Tencent Real-Time Communication

FAQs

Product Documentation

Copyright Notice

Trademark Notice

Tencent Cloud

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

FAQs

Migration Guide

Twilio Video to TRTC

iOS

Last updated：2024-01-18 12:04:21

Overview

Welcome to our comprehensive guide designed to assist developers in migrating from Twilio Video to Tencent RTC (TRTC) on the iOS platform. This document provides a detailed comparison at the API level between Twilio and TRTC, and outlines each step required to implement basic RTC video functionality.

For those planning to develop a new application, you can directly visit the Tencent RTC website and documentation for all the necessary resources and information.

Create a TRTC Application

Create a new application of Tencent RTC (TRTC), then you can get the following necessary information for the authentication of TRTC service.

SDKAppID

SDKAppID is used to uniquely identify a TRTC application. It is generated automatically when an application is created.

SDKSecretKey

SDKSecretKey is one of the key parameters to be used to generate the security signature. The generated signature ensures the access of TRTC service when calling the initialization and login API of the SDK.

Install and Set Up the SDK

Environment Requirements

Xcode 9.0 or later

iPhone or iPad with iOS 9.0 or later

A valid developer signature for your project

Install the SDK

You can either use CocoaPods or manually download and import the SDK into your project. For using CocoaPods, make sure the CocoaPods is installed and the Podfile is created by the following commands:

```
sudo gem install cocoapods
pod init
```

Edit the created Podfile and fill a TRTC version according to your project needs:

```
pod 'TXLiteAVSDK_TRTC', 'version'
```

Then update the local repository and install the SDK by the following commands:

```
pod install
pod update
```

Project Configuration

To use the audio/video features of the SDK, you need to grant the application mic and camera permissions. Add the two items(Privacy - Microphone Usage Description and Privacy - Camera Usage Description) to the Info.plist of your Xcode application.

Import the SDK

After completing the above steps, you can import the APIs provided by the TRTC SDK into your project by the following ways based on your needs.

1. Using Objective-C or Swift APIs

```
// Import the SDK module
@import TXLiteAVSDK_TRTC;
```

```
// Import the header file
#import "TXLiteAVSDK_TRTC/TRTCCloud.h"
```

2. Using C++ APIs

```
#include "TXLiteAVSDK_TRTC/cpp_interface/ITRTCCloud.h"
```

Join a Room

Twilio

```
@IBAction func createARoom(sender: AnyObject) {
    let connectOptions = ConnectOptions(token: accessToken) { (builder) in
        builder.roomName = "my-room"
    }
    room = TwilioVideoSDK.connect(options: connectOptions, delegate: self)
}

// MARK: RoomDelegate

func roomDidConnect(room: Room) {
    print("Did connect to Room")

    if let localParticipant = room.localParticipant {
        print("Local identity \\(localParticipant.identity)")

        // Set the delegate of the local particiant to receive callbacks
        localParticipant.delegate = self
    }
}
```

TRTC

```
// Create trtc instance(singleton)  and set up event listeners
self.trtcCloud = [TRTCCloud sharedInstance];
self.trtcCloud.delegate = self;

// Package the room entry parameter
// Please replace each field in TRTCParams with your own parameters
TRTCParams *params = [[TRTCParams alloc] init];
params.sdkAppId = 1400000123;  // Please replace with your own SDKAppID
params.roomId = 123321; // Please replace with your own room number
params.userId = @"denny";   // Please replace with your own userid
params.userSig = @"xxx"; // Please replace with your own userSig
params.role = TRTCRoleAnchor;

// If your application scenario is a video call between several people, please use
[self.trtcCloud enterRoom:params appScene:TRTCAppSceneLIVE];

// Listen to the onEnterRoom event of the SDK and get notification whether the room
- (void)onEnterRoom:(NSInteger)result {
    if (result > 0) {
        [self toastTip:@"Enter room succeed!"];
    } else {
        [self toastTip:@"Enter room failed!"];
    }
}
```

Publish Local Video/Audio

Twilio

```
T// start camera
var cameraSource = CameraSource(delegate: self),
let captureDevice = CameraSource.captureDevice(position: .front)

cameraSource.startCapture(device: captureDevice, completion: nil)

// Use CameraSource to produce video from the device's front camera.

if let camera = TVICameraCapturer(source: .frontCamera),
    let videoTrack = TVILocalVideoTrack(capturer: camera) {

    // TVIVideoView is a TVIVideoRenderer and can be added to any TVIVideoTrack.
    let renderer = TVIVideoView(frame: view.bounds)

    // Add renderer to the video track
    videoTrack.addRenderer(renderer)

    self.localVideoTrack = videoTrack
    self.camera = camera
    self.view.addSubview(renderer)
}
```

TRTC

```
self.trtcCloud = [TRTCCloud sharedInstance];
// Set the preview mode of the local video image: Enable horizontal mirroring and s
TRTCRenderParams *param = [[TRTCRenderParams alloc] init];
param.fillMode    = TRTCVideoFillMode_Fill;
param.mirrorType = TRTCVideoMirrorTypeAuto;
[self.trtcCloud setLocalRenderParams:param];

// Enable local camera preview(`localCameraVideoView` is used to render the local v
[self.trtcCloud startLocalPreview:YES view:localCameraVideoView];

// Enable mic and set `quality` to `SPEECH` for the voice mode
[self.trtcCloud startLocalAudio:TRTCAudioQualitySpeech];

[self.trtcCloud startLocalAudio:TRTCAudioQualitySpeech];
```

Subscribe Remote Video/Audio

Twilio

```
// MARK: RemoteParticipantDelegate

/*
 * In the Participant Delegate, we can respond when the Participant adds a Video
 * Track by rendering it on screen.
 */
func didSubscribeToVideoTrack(videoTrack: RemoteVideoTrack,
                              publication: RemoteVideoTrackPublication,
                              participant: RemoteParticipant) {

    if let remoteView = VideoView.init(frame: self.view.bounds,
                                       delegate:self) {

        videoTrack.addRenderer(remoteView)
        self.view.addSubview(remoteView)
        self.remoteView = remoteView
    }
}

// MARK: VideoViewDelegate

// Lastly, we can subscribe to important events on the VideoView
func videoViewDimensionsDidChange(view: VideoView, dimensions: CMVideoDimensions) {
    self.view.setNeedsLayout()
}
```

TRTC

```
self.trtcCloud = [TRTCCloud sharedInstance];
// Play back the camera (primary stream) image of `denny`
[self.trtcCloud startRemoteView:@"denny" streamType:TRTCVideoStreamTypeBig view:cam
```

Leave a Room

Twilio

```
//TodisconnectfromaRoom,wecall:
room?.disconnect()

//ThisresultsinacallbacktoRoomDelegate#roomDidDisconnect(room:Room,error:Error?)

//MARK:RoomDelegate

funcroomDidDisconnect(room:Room,error:Error?){
print("Disconnectedfromroom\\(room.name)")
}
```

TRTC

```
self.trtcCloud = [TRTCCloud sharedInstance];
// Exit the current room
[self.trtcCloud exitRoom];

// Listen the `onExitRoom` callback to get notification
- (void)onExitRoom:(NSInteger)reason {
    if (reason == 0) {
        NSLog(@"Exit current room by calling the 'exitRoom' api of sdk ...");
    } else if (reason == 1) {
        NSLog(@"Kicked out of the current room by server through the restful api...
    } else if (reason == 2) {
        NSLog(@"Current room is dissolved by server through the restful api...");
    }
}
```

Conclusion

Compared to Twilio Video, this migration guide outlines how to use Tencent RTC (TRTC) to build a basic video RTC experience for the iOS platform. The API level "mapping" for each step in this guide will facilitate the developers to switch from Twilio Video to TRTC in a quick and straightforward way.

TRTC has more feature sets that can help developers implement cost-effective, low-latency and high-quality interactive audio/video services. For comprehensive details on TRTC features and the implementation specification, please refer to the Tencent RTC website. If you require developer support or any further assistance regarding the TRTC integration, feel free to contact us, or you can join our Discord and Telegram. We will ensure a smooth integration and address any queries you may have.

Android

Last updated：2024-01-18 11:58:22

Overview

Welcome to our comprehensive guide designed to assist developers in migrating from Twilio Video to Tencent RTC (TRTC) on the Android platform. This document provides a detailed comparison at the API level between Twilio and TRTC, and outlines each step required to implement basic RTC video functionality.

For those planning to develop a new application, you can directly visit the Tencent RTC website and documentation for all the necessary resources and information.

Create a TRTC Application

Create a new application of Tencent RTC (TRTC), then you can get the following necessary information for the authentication of TRTC service.

SDKAppID

SDKAppID is used to uniquely identify a TRTC application. It is generated automatically when an application is created.

SDKSecretKey

SDKSecretKey is one of the key parameter to be used to generate the security signature. The generated signature ensures the access of TRTC service when calling the initialization and login API of the SDK.

Install and Set Up the SDK

Environment Requirements

Android Studio 3.5 or later.

Android 4.1 (SDK API level 16) or later

Install the SDK

TRTC SDK has been released to the mavenCentral repository, and you can configure Gradle to download the SDK by the following steps,

1. Add the TRTC SDK dependency to dependencies.

```
dependencies {
    implementation 'com.tencent.liteav:LiteAVSDK_TRTC:latest.release'
}
```

2. In defaultConfig, specify the CPU architecture to be used by your application.

```
defaultConfig {
    ndk {
        abiFilters "armeabi-v7a", "arm64-v8a"
    }
}
```

3. Click sync to download the SDK and integrate them into your project.

Project Configuration

Configure application permissions in AndroidManifest.xml. The TRTC SDK requires the following permissions.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera.autofocus" />
```

Set Obfuscation Rules

In the file proguard-rules.pro, add the classes related to the TRTC SDK to the "do not obfuscate" list.

```
-keep class com.tencent.** { *;}
```

Join a Room

Twilio

```
// Join room
val connectOptions = ConnectOptions.Builder(accessToken)
    .roomName(roomName)
    .audioTracks(localAudioTracks)
    .videoTracks(localVideoTracks)
    .dataTracks(localDataTracks)
    .build()
val room = Video.connect(context, connectOptions, roomListener)
```

TRTC

```
//Initialize TRTC
// Create trtc instance(singleton)  and set up event listeners
val mCloud = TRTCCloud.sharedInstance(getApplicationContext());
mCloud.setListener(this);

// Join room
// Please replace each field in TRTCParams with your own parameters
val params = TRTCParams()
params.sdkAppId = 1400000123 // Please replace with your own SDKAppID
params.userId = "denny" // Please replace with your own userid
params.roomId = 123321 // Please replace with your own room number
params.userSig = "xxx" // Please replace with your own userSig
params.role = TRTCCloudDef.TRTCRoleAnchor

// If your application scenario is a video call between several people, please use
mCloud.enterRoom(params, TRTCCloudDef.TRTC_APP_SCENE_LIVE)

// Listen for the `onEnterRoom` event of the SDK to get the room entry result
// Listen for the `onEnterRoom` event of the SDK and learn whether the room is succ
@Override
public void onEnterRoom(long     result) {
    if (result > 0) {
        Log.d(TAG, "Enter room succeed");
    } else {
        Log.d(TAG, "Enter room failed");
    }
}
```

Publish Local Video/Audio

Twilio

```
// Start video
val cameraEnumerator = Camera2Enumerator(context)
cameraEnumerator.deviceNames.firstOrNull { cameraEnumerator.isFrontFacing(it) }?.le
    val cameraCapturer = Camera2Capturer(context, it)
    val localVideoTrack = LocalVideoTrack.create(context, true, cameraCapturer)
    cameraCapturer.startCapture(width, height, framerate)
}
```

```
// Stop video
cameraCapturer.stopCapture()
```

TRTC

```
// Set the preview mode of the local video image: Enable horizontal mirroring and s
TRTCCloudDef.TRTCRenderParams param = new TRTCCloudDef.TRTCRenderParams();
param.fillMode   = TRTCCloudDef.TRTC_VIDEO_RENDER_MODE_FILL;
param.mirrorType = TRTCCloudDef.TRTC_VIDEO_MIRROR_TYPE_AUTO;
mCloud.setLocalRenderParams(param);


// Enable local camera preview (`localCameraVideo` is the control used to render th
TXCloudVideoView cameraVideo = findViewById(R.id.txcvv_main_local);
mCloud.startLocalPreview(true, cameraVideo);

// Use `TXDeviceManager` to enable autofocus and turn on the flash
TXDeviceManager manager = mCloud.getDeviceManager();
if (manager.isAutoFocusEnabled()) {
    manager.enableCameraAutoFocus(true);
}
manager.enableCameraTorch(true);

// Enable mic and set `quality` to `SPEECH` for the voice mode
mCloud.startLocalAudio(TRTCCloudDef.TRTC_AUDIO_QUALITY_SPEECH );
```

Subscribe Remote Video/Audio

Twilio

```
 private lateinit var remoteVideoView: VideoView
    private fun setRoomListener() {
        remoteVideoView = findViewById(R.id.remote_video_view)
        val roomListener = object : Room.Listener {
            override fun onConnected(room: Room) {
                room.remoteParticipants.forEach { remoteParticipant ->
                    addRemoteParticipant(remoteParticipant)
                }
            }

            override fun onParticipantConnected(room: Room, remoteParticipant: Remo
                addRemoteParticipant(remoteParticipant)
            }
        }
    }
    private fun addRemoteParticipant(remoteParticipant: RemoteParticipant) {
        remoteParticipant.setListener(object : RemoteParticipant.Listener {
            override fun onVideoTrackSubscribed(
                remoteParticipant: RemoteParticipant,
                remoteVideoTrackPublication: RemoteVideoTrackPublication,
                remoteVideoTrack: RemoteVideoTrack
            ) {
                remoteVideoTrack.addSink(remoteVideoView)
            }

            override fun onVideoTrackUnsubscribed(
                remoteParticipant: RemoteParticipant,
                remoteVideoTrackPublication: RemoteVideoTrackPublication,
                remoteVideoTrack: RemoteVideoTrack
            ) {
                remoteVideoTrack.removeSink(remoteVideoView)
            }

        })
    }
```

TRTC

```
// Play back the camera (primary stream) image of `denny`
mCloud.startRemoteView("denny", TRTCCloudDef.TRTC_VIDEO_STREAM_TYPE_BIG, cameraView
```

Leave a Room

Twilio

```
room.disconnect()
```

TRTC

```
// Exit the current room
mCloud.exitRoom();

// Listen for the `onExitRoom` callback to get the reason for room exit
@Override
public void onExitRoom(int reason) {
    if (reason == 0) {
        Log.d(TAG, "Exit current room by calling the 'exitRoom' api of sdk ...");
    } else if (reason == 1) {
        Log.d(TAG, "Kicked out of the current room by server through the restful ap
    } else if (reason == 2) {
        Log.d(TAG, "Current room is dissolved by server through the restful api..."
    }
}
```

Conclusion

Compared to Twilio Video, this migration guide outlines how to use Tencent RTC (TRTC) to build a basic video RTC experience for the Android platform. The API level "mapping" for each step in this guide will facilitate the developers to switch from Twilio Video to TRTC in a quick and straightforward way.

TRTC has more feature sets that can help developers implement cost-effective, low-latency and high-quality interactive audio/video services. For comprehensive details on TRTC features and the implementation specification, please refer to the Tencent RTC website. If you require developer support or any further assistance regarding the TRTC integration, feel free to contact us, or you can join our Discord and Telegram. We will ensure a smooth integration and address any queries you may have.

Web

Last updated：2024-01-18 12:06:31

Overview

Welcome to our comprehensive guide designed to assist developers in migrating from Twilio Video to Tencent RTC (TRTC) on the Web platform. This document provides a detailed comparison at the API level between Twilio and TRTC, and outlines each step required to implement basic RTC video functionality.

For those planning to develop a new application, you can directly visit the Tencent RTC website and documentation for all the necessary resources and information.

Create a TRTC Application

Create a new application of Tencent RTC (TRTC), then you can get the following necessary information for the authentication of TRTC service.

SDKAppID

SDKAppID is used to uniquely identify a TRTC application. It is generated automatically when an application is created.

SDKSecretKey

SDKSecretKey is one of the key parameters to be used to generate the security signature. The generated signature ensures the access of TRTC service when calling the initialization and login API of the SDK.

Install and Set Up the SDK

By npm

1. Use npm to install trtc-sdk-v5 package in your project.

```
npm install trtc-sdk-v5 --save
```

2. Import the module in the project script.

