID is customized through the SDK API, the old cloud stream mixing method will fail.
4. If onSetMixTranscodingConfig() returns success but pulling the stream from the bypass CDN still has no effect, it may be caused by the lack of configuration of the playback domain. It is recommended to check the relevant configuration of the playback domain.

## New Version of Cloud Recording Issues

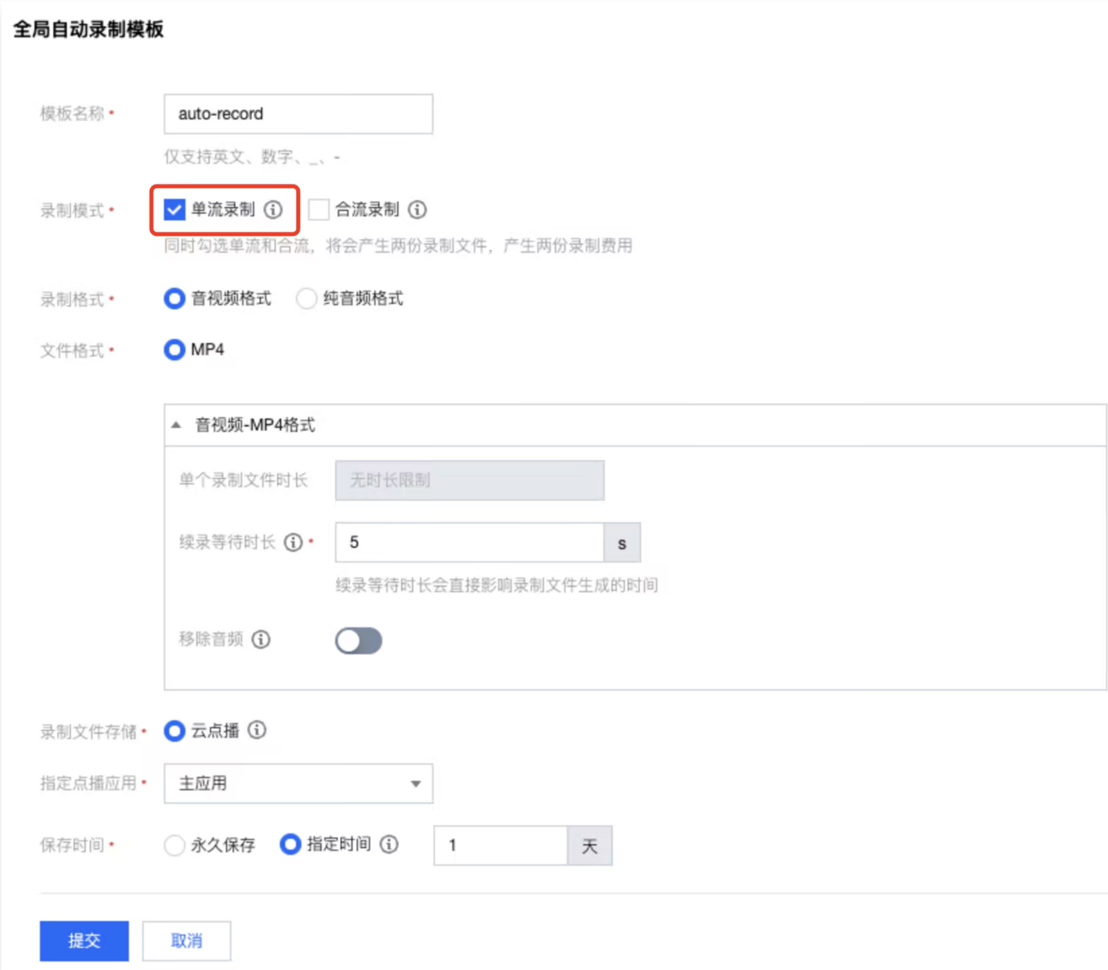
Real-time audio and video launched a brand-new cloud recording capability on August 1, 2022. After August 1, 2022, newly created applications will be upgraded to new version cloud recording capability by default. At the same time, in order to be compatible with old applications (sdkappid), legacy version cloud recording capability will be retained. An application (sdkappid) only supports one of the recording capabilities. When using the recording capability and consulting related recording capability issues, please first confirm the type of cloud recording capability of the current application. See Determine Cloud Recording Capability Type.

## What Are the Control Methods of the New Version of Cloud Recording Capacity?

TRTC provides two cloud recording solutions, namely API call recording and global automatic recording. These two solutions do not conflict and can be used simultaneously. Manual API Recording has the advantages of flexible recording and complete features compared to global automatic recording. Customers can specify the streamers to be recorded in the subscribed room, customize the mixed-stream layout, and update the layout and subscription during recording. The advantage of Global Automatic Recording is that recording does not require customers to start and stop. TRTC backend manages recording tasks. You can complete automatic recording of all audio and video streams in the room according to the preset recording template, reducing the access threshold.

## How to Record a Single Stream?

- **Under Manual API Recording:** You need to achieve manual single-stream recording by setting RecordMode to single-stream recording in RecordParams > RecordMode through the **Start Cloud Recording** API: CreateCloudRecording.
- **Under Global Automatic Recording:** You need to set the recording mode to **single stream recording** in the global automatic recording template configured in the console. For details, see global automatic recording.




## How to Record Stream Mixing (Merge)?

Under Manual API Recording: You need to achieve manual single-stream recording by setting RecordMode to merge recording in RecordParams -> RecordMode through the **Start Cloud Recording** API: CreateCloudRecording.

Under Global Automatic Recording: You need to set the recording mode to **Mixed Stream Recording** in the global automatic recording template configured in the console. For details, see Global Automatic Recording.

## How to Get Recording Callback?

1. Set callback address: For the new version of cloud recording callback address, please go to Recording Callback under **Callback Configuration** of the corresponding application in the console to set it. For details, see Callback Configuration.


2. Initiate a recording task, receive callback information. The new version of cloud recording provides the following cloud recording events. For details, see Cloud Recording Callback Description.

| Field Name | Type | Meaning |
|---|---|---|
| EVENT_TYPE_CLOUD_RECORDING_RECORDER_START | 301 | Cloud-based recording module startup |
| EVENT_TYPE_CLOUD_RECORDING_RECORDER_STOP | 302 | Exit the cloud-based recording module |
| EVENT_TYPE_CLOUD_RECORDING_FAILOVER | 306 | Triggered when cloud recording is migrated and the original recording task is migrated to a new load. |
| EVENT_TYPE_CLOUD_RECORDING_DOWNLOAD_IMAGE_ERROR | 309 | Error occurred when decoding the downloaded image file of cloud-based recording. |
| EVENT_TYPE_CLOUD_RECORDING_VOD_COMMIT | 311 | Cloud-based recording: Upload of media content for VOD recording task completed. |
| EVENT_TYPE_CLOUD_RECORDING_VOD_STOP | 312 | Cloud-based recording: End of VOD recording task. |

> **Note:**
> Cloud recording resource upload end event 311. Depending on the individual file size of the files recorded by your recording task, there may be a delay in the upload completion callback. The specific delay is affected by the file size.

