Tencent Real-Time Communication

SDK Download

Product Documentation

Contents

SDK Download
- SDK Download
- Release Notes (App)
- Release Notes (Desktop Browser)
- Release Notes (Electron)

SDK Download

SDK Download

Last updated : 2021-08-06 22:33:14

Lite Edition (TRTC)

The lite edition only includes TRTC and TXLivePlayer and has the smallest storage footprint. It is suitable for users focusing on TRTC-related features.

Environment	ZIP File	GitHub	Gitee	How to Run Demo	SDK Integration Guide	Impact on Installation Package Size
iOS	DOWNLOAD	GitHub	Gitee	DOC	DOC	2.9 MB (arm64)
Android	DOWNLOAD	GitHub	Gitee	DOC	DOC	jar: 820 KB so (armv7): 4.8 MB so (arm64): 5.7 MB
Windows (C++)	DOWNLOAD	GitHub	Gitee	DOC	DOC	12.7 MB (C++ x86) 15.6 MB (C++ x64)
Windows (C#)	DOWNLOAD	GitHub	Gitee	DOC	DOC	13.8 MB (C# x64) 13.3 MB (C# x86)
Unity	DOWNLOAD	Github	N/A	DOC	DOC	N/A
Mac	DOWNLOAD	GitHub	Gitee	DOC	DOC	2.05 MB (arm64)
web	DOWNLOAD	GitHub	Gitee	DOC	-	N/A
Electron	DOWNLOAD	GitHub	Gitee	DOC	DOC	N/A
WeChat Mini Program	DOWNLOAD	GitHub	Gitee	-	-	N/A

Note :
Read How to Downsize Installation Package to learn how to reduce the storage footprint of the SDK.

Professional Edition
In addition to TRTC, the professional edition incorporates various audio/video-related features including Superplayer (Player+), MLVB, and UGSV. Because the core modules are highly reusable, the professional edition has a lower storage footprint than the two individual SDKs combined and also avoids symbol duplicate.

Environment	ZIP File	GitHub	64-bit Support	How to Run Demo	SDK Integration Guide	Impact on Installation Package Size
iOS	DOWNLOAD	GitHub	Yes	DOC	DOC	4.08 MB (arm64)
Android	DOWNLOAD	GitHub	Yes	DOC	DOC	jar: 1.5 MB so (armeabi): 6.5 MB so (armv7): 6.1 MB so (arm64): 7.3 MB

- For Windows and Mac, only one unified edition of the SDK is available.
- Since LiteAV SDKs all use the same core modules, symbol duplicate can occur if your project integrates more than two LiteAV SDKs. Using the professional edition of LiteAV SDK can avoid this problem.
- Read How to Downsize Installation Package to learn how to reduce the storage footprint of the SDK.