```
import TRTC from 'trtc-sdk-v5';
```

By script

Download SDK file trtc.js from Github and add the following code to your webpage:

```
<script src="trtc.js"></script>
```

Join a Room

Twilio

```
// Replace Twilio Video import
import * as TwilioVideo from 'twilio-video'

var twilioVideo = TwilioVideo
var twilioRoom

twilioRoom = await twilioVideo.connect(TOKEN, { name: 'yourName', audio: false, vid
```

TRTC

```
const trtc = TRTC.create();

try {
  await trtc.enterRoom({ roomId: 8888, scene:'rtc', sdkAppId, userId, userSig });
  console.log('Entered the room successfully');
} catch (error) {
  console.error('Failed to enter the room ' + error);
}
```

Publish Local Video/Audio

Twilio

```
// video
let localVideoTrack = await twilioVideo.createLocalVideoTrack({
   height: { ideal: 720, min: 480, max: 1080 },
   width:  { ideal: 1280, min: 640, max: 1920 },
   aspectRatio: 16/9,
})

twilioRoom.localParticipant.publishTrack(localVideoTrack)
const localMediaContainer = document.getElementById('video-container-div')
localMediaContainer!.appendChild(localVideoTrack.attach())
```

```
// audio
let localAudioTrack = await twilioVideo.createLocalAudioTrack()

twilioRoom.localParticipant.publishTrack(localAudioTrack);
const audioElement = localAudioTrack.attach();
document.body.appendChild(audioElement);
```

TRTC

```
const view = document.getElementById('localVideo');
const cameraList = await TRTC.getCameraList();
if (cameraList[0]) {
  await trtc.startLocalVideo({
    view,
    option: {
      cameraId: cameraList[0].deviceId,
    }
  });
}

// To preview the camera image, you need to place an HTMLElement in the DOM, which
const view = 'local-video';
await trtc.startLocalVideo({ view });

await trtc.startLocalAudio();
```

Subscribe Remote Video/Audio

Twilio

```
twilioRoom.on('participantConnected', (participant) => {
   participant.on('trackSubscribed', (track) => {
       // a user turned on their video, render it
       document.getElementById('remote-video-container-div').appendChild(track.atta
   });

   participant.on('trackUnsubscribed', (track) => {
       // a user turned off their video, stop rendering it
       var selfTwilioVideo = document.getElementById('remote-video-container-div')
       selfTwilioVideo.querySelector('video').remove()
   })
})
```

TRTC

```
// Listen for the TRTC.EVENT.REMOTE_VIDEO_AVAILABLE event before entering the room
trtc.on(TRTC.EVENT.REMOTE_VIDEO_AVAILABLE, ({ userId, streamType }) => {
  // To play the video image, you need to place an HTMLElement in the DOM, which ca
  const view = `${userId}_${streamType}`;
  trtc.startRemoteVideo({ userId, streamType, view });
});
```

Leave a Room

Twilio

```
twilioVideo.disconnect()
```

TRTC

```
await trtc.exitRoom();
// After the exit is successful, if you do not need to use the trtc instance later,
trtc.destroy();
```

Conclusion

Compared to Twilio Video, this migration guide outlines how to use Tencent RTC (TRTC) to build a basic video RTC experience for the Android platform. The API level "mapping" for each step in this guide will facilitate the developers to switch from Twilio Video to TRTC in a quick and straightforward way.
TRTC has more feature sets that can help developers implement cost-effective, low-latency and high-quality interactive audio/video services. For comprehensive details on TRTC features and the implementation specification, please refer to the Tencent RTC website. If you require developer support or any further assistance regarding the TRTC integration, feel free to contact us, or you can join our Discord and Telegram. We will ensure a smooth integration and address any queries you may have.

Billing

Last updated：2024-08-21 11:16:24

Are there discounts for high-demand customers?

If your monthly spending on TRTC services exceeds 3,000 USD, you can contact us for contractual discounts.

How are TRTC services billed?

TRTC offers basic services and value-added services. For details about their billing, see Billing Overview. We provide users with a 10,000-minute free package each month.

Price calculator

You can use the TRTC Price Calculator to estimate your cost.

How do I view my bills and transaction history?

You can view your bills and transaction history in Billing Center > Bill Details.

How do I view the details of my billable durations?

Real-time durations: You can find a usage graph and view your usage details in Usage Statistics of the TRTC console. If you select a single day, the page will show usage statistics on a 5-minute basis. If you select multiple days, it will show usage statistics on a daily basis. The statistics are accurate to the minute.

Billable durations: You can download an Excel file of your billable durations in Tencent Cloud’s billing center. The file shows your usage on a 5-minute as well as daily basis. The statistics are accurate to the second.

Note:

Real-time usage statistics may be slightly different from the durations you are actually billed for. In case of conflicts, **the statistics in your bills shall apply**.

How do I view the remaining minutes in my package?

Durations are deducted from your package in real time, and the number of remaining minutes in your package is updated every 5 minutes. You can view your remaining minutes in Package Management.

Given that billable durations are calculated in seconds and rounded up to the nearest minute, will more minutes than I actually use be deducted from my package?

No. Deductions are based on your cumulative usage in a day.

Below is an example of how deductions work:

Calculation Period	Period Usage	Cumulative Usage	Cumulative Billable Duration	Period Deduction	Cumulative Deduction

00:00:00 - 00:04:59	30 sec	30 sec	1 min	1 min	1 min
00:05:00 - 00:09:59	20 sec	50 sec	1 min	0 min	1 min
00:10:00 - 00:14:59	40 sec	90 sec	2 min	1 min	2 min

How do I estimate my usage of TRTC services and the cost?

You can use the TRTC Price Calculator to estimate your usage and cost.

Why does a video call or video streaming session generate audio durations?

In most cases, if a user subscribes to a stream, they receive both audio and video data. However, in cases where the sender’s camera is turned off, the recipient disables remote video or encounters a network problem, or there is only one user in the room, the user will receive no video data. To reduce your expenses, TRTC bills a period during which a user does not receive video data as audio duration.

How are free Audio & Video Minutes and free MAU calculated in multiple subscription packages with the same SDKAppID？

The calculation of our audio and video hours and MAUs with the same SDKAppID follows an aggregation logic. If you purchase multiple products such as Chat, Live, Call or Conference, the resources in each product package are added together. For example, if you have purchased the Live Standard package for the same SDKAppID, which includes 10,000 free MAUs and 300,000 free audio/video minutes, and the Call 1v1 package, which includes 5,000 free MAUs and 100,000 free audio/video minutes, you have 15,000 free MAUs and 100,000 free audio/video minutes for the same month. If you have purchased the Call 1v1 package, which includes 5000 free MAUs and 100,000 free minutes, your free MAUs for the month will be 15,000 and your free minutes will be 400,000.

How is MAU (Monthly Active Users) calculated?

MAU calculates the number of unique users that log in to app in a given month. After a user logs in successfully by calling the login function of the SDK to establish a persistent connection with the backend, MAU will be increased by one. A user that logs in repeatedly in the same month is considered as one user only for MAU. Use the login function appropriately according to business scenarios to avoid an excessively high MAU. In addition, if a user simultaneously uses Chat, Live, Call, or any combination of the products, they will only be counted once as one MAU.

How is screen sharing billed?

Screen sharing data is published as a separate stream. If a user receives a screen sharing stream, the user’s duration will be billed as video duration.

How is relaying to CDN billed?

TRTC leverages the capabilities of CSS to relay streams to CDNs. **CSS** charges you according to the billing rules described in CDN Relayed Live Streaming.

Are fees charged if there is only one user in a room?

A room with only one user consumes TRTC’s resources, even if no streams are pushed (no upstream data). The only user in a room cannot subscribe to other users’ streams and therefore won’t receive video data. As a result, the duration is billed as **audio duration**.

Why is my service status "Disabled"?

Overdue payments will lead to service suspension. The services will be resumed automatically after you make the payment.

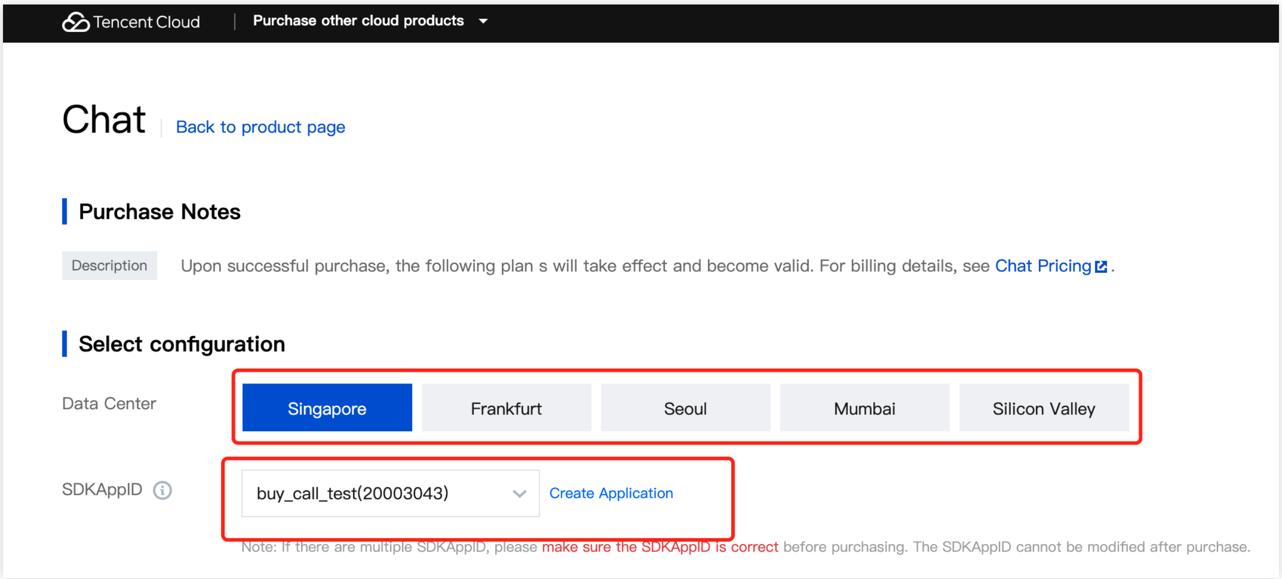
If you manually disabled an application, you can click **Enable Application** to resume TRTC services for the application.

When purchasing Call, why can't I find the application(SDKAppId) I want to select?

Currently, TRTC purchase page only supports purchasing Prepaid applications in the Chat Singapore Data Center. Other types of application purchase methods are as follows:

If your application is a pay-as-you-go Chat application, you can refer to the Purchase Guide to proceed with the purchase.

If your application is a Prepaid Chat application, but not in the Chat Singapore Data Center, you can go to the Chat purchase page, select the corresponding data center and application within the page, then check "Audio/Video call", and choose the Call version to purchase.



Chat | Back to product page

Purchase Notes

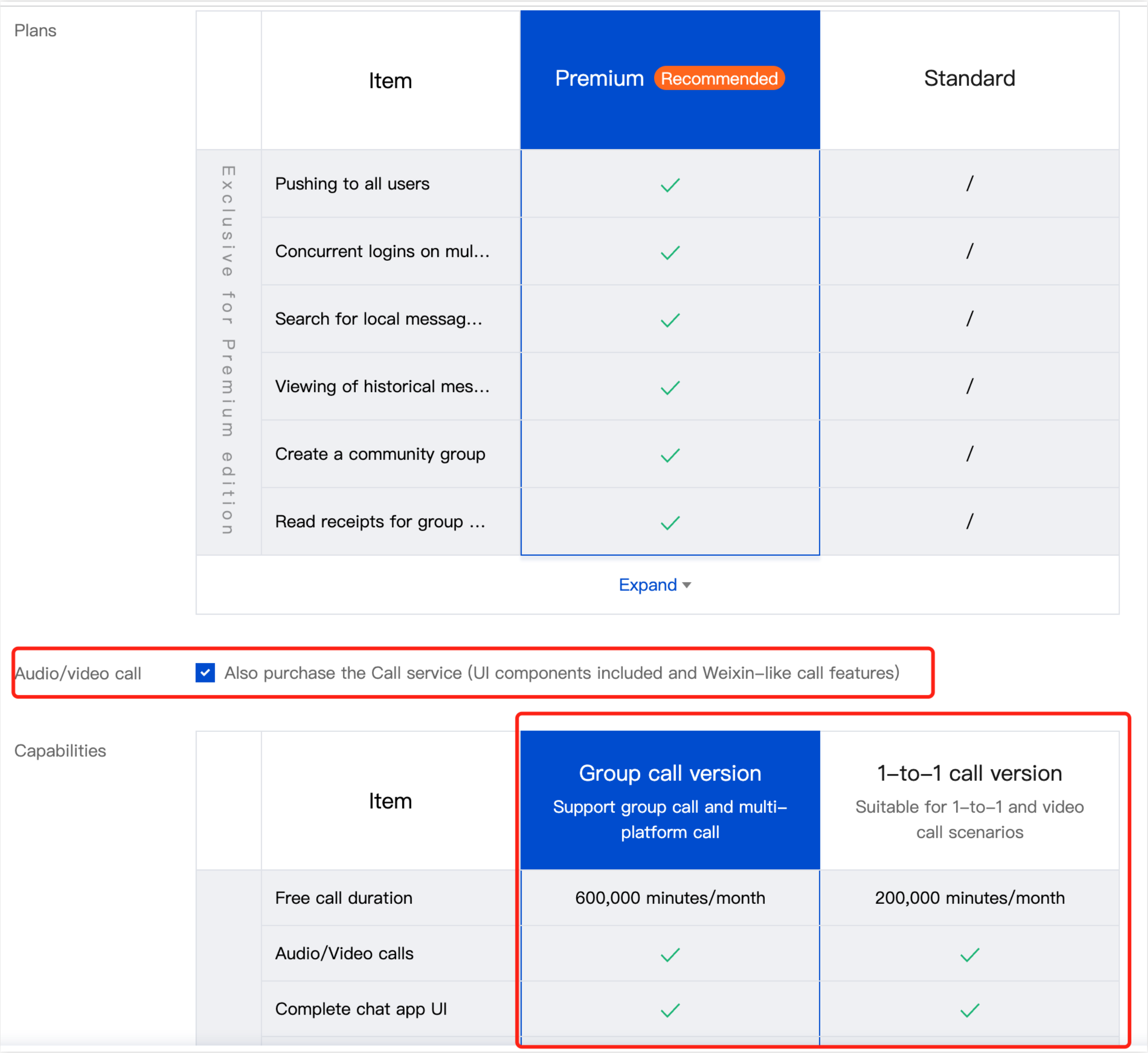
Description Upon successful purchase, the following plan s will take effect and become valid. For billing details, see Chat Pricing.

Select configuration

Data Center: Singapore | Frankfurt | Seoul | Mumbai | Silicon Valley

SDKAppID: buy_call_test(20003043) Create Application

Note: If there are multiple SDKAppID, please make sure the SDKAppID is correct before purchasing. The SDKAppID cannot be modified after purchase.



Plans

	Item	Premium Recommended	Standard
Exclusive for Premium edition	Pushing to all users	✓	/
	Concurrent logins on mul…	✓	/
	Search for local messag…	✓	/
	Viewing of historical mes…	✓	/
	Create a community group	✓	/
	Read receipts for group …	✓	/

Expand ▾

Audio/video call ☑ Also purchase the Call service (UI components included and Weixin-like call features)

Capabilities

	Item	Group call version Support group call and multi-platform call	1-to-1 call version Suitable for 1-to-1 and video call scenarios
	Free call duration	600,000 minutes/month	200,000 minutes/month
	Audio/Video calls	✓	✓
	Complete chat app UI	✓	✓

Features

Last updated：2024-06-18 17:19:13

What is Tencent Real-Time Communication?

Leveraging Tencent's many years of experience in network and audio/video technologies, Tencent Real-Time Communication (TRTC) offers solutions for group audio/video calls and low-latency interactive live streaming. With TRTC, you can quickly develop cost-effective, low-latency, and high-quality interactive audio/video communication apps with a wide range of features. To learn more, see Product Introduction > Overview.

How can I experience TRTC demos?

You can go to the Free Demo page to find TRTC demos for different platforms.

How can I quickly get started with TRTC?

TRTC provides you with demo source code for various platforms that you can use to quickly build your own applications. For more information, see User Tutorial.

What is `RoomID` in TRTC? What is its value range?

`RoomID` uniquely identifies a room and can range from 1 to 4294967295. You are responsible for maintaining and assigning the room IDs of your applications.

What is `UserID` in TRTC? What is its value range?

`UserID` uniquely identifies a user in a TRTC application. It can contain letters (case sensitive), digits, and underscores, preferably not longer than 32 bytes.

How long is the lifecycle of a TRTC room?

The first user who enters a room is the owner of the room. Room owners cannot close rooms manually.
In call modes, TRTC closes a room when all users exit the room.
In live streaming modes, if the last user who exits a room is an anchor, TRTC will close the room immediately; if the user is an audience member, TRTC will close the room after 10 minutes.
A user is removed from a room 90 seconds after an unexpected disconnection occurs. If all users are disconnected, the room is closed after 90 seconds. **The waiting time after a disconnection occurs is also billed**.
If a user attempts to enter a room that does not exist, TRTC will automatically create a room with the ID entered.

Can users not subscribe to audio/video streams?

To enable instant streaming, TRTC subscribes users to audio/video streams by default upon room entry. You can call the `setDefaultStreamRecvMode` API to switch to the manual subscription mode.

How can I make it so users are not automatically subscribed to audio/video streams upon room entry?

Yes. You can specify a `streamId` via `TRTCParams` of `enterRoom` or call the `startPublishing` API to pass in the `streamId` .

What roles are supported during live streaming in TRTC? How do they differ from each other?

The live streaming scenarios (`TRTCAppSceneLIVE` and `TRTCAppSceneVoiceChatRoom`) support two roles: `TRTCRoleAnchor` (anchor) and `TRTCRoleAudience` (audience). An anchor can both send and receive audio/video data, but audience members can only receive and play others' data. You can call `switchRole()` to switch roles.

What application scenarios are supported in TRTC rooms?

The following application scenarios are supported:
TRTCAppSceneVideoCall: The video call scenario is suitable for one-to-one video calls, video conferences with up to 300 participants, online medical consultation, video chat, and video interviews.
TRTCAppSceneLIVE: The interactive video streaming scenario is suitable for low-latency live video streaming, interactive classroom for up to 100,000 participants, live video competition, video dating, remote training, and mega conferences.
TRTCAppSceneAudioCall: The audio call scenario is suitable for one-to-one audio calls, audio conferences with up to 300 participants, audio chat, and online party games.
TRTCAppSceneVoiceChatRoom: The interactive audio streaming scenario is suitable for low-latency audio streaming, live audio co-anchoring, audio chat rooms, karaoke, and radio.

What platforms does TRTC support?

TRTC supports platforms including iOS, Android, Windows (C++), Unity, macOS, web, and Electron. For more information, see Supported Platforms.

What are the differences between TRTC Lite and TRTC Professional?

See Edition Comparison.

Does TRTC support co-anchoring during live streaming?

Yes. For detailed instructions, see the following documents:
Live Streaming Mode (iOS and macOS)
Live Streaming Mode (Android)
Live Streaming Mode (Windows)
Live Streaming Mode (Electron)
Live Streaming Mode (Web)

How many rooms can there be in TRTC at the same time?

There can be up to 4,294,967,294 concurrent rooms in TRTC. There is no limit to the number of non-concurrent rooms.

How do I create a room?

A room is automatically created by TRTC when a user enters a room. Therefore, you do not need to manually create a room. Just call the client API for room entry.

iOS & macOS > enterRoom

Android > enterRoom

Windows（C++） > enterRoom

Windows (C#) > enterRoom

Electron > enterRoom

Web > join

What is the upper limit on the bandwidth used for the video service of TRTC?

There isn’t a limit.

Can TRTC be deployed on-premises?

Private deployment of TRTC is not commercially available yet. If you have questions about it or want to use it, please contact us at info_rtc@tencent.com.

To enable relay to CDN in TRTC, do I need to register my domain name with an ICP filing number?

Yes, according to relevant regulations, playback domains must be registered.

How long is the average latency in TRTC?

The average end-to-end latency of TRTC around the globe is less than 300 milliseconds.

Does TRTC support active calling?

You can enable this feature using signaling channels. For example, you can use the custom message feature of Chat to enable active calling. For more information, see the scenario-specific demos in the SDK source code.

Can I use Bluetooth earphones when having one-to-one video calls in TRTC?

Yes, you can.

Does TRTC support screen sharing on PCs?

Yes. For details, see the following documents:

Real-Time Screen Sharing (Windows)

Real-Time Screen Sharing (macOS)

Real-Time Screen Sharing (Web)

For more information on the screen sharing APIs, see Client APIs > All Platforms (C++). You can also use Electron APIs.

Can I share local video files in TRTC?

Yes. You can achieve this using the Custom Capturing and Rendering feature.

Can I record live streaming sessions and save the recording files locally on my phone?

You cannot save recording files directly to your phone. Recording files are saved to VOD. You can download them from VOD and save them to your phone.

Does TRTC support audio-only streams?

Yes, it does.

Can there be more than one screen sharing image in the same room?

Currently, only one screen sharing substream is allowed in a room.

When a specified window is shared (`SourceTypeWindow`), if the window size changes, will the resolution of the video stream change accordingly?

By default, the SDK automatically adjusts encoding parameters according to the size of the shared window.
If you want a fixed resolution, call the `setSubStreamEncoderParam` API to set encoding parameters for screen sharing or specify the parameters when calling the `startScreenCapture` API.

Does TRTC support 1080p videos?

Yes. You can set the resolution through `setVideoEncoderParam` , the video encoding parameter of the SDK.

Can I customize data capturing in TRTC?

You can on some platforms. For details, see Custom Capture and Rendering.

Is communication between TRTC and the ILVB SDK possible?

No, it's not.

Is communication between TRTC and MLVB possible?

TRTC and MLVB have different backend architectures and therefore cannot communicate with each other. However, you can relay streams from TRTC to CDNs.

How do different room entry modes (`AppScene`) vary from one another in TRTC?

TRTC has four room entry modes. Video call (`VideoCall`) and audio call (`VoiceCall`) are the call modes, and interactive video live streaming (`Live`) and interactive audio live streaming (`VoiceChatRoom`) are the live streaming modes.
The call modes allow a maximum of 300 users in each TRTC room, and up to 50 of them can speak at the same time. The call modes are suitable for scenarios such as one-to-one video calls, video conferences with up to 300 participants, online medical consultation, video interviews, video customer service, and online party games.
The live streaming modes support a maximum of 100,000 concurrent users in each room and allow smooth mic on/off. Co-anchoring latency is kept below 300 milliseconds and viewing latency below 1,000 milliseconds. The live streaming modes are suitable for scenarios such as low-latency interactive live streaming, interactive classrooms for up to 100,000 participants, video dating, online education, remote training, and mega conferences.

Can I use the hands-free mode during video calls in TRTC?

Yes. You can enable the hands-free mode by setting audio routes. In a native SDK, use the `setAudioRoute` API to switch routes.

Does TRTC support volume reminders?

Yes. You can call the `enableAudioVolumeEvaluation` API to enable volume reminders.

Does TRTC support mirror images?

Yes. You can call the `setLocalViewMirror` API to set the mirroring mode for the preview image of the local camera or call `setVideoEncoderMirror` to set the mirroring mode for encoded images.

Can I record the audio of a TRTC call and save the recording file locally?

Yes. You can call `startAudioRecording` to record all audios of a call, including that of the local user, remote users and the background music, into a single file in the format of PCM, WAV, or AAC.

Can I record the video of a TRTC call into a file?

TRTC supports audio/video recording on a local server. To use this feature, please submit a ticket for the SDK and instructions.
You can also use the on-cloud recording feature to record videos.

Does TRTC support floating windows (like those in WeChat) or switching between a big and small window?

These features are part of UI design, for which the TRTC SDK sets no restrictions. Our official demo provides sample code for image overlaying and the grid layout and supports floating windows, big/small window switch, and window dragging. For more information, see the TUICalling demo.

How do I make an audio-only call in TRTC?

TRTC does not use separate channels for audio and video. You can make an audio-only call by calling only `startLocalAudio` and not `startLocalPreview.

How do I enable relay to CDN and recording for an audio-only call in TRTC?

In TRTC SDK earlier than v6.9, you need to construct `json{\\"Str_uc_params\\":{\\"pure_audio_push_mod\\":1}}` and pass it in `TRTCParams.businessInfo` during room entry. `1` means relay to CDN, and `2` means relay to CDN and recording.
In TRTC SDK 6.9 or later, just set the scene parameter to `TRTCAppSceneAudioCall` or `TRTCAppSceneVoiceChatRoom` during room entry.

Can I kick a user out, forbid a user to speak, or mute a user in a TRTC room?

Yes, you can.
To enable the features through simple signaling operations, use `sendCustomCmdMsg` , the custom signaling API of TRTC, to define your own control signaling, and users who receive the message will perform the action expected. For example, to kick out a user, just define a kick-out signaling, and the user receiving it will exit the room.
If you want to implement a more comprehensive operation logic, we recommend that you use Tencent Cloud Chat to map the TRTC room to a Chat group and enable the features via the sending/receiving of custom messages in the group.

Can TRTC pull and play back RTMP/FLV streams?

Yes. The TRTC SDK has integrated TXLivePlayer. If you need more player features, consider using the full-featured LiteAVSDK_Professional.

How many people can there be in a TRTC call?

In call scenarios, each room can accommodate up to 300 concurrent users, and up to 50 of them can turn on their cameras or mics at the same time.
In live streaming scenarios, each room can accommodate up to 100,000 concurrent users, and up to 50 of them can be assigned the anchor role and turn on their cameras or mics at the same time.

How do I start a live streaming session in TRTC?

TRTC offers a dedicated low-latency interactive live streaming solution that allows up to 100,000 participants. Co-anchoring latency is kept as low as 200 milliseconds, and viewing latency is below one second. It adapts excellently to poor network conditions and is optimized for complex mobile network environments.
For more information, see Live Streaming Mode.

Can I use the custom message sending API of TRTC to implement features such as chat and on-screen comments?

No. Custom message sending is intended for simple and low-frequency signaling scenarios. For details, see Sending Custom Messages > Use Limits.

Can I loop background music in TRTC? Can I adjust the playback progress of background music?

Yes. You can call the playback API again in the playback completion callback to loop background music. `seekMusicToPosInMS` of `TXAudioEffectManager` can be used to set the playback progress.

Note

`setBGMPosition()` of `TXAudioEffectManager` has been replaced with `seekMusicToPosInMS` since version 7.3.

Can I listen for the entry/exit of users through callbacks in TRTC? Can I use `onUserEnter` or `onUserExit` ?

Yes. You can use `onRemoteUserEnterRoom` and `onRemoteUserLeaveRoom` to listen for the entry/exit of users, but callbacks are triggered only for users who can send data.

Note

`onUserEnter` and `onUserExit` have been replaced with `onRemoteUserEnterRoom` and `onRemoteUserLeaveRoom` since version 6.8.

How do I listen for network disconnection and reconnection in TRTC?

You can listen for the events through the following callbacks:

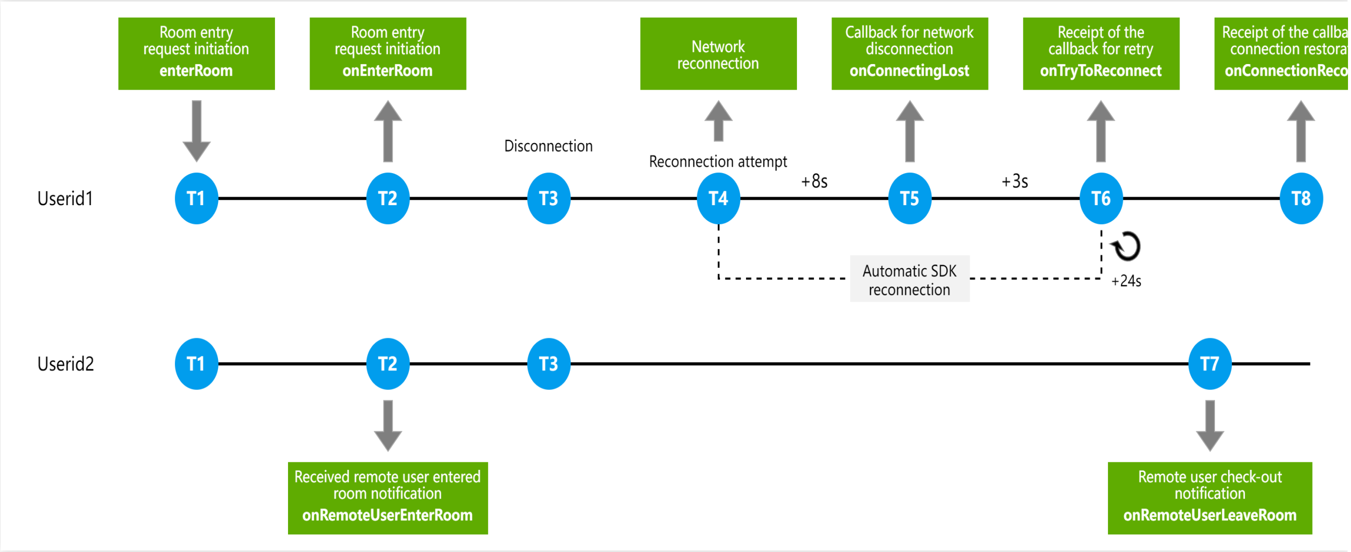
`onConnectionLost` : The SDK is disconnected from the server.

`onTryToReconnect` : The SDK is reconnecting t to the server.

`onConnectionRecovery` : The SDK is reconnected to the server.

Does the TRTC SDK support automatic reconnection?

The SDK reconnects a user automatically after a disconnection. If it fails to reconnect the user within 30 minutes, it will remove the user from the room and return the -3301 error.The figure below shows the callbacks triggered when `Userid1` enters a room, is disconnected from the SDK, and re-enters the room:


Room entry request initiation enterRoom
Room entry request initiation onEnterRoom
Network reconnection
Callback for network disconnection onConnectingLost
Receipt of the callback for retry onTryToReconnect
Receipt of the callba connection restora onConnectionReco
Disconnection
Reconnection attempt
+8s
+3s
Userid1
T1
T2
T3
T4
T5
T6
T8
Automatic SDK reconnection
+24s
Userid2
T1
T2
T3
T7
Received remote user entered room notification onRemoteUserEnterRoom
Remote user check-out notification onRemoteUserLeaveRoom

Description:

T1: The user calls the `enterRoom` API to enter a room.

T2: `Userid1` receives the `onEnterRoom` callback, and `Userid2` receives the `onRemoteUserEnterRoom` callback after about 300 milliseconds (latency).

T3: `Userid1` is disconnected due to a network problem, and the SDK tries to reconnect the user.

T4: If `Userid1` is not reconnected within the first eight seconds, the user will receive the `onConnectionLost` callback.

T5: If three more seconds elapse and `Userid1` is still not reconnected, the user will receive the `onTryToReconnect` callback.

T6: `Userid1` will then receive `onTryToReconnect` callback every 24 seconds.

T7: 90 seconds after the `onConnectionLost` callback, `Userid2` receives the `onRemoteUserLeaveRoom` callback, which indicates that `Userid1` is offline.

T8: If reconnection succeeds at any point during the 90 seconds, `Userid1` will receive the `onConnectionRecovery` callback.

Is there a callback for first frame rendering? Can I listen for the start of image rendering or audio playback?

Yes. You can use `onFirstVideoFrame` and `onFirstAudioFrame` to listen for the events.

Can I take a screenshot of a video in TRTC?

Currently, you can call `snapshotVideo()` on iOS and Android to take screenshots of local and remote videos.

Why do I fail to connect peripheral devices such as Bluetooth earphones to TRTC?

Currently, TRTC supports mainstream Bluetooth earphones and peripherals, but for some devices, there are still compatibility issues. We recommend that you use our official demos and QQ audio/video calls to test the compatibility of a device.

How do I get information such as the upstream/downstream bitrate, resolution, packet loss rate, and audio sample rate of a TRTC audio/video call?

You can call the `onStatistics()` API of the SDK to get the statistics.

Does TRTC’s background music API `playBGM()` support online music?

No. Currently, it supports only local music. You can download an online music file and then call `playBGM()` to play it.

Can I set the local audio capturing volume or the playback volume of each remote user?

Yes. You can call `setAudioCaptureVolume()` to set the audio capturing volume of the SDK and `setRemoteAudioVolume()` to set the playback volume of a remote user.

What are the differences between `stopLocalPreview` and `muteLocalVideo` ?

`stopLocalPreview` is used to stop local video capturing. If you call this API, both you and other users will not see your image.
`muteLocalVideo` is used to stop the sending of local video images. If you call this API, other users will not see your image, but you can still preview your own image.

What are the differences between `stopLocalAudio` and `muteLocalAudio` ?

`stopLocalAudio` is used to disable the capturing and sending of local audio.
When `muteLocalAudio` is called, TRTC does not stop the sending of audio/video data. It continues to send muted data packets at extremely low bitrate.

What resolutions does the TRTC SDK support?

We recommend that you set the resolution as instructed in Setting Video Quality for better image quality.

How do I set the upstream video bitrate, resolution, and frame rate in the TRTC SDK?

Call the `setVideoEncoderParam()` API of `TRTCCloud` and set `videoResolution` (resolution), `videoFps` (frame rate), and `videoBitrate` (bitrate) in `TRTCVideoEncParam` .

How do I rotate videos in the TRTC SDK?

See Rotating Videos.

How do I make a video call in the landscape mode?

See Rotating Videos.

How do I match the rotation of the local and remote videos if they are different?

See Rotating Videos.

What image quality-related settings (bitrate, resolution, and frame rate) does TRTC recommend?

See Setting Video Quality > Recommended Configuration.

Can I test my network speed in TRTC? How?

Yes, you can. For details, see Testing Network Quality.

Can I control access to a TRTC room to allow only authorized users to enter the room?

Yes. For details, please see Enabling Advanced Permission Control.

Can TRTC pull and play back RTMP/FLV streams?

Yes.

What formats does TRTC support for custom rendering?

iOS: I420, NV12, and BGRA.
Android: I420 and Texture2D.

How do I enable on-cloud recording and playback in TRTC?

Please see On-Cloud Recording and Playback.

Does TRTC support beauty filters?

Yes, it does. TRTC offers various effects based on face recognition technologies, such as AI beauty filters, makeup effects, facial feature adjustment, and green screen keying.
To use beauty filters on web, see Enabling Beauty Filters.
The AI beauty filters in native TRTC SDKs are a value-added service which is charged by Tencent Effect SDK.

Note

Currently, only TRTC Professional for iOS and Android support AI beauty filters.

Can I use TRTC outside the Chinese mainland?

In addition to the Chinese mainland, you can also use TRTC in Hong Kong and other regions.

Note

TRTC offers reliable and secure network connection across the globe. It uses Tencent Cloud’s proprietary multi-level addressing algorithm and can connect to nodes across the entire network. Abundant high-bandwidth resources and globally distributed edge servers allow it to keep its average end-to-end latency below 300 milliseconds globally. International connection may be subject to actual local conditions and application scenarios.

Does TRTC support detecting inappropriate content in images?

TRTC blocks pornographic, politically sensitive, and other inappropriate content during live streaming.

How do I query the information of all users in a room?

You can view the currently online rooms and users through Console - Monitoring Dashboard - Real-Time Monitoring.

Can TRTC receive other RTSP streams?

No, it can’t, but it does support RTMP streaming. For details, see Publishing over RTMP.

Does TRTC support dual-channel encoding?

Yes, TRTC supports dual-channel audio.

When publishing streams, does TRTC package or encode streams first?

After data capturing, TRTC encodes streams first before packaging.

Does the TRTC SDK use Swift?

The model layer uses Objective-C and the UI layer uses Swift.

Can I use TRTC if my account is a personal account?

Yes, you can.

Does TRTC SDK support IPv6?

Yes, it is recommended to upgrade to version 10.6 and above for further optimization of IPv6 room entry speed.

How to use the Region of Interest video codec feature in TRTC Monthly Package?

To use the Region of Interest video codec feature, you need to call the experimental interface (callExperimentalAPI, taking Android as an example) to set the region of interest. Here is an example:

```
//Set ROI parameters
{
    "api":"UpdateRoiConfig",
    "params":{
          "roiConfigArray":[
              {
                    "stream":1,
                    "x":0,
                    "y":0,
                    "width":100,
                    "height":100,
                    "level":2
               },
                {
                        "stream":1,
                        "x":20,
                        "y":200,
                        "width":100,
                        "height":100,
                        "level":2
                }
            ]
        }
}
```

Field Name	Description
stream	Stream Type: 0 for main stream, 1 for substream, 2 for auxiliary stream.
x,y,w,h	ROI coordinate points, with the codec output resolution as a reference.
level	[0, 12], which is the intensity of the ROI. The higher the value, the more obvious the effect in the ROI area, but the non-ROI area may be more blurry.

UserSig

Last updated：2025-04-09 17:56:44

Overview

This document outlines two authentication methods for Tencent-RTC services, with a focus on UserSig, a security signature by Tencent Cloud to safeguard against unauthorized access. For basic cloud service usage, provide SDKAppID, UserID, and UserSig during SDK initialization or login.

SDKAppID is used to identify your application.

UserID is used to identify your user.

UserSig is a security signature calculated based on the first two using the **HMAC SHA256** encryption algorithm. As long as attackers cannot forge the UserSig, they cannot steal your cloud service traffic.

How to calculate UserSig during the debugging phase?

you can calculate and obtain UserSig using either Client Sample Code or the Console. Please refer to the following introduction for details.

Unsafe:

Note that the following two UserSig acquisition calculation schemes are only suitable for debugging. If the product is to be officially launched, **using these schemes is not recommended** because the SECRETKEY in the client code (especially on the Web) is easily vulnerable to decompilation and reverse engineering.

Client calculation of UserSig

1. Get SDKAPPID and Key:

Log in to Tencent-RTC Console > App Management.

Locate the app with the desired SDKAppID, click its name for details.

Click on SDKSecretKey to reveal and copy.

2. Calculate UserSig:

To facilitate client use, we provide source files for calculating UserSig on various platforms. You can download and use them directly:

Android	iOS	Web	Windows(C++)	Windows(C#)	Flutter	Mac

Github	Github	Github	Github	Github	Github	Github

The sample code is as follows (of course, you can also refer to the demo projects of our products, see the development documentation of each product):

Android

iOS

Web

Window(C++)

Window(C#)

Flutter

Mac

```
// Step 1: Import the source file
import com.xxx.xxx.GenerateTestUserSig;

// Step 2: Fill in the SDKAppID and SDK key obtained from the previous steps
GenerateTestUserSig.SDKAPPID = xxxxxx;
GenerateTestUserSig.SECRETKEY = "xxxxxx";

// Step 3: Generate userSig based on userID
String userSig = GenerateTestUserSig.genTestUserSig("userID");
```

```
// Step 1: Import the header file
#import "GenerateTestUserSig.h"

// Step 2: Fill in the SDKAppID and SDK key obtained from the previous steps
[GenerateTestUserSig setSDKAPPID:xxxxxx];
[GenerateTestUserSig setSECRETKEY:@"xxxxxx"];

// Step 3: Generate userSig based on userID
NSString *userSig = [GenerateTestUserSig genTestUserSig:@"userID"];
```

```
// Step 1: Import the module
<script src='js/libs/lib-generate-test-usersig.min.js'></script>
<script src='js/libs/generateTestUserSig.js'></script>

// Step 2: Fill in the SDKAppID and SDK key obtained from the previous steps, enter
const {sdkAppId, userSig } = genTestUserSig({
        sdkAppId: xxxxxx,
        userId: 'xxxxxx',
        sdkSecretKey: 'xxxxxx',
}
```

```
// Step 1: Import the header file
#include "GenerateTestUserSig.h"

// Step 2: Fill in the SDKAppID and SDK key obtained from the previous steps
const int SDKAPPID = xxxxxx;
const char* SECRETKEY = "xxxxxx";

// Step 3: Generate userSig based on userID
const char* userSig = GenerateTestUserSig::genTestUserSig("userID", SDKAPPID, SECRE
```

```
// Step 1: Import the header file
using GenerateTestUserSig;

// Step 2: Fill in the SDKAppID and SDK key obtained from the previous steps
GenerateTestUserSig.SDKAPPID = xxxxxx;
GenerateTestUserSig.SECRETKEY = "xxxxxx";

// Step 3: Generate userSig based on userID
string userSig = GenerateTestUserSig.GetInstance().GenTestUserSig("userID");
```

```
// Step 1: Import the source file
import 'package:xxx/GenerateTestUserSig.dart';

// Step 2: Fill in the SDKAppID and SDK key obtained from the previous steps
GenerateTestUserSig.SDKAPPID = xxxxxx;
GenerateTestUserSig.SECRETKEY = "xxxxxx";