## How to Get Recording Files?

The updated cloud recording feature will store the recorded files in the Video on Demand (VOD) you specify. You can find them in the following ways:

### Method 1: Manual Search in the VOD Console

1. Log in to VOD Console and select **Media Asset Management** in the left sidebar.
2. Click **Prefix Search** at the top of the list, select **Prefix Search**, enter keywords in the search box, fill in according to Recording File Naming Rules, for example, enter `1400000123_1001` under combined stream recording, click , and video files with matching prefixes of video names will be displayed.
    - **Single stream recording** MP4 filename rules: `<SdkAppId>_<RoomId>_UserId_s_<UserId>_UserId_e_<MediaId>_<Index>.mp4`
    - **Mixed-stream recording** MP4 filename rules: `<SdkAppId>_<RoomId>_<Index>.mp4`

### Method 2: Find Through Video-On-Demand REST API

Tencent Cloud VOD system provides a series of REST APIs to manage audio and video files on it. You can query files on the VOD system through the Search media information REST API. You can perform matching searches through the `NamePrefixes` parameter.
REST request example:

```
https://vod.tencentcloudapi.com/?Action=SearchMedia
&NamePrefixes.0=1400000123_1001xxxx
&Sort.Field=CreateTime
&Sort.Order=Desc
&<common request parameters>
```

# Old Version of Cloud Recording Issues

## How to Record a Single Stream?

- Configure **global automatic recording** in the console. When streaming, every single stream in the room will be automatically recorded. For details, see global automatic recording.

- Configure **specified user recording** in the console. For the stream to be recorded, set the userDefineRecordId parameter in TRTCParams when entering the room. For details, see specified user recording (SDK API).
- If you do not want global automatic recording and the platform does not support SDK API, you can use live streaming recording of Cloud Streaming Services for separate recording.

## How to Record Stream Mixing?

- If **global automatic recording** is configured, stream mixing will also be automatically recorded.
- If **specified user recording** is configured and server REST API is used to trigger stream mixing, specify the parameter OutputParams.RecordId in the stream mixing API to enable stream mixing recording. For details, see stream mixing parameters - OutputParams.
- If **specified user recording** is configured and stream mixing is triggered using the Client SDK API, stream mixing will be recorded when the anchor sets the userDefineRecordId parameter in TRTCParams upon entering the room. For details, see specified user recording (SDK API).

## When Does the Recording Start?

- Single stream: Start recording a few seconds (network latency and key frame waiting) after starting streaming.
- Stream mixing: Start recording a few seconds (network latency and key frame waiting) after starting stream mixing.

## When Does the Recording End?

- Separate stream recording ends after a stream stops. If a resumption timeout period is specified, the recording ends after the timeout period elapses.
- If stream mixing triggered by calling the client SDK setMixTranscodingConfig(), it ends when the anchor stream exits or when setMixTranscodingConfig() is called again and the parameter is set to null.
- If stream mixing triggered by calling the server REST API StartMCUMixTranscode, it stops automatically after all users check out, or stops manually by calling StopMCUMixTranscode midway.

## When Do Recording Files and Callbacks Occur?

- The recording file will be transferred to the VOD platform and trigger a callback 5 minutes after the recording is completed.
- If resumption time is set, the resumption timeout time needs to be added on top of the waiting time. If resumption time is set, recording will not end during stream interruption,

and no recording files or callbacks will be generated.
- For MP4, FLV and AAC types, there is a limit on the maximum duration of a single file (configurable in the console). After exceeding the maximum duration, a recording file and a callback will be generated, and then a new recording file will continue to be generated.

## How to Get Recording Files?

- You can manually search through the VOD console or search via the VOD REST API. For details, see Search for recording files.
- You can obtain the download URL of the recording file in a timely manner through callback. For details, see Receive recording files.

## How to Troubleshoot When No Recording Files Are Generated?

- The call is not established normally or the upstream streaming time is too short (it is recommended to stream for more than 30 seconds), and no recording file may be generated. You can check whether the upstream data is normal through the dashboard.
- If **specified user recording** is configured and userDefineRecordId in TRTCParams is not set when entering the room, the single stream will not be recorded.
- If **specified user recording** is configured and server REST API is used to trigger stream mixing, and the parameter OutputParams.RecordId is not specified in the stream mixing API, the stream mixing will not be recorded.
- If **specified user recording** is configured and stream mixing is triggered using the Client SDK API, stream mixing will not be recorded if the anchor does not set userDefineRecordId.

## How to Troubleshoot When No Recording Callback Is Received?

- First, check whether the recording file is generated through the console. If it is not generated, you can perform preliminary troubleshooting according to the previous method. For file search, see Find recording files.
- If the recording file is generated but no callback is received, first check whether the callback has been correctly configured. For callback configuration, see Receive recording files.
- If the callback has been correctly configured, you can further check whether the server can process the callback normally, for example, by using CURL to perform simulation testing to see if the callback request can be processed normally.

## Why Are There So Many Generated Recording Files?

- If **global automatic recording** is configured, every stream in the room will be automatically recorded.

- If resumption time is not configured, a new recording file will be generated after each stream interruption and re-streaming.
- For MP4, FLV and AAC types, there is a limit on the maximum duration of a single file. After exceeding the maximum duration, a recording file and a callback will be generated, and then a new recording file will continue to be generated.
- If stream mixing is triggered multiple times in a room, multiple stream mixing recording files may be generated.

# Audio and Video Quality Related

Last updated：2025-03-17 15:27:08

## Video Issues

### How to Remove the Black Border From TRTC Video Footage?

Set TRTCVideoFillMode_Fill (fill) to solve this problem. TRTC video rendering modes are divided into fill and adapt. Local rendering images can be set through setLocalViewFillMode(), and remote rendering images can be set through setRemoteViewFillMode:

- TRTCVideoFillMode_Fill: The image fills the screen, and the part of the video that exceeds the display window will be cropped, so the picture may not be complete.
- TRTCVideoFillMode_Fit: The long side of the image fills the screen, and the short side area will be filled with black, but the content of the picture is definitely complete.

### How to Troubleshoot Lag in TRTC?

You can view the call quality through the corresponding RoomID and UserID on the Dashboard page of the Tencent Real-Time Communication console:

- View the situation of sender and recipient users from the perspective of the recipient.
- Check whether the packet loss rate of the sender and recipient is high. If the packet loss rate is too high, it is generally caused by unstable network conditions, resulting in Lag.
- Check the frame rate and CPU occupancy rate. Low frame rate and high CPU utilization can both lead to Lag.

### How to Troubleshoot Poor Video Quality in TRTC, Such As Blurriness and Mosaic?

- Clarity is mainly related to bitrate. Check whether the SDK bitrate is configured relatively low. If the resolution is high but the bitrate is low, it is easy to produce a mosaic phenomenon.
- TRTC will dynamically adjust the bitrate and resolution through the cloud-based QOS flow control policy according to network conditions. When the network is poor, it is easy to reduce the bitrate, resulting in decreased clarity.
- Check whether the VideoCall mode or Live mode is used when entering the room. For call scenarios, the VideoCall mode focuses on low latency and smoothness. Therefore, in weak network conditions, it is more likely to sacrifice video quality to ensure smoothness. For scenarios where video quality is more important, it is recommended to use the Live mode.