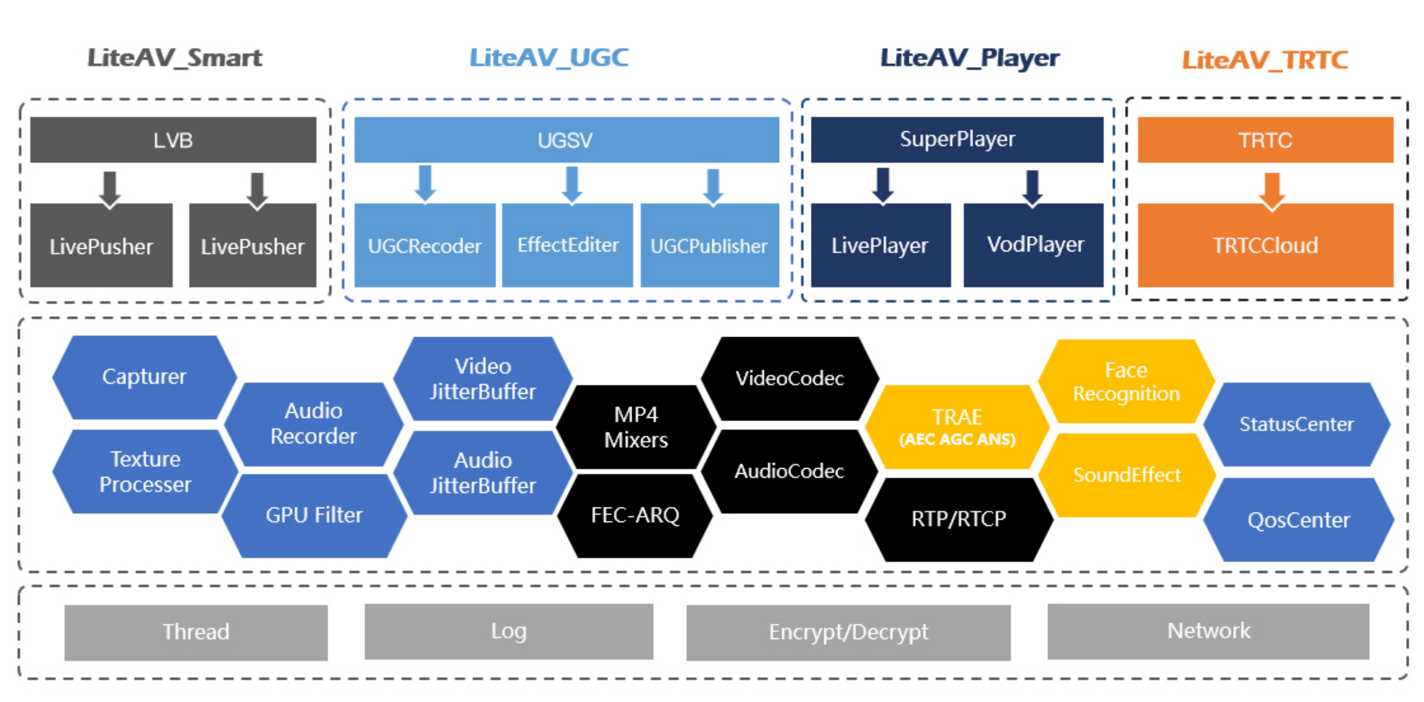
Enterprise Edition

In addition to all the features of the professional edition, the enterprise edition includes a set of AI beauty special effect components such as eye enlarging, face slimming, beauty filter, and dynamic stickers.

Environment	ZIP File	64-bit Support	How to Run Demo	SDK Integration Guide	Impact on Installation Package Size
iOS	DOWNLOAD	Yes	DOC	DOC	6.15 MB (arm64)
Android	DOWNLOAD	Yes	DOC	DOC	jar: 2.3 MB so (armeabi): 20.4 MB

- For Windows and Mac, no SDK with AI beauty special effect components is available and only one unified edition is provided.
- Read How to Downsize Installation Package to learn how to reduce the storage footprint of the SDK.

Differences Between Editions



Module	Features	LiteAV_Smart	LiteAV_UGC	LiteAV_TRTC	LiteAV_Player
Live Push	Camera Push	✓	-	-	-

	Screencapturing Push	✓	-	-	-
LVB Playback	RTMP Protocol	✓	-	-	✓
	HTTP-FLV	✓	-	✓	✓
	HLS (m3u8)	-	-	-	✓
VOD Playback	MP4 Format	-	-	-	✓
	HLS (m3u8)	-	-	-	✓
	DRM	-	-	-	✓
Beauty Filter	Basic Beauty	✓	✓	✓	-
	Basic Filter	✓	✓	✓	-
LVB Co-anchoring	Co-anchoring	✓	-	✓	-
	Cross-room competition	✓	-	✓	-
Video Call	Two-person Call	-	-	✓	-
	Video Conference	-	-	✓	-
UGSV	Recording/Shooting	-	✓	-	-
	Clipping/Splicing	-	✓	-	-
	TikTok-like Special Effects	-	✓	-	-
	Video Upload	-	✓	-	-
AI Beauty Special Effects	Eye Enlarging/Face Slimming	-	-	-	-
	Face V-shaping/Nose Beautifying	-	-	-	-
	Dynamic Stickers	-	-	-	-
	Green Screen Keying	-	-	-	-

Release Notes (App)

Last updated : 2021-08-13 16:08:56

Version 8.7 Released on May 25, 2021

New features

- All platforms: supported anomaly detection for peripheral audio devices. After registering `onStatistics` , you can detect in real time when there is no audio for a long time and when audio cracks or is interrupted via the `audioCaptureState` field in TRTCLocalStatistics.
- Windows: supported RGBA video data for custom capturing.

Quality improvement

- All platforms: optimized background music management to release memory resources in a timely manner.
- All platforms: ensured that audience receive the onUserVideoAvailable(false) notification in a timely manner after stream publishing is paused because the application is switched to the background.
- macOS: reduced the CPU and memory usage of mouse cursor capturing during screen sharing.

Bug fixing

- Android: fixed the issue where `setRemoteViewFillMode` doesn't work on some devices.
- iOS & macOS: fixed the memory release issue of custom beauty filters after they are disabled.

Version 8.6 Released on May 8, 2021

- All platforms: optimized the QoS control algorithm, enhancing audio/video transmission quality.
- All platforms: improved audio playback smoothness when users switch between the anchor and audience roles.
- iOS & macOS & Windows: optimized the audio processing module, improving audio quality in the speech and default modes.
- iOS & macOS: improved the adaptability of custom audio capturing to situations of high CPU usage.
- iOS & Android: supported publishing screen recording data via the substream, as in SDKs for desktop platforms.