// Step 3: Generate userSig based on userID
String userSig = GenerateTestUserSig.genTestUserSig("userID");
```

```
// Step 1: Import the header file
#import "GenerateTestUserSig.h"

// Step2: Enter the SDKAppID and SDK secret obtained in the Back step
[GenerateTestUserSig setSDKAPPID:xxxxxx];
[GenerateTestUserSig setSECRETKEY:@"xxxxxx"];

// Step 3: Generate userSig based on userID
NSString *userSig = [GenerateTestUserSig genTestUserSig:@"userID"];
```

Get UserSig from the console

Log in to Tencent-RTC Console, navigate to Development Tools > UserSig Tools.
Under the UserSig Generation Tool, select the corresponding SDKAppID and UserID.
Click the Generate button to compute the corresponding UserSig.

How to calculate UserSig during the official operation phase?

During the official operation phase, Tencent-RTC provides a more secure server-side UserSig calculation solution. This maximizes the protection of the key used to calculate UserSig from being leaked, as compromising a server is more difficult than reverse engineering an app. The specific implementation process is as follows:

1. Before your app calls the SDK initialization function, it must first request UserSig from your server.
2. Your server calculates UserSig based on SDKAppID and UserID. Refer to the first part of the documentation for the source code.
3. The server returns the calculated UserSig to your app.
4. Your app passes the obtained UserSig to the SDK via a specific API.
5. The SDK submits `SDKAppID + UserID + UserSig` to Tencent CVM for verification.
6. Tencent Cloud verifies UserSig to confirm its validity.
7. After verification, Tencent-RTC services will be provided to the Tencent-RTC SDK.

To simplify your implementation process, we provide UserSig Calculation Source Code and examples in multiple language versions:

Language Version	Signature algorithm	Source Code	Usage Examples
Java	HMAC-SHA256	genSig	Github
GO	HMAC-SHA256	genSig	Github
PHP	HMAC-SHA256	genSig	Github
Node.js	HMAC-SHA256	genSig	Github
Python	HMAC-SHA256	genSig	Github
C#	HMAC-SHA256	GenSig	Github

Firewall Restrictions

Last updated：2024-02-29 11:02:59

What ports and domain names should I add to the allowlist of my firewall for a native SDK?

Add the following ports to the allowlist:

TRTC SDK (Native)	Ports
TCP	443, 20166, 10443, 10444, 10445, 10446, 10447, 10448, 10449, 10450, 10451, 13275, 23275, 33000, 37528
UDP	8000, 8080, 8001, 8002, 8003, 8004, 8005, 8006, 8007, 8008, 8009, 16285, 9000

Add the following domain names to the allowlist:

```
intl-query.trtc.tencent-cloud.com
intl-accelerate.trtc.tencent-cloud.com
trtc-client-log-overseas-1258344699.cos.ap-singapore.myqcloud.com
intl-sdklog.trtc.tencent-cloud.com
sdkdc.live.qcloud.com
speedtestint.trtc.tencent-cloud.com
intl-query.trtc.tencent-cloud.com
hwapi.im.qcloud.com
videoapi-sgp.im.qcloud.com
trtc-sdk-config-1258344699.file.myqcloud.com
```

Note:

As the Tencent Cloud service-side IP address/domain is dynamic, and not fixed to a certain batch of IP addresses/domains, we are unable to provide you with a fixed list.

What ports and domain names should I add to the allowlist of my firewall for WebRTC?

Add the following ports to the allowlist:

WebRTC (H5)	Ports
TCP	443
UDP	8000, 8080, 8800, 843, 443, 16285

Add the following domain names to the allowlist:

```
signailing.rtc.tencentcloud.com
schedule.rtc.tencentcloud.com
*.rtc.tencentcloud.com
*.rtc-web.com
*.rtc-web.io
```

How do I configure proxies for clients to access the TRTC SDK for web from a private network?

You can use the Nginx + Coturn scheme. For detailed directions, see Handle Firewall Restriction.

Solution	Application Scenario	Network Requirements
Solution 1	Allow clients to access specific proxy servers in the public network	Allow clients to access the proxy servers configured in the public network
Solution 2	Allow clients to access the public network via proxy servers configured in the private network	Allow the proxy servers to access the public network

How to Downsize Installation Package

Last updated：2023-10-13 11:21:36

How much will the file increment be after the TRTC SDK is integrated?

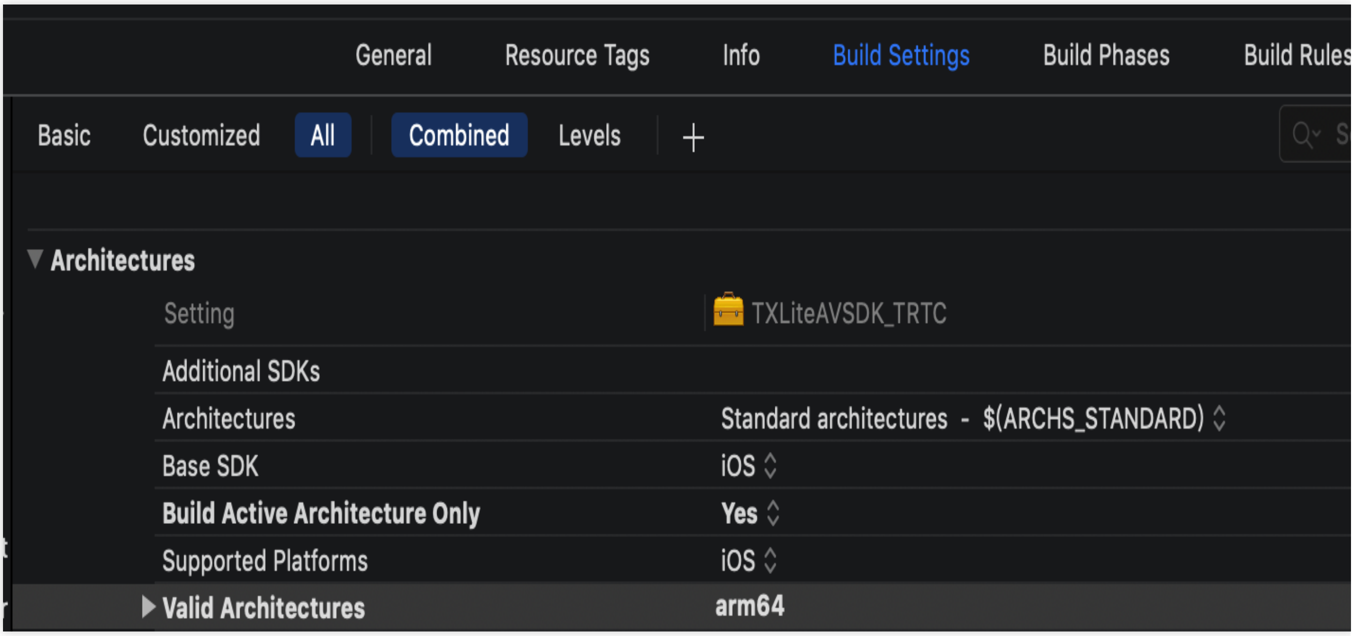
The file size increment varies by TRTC SDK version. For more information, please see SDK Download.

How do I reduce the size of an installation package for iOS?

Method 1. Only package the ARM64 architecture (recommended)

Method 2. Enable Bitcode

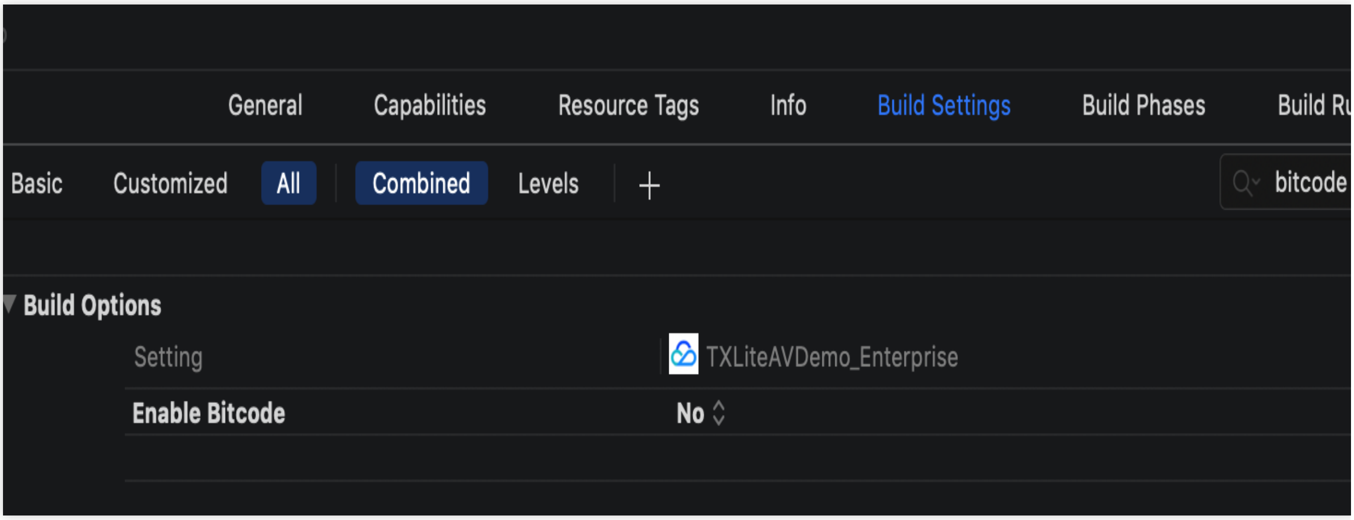
For iPhone models after 5s, you can just package x64 architecture by setting "Build Active Architecture Only" in "Build Settings" in Xcode to "Yes" and write only `arm64` into "Valid Architectures". The single-architecture IPA increment of the TRTC SDK will be 1.9 MB only.



▼ Architectures

Setting	TXLiteAVSDK_TRTC
Additional SDKs	
Architectures	Standard architectures - $(ARCHS_STANDARD)
Base SDK	iOS
Build Active Architecture Only	**Yes**
Supported Platforms	iOS
▶ **Valid Architectures**	**arm64**

For iPhone 5s and older models, **if all third-party libraries in the project support Bitcode,** you can enable Bitcode to reduce the size of the installation package. Toggle on "Enable Bitcode" in "Build Settings" > "Build Options" to enable bitcode.



Setting	TXLiteAVDemo_Enterprise
Enable Bitcode	No

From 2016 on, Apple started to support Bitcode compilation in the Xcode development environment. After Bitcode is enabled, the compiler will generate the application's intermediate code instead of the actual assembly code, and users will download and install the machine code generated for the specific mobile CPU architecture from App Store, which greatly reduces the installation package size.

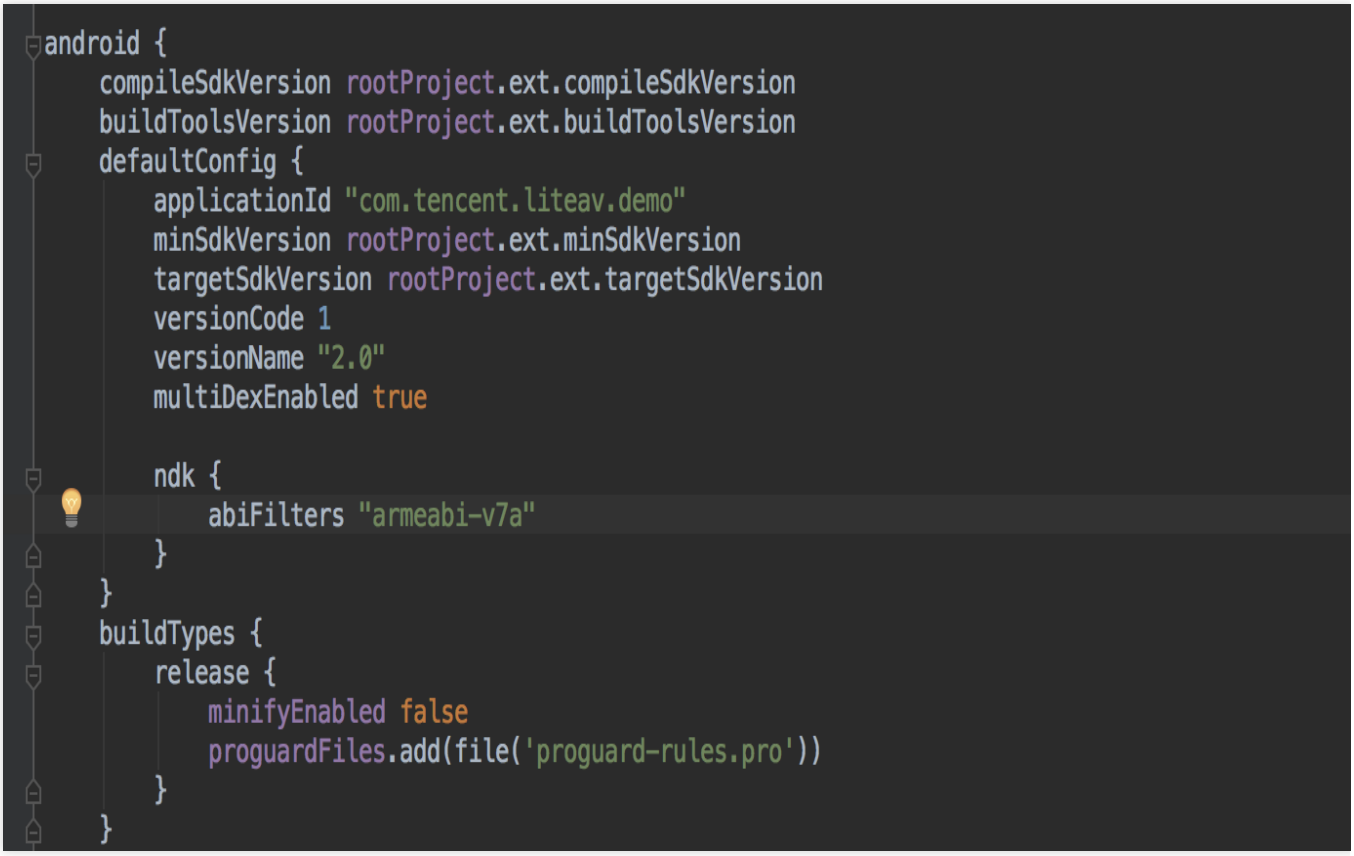
How do I reduce the size of an installation package for Android?

Method 1. Only package certain .so files

Method 2. Only package JAR files (i.e., .so files will be downloaded after installation)

If your application is used in the Chinese mainland only, you can just package the .so files for the `armeabi-v7a` architecture to reduce the increment in the installation package size to below 5 MB. If you want to offer your application on Google Play, you can package the .so files for the `armeabi-v7a` and `arm64-v8a` architectures. **Directions:** add `abiFilters "armeabi-v7a"` to `build.gradle` of the current project to specify to package the .so files in a single architecture only or add `abiFilters "armeabi-v7a","arm64-v8a"` to specify to package .so files in two architectures.

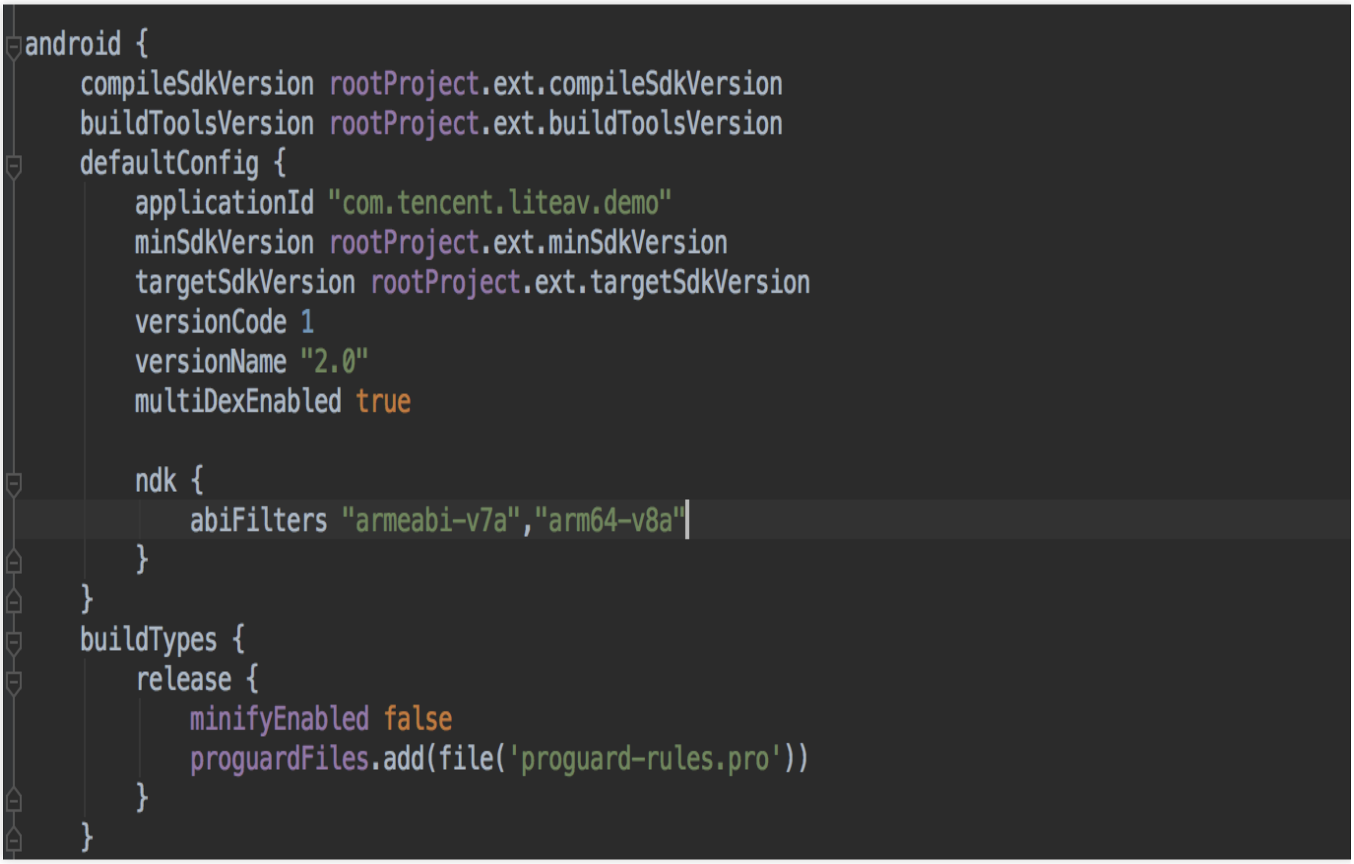
If only .so files in `armeabi-v7a` architecture are packaged (i.e., your application is not offered on Google Play):



```
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-v7a"
        }
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles.add(file('proguard-rules.pro'))
        }
    }
```

If .so files in `armeabi-v7a` and `arm64-v8a` architectures are packaged (i.e., your application is offered on Google Play):