## Why Are the Local Screen and Remote Screen in TRTC Reversed Left and Right?

The locally collected image is mirrored by default. On the App side, you can set it through the setLocalViewMirror API, which only changes the mirror mode of the local camera preview image; you can also set the mirror mode of the encoder output image through the setVideoEncoderMirror API, which does not change the local camera preview image but will change the image effect seen by the user on the other end (as well as the server-recorded image). On the Web side, you can modify the mirror parameter through the createStream API to set it.

## TRTC Setting Video Encoding Output Orientation Has No Effect?

You need to set setGSensorMode() to TRTCGSensorMode_Disable to disable the gravity sensor. Otherwise, after calling setVideoEncoderRotation, the picture seen by the remote user will not change.

## TRTC Normal Uplink Has Data, Why Does Bypass Streaming Fail and No Image Can Be Seen?

Please confirm whether automatic bypass push stream has been enabled in Application Management > **Feature Configuration**.

## Preview/Playback Image Appears Rotated?

- **Use TRTCSDK camera collection:**
  - It is recommended to update the SDK version to the latest version.
  - For special devices, you can use the local preview frame rotation API `setLocalViewRotation`, the remote video frame rotation API `setRemoteViewRotation`, and the encoder output frame rotation API `setVideoEncoderRotation` for adjustment. For detailed instructions on using these APIs, see video frame rotation.
- **Use custom video capture feature**
  - It is recommended to update the SDK version to the latest version.
  - Confirm whether the screen angle when collecting video is correct.
  - Fill in the video data for TRTCSDK and check whether the rotation angle is set for `TRTCCloudDef.TRTCVideoFrame`.
  - For special devices, you can use the local preview frame rotation API `setLocalViewRotation`, the remote video frame rotation API `setRemoteViewRotation`, and the encoder output frame rotation API `setVideoEncoderRotation` for adjustment. For detailed instructions on using these APIs, see video frame rotation.

## Video Has a Mirror Problem?

When using the front camera for a video call, there will be a mirror effect, so the local preview and the remote viewer's picture are reversed left to right. If you want the pictures on both ends to be consistent, please refer to When using the front camera for a video call, the local preview and the remote viewer's picture are reversed left to right.

## How to Stream in Landscape Mode?

Developers may use devices such as TVs or need to do horizontal streaming due to scene requirements. For specific implementation, please refer to Android TRTC Implements Landscape Video Call.

## What Causes a Black Screen During Live Stream Playback?

- Playback failure or decoding failed. See Solutions for playback failure.
- Metadata issues, such as only audio stream information in metadata but both audio and video in actual data; or only audio at the start of the data and then video information added after a period of playback. In such cases, it is generally recommended to modify the metadata information of the source stream.
- When there is no picture information in the video encoding data and only frames like SEI are present, there will be no picture during decoding, resulting in a black screen. This is generally custom video data.

## What Are the Reasons For Screen Glitches or Green Screens During Live Streams?

- Generally, it is caused by the loss of I-frame, because the decoding of P-frames and B-frames depends on I-frames. If the I-frame is lost, both P-frames and B-frames will fail to decode, resulting in screen glitches, ghosting, green screens, etc. First, use different players such as ffplay, VLC, and Potplayer to play the same stream simultaneously. If all players show screen glitches or green screens, usually there is a problem with the audio and video source stream, and the source stream needs to be checked.
- Metadata changes. Most players generally parse the metadata settings and decoding parameters only once before starting decoding. When the picture changes, for example, the resolution changes, but the player's decoding parameters are not reconfigured, it may cause screen glitches or green screen. In this case, the best way is for the streaming side not to change the encoding parameters during live streaming, so as not to cause modification of metadata information.
- Compatibility issues of hardware encoding and decoding. This kind of situation usually occurs on Android devices. The hardware encoders and decoders of some Android

devices are poorly implemented and have poor compatibility. In this case, the best way is to switch to software encoding and decoding for comparison.

- When the color formats of the streaming side and the playback side are inconsistent, for example, if the streaming side uses NV12 while the playback side supports I420, screen glitches or green screens may occur during decoding due to the color format inconsistency. In this case, unify the color formats of the streaming and playback ends.

# Audio Issues

## Why Is the Sound Very Low When Using the On-Demand Player TXVodPlayer During a TRTC Call?

Set the system volume type used during a call through the setSystemVolumeType API. Set it to the media volume mode TRTCSystemVolumeTypeMedia to solve the problem.

## How to Select Media Volume and Call Volume?

Through the setSystemVolumeType API, you can choose call volume and media volume independently.

- TRTCAudioVolumeTypeAuto: Default type, call volume when speaking, media volume when not speaking.
- TRTCAudioVolumeTypeVOIP: Always use call volume.
- TRTCAudioVolumeTypeMedia: Always use media volume.

## Handling No Sound in TRTC Upstream and Downstream

For details, please refer to Upstream and Downstream Silent Troubleshooting.

## Handling Low Volume

- If **all the audience hear low sound**, it is caused by uplink factors:
  - Check whether the volume value of the setCurrentDeviceVolume API for Windows and mac and the setAudioCaptureVolume API for all platforms is less than 50. You can appropriately increase the volume.
  - Check whether the AGC automatic gain of 3A processing is enabled.
  - Check whether it is caused by Bluetooth headphones.
- If **some of the audience hear low sound**, it is caused by downlink factors:
  - Check whether the volume value of the setAudioPlayoutVolume API and the setCurrentDeviceVolume API is less than 50. You can appropriately increase the volume.

- Mobile app can check whether the setAudioRoute API has been called to switch to receiver playback.

## Is There Sound Lag, Interruption or Intermittence?

Open Monitoring Dashboard and view in the Audio tab:
- If the CPU of the "device status" of the recipient and the sender exceeds 90%, it is recommended to close other background programs.
- If there is significant packet loss in audio upstream and downstream and the rtt value fluctuates greatly, it indicates that the current user's network quality is poor. It is recommended to switch to a stable network.

## Why Is There an Echo?

When the devices of both parties are too close to each other, it is normal. Please keep a greater distance from each other during testing; whether the AEC echo cancellation of 3A processing has been accidentally turned off.

## Poor Sound Quality or Fluctuating Volume?

If you have connected an external sound card and enabled in-ear monitoring, this issue may occur during mic connection. It is recommended to disable in-ear monitoring when using an external sound card, as the sound card generally has its own in-ear monitoring function.

## Echo, Noise, or Low Volume During Web Call?

When the devices of both parties are too close to each other, it is normal. Please keep A greater distance from each other during testing. When the other end hears echo, noise, or interference in the sound from the Web end, it indicates that the 3A processing on the Web end is not effective. If you use the browser native getUserMedia API for custom capture, you need to manually set the 3A parameters:
- echoCancellation: Echo cancellation switch.
- noiseSuppression: Noise suppression switch.
- autoGainControl: Automatic gain control switch. For detailed settings, see Media tracking constraints.

If you use the TRTC.createStream API for capture, there is no need to manually set the 3A parameters. The SDK enables 3A by default.

# Others

## How Does TRTC Monitor Network Status and Implement the Feature of Displaying Signal Strength?

You can use onNetworkQuality() to monitor the uplink and downstream quality of the current network. Taking Android as an example, see TRTC-API-Example to implement the signal strength feature.