- macOS: added native support for Apple M1.
- Windows: optimized the memory allocation logic to enhance stability.

Version 8.5 Released on March 24, 2021

New features

- macOS: optimized the screen sharing feature. You can now share other windows along with the target window. For details, see the API addIncludedShareWindow.
- All platforms: supported publishing VOD content. You can now bind TXVodPlayer with `TRTCCloud` and publish the content played by VOD via TRTC's substream.
- All platforms: supported custom capturing of substream data. For details, see the API sendCustomVideoData.
- All platforms: supported custom audio mixing. You can feed a local audio track into the SDK's audio processing. The SDK will mix the two tracks before publishing. For details, see the API mixExternalAudioFrame.
- All platforms: supported mixing only video streams, allowing for more flexible stream mixing control.

Quality improvement

- macOS: enabled the `startSystemAudioLoopback` API to support dual sound channels.
- Windows: supported automatic switch to the slideshow window when a slideshow is selected for screen sharing.
- All platforms: included end-to-end latency in the information returned via the status callback.

Bug fixing

- iOS: fixed occasional crash when images are rendered using OpenGL in the background.
- iOS: fixed playback failure when shared images are static.

Version 8.4 Released on February 8, 2021

New features

- macOS: supported capturing system audio, i.e., the system loopback feature that is enabled on Windows. The feature allows the SDK to capture system audio so that anchors can stream local audio or video files to other users.

- macOS: supported local preview for screen sharing. You can now display screen sharing preview in a small window.
- Windows: supported setting the volume of the current process. You can now use setApplicationPlayVolume to set the volume of the volume mixer.
- All platforms: supported local recording. An anchor can now record local audio and video into an MP4 file during streaming. For details, see startLocalRecording.

Quality improvement

- All platforms: improved audio quality in the music mode, which makes it more suitable for Clubhouse-like audio streaming scenarios.
- All platforms: improved the adaptability to poor network conditions. Smooth audio and video can be delivered even when the packet loss rate reaches 70%.
- Windows: improved audio quality in some streaming scenarios by significantly reducing audio damage.
- Windows: improved performance by 20%-30% in some scenarios.

Bug fixing

- Windows: fixed crash when the desktop is shared on Windows Server 2019 Datacenter x64.
- Windows: fixed the issue where screen sharing sometimes ends unexpectedly when the target window is resized during screen sharing.
- Windows: fixed image capturing failure with some cameras.
- iOS: fixed the issue where `snapvideoshot` causes stuttering with CAAnimation.
- iOS & macOS: fixed the black screen issue when the same view is used to display camera and screen sharing images in turn.
- iOS: fixed the blurry screen issue on iPhone 6S when a third-party beauty filter component is used.
- iOS: fixed the issue where, in cases where TRTC and VOD are used at the same time, the SDK occasionally crashes after VOD is stopped.
- Android: fixed the issue where audio is played via the speaker after a user using Bluetooth earphone rejects an incoming call.

Version 8.3 Released on January 15, 2021

New features

Optimized the business logic of custom capturing:

- iOS & Android & macOS: optimized the audio module to ensure acoustic echo cancellation (AEC) and active noise suppression (ANS) effects when you use enableCustomAudioCapture to capture

audio data and send it to the SDK for processing.
- iOS & Android: if you need to add your own audio effects and audio processing logic in addition to those of the TRTC SDK, we recommend you use version 8.3, with which you can use setCapturedRawAudioFrameDelegateFormat and other APIs to set what to include in the audio data callback, for example, the audio sample rate, the number of sound channels, and the number of samples, so that you can process audio data in your preferred format.
- All platforms: if you collect video data by yourself and use the audio module of the TRTC SDK at the same time, lip-sync errors may occur. This is because the SDK has its own timeline control logic. To solve this problem, we have provided the generateCustomPTS API. When a video image frame is captured, call this API and record the PTS (timestamp), and provide the timestamp when you call sendCustomVideoData.
- Windows: supported SOCKS5 proxy addresses for domain names.

Bug fixing

- All platforms: fixed occasional lip-sync errors for recorded content due to timestamp exceptions in audio data.
- Windows: improved the compatibility of window sharing with high DPI displays.
- Windows: added minimized windows to the shareable window list. The thumbnails of minimized windows are their application icons.
- Windows: fixed the issue of unnecessary CPU usage by DXGI after the SDK is started.
- iOS: fixed the ANR error caused by manual focus setting.
- iOS: fixed occasional failure to switch between the front and rear cameras.
- iOS: fixed VODPlayer crash when video is played back in slow motion.
- iOS: fixed the issue where audio is occasionally played via the speaker after room entry.
- iOS & Android: optimized the AEC and ANS effects and supported reverb effects for in-ear monitoring.
- Android: fixed occasional green or blurry screen during hardware decoding.
- macOS: fixed the issue where, during screen sharing with the highlighting feature enabled, the highlighted borders of the shared window flash when the window is moved near the edge of the screen.
- macOS: fixed black screen when the view rendered is moved.