```
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-v7a","arm64-v8a"
        }
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles.add(file('proguard-rules.pro'))
        }
    }
```

notice

If you want to offer your application on Google Play, please do not use this method, as it may cause a failure in offering the application.

The size of .so files takes the greatest proportion of the total size of the SDK for Android. If you want to reduce your installation package to below 1 MB, you can use the method of downloading .so files after installation:

1. In the folder on GitHub, click and download the package named in the format of `LiteAVSDK_TRTC_x.x.xxx.zip` , decompress it, and find the .so files for the specified architecture.
2. Upload the .so files downloaded in step 1 to your server (or Tencent Cloud COS) and record the download address such as `http://xxx.com/so_files.zip` .
3. Before an SDK feature such as video playback is started by the user, use a loading animation to prompt the user that the relevant feature module is being loaded.
 When the user is waiting, the application can download the .so files from `http://xxx.com/so_files.zip` and store the files in the application directory (such as the `files` folder in the application's root directory). To ensure that this process is not affected by ISP DNS hijacking, please verify the integrity of the .so files after download to check whether the zip package has been tampered with the ISP.
4. After all .so files are ready, call the `setLibraryPath()` API in the `TXLiveBase` class (the earliest basic module of `LiteAVSDK`) to set the target paths of the downloaded .so files to the paths in the SDK, so that the SDK can load the required .so files at those paths and start the relevant features.

Android and iOS

Last updated：2024-03-11 17:04:23

What system volume types does TRTC support on mobile devices (Android and iOS)?

It supports two system volume types: call volume and media volume.
Call volume is designed for call scenarios. It uses phones' built-in acoustic echo cancellation (AEC) technique and has lower audio quality than media volume. In the call volume mode, the volume cannot be turned to 0 with the volume buttons, but the mics of Bluetooth earphones can be used for audio capturing.
Media volume is designed for music scenarios. It has higher audio quality than call volume. In the media volume mode, the volume can be turned to 0 with the volume buttons, and if you want to enable the AEC feature, the SDK will use its built-in acoustic processing algorithm to process the audio. With media volume, only the mics on phones can be used to capture audio.

How do I set the resolution for push to 1080p in the SDK on mobile devices?

1080p is defined as `114` in `TX_Enum_Type_VideoResolution` . You can use the resolution simply by passing in the enumerated value.

Can I enter a room created on Mini Program from a mobile client?

Yes. TRTC supports communication across all platforms.

How do I use the screen sharing feature in TRTC on mobile devices?

Android: 7.2 and later versions support screen sharing. For detailed directions on how to implement the feature, see Real-Time Screen Sharing (Android).
iOS: 7.2 and latter versions support in-app screen sharing. 7.6 and later versions support both system-level and in-app screen sharing. For detailed directions on how to implement the feature, see Real-Time Screen Sharing (iOS).

Does TRTC support the 64-bit arm64-v8a architecture on Android?

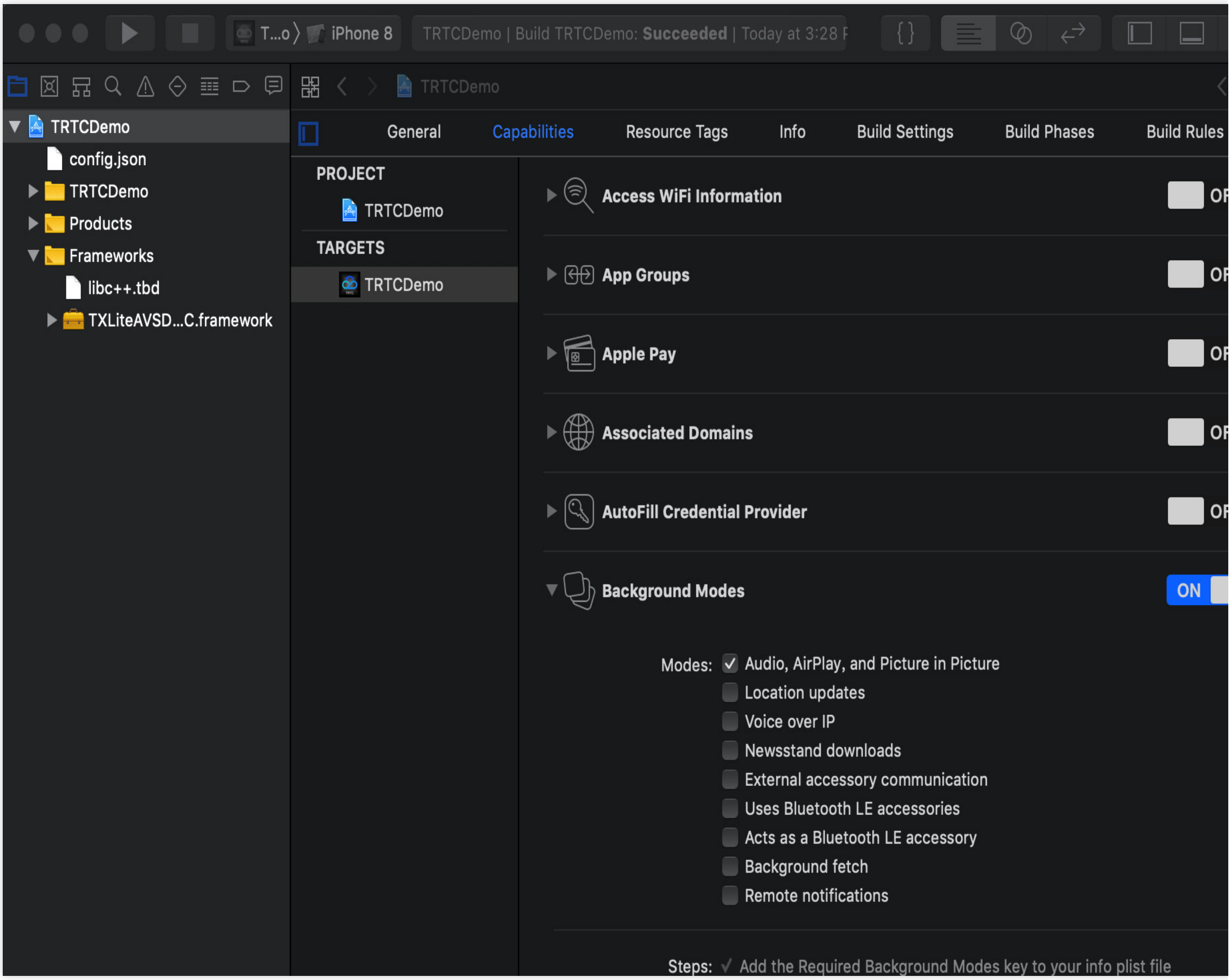
TRTC has supported the arm64-v8a ABI since v6.3.

Does TRTC support Swift integration on iOS?

Yes. Just integrate the SDK in the same steps as you do a third-party library or by following the steps in Demo Quick Start (iOS & macOS).

Can I run the TRTC SDK in the background on iOS?

Yes, you can. To run the SDK in the background, select your project, set **Background Modes** under **Capabilities** to **ON**, and select **Audio, AirPlay, and Picture in Picture**.



Access WiFi Information OFF

App Groups OFF

Apple Pay OFF

Associated Domains OFF

AutoFill Credential Provider OFF

Background Modes ON

Modes:
☑ Audio, AirPlay, and Picture in Picture
□ Location updates
□ Voice over IP
□ Newsstand downloads
□ External accessory communication
□ Uses Bluetooth LE accessories
□ Acts as a Bluetooth LE accessory
□ Background fetch
□ Remote notifications

Steps: ✓ Add the Required Background Modes key to your info plist file

Can I listen for room exit by remote users on iOS?

You can use onRemoteUserLeaveRoom to listen for the room exit event. In the `VideoCall` mode, this callback is triggered when any user leaves the room; in the `LIVE` mode, it is triggered only when an anchor leaves the room.

How to call a user whose phone screen is locked or when the application is in the background or closed on the user’s phone?

You can achieve this using the offline answering feature. For details, see Audio Call (Android).

Can Android and web users call each other?

Yes. They just need to use the same SDKAppID and enter the same room. For more information, see the documents below:

Demo Quick Start (Android)

Demo Quick Start (Web)

Can both anchors and audience members initiate co-anchoring during live streaming?

Yes. Both anchors and audience can initiate co-anchoring using the same logic. For more information, see Live Streaming Mode (Android).

In the audio/video conference scenario, can mobile users and web users enter the same room?

Yes, make sure the SDKAppID and room ID are the same while the user IDs are different.

Can I create N TRTC objects and log in to N rooms with N user IDs on the same page?

Yes. Since version 7.6, a user can enter multiple rooms.

How do I view the latest version number of the SDK?

In the case of automatic loading, `latest.release` will load the latest version automatically. You don’t need to modify the version number. For detailed directions on integration, see SDK Quick Integration.

You can find the latest version number of the SDK on the release notes page.

For iOS & Android, see Release Notes (App).

For web, see Release Notes (Web).

For Electron, see Release Notes (Electron).

Web

Last updated：2024-11-14 11:48:05

1. Basics

What browsers does the TRTC Web SDK support?

Refer to Supported Platforms.

How do I test my audio/video devices before making a call?

Please see Detect Capabilities.

How do I test my current network quality?

Please see Detect Network Quality.

I can use the TRTC web SDK in a local development environment but not after it is deployed online. What should I do?

To ensure data security and protect users' privacy, browsers allow access to the mic and camera only in secure environments, for example, when `https` , `localhost` , or `file://` is used for access. Because HTTP is not secure, browsers block access to media devices when HTTP is used.
If you can use the SDK in a local development environment but cannot capture data from the camera or mic after deploying it to the web, check whether your webpage is deployed using HTTP. If so, use HTTPS instead and make sure you have a valid HTTPS certificate.
For more information, see Domain Name and Protocol Support.

Does the web SDK support the dual-stream mode, watermarking?

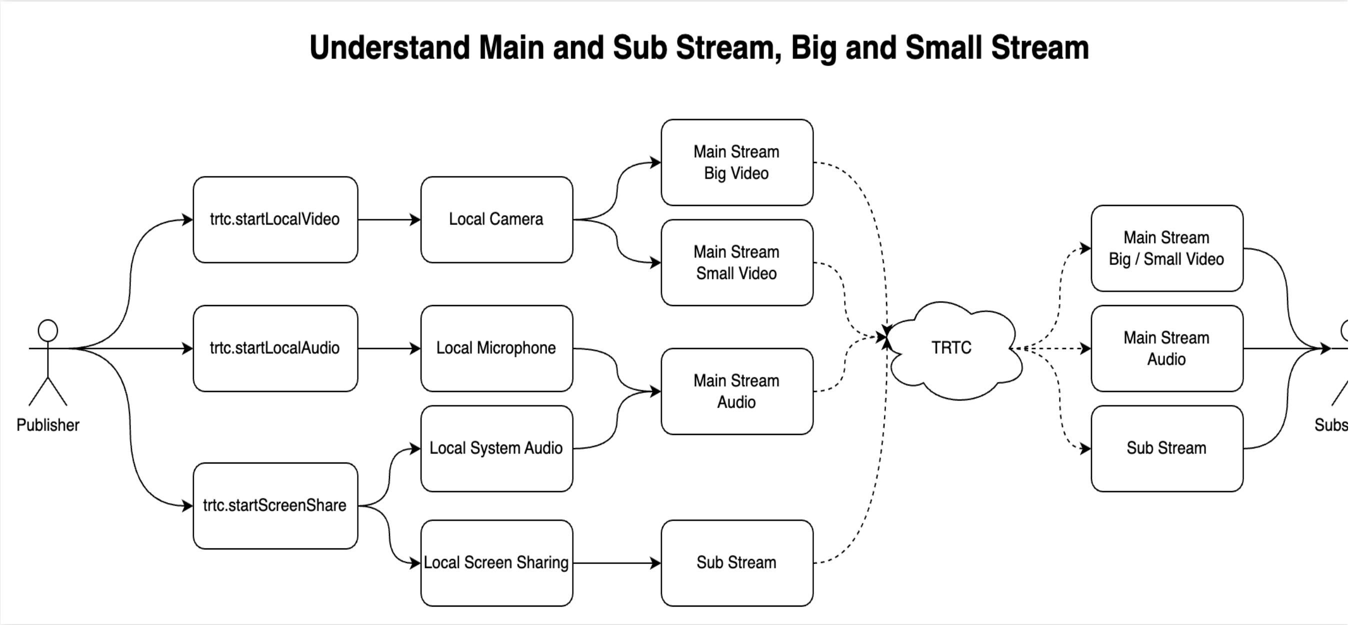
You can refer to Enable Dual-Channel Mode, and Enable Watermarking to enable the advanced features.

What are the known issues of WebRTC?

Refer to WebRTC Known Issues and Solutions.

What is the main/sub video stream?

1. TRTC has a main video stream (main stream) and an sub video stream (sub stream).
2. The camera is published through the main stream, and the screen sharing is published through the sub stream.
3. The main video stream includes: big stream and small stream. By default, TRTC.startRemoteVideo plays the big stream, and the small stream can be played through the small parameter. Refer to: Optimize Multi-Person Video Calls.



2. Capturing and Publishing

Why can I turn on the camera or microphone?

Refer to DEVICE_ERROR.

Why am I unable to publish or play streams on some mobile browsers?

Refer to Supported Platforms.

If your browser is not listed in the above document, you can open the TRTC compatibility check page with the browser to test whether it fully supports WebRTC.

Can I change the video profile(resolution, frameRate, bitrate)?

Refer to Set Encoding Profile.

Why does the camera support capturing 1080p, but the captured resolution is less than 1080p?

This situation is generally due to the fact that the camera is occupied by low-resolution ahead of time, resulting in the SDK not being able to capture the target resolution.
Solution: Check the business side logic, if the camera is captured in advance, please release the camera in time when it is no longer in use.

Can I modify the style of screen sharing pop-up window in the TRTC Web SDK?

No. The style of screen sharing is controlled by browsers and cannot be modified.

How do I capture system audio during screen sharing in the TRTC Web SDK?

Refer to Screen Sharing.

How do I switch the camera/mic?

Refer to Media Device.

How to detect camera added and removed?

Refer to Media Device.

What should I do if the error "Permission denied" occurs when I use the TRTC web SDK in iframes?

To use WebRTC in iframes, you need to add the following attribute to the iframe tag to obtain the permissions needed.
Mic, camera, and screen sharing permissions:

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

3. Playing

Why is there video but no audio during a call?

It may be because of the browser’s autoplay policy, which causes the “PLAY_NOT_ALLOWED” error. To solve the problem, please see Handle Autoplay Restriction.

Why does the sound output change from speaker to handset after microphone being captured?

On Android, there are usually multiple microphones, and the label list is: ['default', 'Speakerphone', 'Headset earpiece'].
If you do not specify the microphone in trtc.startLocalAudio, the browser default microphone may be the Headset earpiece and the sound will come out of the handset.
If you need to play out through the speaker, you need to specify the microphone with the label 'Speakerphone'.

4. Others

How to get current volume of microphone?

Refer to Audio Volume.

What triggers the trtc.on(TRTC.EVENT.KICKED_OUT) event?

The event is triggered when a user is removed from a room, for example, when the same user ID is used to log in from different devices or when the RESTful API RemoveUser is called to remove a user. For more information, see KICKED_OUT。

Note:

Repeated login is not allowed by the SDK (it may cause call exceptions) and should be avoided at the business layer.

Causion for Using Vue 3

When the Vue3 framework is used, it is necessary to use the markRaw interface to mark the trtc instance to avoid the conversion of trtc into the Proxy object by Vue, which may cause some unexpected problems.