## Why Does an Exception Such As the Device Camera or Microphone Being Occupied Occur?

Calling the exitRoom() API will execute the relevant logic for exiting the room, such as releasing audio and video device resources and codec resources. The release of hardware devices is an asynchronous operation. After the resources are released, the SDK will notify the upper layer through the onExitRoom() callback in TRTCCloudListener. If you want to call enterRoom() again or switch to other audio and video SDKs, wait for the onExitRoom() callback to arrive before performing relevant operations.

## How to Judge If the Camera Is Successfully Turned On?

Through the callback method onCameraDidReady, when this callback is received, it indicates that the camera is ready.

## How to Judge If the Microphone Is Successfully Turned On?

Through the callback method onMicDidReady, when this callback is received, it indicates that the microphone is ready.

## Failed to Open the Camera?

- Confirm whether the camera permission is granted
- If the device is a TV, box, etc., the camera used is external. Currently, TRTCSDK supports recognizing external cameras. Therefore, it is necessary to confirm whether the camera connector is in good contact with the device.

## What Are the Technical Statistics Indicators of TRTC?

> **Note:**
> This scenario is applicable to iOS/Mac, Android, and Windows platforms.

The SDK provides a callback method onStatistics(TRTCStatistics statics), which reports technical metrics every 2 seconds. This includes the current appCpu (CPU utilization of the App), systemCpu (CPU utilization of the current system), rtt (latency), upLoss (uplink packet loss rate), downLoss (downlink packet loss rate), and audio/video statistics for local and remote members. For specific parameters, see TRTCStatistics type description.

# Other Issues

Last updated：2025-03-17 15:27:16

## What Are the Differences and Relationships Between Live Stream, Interactive Live Streaming, Tencent Real-Time Communication (TRTC) and Bypass Live Streaming?

- **Live stream** (Keywords: one-to-many, RTMP/HLS/HTTP-FLV, CDN)
  Live streaming consists of the streaming side, playback side, and live streaming cloud service. The cloud service uses CDN to distribute the live stream. The streaming uses the general standard protocol RTMP. After distribution by CDN, playback can generally choose RTMP, HTTP-FLV, or HLS (supported by H5) methods for viewing.
- **Interactive live streaming** (Keywords: mic connection, PK)
  Interactive live streaming is a business form that refers to a type of live streaming where anchors interact with audiences through mic connections and anchors interact with each other through PK.
- **TRTC** (Keywords: multi-person interaction, UDP-based proprietary protocol, low-latency)
  TRTC (Tencent Real-Time Communication) is mainly used in audio and video interaction and low-latency live streaming. It uses a UDP-based proprietary protocol with latency as low as 100 ms. Typical scenarios include QQ calls, Tencent Meeting, and large classes. Tencent Cloud TRTC covers all platforms. Besides iOS/Android/Windows, it also supports mini-programs and WebRTC interconnection, and enables side-streaming live broadcasts to CDN through cloud-based mixed-streaming.
- **Bypass live streaming** (Keywords: cloud stream mixing, RTC bypass push, CDN)
  Bypass live streaming is a technology that refers to copying multiple push stream images from low-latency mic connection rooms, mixing the images into one in the cloud, and pushing the mixed stream to live streaming CDN for distribution and playback.

## Why Can'T Two Devices Running the Demo Simultaneously See Each Other'S Visual?

Ensure that two devices use different UserIDs when running the Demo. TRTC does not support the same UserID (unless the SDKAppID is different) being used on two devices simultaneously.

## Why Is the Visual Blurry and Choppy When There Is Only One Person in the Room While Watching Live Stream Via CDN?

Specify the TRTCAppScene parameter in `enterRoom` as **TRTCAppSceneLIVE**.
The VideoCall mode is optimized for video calls. Therefore, when there is only one user in the room, the picture will maintain a lower bitrate and frame rate to save the user's network traffic, which may appear choppy and blurry.

## Why Can'T I Enter Any Online Rooms?

This may be because room entry permission protection has been enabled. After room entry permission protection is enabled, rooms under the current SDKAppID will require setting privateMapKey in TRTCParamEnc to enter. If your online business is in operation and the online version does not include logic related to privateMapKey, do not enable this feature. For more details, see Room Entry Permission Protection.

## How to View TRTC Logs?

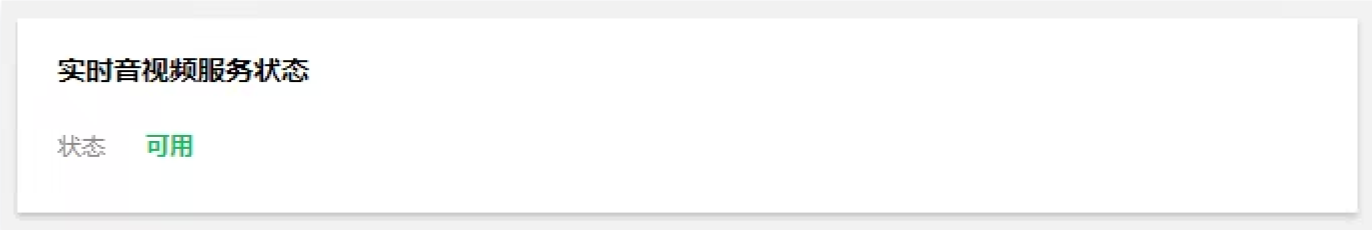
TRTC logs are compressed and encrypted by default, with the suffix .xlog. Whether logs are encrypted can be controlled through setLogCompressEnabled. Files containing C (compressed) in the generated file name are encrypted and compressed, while those containing R (raw) are plaintext.

- iOS & Mac: `sandbox's Documents/log`
- Android:
  - Version 6.7 and earlier: `/sdcard/log/tencent/liteav`
  - Version after 6.8: `/sdcard/Android/data/package name/files/log/tencent/liteav/`
  - Later than v8.5: `/sdcard/Android/data/package name/files/log/liteav/`
- Windows
  - For versions before 8.8: `%appdata%/tencent/liteav/log`
  - For versions 8.8 and later: `%appdata%/liteav/log`
- Web: Open the browser console or use vConsole to record SDK print information.
- Mini program: Enable the debug attribute of <live-pusher> and <live-player> tags, and use vConsole to record printed information.

> **Note:**
> - To view.xlog files, download decryption tool, place it in the same directory as the xlog file in the Python 2.7 environment, and run `python decode_mars_log_file.py`.
> - To view.clog files (new log format after version 9.6), download decryption tool, place it in the same directory as the clog file in the Python 2.7 environment, and run `python decompress_clog.py`.
> - For more log-related settings, refer to: Log Output Configuration.

## How to Handle the 10006 error?

If "Join room failed result: 10006 error: service is suspended, if charge is overdue, renew it" appears, please confirm whether the service status of your TRTC application is available. Log in to the **TRTC console** > Application Management, select the application you created, and click **Application Information**. You can confirm the service status in the application information panel.




## What Causes the Error Code -100018 When Entering a Room?

The reason is UserSig verification failure. Possible situations are as follows:

- Incorrect parameter SDKAppID input. You can log in to the TRTC console, select Application Management, and view the corresponding SDKAppID.
- The verification signature UserSig corresponding to parameter UserID is incorrect. You can log in to the TRTC console, select **Development Assistance** > UserSig Generation & Verification to verify UserSig.