Version 8.2 Released on December 23, 2020

New features

- iOS & Android: supported callback of a combination of locally captured audio and all played back audio, making local recording easier.
- Android: the video rendering component TXCloudVideoView supported using `TextureView` for local rendering through the calling of the `addVideoView(new TextureView(getApplicationContext()))` API.
- Android: supported RGBA video data for custom rendering.
- Windows: supported taking screenshots of locally captured video and played back remote video. For details, see ITRTCCloud.snapshotVideo.
- Windows: supported using `addExcludedShareWindow` and `addIncludedShareWindow` to exclude or include windows you specify, increasing the flexibility of screen sharing.
- macOS & iOS: supported calling `TRTCCloud.snapshotVideo` to take screenshots of video in the custom rendering mode.

Quality improvement

- Android: improved encoding quality for live streaming, enabling clearer video images.
- Windows: improved the AEC algorithm.

Bug fixing

- iOS: fixed occasional audio playback errors when VODPlayer and TRTC are used at the same time.
- Android: fixed black screen when custom beauty filters are used.
- Windows: fixed occasional failure to exit the current process.

Version 8.1 Released on December 3, 2020

New features

- All platforms: added statistics on remote video stuttering to `onStatistics`.
- All platforms: supported using the volume adjustment API `setAudioPlayoutVolume` (100-150) to enable audio gain.
- iOS & Android: added the `setLocalVideoProcessListener` API to better support the integration of third-party beauty filters.
- C#: upgraded to the latest APIs.

Quality improvement

- All platforms: optimized the audio processing algorithm when earphones are used to deliver better audio quality.
- Android: optimized the audio pre-processing algorithm to reduce the impact of AEC, ANS, and AGC on audio quality.

Bug fixing

- iOS: fixed occasional crash when the application is force killed.
- Android: fixed the issue where beauty filters do not produce desired results when the frame rate of captured video is high.
- Windows: fixed occasional crash during screen sharing when high DPI displays are used.

Version 8.0 Released on November 13, 2020

New features

- All platforms: added cross-platform C++ APIs. For more information, see cpp_interface/ITRTCCloud.h.
- All platforms: supported string-type room IDs. For more information, see `TRTCParams.strRoomId` .
- All platforms: added the device management class `TXDeviceManager` .
- All platforms: added the `TRTCCloud.switchRoom` API to allow room switching with capturing uninterrupted.
- All platforms: added the `TRTCCloud.startRemoteView` API to start rendering remote video images.
- All platforms: added the `TRTCCloud.stopRemoteView` API to stop rendering remote video images.
- All platforms: added the `TRTCCloud.getDeviceManager` API to get the device management class.
- All platforms: added the `TRTCCloud.startLocalAudio` API to enable local audio capturing and upstream data transfer.
- All platforms: added the `TRTCCloud.setRemoteRenderParams` API to configure rendering for remote images.
- All platforms: added the `TRTCCloud.setLocalRenderParams` API to configure rendering for local images.

Optimization

- Android: optimized the logic for switching between software and hardware decoding.
- Windows: improved audio quality and AEC for system loopback.
- Windows: optimized the audio device selection logic to reduce cases of no audio.
- Windows: reduced audio loss in double-talk scenarios.
- All platforms: optimized instant streaming for role switching in the manual subscription mode.
- All platforms: optimized the audio receiving logic, improving audio quality.
- All platforms: improved the reliability of `sendCustomCmdMsg` .

Bug fixing

- iOS: fixed the issue where the calling of `muteLocalVideo` suspends local video rendering.

- iOS: fixed the issue where the application occasionally freezes when a component is called during foreground-background switch.
- iOS: fixed intermittent in-ear monitoring audio when audio effects are enabled.
- Android: fixed the issue where audio effects played in the call volume mode do not stop when there is an incoming call.
- Android: fixed occasional failure to enable audio capturing.
- Windows: fixed occasional black screen during local video rendering.
- Windows: fixed occasional crash when users exit the app.
- Windows: improved support for Bluetooth earphones and fixed the no audio issue.
- Windows: fixed the focus stealing problem that occurs when screen sharing stops.
- All platforms: fixed failure to collect statistics on packet loss rate for status callback.