```
import { markRaw } from 'vue';
const trtc = markRaw(TRTC.create());
```

Why mobile phone get hot when make a multi-person video call

Refer to Optimize Multi-Person Video Calls.

Flutter

Last updated：2025-03-11 17:29:40

The demo is running on two mobile phones, but why can't they display the images of each other?

Make sure that the two mobile phones use different `UserIDs` . With TRTC, you cannot use the same `UserID` on two devices simultaneously unless the `SDKAppIDs` are different.

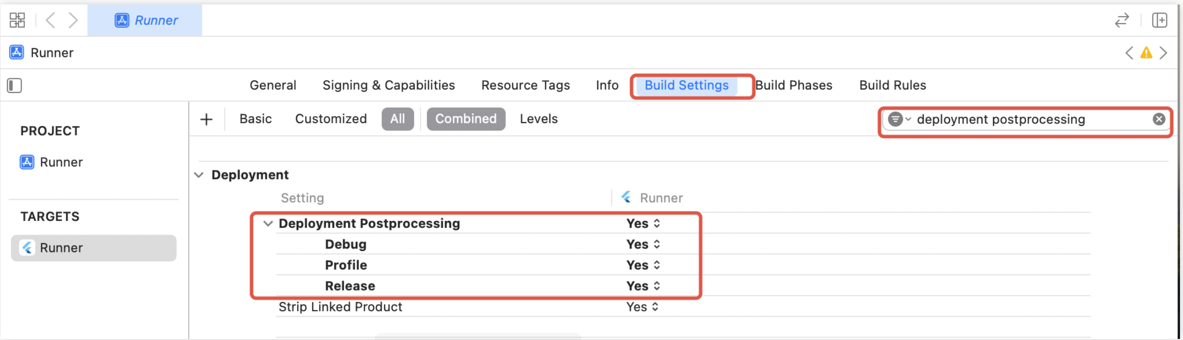
What firewall restrictions does TRTC face?

The SDK uses the UDP protocol for audio/video transmission and therefore cannot be used in office networks that block UDP. If you encounter such a problem, see How to Deal with Firewall Restrictions to troubleshoot the issue.

IOS Release Package Runtime [Symbol Not Found]?

Since Tencent_RTC_SDK calls APIs through Flutter FFI, Xcode's symbol clipping optimization during the iOS Release build may mistakenly remove TRTC's C symbols, causing a `symbol not found` error. The solutions are as follows:

1. In the project's Build Settings, find `deployment postprocessing` and set it to **Yes**.

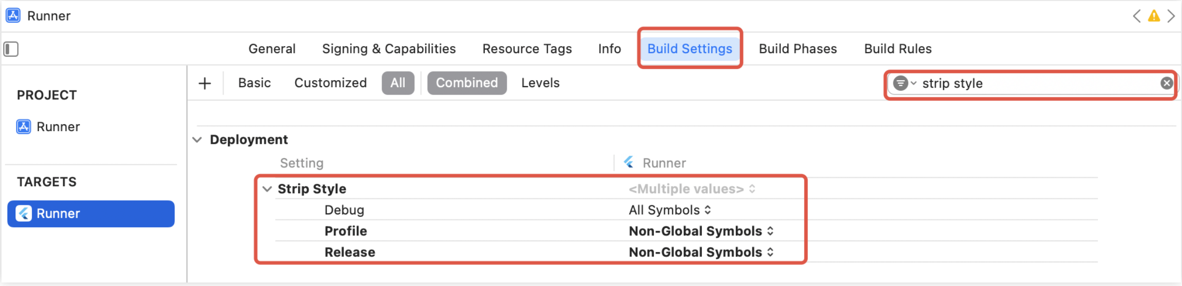


deployment postprocessing

Deployment

Setting	Runner
Deployment Postprocessing	Yes
Debug	Yes
Profile	Yes
Release	Yes
Strip Linked Product	Yes

2. In the project's Build Settings, find `strip style` and set the value for Release to **Non-Global Symbols**.



strip style

Deployment

Setting	Runner
Strip Style	<Multiple values>
Debug	All Symbols
Profile	Non-Global Symbols
Release	Non-Global Symbols

What should I do if the iOS app crashes when I build and run it?

Check if it is caused by the debug mode issue on iOS 14 and above. For details, see this Flutter document.

What should I do if videos do not show on iOS but do on Android?

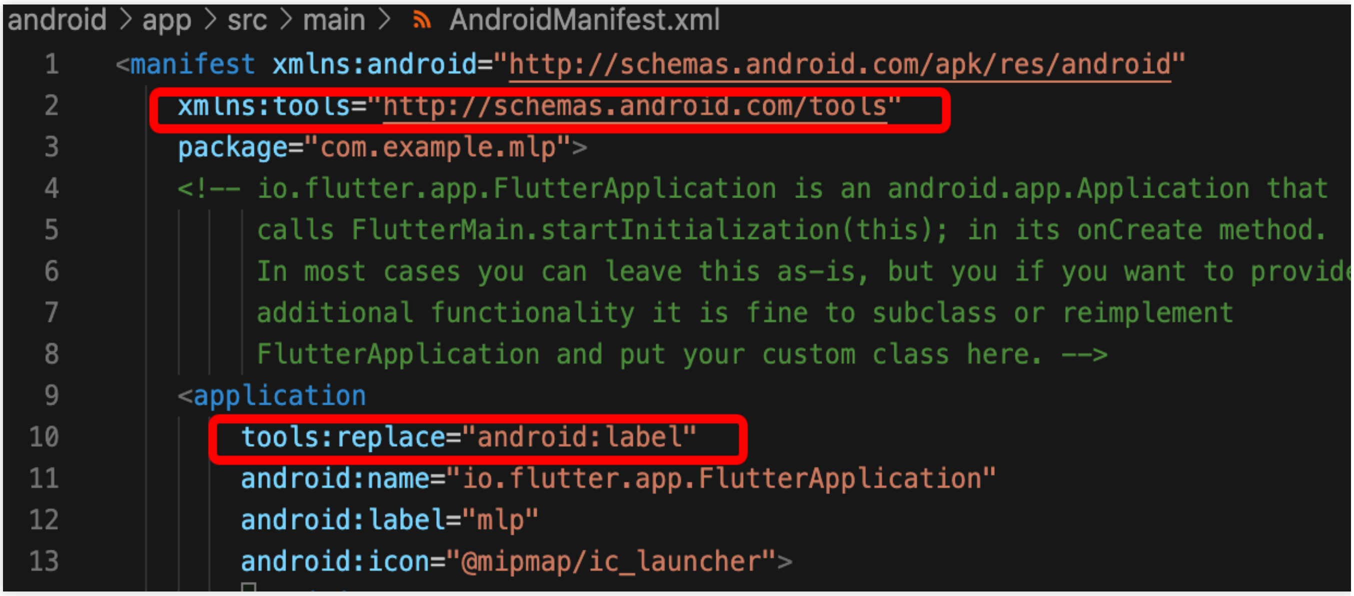
Make sure that in `info.plist` of your project, the value of `io.flutter.embedded_views_preview` is `YES` .

What should I do if an error occurs when I run CocoaPods for my iOS project after updating to the latest version of the SDK?

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

What should I do if Android Studio fails to build my project with the error “Manifest merge failed”?

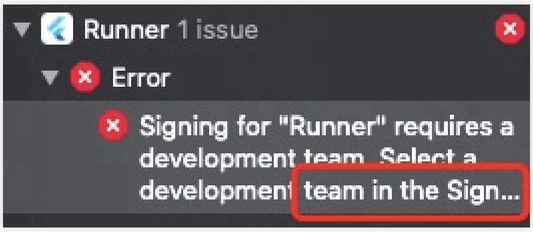
1. Open `/example/android/app/src/main/AndroidManifest.xml` .
2. Add `xmlns:tools="http://schemas.android.com/tools"` to `manifest` .
3. Add `tools:replace="android:label"` to `application` .



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.mlp">
    <!-- io.flutter.app.FlutterApplication is an android.app.Application that
         calls FlutterMain.startInitialization(this); in its onCreate method.
         In most cases you can leave this as-is, but you if you want to provide
         additional functionality it is fine to subclass or reimplement
         FlutterApplication and put your custom class here. -->
    <application
        tools:replace="android:label"
        android:name="io.flutter.app.FlutterApplication"
        android:label="mlp"
        android:icon="@mipmap/ic_launcher">
```

What should I do if an error occurs due to the absence of signatures when I debug my project on a real device?

If the error message is as shown below:



Runner 1 issue
Error
Signing for "Runner" requires a development team. Select a development team in the Sign...

1. Purchase an Apple certificate and you will be able to debug on a real device after configuration and signing.
2. Configure in `target > signing & capabilities` after purchase.

Why can't I find the corresponding file after deleting/adding content in the swift file of the plugin?

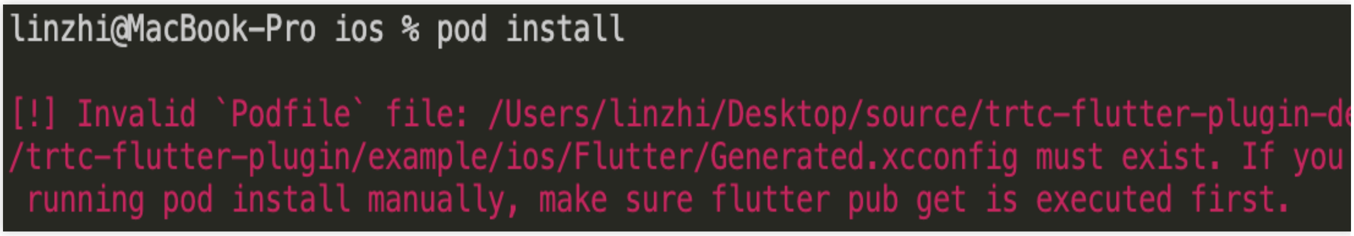
In the directory of your main project, run `pod install` in the folder of `/ios` .

What should I do if the error "Info.plist, error: No value at that key path or invalid key path: NSBonjourServices" occurs when I run my project?

Run `flutter clean` and run the project again.

What should I do if an error occurs when I run `pod install` ?

If the error message is as shown below:



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

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

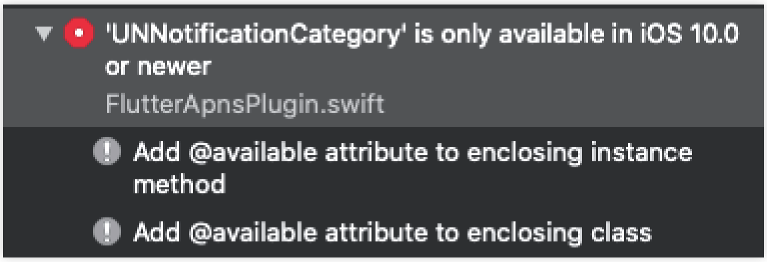
According to the message, the error is caused by the absence of `generated.xcconfig` , and to fix the problem, you **need to run flutter pub get**.

Note:

This problem occurs after compilation with Flutter. You won't run into the problem if you have a new project or have run `flutter clean` .

What should I do if a dependency error occurs when I run my project on iOS?

If the error message is as shown below:



'UNNotificationCategory' is only available in iOS 10.0 or newer
FlutterApnsPlugin.swift

Add @available attribute to enclosing instance method

Add @available attribute to enclosing class

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 capturing or rendering?

No, it doesn’t for the time being. For more information on platforms that support custom capturing and rendering, please see Custom Capturing and Rendering > Supported Platforms.

Upgrade from a version below 1.8.0 to a version 1.8.0 and above, resulting in a compilation error or unable to load the page problem fix?

If you are upgrading from below 1.8.0 to 1.8.0 and above, you need to check whether the following steps are normal. Add navigatorObservers to MateriaApp. The purpose is to navigate to a TUICallKit page when a call invitation is received. The sample code is as follows:

```
import 'package:tencent_calls_uikit/tuicall_kit.dart';

MaterialApp (
      navigatorObservers: [TUICallKit.navigatorObserver],
      ...
)
```

The imported files in the **tencent_calls_engine** plugin are uniformly replaced with new ones.

```
import 'package:tencent_calls_engine/tuicall_engine.dart';
import 'package:tencent_calls_engine/tuicall_observer.dart';
import 'package:tencent_calls_engine/tuicall_define.dart';
```

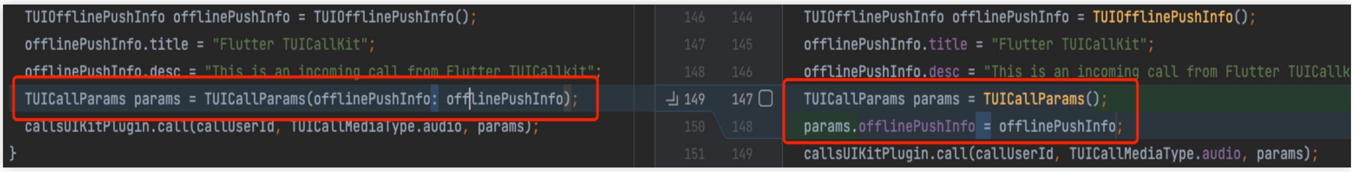
Replace with:

```
import 'package:tencent_calls_engine/tencent_calls_engine.dart';
```

The login API adjustment is more standardized, no need to specify parameters.

```
Future<TUIResult> login(int sdkAppId, String userId, String userSig)
```

Offline push parameter construction optimization.



```
TUIOfflinePushInfo offlinePushInfo = TUIOfflinePushInfo();
offlinePushInfo.title = "Flutter TUICallKit";
offlinePushInfo.desc = "This is an incoming call from Flutter TUICallkit";
TUICallParams params = TUICallParams(offlinePushInfo: offlinePushInfo);
callsUIKitPlugin.call(callUserId, TUICallMediaType.audio, params);
}
```

```
TUIOfflinePushInfo offlinePushInfo = TUIOfflinePushInfo();
offlinePushInfo.title = "Flutter TUICallKit";
offlinePushInfo.desc = "This is an incoming call from Flutter TUICallk
TUICallParams params = TUICallParams();
params.offlinePushInfo = offlinePushInfo;
callsUIKitPlugin.call(callUserId, TUICallMediaType.audio, params);
```

Electron

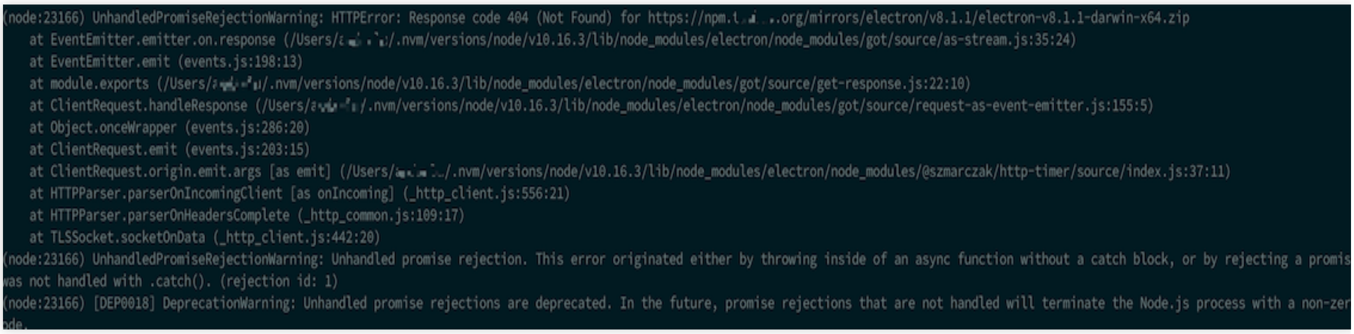
Last updated：2023-10-13 11:24:27

Installation

Is `trtc-electron-sdk` compatible with Electron v12.0.1?

Yes, it is. `trtc-electron-sdk` does not rely on the SDK of Electron and therefore does not have version requirements.

What should I do if a 404 error occurs when I download Electron?



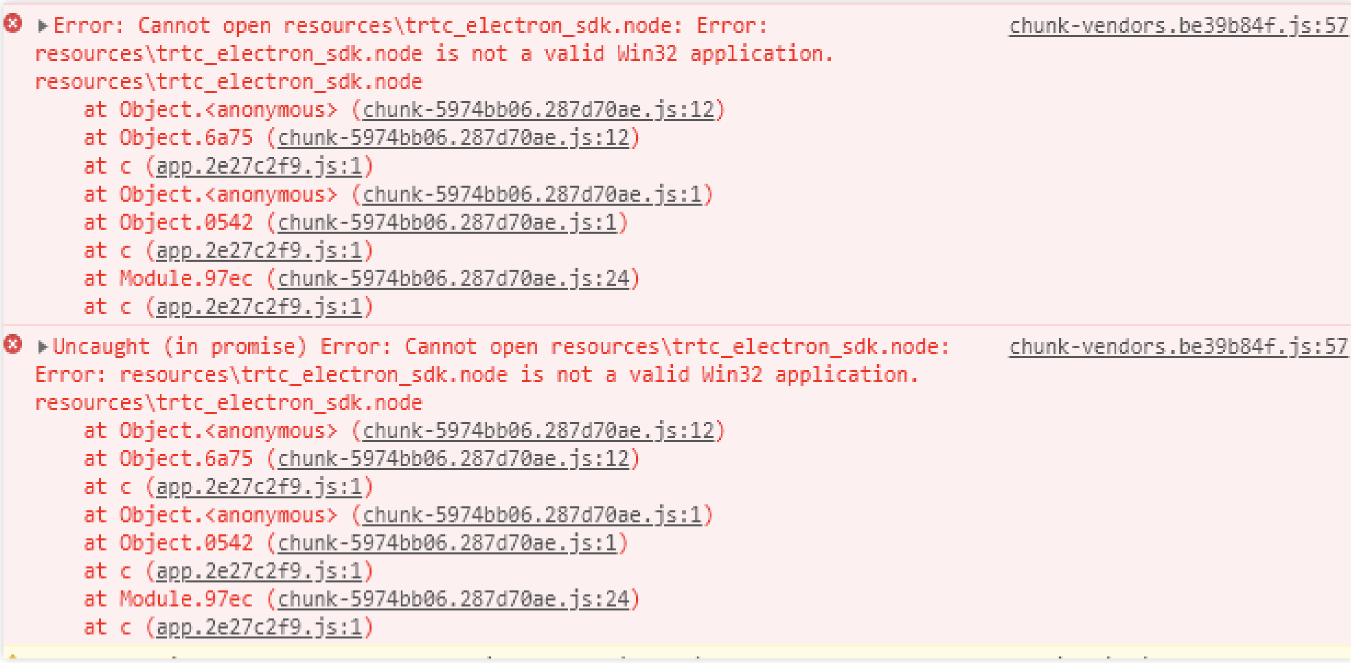
```
(node:23166) UnhandledPromiseRejectionWarning: HTTPError: Response code 404 (Not Found) for https://npm.[redacted].org/mirrors/electron/v8.1.1/electron-v8.1.1-darwin-x64.zip
    at EventEmitter.emitter.on.response (/Users/[redacted]/.nvm/versions/node/v10.16.3/lib/node_modules/electron/node_modules/got/source/as-stream.js:35:24)
    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 promis
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-zer
ode.
```

Type the following command in the terminal

```
$ npm config set electron_custom_dir 8.1.1 # Determined by the actual version number
```

Running

When I ran the SDK on 32-bit Windows, the error “Error：resource\\trtc_electron_sdk.node is not a valid Win32 application” occurred and the system said I should use 32-bit trtc_electron_sdk.node. What should I do?



```
▸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. Go to the directory of `trtc-electron-sdk` (xxx/node_modules/trtc-electron-sdk) in your project and run the following command:

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

2. Download the 32-bit `trtc_electron_sdk.node` and build your project again.

I launched the Electron demo on the VS Code terminal, and a blank screen was displayed after room entry. What should I do?

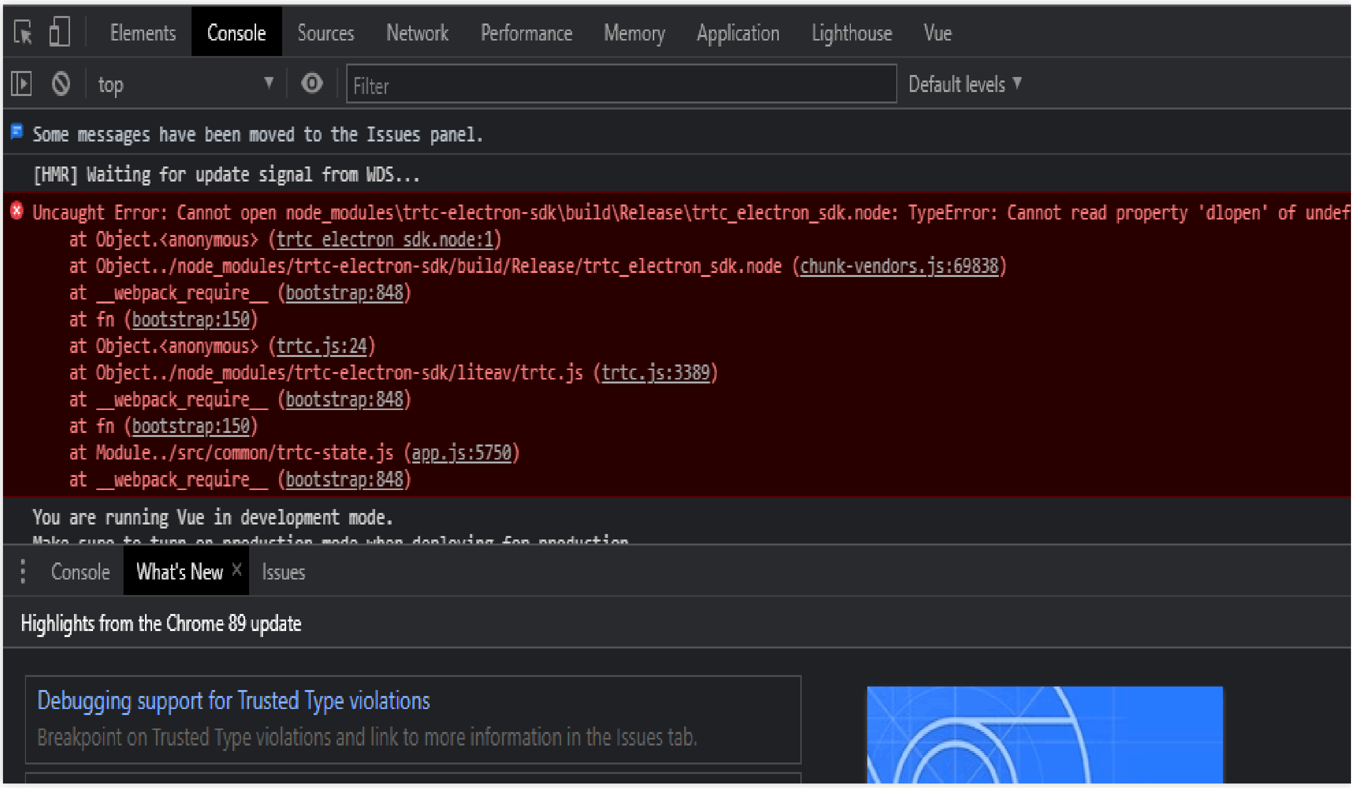
You need to request access to the camera from VS Code. Use the code below:

```
    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

    # for BigSur
    INSERT into access VALUES('kTCCServiceCamera',"com.microsoft.VSCode",0,1,1,1,NU
```

What should I do if an undefined null pointer error “cannot read property 'dlopen' of undefined” occurs when I run the demo?