## How to Cross-Room Mic Connection (Streamer PK)?

The connectOtherRoom API can be used. After the streamer calls connectOtherRoom(), the result of cross-room PK can be obtained through the onConnectOtherRoom callback. Everyone in the room where streamer one is located will be notified of streamer two's entry into the room through the onUserEnter callback. Everyone in the room where streamer two is located will also be notified of streamer one's entry into the room through the onUserEnter callback.

## Is It Necessary to Call the exitRoom() API?

Regardless of whether entering the room is successful or not, enterRoom must be used in conjunction with exitRoom. Calling the enterRoom function again before calling exitRoom may cause unexpected error issues.

## What Is the Format of Recording Files Generated in Various Scenarios of Bypass Recording?

TRTC Console shall prevail in configuring the recording file format.

## How to Judge Whether Audio and Video Call Streaming Is Successful?

Through the callback method onSendFirstLocalVideoFrame, start camera capturing after entering the room and starting local preview successfully, and encode the captured picture. When the SDK successfully sends the first frame of video data to the cloud, this callback event will be thrown.

## How to Judge Whether Audio-Only Call Streaming Is Successful?

Through the callback method onSendFirstLocalAudioFrame, start mic capturing after entering the room and starting local preview successfully, and encode the captured sound. When the SDK successfully sends the first frame of audio data to the cloud, this callback event will be thrown.

## Can I Query All UserIDs?

Currently, it does not support counting all UserIDs. You can write user information into SQL after successful client user account registration for management or querying.

## Can the Same UserID Enter Multiple Rooms Simultaneously?

TRTC does not support two identical userIDs entering the same room at the same time; otherwise, they will interfere with each other.

## Why Does Calling setAudioRoute to Set the Audio Route (Receiver/Speaker) Not Take Effect?

The earpiece/speaker can only be switched in Call Volume Mode, that is, it is effective only when being invoked during mic connections of two or more users.

## Does TRTC Only Support Automatic Recording Enabled in the Tencent Cloud Console? How to Manually Enable Recording?

TRTC supports manual recording. The specific operation methods are as follows:

1. Enter Application Management > **Feature Configuration**, enable **Automatic Bypass Streaming**, and do not enable **Start Cloud Recording**.
2. After the user enters the room, calculate the streamid corresponding to the userid according to the generation rule of stream ID.
3. Use the Create Recording Task API of Cloud Streaming Services (CSS) to start a recording task for the streamid.
    - DomainName is `[bizid].livepush.myqcloud.com`.
    - AppName is `trtc_[sdkappid]`.
    - `StreamName is streamid`.
4. After the recording task is completed, CSS will save the file in VOD and notify you via recording callback event.

## How to Verify the Correctness of the Generated UserSig in TRTC? How to Troubleshoot Room Entry Errors -3319 and -3320?

You can log in to the TRTC console and select **Development Assistance** > UserSig Generation & Verification to verify UserSig.

## How Does TRTC View Call Duration and Usage?

You can view it on the Usage Statistics page of the TRTC console.

## How Does TRTC Maintain a List of Users and Count the Number of Viewers in a Live Streaming Room?

If you have integrated IM into your project, you can use the IM group user counting API to calculate the number of users in a room. However, such calculation is not always accurate. You may use this method if you don't have a high requirement on accuracy.
If you do have a high requirement on the accuracy of the calculation, we recommend you implement the following calculation logic:

1. Increase the number of viewers (Client -> Server). When a new viewer joins, it means that the number of viewers in a room needs to be increased by 1. You can let the viewer end of the App send a request count increasing to the Server when entering the room.
2. Decrease the number of viewers (Client -> Server). When a viewer exits a room, it means the number of viewers in a room needs to be decreased by 1. The viewer end of the App can send a decreasing request count to the Server when exiting the room.

## Room Entry Reports Error Code -100013 With Error Information ERR_SERVER_INFO_SERVICE_SUSPENDED. What'S the Problem?

This error indicates that the service is unavailable. Check:

- Whether the number of remaining minutes in the package is greater than 0.
- Whether the Tencent Cloud account is in arrears.

## Troubleshooting When TRTC Starts Cloud Recording but No Recording File Is Generated

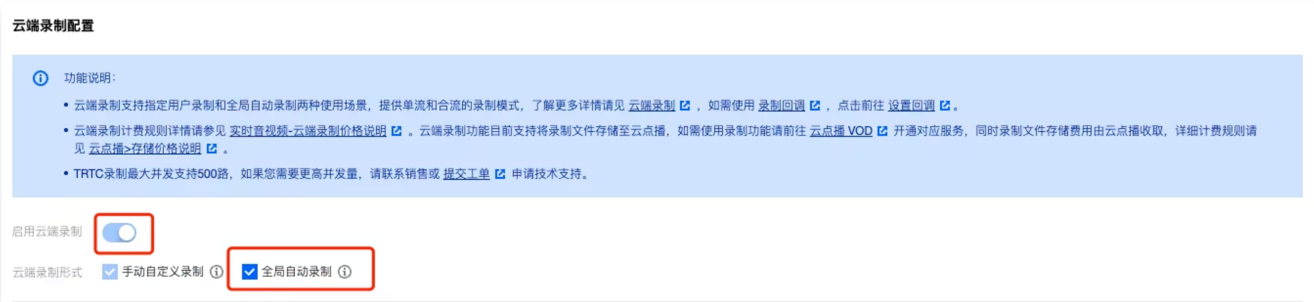
First, confirm the recording feature type of the current application (SDKAppId), How to distinguish cloud recording feature types.

**For troubleshooting the new version of cloud recording, refer to the following methods:**

1. Distinguish the initiation method of the recording feature, manual API recording or global auto-recording.
2. Manual API recording:

- Confirm whether the call to the cloud recording API CreateCloudRecording is correct.
- There are users' normal uplink audio and video data in the room (the uplink duration should exceed 30 s. If it is too short, the recording will fail).
- After the recording in a room ends, confirm whether the callbacks for recording completion and file upload are received. The recording file will be uploaded to VOD. You can go to VOD Console to check if the file exists.

3. Global automatic recording:
   - Confirm that the global auto-recording template is created in the console and the global recording switch is turned on.




   - There are users' normal uplink audio and video data in the room (the uplink duration should exceed 30 s. If it is too short, the recording will fail).
   - After the recording in a room ends, confirm whether the callbacks for recording completion and file upload are received. The recording file will be uploaded to VOD. You can go to VOD Console to check if the file exists.

**For troubleshooting the old version of cloud recording, refer to the following methods:**

1. Ensure that **Automatic Bypass Streaming** and **Start Cloud Recording** have been enabled in TRTC Console.
2. Recording starts only when there are users uploading audio and video data normally in the TRTC room.
3. Recording files are generated only when pulling streams through bypass CDN is normal.
4. If there is only audio at the beginning and the video is switched in the middle, depending on the recording template, only the recording files for the video time period or only the recording files for the audio time period may be generated.

## How to Inform the Invited Guest of the Room Number For Connecting?

The operation of informing guests of the room number can be added to the custom message. Parse the message content to obtain the roomid. For related instructions, see Create a Custom Message and TIMMsgSendNewMsg.