Version 7.9 Released on October 27, 2020

New features

- macOS: supported filtering out selected windows from screen sharing. Users can exclude windows they do not want to share, better ensuring privacy.
- Windows: supported configuring the border color and width of the "Sharing" message box during screen sharing.
- Windows: supported the high performance mode during desktop sharing.
- All platforms: supported custom encryption, allowing users to process encoded audio/video data using an exposed C API.
- All platforms: added audio stuttering information `audioTotalBlockTime` and `audioBlockRate` to `TRTCRemoteStatistics` .

Optimization

- iOS: shortened the startup time of the audio module, allowing quicker capturing and sending of the first audio frame.
- Windows: optimized the AEC algorithm for system audio loopback.
- Windows: allowed users to filter out certain windows from screen sharing to prevent the target window from being covered.
- Android: optimized the in-ear monitoring effect for most Android devices, reducing in-ear monitoring latency to a more acceptable level.
- Android: reduced end-to-end delay in the music mode (specified in `startLocalAudio`).
- All platforms: enhanced audio smoothness when a user switches between the anchor and audience role in the manual subscription mode.

- All platforms: improved audio/video call performance and audio smoothness in poor network conditions.

Bug fixing

- iOS: fixed occasional failure to render video images in certain scenarios.
- iOS: fixed occasional noise when users use earphones in the default mode.
- iOS: fixed known memory leak issues.
- iOS: fixed occasional crash after ReplayKit screen recording ends.
- iOS: fixed compilation problems on simulators.
- Android: fixed occasional lip-sync errors on certain phones after the application remains in the background for a long time.
- Android: fixed the issue where the mic is not released after the application is switched to the background.
- Android: fixed the issue where certain OpenGL resources in the SDK are not released in time.
- Windows: fixed occasional noise in some scenarios.
- All platforms: fixed occasional crash, improving the SDK’s performance stability.

Version 7.8 Released on September 29, 2020

New features

- Mac: added the callback of system volume change. For details, please see TRTCCloudDelegate.onAudioDevicePlayoutVolumeChanged.
- Windows: supported specifying content for screen sharing across screens.
- Windows: supported filtering out specified windows from screen sharing to prevent the target window from being covered. For more information, please see TRTCCloud.addExcludedShareWindow and TRTCCloud.removeExcludedShareWindow.
- Windows: added the callback of system volume change. For details, please see ITRTCCloudCallback.onAudioDevicePlayoutVolumeChanged.

Optimization

- iOS: allowed using VODPlayer and TRTC at the same time with AEC enabled.
- iOS & macOS: supported pushing a specified image when stream pushing pauses. For more information, please see TRTCCloud.setVideoMuteImage.
- Android: supported pushing a specified image when stream pushing pauses. For more information, please see TRTCCloud.setVideoMuteImage.

- Android: optimized the audio routing policy to make sure that audio is always played back via earphones when earphones are connected.
- Android: allowed low-delay capturing and playback in certain systems, reducing call delay.
- Android: supported using VODPlayer and TRTC at the same time with AEC enabled.
- Windows: made the SDK compatible with the virtual webcam e2eSoft VCam.
- Windows: allowed calling `startLocalPreview` and `startCameraDeviceTest` at the same time.
- Windows: allowed publishing screen sharing images via the primary stream and at the same time calling `startLocalPreview` to enable local preview.
- Windows: fixed long audio delay caused by the playback buffer of the SDK.
- Windows: optimized the audio enablement logic to prevent mic occupation in the playback-only mode.

Bug fixing

- iOS: fixed low playback volume on iPhone SE.
- iOS: fixed crash when a sub-instance (`TRTCCloud.createSubCloud`) calls `muteRemoteAudio` .
- iOS: fixed occasional crash during rendering.
- iOS: fixed the issue where video rendering on certain iPad devices occasionally causes the main thread to crash during foreground/background switching.
- iOS: fixed known memory leak issues.
- iOS: fixed the issue where iOS 14 prompts "the app would like to find and connect to devices on your local network".
- macOS: fixed the issue where `getCurrentCameraDevice` returns `nil` .
- macOS: fixed the issue where certain USB cameras cannot be turned on.
- macOS: fixed crash when the area of shared content is set to `0` .
- Android: fixed crash on Android 5.0 when the `READ_PHONE_STATE` permission is not granted.
- Android: fixed audio capturing and playback exceptions after Bluetooth earphones are disconnected and connected again.
- Android: fixed known crash issues.
- Windows: fixed crash on 64-bit Windows when screen sharing is enabled and disabled multiple times.
- Windows: fixed crash on certain systems when OpenGL is used.

Version 7.7 Released on September 8, 2020

Optimization

- All platforms: improved instant streaming performance of the substream (screen sharing images).