```
Some messages have been moved to the Issues panel.
[HMR] Waiting for update signal from WDS...
Uncaught Error: Cannot open node_modules\trtc-electron-sdk\build\Release\trtc_electron_sdk.node: TypeError: Cannot read property 'dlopen' of undef
    at Object.<anonymous> (trtc_electron_sdk.node:1)
    at Object../node_modules/trtc-electron-sdk/build/Release/trtc_electron_sdk.node (chunk-vendors.js:69838)
    at __webpack_require__ (bootstrap:848)
    at fn (bootstrap:150)
    at Object.<anonymous> (trtc.js:24)
    at Object../node_modules/trtc-electron-sdk/liteav/trtc.js (trtc.js:3389)
    at __webpack_require__ (bootstrap:848)
    at fn (bootstrap:150)
    at Module../src/common/trtc-state.js (app.js:5750)
    at __webpack_require__ (bootstrap:848)
You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
```

Highlights from the Chrome 89 update

Debugging support for Trusted Type violations

Breakpoint on Trusted Type violations and link to more information in the Issues tab.

Solution:

Context isolation is enabled by default in Electron 12. Disable it by setting `contextIsolation` to `false` .

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

What should I do with the frequent reentry issue in the SDK for Electron?

This issue should be dealt with case by case. Possible causes include:

The client has bad network conditions (network disconnection triggers reentry).

The room entry signaling message is sent twice consecutively.

The device is overloaded, which results in decoding failure.

The same UID is used to log in on different devices.

What should I do if the error “Electron failed to install correctly” occurs in the terminal?

After download, when you run your project, the following error may occur in the terminal:

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

Follow the steps below to **manually download** Electron:

1. Run `npm config get cache` to view the cache path.
2. Manually download Electron to the cache folder.
3. Run `npm install` again.

What should I do if VS Code crashes on macOS when the SDK starts the camera and mic?

If you launch your project using the VS Code terminal, the process may crash when the SDK starts the camera and mic:

Solution A: Use an authorized terminal to run your project.

Solution B: Go to **System Preferences** > **Security & Privacy** and allow VS Code to load.

Solution C: Follow the steps below to disable SIP:

1.1 Restart your computer, **holding down** Command + R **until** it enters the recovery mode.

1.2 Open the terminal and enter `csrutil disable` .

1.3 Restart your computer in the normal mode. You can now use the VS Code terminal to start your project.

1.4 To switch SIP back on, enter `csrutil enable` in step 2.

What should I do if Electron reports the error “xx is not defined” in the console?

When you run your project, Electron may report the error “xx is not defined” in the console (`xx` is the node module):

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

Set `nodeIntegration` to `true` in the `main.js` file of Electron:

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

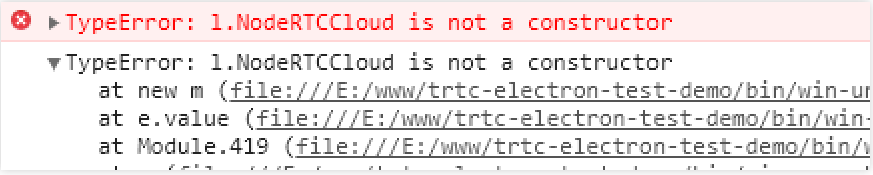
Packaging

What should I do with a .node module loading issue?

Errors

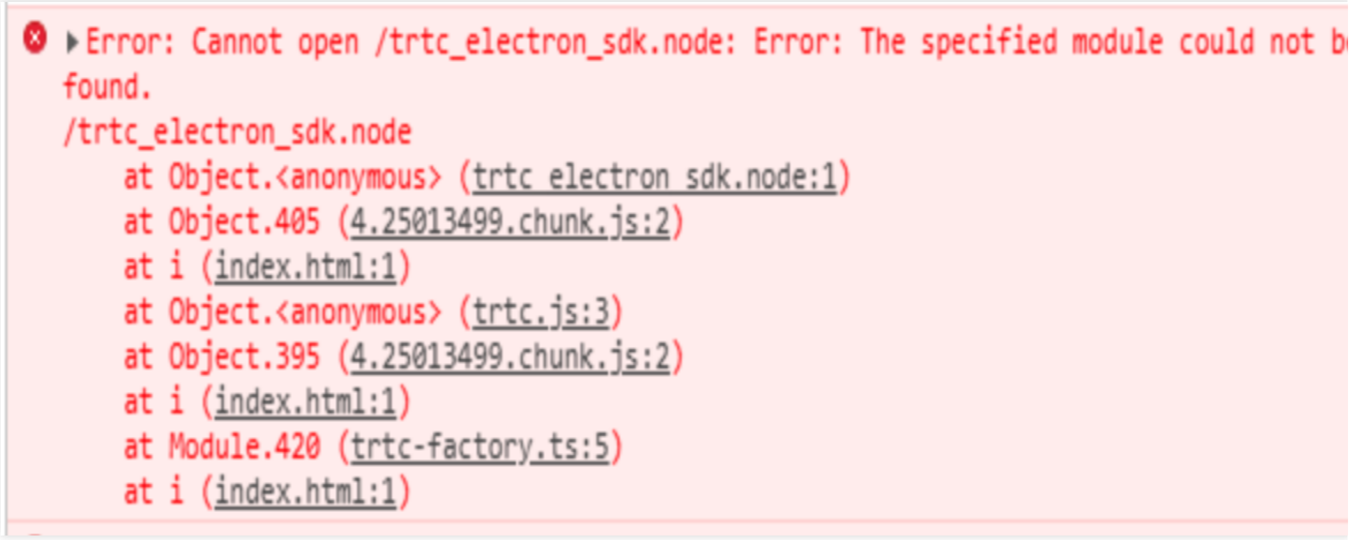
You may see one of the following error messages when running your project after compilation:

`NodeRTCCloud is not a constructor`



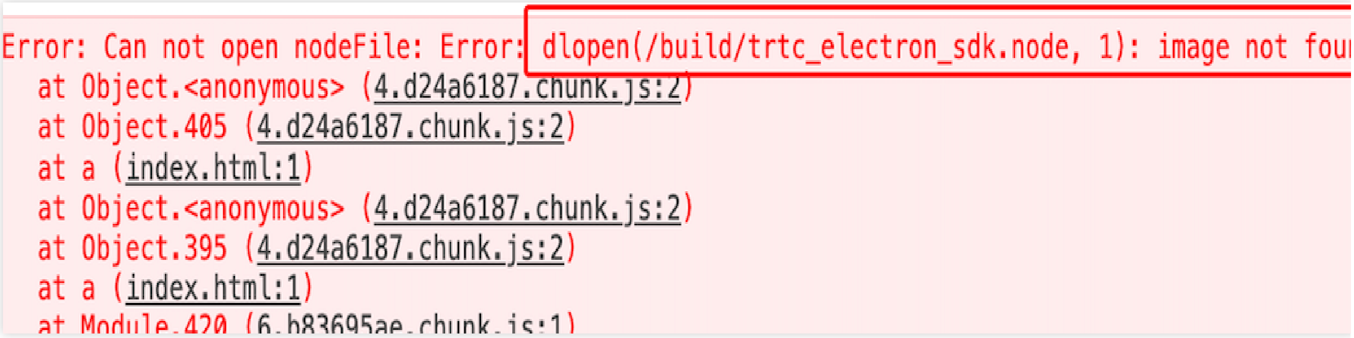
```
▶TypeError: 1.NodeRTCCloud is not a constructor
▼TypeError: 1.NodeRTCCloud is not a constructor
    at new m (file:///E:/www/trtc-electron-test-demo/bin/win-ur
    at e.value (file:///E:/www/trtc-electron-test-demo/bin/win-
    at Module.419 (file:///E:/www/trtc-electron-test-demo/bin/w
```

`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 fou
    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

The errors indicate that the `trtc_electron_sdk.node` module hasn’t been built into your project successfully. To fix the problem, follow these steps:

1. Install `native-ext-loader` .

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

2. Modify webpack configuration.

2.1 Add the following code before `module.exports` to pass the `--target_platform` command line parameter to `webpack.config.js` so that your project can be packaged correctly for its target platform.

```
const os = require('os');
// If you don’t pass `target_platform`, your project will be packaged for the current platform.
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` indicates Windows, including 32-bit and 64-bit. `darwin` indicates macOS.
 if (!['win32', 'darwin'].includes) target = os.platform();
 return target;
})();
```

2.2 Add rules:

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

notice

If you create your project with `vue-cli` , webpack configuration can be found in the `configureWebpack` option of `vue.config.js` .

If you create your project with `create-react-app` , the path to the webpack configuration file is `[Project directory]/node_modules/react-scripts/config/webpack.config.js` .

3. Add packaging and building scripts to `packages.json` .

3.1 Add packaging configuration for `electron-builder` (case sensitive):

```
"build": {
 "Omit": "...",
 "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 For projects created with `create-react-app` , add building and packaging scripts as follows:

```
"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 projects created with `vue-cli` , add building and packaging scripts as follows:

```
"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"
}
```

What should I do if I cannot find the entry point file?

When you use `electron-builder` to build a project created with `create-react-app` , the following error may occur:

```
    $ 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` indicates that the entry point file was not found.