## Can Recording Start Only When At Least Two People Enter the Room?

Yes. If you need to obtain the audio data after cloud stream mixing, you can specify the output stream ID after starting cloud stream mixing and call the live streaming API Creating a Recording Task.

## How Does the Windows Client Collect the Sound Played By the Shared Application?

By calling the startSystemAudioLoopback API, system sound capture can be enabled.

## How to Implement the Feature of Streamers Initiating Audio and Video Connections With the Audience in Windows Meeting Mode?

It is necessary to pair with another Cloud Service Instant Messaging (IM) to meet the connecting line requirements.
The general logic of calling: A sends a custom message X to B and launches the call page. Handle the display effect of X by yourself. After receiving X, B launches the called page. B clicks enterRoom to enter the room and sends a custom message X1 to A. After receiving X1 (decide whether to display it by yourself), A also calls enterRoom to enter the room. Use IM to send custom messages.

## How Can the Audience View the Connected Screen in the Room?

When the audience uses live mode, they will be informed of the anchor's userid when entering the room to watch through the onUserVideoAvailable callback in TRTCCloudDelegate (the person doing mic connection will also enterRoom into the room, and for the audience, they are also considered as anchors). Then the audience can call the startRemoteView method to display the anchor's video footage.

## Does TRTC Support Screen Sharing During Video Calls or Mic-Connection Interactions?

Supported. During TRTC mic-connection interaction or video call, the camera-captured picture serves as the mainstream picture, and screen sharing is also supported as the auxiliary stream picture. The shared screen is the current mobile phone screen picture, including the window of mic-connection interaction or video call.

## Help Instructions For Migrating Interactive Live Streaming (iLVB) to Real-Time Audio and Video (TRTC)

Tencent Cloud **Interactive Live Streaming iLVB** product was discontinued in October 2019 and can no longer meet new business needs. To better provide you with audio and video related services, it is recommended that you migrate services to **TRTC Practices** product in a timely manner. Audio and video interaction related needs are provided with subsequent service

support by **TRTC Practices** product. In addition to fully compatible interactive live streaming basic audio and video communication functions, **TRTC Practices** also provides rich terminal capabilities and cloud characteristics, including but not limited to AI noise reduction, cloud recording, content recognition and review, Mini Program Acceleration Service, RTMP streaming into room, etc.
You can quickly create and integrate services through the TRTC console. The existing Instant Messaging (IM) service of your application will not be affected. (Alternatively, you can also replace the IM service with a new SDKAppID.) TRTC provides two solutions: Integration (UI Included) and Integration Solution (No UI). You can choose and access either of them according to your development capability and business requirement.

# Dealing With Firewall Limitations

Last updated：2025-03-17 15:27:24

## What Ports or Domain Names Does the Client Native SDK Need to Configure As an Allowlist?

Firewall ports are shown in the table below:

| TRTC SDK (Native) | Allowlist Items |
|---|---|
| TCP port | 443, 20166 |
| UDP port | 8000, 8080, 8001, 8002, 8003, 8004, 8005, 8006, 8007, 8008, 8009, 16285, 9000 |

Add the following domain names to the allowlist:

```
cloud.tim.qq.com
gz.file.myqcloud.com
avc.qcloud.com
yun.tim.qq.com
dldir1.qq.com
mlvbdc.live.qcloud.com
query.tencent-cloud.com
*.trtc.tencent-cloud.com
```

## What Ports or Domain Names Does WebRTC Need to Configure As an Allowlist?

Firewall ports are shown in the table below:

| WebRTC (H5) | Allowlist Items |
|---|---|
| TCP port | 8687 |
| UDP port | 8000; 8080; 8800; 843; 443; 16285 |

Domain name allowlist

```
*.rtc.qcloud.com
```

```
*.rtc.qq.com
yun.tim.qq.com
```

## How to Set Up a Proxy in the TRTC web End Private Network Environment?

You can use the Nginx+coturn proxy solution. For details, see Enterprise Private Network Proxy Solution.

| Solution Name | Applicable Scenario | Network Requirements |
| --- | --- | --- |
| Option I | Allow clients to access specific public network proxy servers | Allow clients to access public network proxy server |
| Option II | Allow clients to access the public network through a private network proxy server. | Allow proxy server to access public network |

## What Domain Names Does WeChat Mini Program Need to Configure As an Allowlist?

<trtc-room> Domain name allowlist:

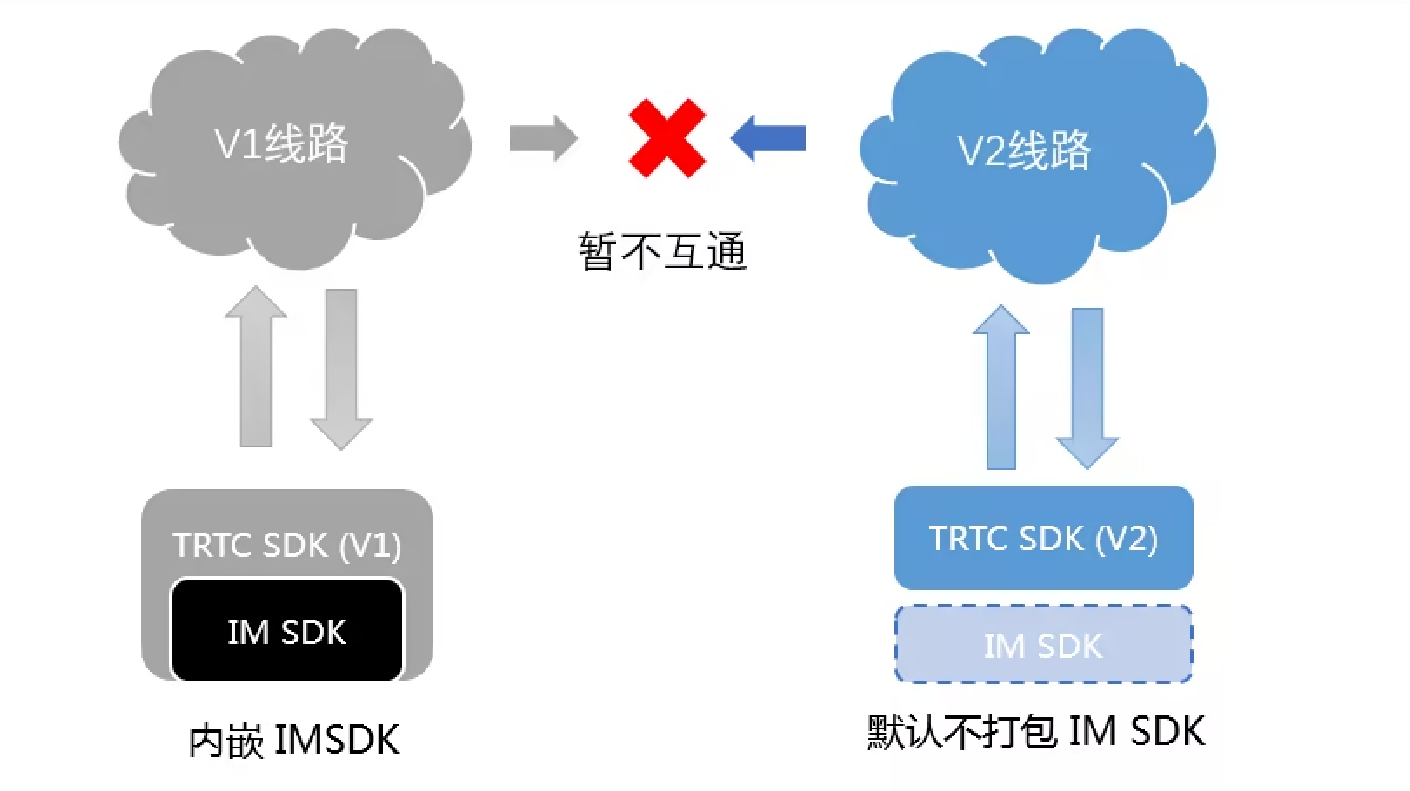
```
https://official.opensso.tencent-cloud.com
https://yun.tim.qq.com
https://cloud.tencent.com
https://webim.tim.qq.com
https://query.tencent-cloud.com
```