- iOS: optimized the internal thread model to improve stability when 30 or more channels of audio/video are played back at the same time.
- iOS & Android: improved the performance of the audio module and reduced the capturing delay of the first audio frame.
- iOS & Android: improved volume and audio quality when VODPlayer and TRTC are used at the same time.
- iOS & Android: supported files in WAV format for audio effects and background music.
- Windows: fixed high CPU usage when low-end cameras are used.
- Windows: optimized the compatibility with multiple USB cameras and mics to make it easier to turn on such devices.
- Windows: optimized the selection policy of cameras and mics to avoid audio/video capturing exceptions caused by the connection/disconnection of cameras and mics.

Bug fixing

- All platforms: fixed occasional playback exceptions when the `muteLocalVideo` and `muteLocalAudio` APIs are called in poor network conditions.
- iOS: fixed occasional failure to play audio effects on earlier generations of iPhone or iPad devices.
- iOS: fixed distorted screen sharing images on iPad Pro.
- iOS: fixed the issue where the application keeps requesting screen recording permission after the user denies it.
- Windows: fixed the issue where onExitRoom fails after laptops or desktops remain in sleep mode for a long time.
- Windows: fixed echo after system audio capturing is enabled via the calling of startSystemAudioLoopback in the music mode.
- Windows: fixed the issue where no audio is played sometimes when a user uses `enterRoom` and `exitRoom` to enter and leave the room in a short period of time.
- Windows: fixed project compilation problems with Visual Studio 2010.
- Windows: fixed the issue where the `onUserVideoAvailable` event callback is returned multiple times in the manual subscription mode (setDefaultStreamRecvMode(false， false)).

Version 7.6 Released on August 21, 2020

New features

- Windows: added the updateLocalView and updateRemoteView APIs to improve user experience in adjusting HWND rendering windows in real time.
- Windows: added the getCurrentMicDeviceMute API to get whether the PC is muted.

- Windows: added the setCurrentMicDeviceMute API to turn on global mute for the PC.
- macOS: added the updateLocalView and updateRemoteView APIs to optimize user experience in adjusting the view rendering area in real time.
- macOS: added the getCurrentMicDeviceMute API to get whether the PC is muted.
- macOS: added the setCurrentMicDeviceMute API to turn on global mute for the PC.
- iOS: added the updateLocalView and updateRemoteView APIs to optimize user experience in adjusting the view rendering area in real time.
- iOS: added the onCapturedRawAudioFrame callback to `TRTCCloudDelegate` , and changed the names of a number of other callback functions. The names used now are onLocalProcessedAudioFrame, onRemoteUserAudioFrame, and onMixedPlayAudioFrame.
- Android: added the onCapturedRawAudioFrame callback to `TRTCCloudListener` , and changed the names of a number of other callback functions. The names used now are onLocalProcessedAudioFrame, onRemoteUserAudioFrame, and onMixedPlayAudioFrame.

Optimization

- All platforms: optimized the protocol policy of `enterRoom` to improve the speed and success rate of room entry.
- All platforms: fixed reduced performance and stuttering when a large number of audio channels are subscribed at the same time.
- macOS: supported sharing specified area of a specified window.

Bug fixing

- All platforms: fixed the issue where the SDK does not trigger the `onEnterRoom` callback when users enter the same room without exiting.
- All platforms: fixed a few internal bugs that may cause a black screen.
- All platforms: fixed failure to display screen sharing images when `startRemoteSubStreamView` is called early.
- Windows: fixed known handle and GDI leaks.
- Windows: fixed known crash issues.
- Windows: fixed the issue where cameras and mics are not started automatically after being disconnected and connected again.
- iOS: fixed crash on iOS 10 when certain file paths are passed in the background music API.
- Android: fixed the occasional no audio issue when `enterRoom` and `exitRoom` are called multiple times in a short period of time.
- Android: fixed occasional black screen during the streaming of screen recording.

Version 7.5 Released on July 31, 2020

New features

- Supported dual-stack IPv6 and IPv6-only.
- Allowed playing back streams in multiple rooms. This feature can be used for ultra-small classes.
- Allowed setting a background image for MCU On-Cloud MixTranscoding (for regulatory purposes, the image must be uploaded to the TRTC console first).
- Added two new modes for MCU On-Cloud MixTranscoding: `A+B=>C` and `A+B=>A` .
- Added the `jitterBufferDelay` field, which indicates the playback buffer time, to the real-time status callback API `onStatistics` .

Optimization

- Reduced end-to-end delay for co-anchoring by 40% from that in version 7.4.
- Reduced in-ear monitoring delay on phones and allowed setting voice change and reverb effects for in-ear monitoring.
- Optimized the algorithm for evaluating network jitter at the player end to reduce playback delay.
- Reduced end-to-end delay for co-anchoring in TRTC SDK for Android.
- Reduced in-ear monitoring delay.
- Optimized the issue where playback view switching causes a black screen.