Solution

1. Move and rename the entry point file:

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

2. Modify the `pacakge.json` file:

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

What should I do if a syntax error of the `fs-extra` module occurs during 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)
```

Update to the latest version of Node.

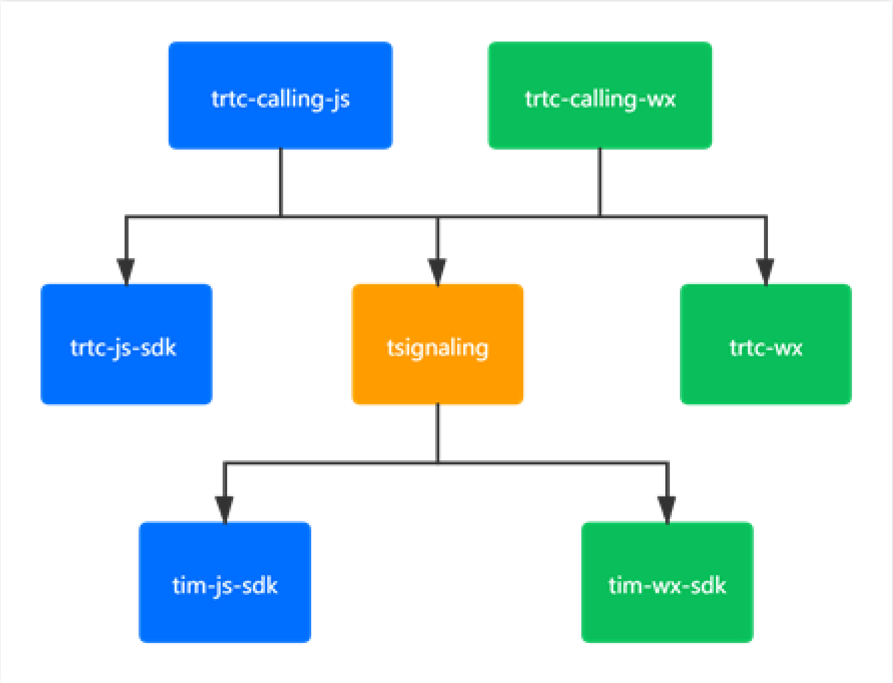
TRTCCalling for Web

Last updated：2024-03-11 17:04:23

Basics

What is TRTCCalling?

TRTCCalling is an RTC solution based on TRTC and TIM. It supports one-to-one and group audio/video calls and allows quick integration.



Does TRTCCalling support string-type `roomID` ?

`roomID` can be a string, but it must be a numeric string.

Environment

What browsers does the TRTC Web SDK support?

For details about browser support, please see Browsers Supported.

If your browser is not listed in the above document, you can open the TRTC compatibility check page with the browser to test whether it fully supports WebRTC.

How do I test my current network quality?

Please see Network Quality Check Before Calls for detailed directions.

I can run the Chat H5 demo successfully locally, but when it is deployed to the server and accessed via an IP address, I can’t make audio/video calls. What should I do?

Background: After running the Chat H5 demo locally, the user can send messages and make audio/video calls using localhost. When the project is deployed to the server and accessed via an IP address, text messages can be sent and received, the server responds to console requests properly, and the console reports no errors, but the user fails to make audio/video calls and cannot obtain video.
Cause: Chat uses the TRTCCalling SDK for audio/video calls, but the user uses HTTP for access.
Solution: Make sure the TRTCCalling SDK is accessed via HTTPS or localhost.

Integration

Why can’t I receive the `NO_RESP` event in the online demo of TRTCCalling?

Cause: The `NO_RESP` event is triggered only when two conditions are met at the same time: the call invitation times out and the callee is not online.
Solution: You will receive the event when both conditions are met.

What should I do if I can’t hear the audio of remote users when accessing TRTCCalling using WeChat’s built-in browser on iOS?

Cause: This problem is caused by the browser’s autoplay policy.
Solution: We have addressed this problem in v1.0.0. Please update TRTCCalling to v1.0.0 or a later version.

What should I do if the error “uncaught (in promise) TypeError: cannot read property 'stop' of null” occurs when I call `hangup()` of TRTCCalling?

Cause: The user calls `hangup()` multiple times in a callback, which causes the API to be triggered again before it is executed.
Solution: You only need to call `hangup()` once. The subsequent operations are performed within TRTCCalling. You only need to pay attention to operations related to your business.

On Chrome 90, `trtccalling.js` prompts “Unsupported TRTCClient. Your browser is not compatible with the application”. What should I do?

Cause: Your Chat version is too old. It does not have a compatibility check mechanism.
Solution: Update Chat to the latest version.

What should I do if the error “TypeError: Cannot read property 'getVideoTracks' of null” occurs when I make a call?

Cause: The error occurs because the callee has not granted the camera/mic access when answering the call.
Solution: Call device-related APIs (such as `startRemoteView` and `startLocalView`) asynchronously, or update TRTCCalling to v1.0.0.

What should I do if the error “TSignaling._onMessageReceived unknown bussinessID=undefined” occurs in an application (`SDKAppID`) that imports TRTCCalling via script?

Details: If the same application (`SDKAppID`) imports TRTCCalling both via script on two clients, the two clients can communicate with each other. However, if the application imports TRTCCalling via script on one client and npm on the other, or if the other client’s application imports the TRTC SDK for Android/iOS, the two clients cannot communicate with each other.
Cause: `bussinessId=undefined` indicates that the TSignaling version is too old. The signaling feature in old TSignaling versions is flawed.
Solution: Update TSignaling and make sure that **the file name of TSignaling is `tsignaling-js` during import.**

What should I do if the error “Uncaught (in promise) Error: You can call `createCustomMessage` only when the SDK is ready” occurs?

Cause: The SDK is not initialized as required.
Solution: Update TRTCCalling to v1.0.0 and call the API after receiving the `SDK_READY` callback.

What should I do if the error “Uncaught (in promise) RTCError: duplicated play() call observed, please stop() firstly <INVALID_OPERATION 0x1001>” occurs?

Cause: This error occurs if you call the `startRemoteView` API during an audio call.
Solution: Do not call `startRemoteView` during an audio call.

What should I do if the error “Uncaught (in promise) Error: inviteID is invalid or invitation has been processed” occurs?

Details: Client A, which uses TRTCCalling for web, calls client B, which uses a native application. Client B answers the call, and client A hangs up before the local camera is started and local preview is displayed. Client B remains on the call page, and the error `Uncaught (in promise) Error: inviteID is invalid or invitation has been processed` is prompted.
Cause: A user can enter a TRTC room without granting access to audio/video devices, but if the user hangs up, a user in a native application will not be notified.
Solution: TRTCCalling 1.0.0 requests camera/mic access beforehand and does not allow users to enter a room before they grant the access. We recommend you update TRTCCalling to v1.0.0 or a later version.

After a call is made, log is printed at the callee side (which indicates that the call invitation is received), but why is the `handleNewInvitationReceived` callback not received?

Cause: The TRTCCalling version is v0.6.0 or earlier, or the TSignaling version is v0.3.0 or earlier.

Solution: Update TRTCCalling and TSignaling to the latest version.

What should I do if I fail to make another call after a call is rejected?

Cause: The calling status is not reset after the call is rejected.

Solution: Update TRTCCalling to v1.0.3 or a later version.

What should I do if the error “Error: TRTCCalling.call - failed to access the user’s device” occurs?

Cause: TRTCCalling has no access to the camera/mic or the camera/mic does not exist.

Solution:

Run the TRTC support level test.

Check in Chrome Settings (chrome://settings/content) whether your Chrome has allowed the website using TRTCCalling to access the camera/mic.

Does TRTCCalling for web support receiving messages offline?

It does not support receiving messages offline.

However, it supports sending offline push notifications. You can set the message to send using offlinePushInfo in `call` or `groupCall` .

Audio and Video Quality

Last updated：2023-10-13 11:25:20

1. Video

How do I remove the black bars in a TRTC video image?

You can solve this problem by setting `TRTCVideoFillMode_Fill` (rendering mode). TRTC has two video rendering modes: fill and fit. You can set the rendering mode of the local image using `setLocalViewFillMode()` and that of a remote image using `setRemoteViewFillMode`.
TRTCVideoFillMode_Fill: The image fills the entire screen, and the excess parts are cropped. The image may not be displayed in whole.
TRTCVideoFillMode_Fit: The image is stretched as large as the long side can go, and the blank area is filled with black bars. The image is displayed in whole.

How do I fix stutter?

You can check call quality by room ID or user ID in Monitoring Dashboard in the TRTC console.
Check the send and receive statistics from the recipient's perspective.
Check the send and receive packet loss. High packet loss suggest that the stutter may be caused by unstable network connections.
Check the frame rate and CPU usage. Both low frame rates and high CPU usage can cause stutter.

How do I fix low-quality, blurry and pixelated videos?

Resolution is mainly associated with bitrate. Check whether the bitrate is set too low. Pixelation tends to occur when resolution is high but bitrate low.
With its on-cloud QoS control policy, TRTC dynamically adjusts bitrate and resolution based on network conditions. It reduces the bitrate in case of poor network connections, which leads to decreased resolution.
Check whether the `VideoCall` or `Live` mode is used during room entry. Because the `VideoCall` mode is designed for calls and features low latency and smoothness, it tends to sacrifice video quality for smoothness when network connections are poor. We recommend that you use the `Live` mode for application scenarios with high requirements on video quality.

Why are the local preview and image seen by remote users horizontally reversed?

Images captured by the local camera are mirrored by default. For native applications, you can use setLocalViewMirror to set the mirror mode of the local preview, or `setVideoEncoderMirror` to set the mirror mode of encoded images, that is, images seen by remote users and recorded by the server. On web, you can set the mirror mode by specifying the `mirror` parameter when calling createStream.

Why doesn’t my orientation setting for encoded video take effect?

You need to set `setGSensorMode()` to `TRTCGSensorMode_Disable` to disable gravity sensing; otherwise, the orientation of video seen by remote users will not change even after you call `setVideoEncoderRotation` .

Upstream data transfer is normal, but why can’t I pull data from CDNs and why can’t remote users see an image?

Make sure you have enabled relayed to CDN in Application Management > **Function Configuration**.

What should I do if preview/playback images are rotated?

If you use the TRTC SDK to capture camera data:

Update the TRTC SDK to the latest version.

If you use special devices, you can call the local preview rotation API `setLocalViewRotation` , remote image rotation API `setRemoteViewRotation` , and encoded image rotation API `setVideoEncoderRotation` to adjust the rotation. For detailed directions, see Video Rotation.

If you capture video data by yourself:

Update the TRTC SDK to the latest version.

Check the original rotation of the video you capture.

Send the video data to the TRTC SDK and check whether a rotation angle is specified for `TRTCCloudDef.TRTCVideoFrame` .

If you use special devices, you can call the local preview rotation API `setLocalViewRotation` , remote image rotation API `setRemoteViewRotation` , and encoded image rotation API `setVideoEncoderRotation` to adjust the rotation. For detailed directions, see Video Rotation.

Why are videos mirrored?

When a user uses the front camera, his or her video will be mirrored, and the local preview and video seen by remote users will be horizontally reversed.

How do I publish streams in landscape mode?

You may want to publish streams in landscape mode in certain scenarios, for example, if you use TVs for live streaming. For how to implement this, see Video Rotation.

What are the causes of black screen during live streaming?

Playback or decoding failure. Please refer to the solutions to playback failure.

Metadata issue. For example, the metadata contains only audio stream information, but there is also video in the actual data or there is only audio at the beginning but video is added later. If this is the case, we recommend you modify the metadata of the source stream.

There are only frames such as SEI but no image information in encoded video data. As a result, there are no images to decode, hence the black screen. This usually occurs with custom video data.

What are the causes of blurry or green screen during live streaming?

Missing of I-frames. Both P- and B-frames rely on I-frames to decode, so if I-frames are missing, the decoding of P- and B-frames will fail, resulting in ghosting, or blurry or green screen. Use different players such as FFplay, VLC, and PotPlayer to play the stream at the same time. If blurry or green screen occurs on all players, the problem probably lies in the source stream, and you need to check your source stream.

Change of metadata. Most players parse metadata only before decoding to configure decoding parameters. In case of video changes, for example, if the resolution changes, but the player doesn’t update the decoding parameters, the blurry or green screen issue may occur. The best solution to this issue is keeping encoding parameters unchanged during live streaming so that the metadata does not change.

Compatibility issues of hardware encoders/decoders. This problem is usually found on Android devices. The hardware encoders/decoders of some Android devices are not well implemented and has poor compatibility. If this is the case, we recommend you switch to software encoders/decoders.

Use of different color formats for publishing and playback. For example, if NV12 is used for publishing and I420 is used for playback, the issue of blurry or green screen may occur during decoding. To solve this, make sure the same color format is used for publishing and playback.

2. Audio

What should I do if the audio volume is low when I use TXVodPlayer to play videos during a TRTC call?

Call the `setSystemVolumeTyp` API to set the system volume type used during the call to `TRTCSystemVolumeTypeMedia` (media volume mode).

How do I select the media or call volume type?

You can call the `setSystemVolumeType` API to select the media or call volume type as needed.

TRTCAudioVolumeTypeAuto (default): The call volume type is used when the mic is on and the media volume type is used when the mic is off.

TRTCAudioVolumeTypeVOIP: The call volume type is always used.

TRTCAudioVolumeTypeMedia: The media volume type is always used.

What should I do if the audio volume is low?

If **the volume is low for all users**, then it is an upstream issue.

Check whether `volume` is set to lower than 50 in the setCurrentDeviceVolume API for Windows and macOS or the setAudioCaptureVolume API for all platforms. If so, set it to a larger value.

Check whether automatic gain control (AGC) is enabled.

Check whether the use of Bluetooth earphones caused the problem.

If **the volume is low for only some users**, then it is a downstream issue.

Check whether `volume` is set to lower than 50 in the setAudioPlayoutVolume or setCurrentDeviceVolume API. If so, set it to a larger value.

On mobile phones, check whether the `setAudioRoute` API was called to switch to the receiver for playback.

What should I do if audio stutters?

Open Monitoring Dashboard, go to the end-to-end details page, and select the **Audio** tab.

In **Device Status**, if the CPU usage of the receiver and sender exceeds 90%, close other processes running in the background.

If there are marked upstream and downstream packet loss and major fluctuations in RTT, it indicates poor network conditions. Change to a different network.

Why do I hear echo?

The echo issue is common if the call participants are close to each other. Please ensure a certain distance between call participants when testing. Also, check whether you have unintentionally disabled acoustic echo cancellation (AEC).

What should I do if audio is of low quality and the volume keeps changing?

This problem occurs if you use an external sound card and enable in-ear monitoring at the same time. This is because sound cards are often built in with in-ear monitoring. Please disable in-ear monitoring when you use an external sound card.

What should I do if there is echo or noise during a call or the volume of a call is small?

The issues are common if the call participants are close to each other. Please ensure a certain distance between call participants when testing. If a non-web client hears echo or noise from a web client, it indicates that 3A is not working on web. If you use the browser’s built-in API getUserMedia for custom capturing, you need to enable 3A manually using the parameters below:

`echoCancellation` : echo cancellation

`noiseSuppression` : noise suppression

`autoGainControl` : automatic gain control

If you use the TRTC.createStream API for capturing, you don't need to set the 3A parameters manually. The TRTC SDK enables 3A by default.

3. Others

How do I monitor network status and display signal strength in TRTC?

You can use `onNetworkQuality()` to monitor the current upstream and downstream network quality. To display signal strength, on Android, for example, refer to TRTC-API-Example.

Why is my camera or mic occupied?

When the `exitRoom()` API is called, logic such as the release of audio/video devices and codecs will be executed. Releasing devices is an async operation. After the release, the SDK will use the `onExitRoom()` callback in `TRTCCloudListener` to notify the upper layer. Please wait until you receive the `onExitRoom()` callback before you call `enterRoom()` again or switch to another audio/video SDK.

How do I know whether the camera is enabled successfully?

If the `onCameraDidReady` callback is received, the camera is ready.

How do I know whether the mic is enabled successfully?

If the `onMicDidReady` callback is received, the mic is ready.

What should I do if the camera fails to be turned on?

Check whether you have granted access to the camera.

The TRTC SDK supports external cameras, which are required if you use TVs or set-top boxes. Check whether the external camera is properly connected to your device.

What technical metrics does TRTC use?

notice

The following applies to iOS, macOS, Android, and Windows.

The TRTC SDK offers the `onStatistics (TRTCStatistics statics)` callback. Every 2 seconds, the callback returns statistics on technical metrics including `appCpu` (app CPU usage), `systemCpu` (system CPU usage), `rtt` (latency), `upLoss` (upstream packet loss), `downLoss` (downstream packet loss), and audio/video statistics of the local user and remote users. For details, see TRTCStatistics.

Others

Last updated：2023-10-13 11:25:51

How do live streaming, interactive live streaming, TRTC, and relayed live streaming differ from and relate to each other?

Live streaming (keywords: one-to-many, RTMP/HLS/HTTP-FLV, CDN)
Live streaming consists of the push end, the playback end, and the cloud live streaming service. Streams are pushed over the universal protocol RTMP, distributed through CDNs, and can be watched over protocols including RTMP, HTTP-FLV, or HLS (for HTML5).
Interactive live streaming (keywords: co-anchoring, cross-room communication)
In interactive live streaming, audience can co-anchor with anchors and anchors from different rooms can compete with each other.
Tencent real-time communication (keywords: multi-person interaction, UDP-based proprietary protocol, low latency)
The main capabilities of TRTC are audio/video interaction and low-latency live streaming. It uses a UDP-based proprietary protocol and can keep the latency as low as 100 ms. Typical applications include QQ calls, VooV Meeting, and online group classes. TRTC offers solutions for mainstream platforms including iOS, Android, and Windows, allows communication with WebRTC, and supports mixing streams in the cloud and relaying them to CDNs.
Relayed live streaming (keywords: on-cloud stream mixing, RTC relayed live streaming, CDN)
The relayed live streaming technology replicates multiple streams in a low-latency co-anchoring room and mixes them into one stream in the cloud before pushing it to a live streaming CDN for distribution and playback.

The demo is running on two devices, but why can't they display the images of each other?

Make sure that the two devices use different `UserID` . With TRTC, unless under different applications (`SDKAppID`), you cannot use the same `UserID` on two devices simultaneously.

When there is only one user in a room, why is CDN playback stuttering and blurry?

Please set the `TRTCAppScene` parameter in `enterRoom` to **TRTCAppSceneLIVE**.
The `VideoCall` mode is optimized for video calls, so when there is only one user in a room, TRTC tends to maintain a low bitrate and frame rate to reduce traffic usage, which makes the video choppy and blurry.

Why can't I enter any online room?

This may be because advanced permission control is enabled. If you enable advanced permission control for an application (`SDKAppID`), users must pass `PrivateMapKey` in TRTCParams ` to enter the rooms under the application. Therefore, if your business is online, and you haven't integrated into it the ` privateMapKey` logic, please do not enable the feature. For more information, see Enabling Advanced Permission Control.

How do I view TRTC logs?

TRTC logs are compressed and encrypted by default with the XLOG extension. You can use `setLogCompressEnabled` to specify whether to encrypt logs. If a log filename contains `C` (compressed), the log is compressed and encrypted; if it contains `R` (raw), the log is in plaintext.

iOS/macOS: `Documents/log` of the application sandbox

Android:

v6.7 or earlier: `/sdcard/log/tencent/liteav`

v6.8-8.5: `/sdcard/Android/data/package name/files/log/tencent/liteav/`

Later than v8.5: `/sdcard/Android/data/package name/files/log/liteav/`

Windows:

Earlier than v8.8: `%appdata%/tencent/liteav/log`

v8.8 and later: `%appdata%/liteav/log`

Web: Open the browser console or use vConsole to log printed SDK information.

Note

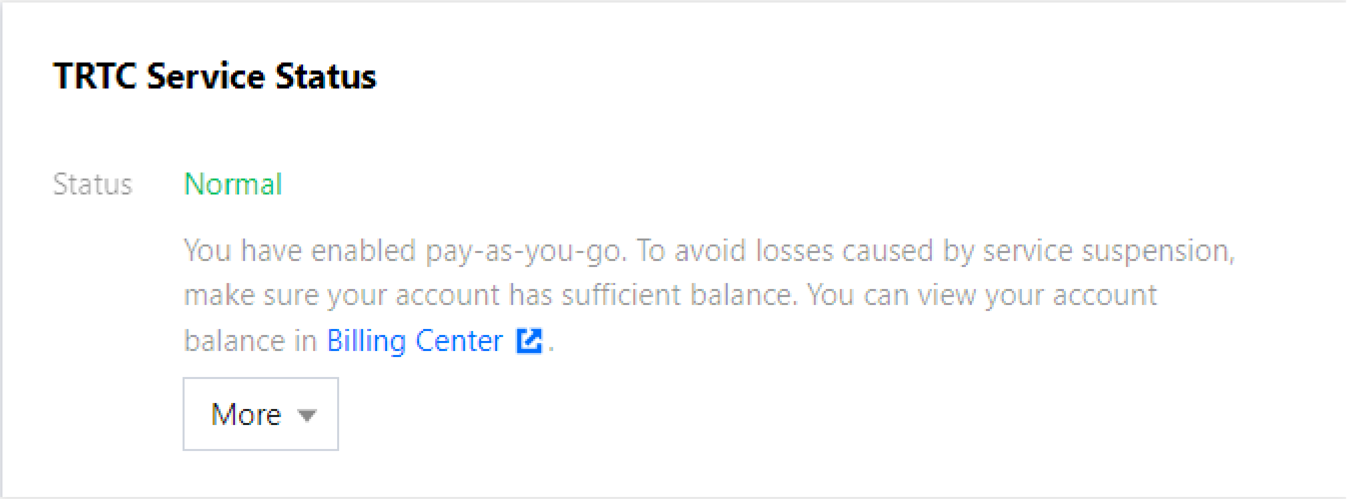
You need to download a decryption tool to view an XLOG file. Place the tool in the same directory as the XLOG file in Python 2.7 and run `python decode_mars_log_file.py` .

You can download the log decryption tool at `dldir1.qq.com/hudongzhibo/log_tool/decode_mars_log_file.py` .

What should I do if a 10006 error occurs?

If the "Join room failed result: 10006 error: service is suspended, if charge is overdue, renew it" occurs, check whether your TRTC application service is available.

Log in to the TRTC console, click Application Management, find the application you created, and click **Application Info** to view the service status.



TRTC Service Status

Status Normal

You have enabled pay-as-you-go. To avoid losses caused by service suspension, make sure your account has sufficient balance. You can view your account balance in Billing Center.

Why is the error code “-100018” returned during room entry?

This error is a result of `UserSig` verification failure, which may be caused by the following reasons:

The `SDKAppID` parameter passed in is incorrect. You can log in to the TRTC console and click Application Management to view the `SDKAppID` .

The `UserSig` passed in, which should match the `UserID` , is incorrect. To verify your `UserSig` , log in to the TRTC console and click **Development Assistance** > UserSig Generation & Verification.

How do I make a cross-room call?

You can make a cross-room call by using the `connectOtherRoom` API. Anchor A calls `connectOtherRoom()` to connect to anchor B and gets the result via the `onConnectOtherRoom` callback. All users in anchor A’s room will be notified via the `onUserEnter` callback that anchor B has entered the room, and all users in anchor B’s room will be notified via the `onUserEnter` callback that anchor A has entered the room.

Do I have to call the `exitRoom()` API?

After you call `enterRoom` , regardless of whether room entry succeeds, you must call `exitRoom` before calling `enterRoom` again; otherwise, an unexpected error will occur.

What are the formats of the recording files generated in different relayed recording scenarios?

Recording files are generated in the formats you specify in the TRTC console.

How do I know whether a stream is published successfully in a video call?

You know when you receive the `onSendFirstLocalVideoFrame` callback. After `enterRoom` and `startLocalPreview` are called successfully, the SDK will capture video from the camera and encode the video captured. It will return this callback after sending the first video frame to the cloud.

How do I know whether a stream is published successfully in an audio call?

You know when you receive the `onSendFirstLocalAudioFrame` callback. After `enterRoom` and `startLocalAudio` are called successfully, the SDK will capture audio from the mic and encode the audio captured. It will return this callback after sending the first audio frame to the cloud.

Can I query all `UserID` values?

Currently, you cannot view the statistics of all user IDs, but you can write user data into the SQL database whenever accounts are created on the client for future management and query.

Can users with the same `UserID` be in the same room at the same time?

In TRTC, users with the same `UserID` cannot be in the same room at the same time because it will cause a conflict.

Why doesn't the audio route (receiver/speaker) configured via the `setAudioRoute` API take effect?

You can switch between the receiver and speaker only in the call volume mode. That is to say, the API works only if two or more users are co-anchoring.

Can I manually enable recording for a call?

You can manually enable recording for a call in the following steps:

1. In the TRTC console, click Application Management > **Function Configuration**, enable **Relay to CDN**, and disable **On-Cloud Recording**.
2. After a user (`userid`) enters the room, splice the user’s `streamid` according to the stream ID generation rule.
3. Use the CreateRecordTask API of CSS to start a recording task for the `streamid` .
Set `DomainName` to `[bizid].livepush.myqcloud.com` .

Set `AppName` to `trtc_[sdkappid]` .

Set `StreamName` to `streamid` .

4. After the recording task is completed, CSS will save the file in VOD and notify you via the recording callback.

How does TRTC verify `UserSig` ? How do I troubleshoot the “-3319” or “-3320” error during room entry?

Log in to the TRTC console and select **Development Assistance** > UserSig Generation & Verification to verify your `UserSig` .

How do I view my call duration and usage?

You can find the information on the Usage Statistics page of the TRTC console.

How do I maintain the user list and count the number of users in a TRTC room?

If you have integrated Chat into your project, you can use the Chat 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 user count (client -> server): Whenever a user enters the room, increase the user count by 1. To achieve this, you can make the user’s client send a count increasing request to the server upon room entry.
2. Decrease the user count (client -> server): When a user leaves a room, decrease the user count by 1. To achieve this, you can make the user’s client send a count decreasing request to the server during room exit.

A “-100013” error is reported during room entry, with the error message “ERR_SERVER_INFO_SERVICE_SUSPENDED”. What should I do?

This error indicates that the service is unavailable. Please check the following:

Whether you have used up your package

Whether your Tencent Cloud account has overdue payment

What should I do if I have enabled on-cloud recording but no recording files are generated?

1. Make sure you have enabled **Relay to CDN** and **On-Cloud Recording** in the TRTC console.
2. TRTC starts recording only if there is a user publishing audio/video data in a room.
3. A recording file will be generated only if CDN pull is successful.

4. If there is only audio in a room at first before video is published, depending on the recording template configured, the recording file generated may contain only the video segment or the audio-only segment.

How do I give the room ID to the co-anchor I invite?

You can insert the room ID in a custom message. The invitee can get the room ID after parsing the message. For details, see Message Sending and Receiving and TIMMsgSendNewMsg.

Is it possible to start audio recording only when there are two or more users in a room?

Yes, it is. If you want to record mixed audio data, call On-Cloud MixTranscoding first, specifying the output stream ID, and call the CreateRecordTask API of CSS.

How do I capture the audio of a shared application on Windows?

You can call startSystemAudioLoopback to enable system audio capturing.

How to implement the feature that allows anchors to invite audience members to co-anchor in conference scenarios on Windows?

You need to use another Tencent Cloud product, Chat, to implement the feature.
This is how it works: A sends a custom message X (you can determine how the message is displayed) to B, and the calling page is shown; B receives X and the called page is shown; B uses enterRoom to enter the room and sends a custom message X1 to A; A receives X1 (you can determine whether to display the message) and uses `enterRoom` to enter the room. The messages are sent via Chat.

How do audience members watch the videos of co-anchors in a room?

In live streaming scenarios, audience get the `userid` of anchors in a room via the onUserVideoAvailable callback in `TRTCCloudDelegate` (co-anchoring users enter the room by calling enterRoom and are also anchors for the audience). They then call startRemoteView to play the videos of the anchors.
For more information, see Live Streaming Mode > Windows.

Is there a TRTC SDK for Linux?

The TRTC Linux SDK is not commercially available yet. If you have questions about the SDK or want to use it, please contact us at colleenyu@tencent.com.

Does TRTC support screen sharing during video calls or interactive live streaming?

Yes, it does. During video calls or interactive live streaming, the video captured by the camera is published as the primary stream. You can also publish the screen as the substream. The shared screen will contain the video call or interactive live streaming window.