**Note:**
Since the IP addresses of Tencent Cloud server-side are dynamically updated and not a fixed set of IP addresses, we are unable to provide you with a fixed list of IP addresses.

# iLiveSDK Migration Related

Last updated：2025-03-17 15:27:31

## Difference Between TRTC V1 (iLiveSDK) and V2 (LiteAVSDK) Versions?




| Difference Item | Old Version V1 | New Version V2 |
| --- | --- | --- |
| Kernel Architecture | iLiveSDK | LiteAVSDK |
| IM SDK | Embedded | Not embedded |
| API Interface | V1 | V2 |
| Streaming on CDN | Use the REST API to enable | Support turning on the client |
| Cloud-based Circuit | V1 Circuit | V2 Circuit |

## How to Upgrade TRTC V1 (iLiveSDK) to V2 (LiteAVSDK)?

- If your **project has never integrated TRTC SDK**, it is highly recommended to directly use V2 (LiteAVSDK), which has advantages in call quality, circuit specifications, access difficulty, and feature expansion.

- If your **project is stable and has no issues**, since the cloud routes of V1 and V2 are currently not interconnected, if your project has entered the stable operation stage, you can temporarily not upgrade.
- If your **project is integrating with Old Version V1**, it is recommended that you can directly integrate with Version V2. The API of Version V2 is newly designed, and the integration time is much shorter compared to the old version.
- If you **are already using Old Version V1 and want to improve call quality**, since the cloud routes of V1 and V2 are currently not interconnected, upgrading to the new version SDK requires going through a process of "SDK integration", "rolling out in phases", and "cloud switching". The general steps are as follows:

1. Integrate the new version of the SDK into the existing project and pass the test.
2. Add the SDK version number field to the room list. The App decides whether to use V1 or V2 based on the server field.
3. Publish a new version of the App and wait for the version to gradually cover your user group.
4. Switch the SDK version number field in the room list from V1 to V2 to complete the circuit switching.

## How to Achieve Seamless Integration of LiteAVSDK and iLiveSDK On the Android Side?

Both iLiveSDK and LiteAVSDK use TRAE for audio processing such as echo cancellation and noise reduction. The version of TRAE used in LiteAVSDK is newer and includes all the functional interfaces used in iLiveSDK. Therefore, you only need to configure your project to use the TRAE library in LiteAVSDK.
Integrate the project using the aar method. Modify the build.gradle file in your sub-project (app directory) and make the following configuration in the android{} node:

> **Note:**
> When adding a reference, LiteAVSDK must be placed before iLiveSDK.

```
android{
//1. Configure packagingoptions in gradle
packagingOptions {
pickFirst 'lib/armeabi-v7a/libTRAECodec.so'
pickFirst 'lib/armeabi-v7a/libstlport_shared.so'
pickFirst 'lib/armeabi/libTRAECodec.so'
```

```
pickFirst 'lib/armeabi/libstlport_shared.so'
}
//2. Import dependencies
implementation(name: 'LiteAVSDK_TRTC_6.4.7108', ext: 'aar') // Note that
TRTC must be placed before iLiveSDK
implementation 'com.tencent.ilivesdk:ilivesdk:1.9.4.6.4'
}
```

## How to Achieve Seamless Integration of LiteAVSDK + iLiveSDK + BeautySDK On the iOS Side?

In TRTC V1, BeautySDK was used to implement features such as beauty effect and motion effect. In TRTC V2, the features of BeautySDK have been embedded into LiteAVSDK, making it more convenient for users. If you have already integrated iLiveSDK and introduced BeautySDK into your project, you will encounter a file conflict. The solutions are as follows:

| Version | Solution |
| --- | --- |
| Basic version of BeautySDK (without photo editing) | You only need to configure the header search paths of BeautySDK in the Xcode project and unlink BeautySDK. |
| BeautySDK advanced edition (with photo editing version) | You need to use the full – feature version of LiteAVSDK and configure the header search paths of BeautySDK in the Xcode project and unlink BeautySDK (The full – feature version of LiteAVSDK has a photo editing component. You can directly use the photo editing license you purchased before and do not need to pay again). |

## How to Achieve Seamless Integration of LiteAVSDK and iLiveSDK On the Windows Client Simultaneously?

LiteAVSDK and iLiveSDK on the Windows client both use TRAE for audio processing such as echo cancellation and noise reduction. However, the TRAE version used by LiteAVSDK is newer and there are differences in feature usage. Therefore, they cannot be directly replaced. You can handle it in the following ways.

### Engineering Structure

It is advisable for your project to adopt the following structure:

```
|
Main program.exe
Other files that Main program.exe depends on
|- iLiveSDK.dll
Other files that iLiveSDK.dll depends on
|- LiteAV
|        |- liteav.dll
Other files that liteav.dll depends on
```

## Initialization Method

When in use, iLiveSDK can be directly linked with.lib or dynamically loaded with the following code:

```
HMODULE hiLive = LoadLibrary("iLiveSDK.dll");
```

When you need to use LiteAVSDK, load and initialize it with the following code:

```
typedef ITRTCCloud* (*getTRTCShareInstanceMtd)();
typedef void(*destroyTRTCShareInstanceMtd)();

TCHAR dllPath[MAX_PATH];
GetModuleFileName(nullptr, dllPath, MAX_PATH);
PathRemoveFileSpec(dllPath);
wcscat(dllPath, L"\\LiteAV\\");
SetDllDirectory(dllPath);
HMODULE hLiteAV = LoadLibrary(L"liteav.dll");
if (!hLiteAV) {
printf("Failed to load liteav.dll: %d", GetLastError());
return;
}

getTRTCShareInstanceMtd pGetTRTCShareInstance =
(getTRTCShareInstanceMtd)GetProcAddress(hLiteAV,
"getTRTCShareInstance");
if (!pGetTRTCShareInstance) {
printf("Failed to load function getTRTCShareInstance");
return;
}
```

```
destroyTRTCShareInstanceMtd pDestroyTRTCShareInstance =
(destroyTRTCShareInstanceMtd)GetProcAddress(hLiteAV,
"destroyTRTCShareInstance");
if (!pDestroyTRTCShareInstance) {
printf("Failed to load function destroyTRTCShareInstance");
return;
}

ITRTCCloud *pTrtcCloud = m_pGetTRTCShareInstance();
if (!pTrtcCloud) {
printf("Failed to create TRTC instance");
return;
}
SetDllDirectory(nullptr);

pTrtcCloud->enterRoom(...);
```

# Enabling Various Cloud Services

Last updated：2025-03-17 15:27:39

## Cloud Streaming Services

### How to Activate Cloud Live Streaming Service?

Enter CSS Management Console, go to the Tencent Cloud Live Service activation page, review the relevant agreements and check **agree**, then click **Apply for Activation** to activate the CSS service.