Bug fixing

- Fixed the issue where playback fails after `playBGM` and pauseBGM` are called successively in a function.
- Fixed the occasional issue where users continue to receive the `onEnterRoom` callback after exiting the room.
- Fixed the issue on certain devices where ultra-low-resolution encoding fails and cannot be recovered.

Version 7.4 Released on June 24, 2020

Optimization

Allowed setting volume for in-ear monitoring.

Bug fixing

- Fixed the issue where the local image flickers during landscape/portrait mode switch on Android.
- Fixed encoding failure when custom videos are published from certain Android phones.

- Fixed occasional crash during audio packet processing.

Version 7.3 Released on June 1, 2020

New features

- Added a new audio effect management API `TXAudioEffectManager` to offer more diverse audio effects while continuing to support the legacy API.
- Added the `minVideoBitrate` option to `setVideoEncoderParam` , the API used to set video encoding parameters. The option is recommended for scenarios with high requirements on video quality.

Optimization

- Supported transient noise reduction, which can be enabled using `setAudioQuality(TRTCAudioQualitySpeech)` .
- Supported asset packages for audio effect files.
- Improved local video clarity.
- Supported custom rendering by texture for the player end, reducing resource consumption.
- Optimized the resolution selection logic for video captured by the camera, enhancing visual experience.
- Optimized echo cancellation.
- Supported 128 Kbps high-quality stereo sound from sender to recipient, which can be set using the `setAudioQuality(TRTCAudioQualityMusic)` API.
- Supported the `SPEECH` audio mode, which provides better ANS capabilities for audio conferencing calls. This mode can be set using the `setAudioQuality(TRTCAudioQualitySpeech)` API.
- Supported playing back multiple tracks of background music at the same time. This feature can be used in karaoke scenarios, where vocals and instrumentals are separate tracks. Loop playback of background music is also supported now.
- Supported calling `muteLocalVideo` before `startLocalPreview` to preview without pushing streams. This can be achieved by calling `startLocalPreview` before `enterRoom` as well.

Bug fixing

- Fixed occasional crash due to OpenGL context error during custom video capturing.
- Fixed the issue where the custom rendering callback is not triggered after `setLocalVideoRenderListener` is called before room entry.
- Fixed the issue where, when a user switches to the front/rear camera in the landscape mode, other users see upside down video of the user.

- Fixed the occasional issue where, if a user calls `startLocalPreview` before room entry, other users see corrupted image of the user.
- Fixed occasional crash of the hardware encoder.
- Fixed occasional intermittent audio for local recording.
- Fixed the issue where, when the user publishing streams enters the room again after the pausing of stream pushing (`muteLocalVideo` , `muteLocalAudio`) causes crash or forces the application to close, streams are not played back automatically at the player end.

Version 7.2 Released on April 16, 2020

New features

Supported screen recording on Android, allowing users to stream screen recording on phones.

Optimization

- Optimized call performance on low-end and mid-range Android phones, improving audio experience.
- Optimized visual effect APIs such as filters and green screen and aggregated them into the `TXCBeautyManager` class so that they share a calling method.

Bug fixing

Fixed the occasional issue where the custom stream ID fails to take effect in time during role switching.

Version 7.1 Released on March 27, 2020

Optimization

- Supported static build of projects using the C++ STL library.
- Enabled ANS and AGC by default in the call volume mode to improve audio quality.
- Improved the usability of the preset stream mixing template.
- Enhanced the success rate of stream mixing.

Bug fixing

- Fixed the issue where all audio processing values become zero when AGC is enabled/disabled frequently during room entry.
- Fixed the issue where speed testing slows down the calling of other APIs.

- Fixed the issue where the volume of upstream data doubles and noise is heard after real-time communication is interrupted by an incoming call.
- Fixed the issue where streams are relayed automatically after room entry.

Version 7.0 Released on March 9, 2020

- Optimized the 3A enabling policy.
- Improved the usability of MCU On-Cloud MixTranscoding.
- Enhanced audio smoothness in poor network conditions.
- Fixed memory leaks caused by frequent room entry and exit.

Version 6.9 Released on January 14, 2020

New features

- Supported Android 10.0.
- Added the `snapshotVideo()` API for taking screenshots of local or remote video.
- Added the `pauseAudioEffect` and `resumeAudioEffect` APIs for pausing and resuming an audio effect.
- Added the `setBGMPlayoutVolume` and `setBGMPublishVolume` APIs for setting the local playback volume and publishing volume of background music respectively.
- Added the `setRemoteSubStreamViewRotation` API for adjusting the rotation of played back substream video.
- Added a global volume mode setting API `setSystemVolumeType(TRTCSystemVolumeTypeVOIP)` , which can ensure that call volume is used all the time. This is mainly to prevent the switch from Bluetooth earphones to the built-in mic during audio capturing.
- Added the `streamId` attribute to the `TRTCParams` parameter of `enterRoom` , which can be used to set the user’s CDN stream ID, making it easier to bind to live streaming CDNs.
- Added the `cloudRecordFileName` attribute to the `TRTCParams` parameter of `enterRoom` , which can be used to set the on-cloud recording filename for a live stream.
- Added the `TRTCAppSceneAudioCall` scenario, which is optimized for audio calls and can be set during the calling of `enterRoom` .
- Added the `TRTCAppSceneVoiceChatRoom` scenario, which is optimized for audio chat rooms and can be set during the calling of `enterRoom` .