.

### How to Enable Stream Hotlink Protection KEY?

The push Hotlink protection KEY is a security protection measure to ensure that only your App users can push streams. It can be modified at any time according to your needs in the Domain Name Management of the live streaming management console. For details, see Playback Authentication Configuration.

### How to Access Authentication KEY Via API?

The API access authentication KEY is required when your backend server calls cloud live streaming-related TencentCloud API. Its purpose is to help Tencent Cloud confirm the legitimacy of the call. For information on obtaining the API access authentication KEY, see API Authorization.

## How to Call Back Event Notification URL?

Tencent Cloud can configure callbacks for some live streaming events. Tencent Cloud services will send notifications to the address you configured in the form of HTTP POST. For modifying the event callback URL, see Callback Configuration.

# Video On Demand

## How to Activate the VOD Service?

Just activate the service in VOD Management Console to use the VOD service. By default, it is billed on a daily basis. You can also purchase suitable resource packages for deductions.

## How to Query My VOD APPID?

Each Tencent Cloud account has a unique APPID corresponding to it, which is located in Account Information.


# Chat (IM)

## How to Activate the IM Service?

Enter Instant Messaging (IM) Management Console.
For a newly authenticated Tencent Cloud account, the application list for Instant Messaging (IM) is empty. Click **Add New Application**, fill in the relevant information, and then click **Yes** to

create a new application:


## What Is the SDK APPID?

The numbers shown in the figure below are the SDKAPPID, which represents a product under your Tencent Cloud account. If you have multiple products, there will be multiple corresponding SDKAPPIDs.


## What Is an Administrator?

IM provides a set of REST APIs that allow your backend server to directly call IM services, such as group creation, sending system messages, removing a user from a group, etc. However, the IM REST API can only be called by administrators, that is, an administrator username (Administrator) and corresponding password (UserSig) are required. For specific usage, see REST API Introduction.


**Note:**
UserSig is the password for users to log in to Chat IM. For specific methods to obtain the key, see Getting Key.

# Cloud Object Storage Service (COS)

## How to Activate the COS Service?

Newly authenticated Tencent Cloud accounts can immediately use COS. Simply enter COS Console, click **Bucket List** > **Creating a Bucket**, fill in the relevant information, and click **Yes** to successfully create a bucket and start using it.


## What Is a Bucket?

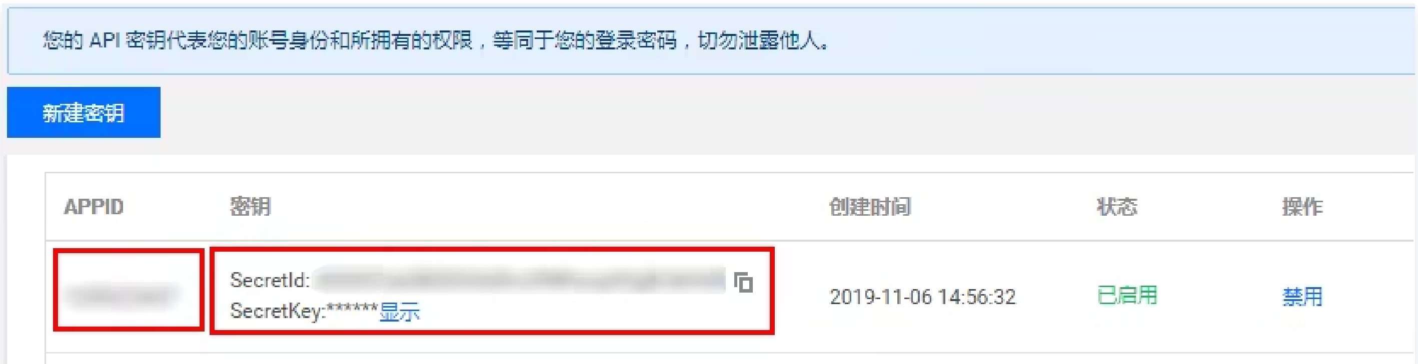
Bucket is a rather technical term, which means "bucket" in Chinese. You can simply understand it as the concept of **Disk Partition**. For example, if you have purchased COS (Cloud Object Storage) service on Tencent Cloud and bought a new hard disk in a mall, before storing data on it, you generally partition and format it first. So we can say that creating a partition on your real-world hard disk is like creating a Bucket in COS (Cloud Object Storage) on Tencent Cloud.

## How to Query My Bucketname?

The name you specify when creating a Bucket is your Bucketname. For example, the `xiaozhibo` specified in example 4.1 is a Bucketname.

## How to Query My COS APPID, SecretId and SecretKey?

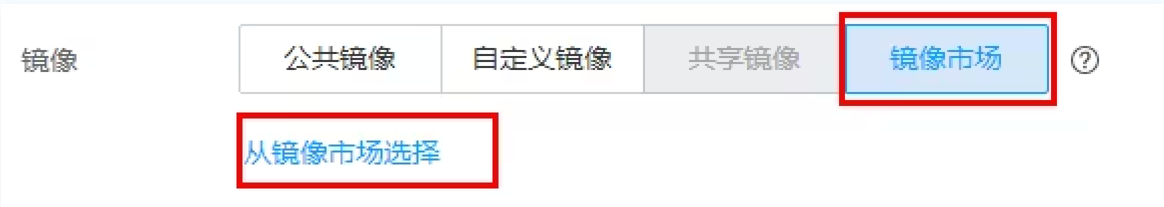
Click the Key Management page on the COS management console and click **Cloud API Key** to obtain relevant information.
COS APPID is bound to SecretId and SecretKey. They are mainly used to access COS APIs. Since COS is a cloud service with high security requirements, if the API does not pass in the correct key, Tencent Cloud will reject these API requests.




# Cloud Virtual Machine (Optional)

You can use your own server as a business server to deploy backend scripts. However, it is recommended that you use Tencent Cloud's CVM to deploy backend scripts, which is more professional and stable. In addition, if you choose Tencent Cloud's cloud database as a distributed database, it must be accessed in conjunction with Tencent Cloud's CVM.
Enter CVM Management Console, select **Instance** in the left sidebar, and click **Creation** to

enter the server purchase page:


> **Note:**
> When selecting a server image, it is recommended that you choose a Linux image with Nginx + PHP + MYSQL from the server market:
>
> 
> 

Follow the guideline for subsequent operations. After the mirror installation is complete, the CVM can be used.

## Cloud Database (Optional)

### How to Enable Cloud Database?

Please refer to Purchase Guide.

### How to Use a Database?

Please refer to:

- Create a MySQL instance
- Initialize a MySQL instance
- Connect to a MySQL instance