Optimization

- Improved the recording feature’s tolerance of video interruption, enabling remote recording of more complete video.
- Fixed lip-sync errors during hardware decoding on certain devices.
- Supported capturing 1080p video, allowing PC audience to watch clearer video published from phones.
- Simplified the error codes for room entry.
- Fixed occasional slow streaming.

Bug fixing

- Fixed occasional crash of the HTTP component.
- Fixed the occasional issue where no callback is returned for the completion of audio effect playback.
- Fixed the occasional issue where the system cannot be recovered from room entry failures.

Version 6.8 Released on November 15, 2019

New features

- Added the in-ear monitoring capability.
- Allowed users to disable automatic stream pulling upon room entry.
- Added the `getBeautyManager` API, which aggregates beauty filter, retouching, and animated effect APIs.
- Added retouching features including skin polishing, eye brightening, teeth whitening, wrinkle removal, and eye bag removal to the Enterprise Edition.
- Added the `onRemoteUserEnterRoom` and `onRemoteUserLeaveRoom` callbacks for the entry and exit of a user.

Optimization

- Optimized the PTS generation mechanism.
- Enabled automatic selection of the best access point after network change.
- Supported calling `startRemoteView` in advance.

Bug fixing

Fixed known crashes.

Version 6.7 Released on September 30, 2019

New features

- Added permission requesting configuration for AAR packaging.
- Supported CPU usage evaluation on Android 8.0 and above.

Optimization

- Sped up relayed push.
- Supported adjusting the playback volume of a specific user.

Version 6.6 Released on August 2, 2019

New features

- Supported local audio recording.
- Added callback APIs for sending the first audio and video frame.
- Added the system volume type setting API.
- Added the audio effect API for playing short audio effects.
- Made the data returned via the custom audio callback modifiable.

Optimization

- Sped up room entry and improved its success rate.
- Added an API to mute remote video.
- Unified room entry error codes, which are returned via `onEnterRoom` . If `result` is smaller than 0, it indicates failure to enter the room.
- Optimized the demo to support low-latency big rooms.
- Added a volume setting API and a volume callback API for the player.
- Supported local rendering for custom video publishing.
- Supported custom capturing and publishing of 1080p video.
- Supported the `SurfaceView` method for local and remote rendering.

Bug fixing

- Fixed the problems with relayed push and stream mixing.
- Fixed incorrect rotation for local preview.

Version 6.5 Released on June 12, 2019

New features

Added the "low-latency big room" feature for the live streaming mode (`TRTCAppSceneLIVE`):

- Adopted a UDP protocol optimized for audio/video, allowing the SDK to better adapt to poor network conditions.
- Reduced the average watch latency to around 1 second, enhancing anchor-audience interaction experience.
- Supported rooms with up to 100,000 users.

Optimization

- Fixed lip-sync errors in poor network conditions.
- Optimized the `onStatistics` status callback. Callbacks are returned for only existing streams now.
- Optimized the playback buffer logic of `TXLivePlayer` , reducing stuttering.
- Sped up playback at the player end after local video is muted via `muteLocalVideo` and unmuted again.
- Optimized the QoE algorithm for high-latency and high-packet-loss network environments.
- Improved decoder performance and fixed increasing delay on earlier generations of Android phones.
- Optimized the volume evaluation algorithm (`enableAudioVolumeEvaluation`), improving accuracy.
- Optimized the QoE algorithm in the video call mode (`TRTCAppSceneVideoCall`), further enhancing the smoothness of one-to-one calls in poor network conditions.

Bug fixing

- Fixed the occasional issue where no callback is returned for `enterRoom` .
- Fixed the issue where a user cannot play audio after disabling audio capturing.
- Fixed green screen after the local rendering view is removed and added again.
- Fixed the issue where the custom rendering callback (`setRemoteVideoRenderDelegate`) is returned only 10 times at most when the resolution of the remote video is 540p or above.

Version 6.4 Released on April 25, 2019

New features

- Added APIs for mirroring local video and encoded video.
- Added a callback for the `setMixTranscodingConfig` API.
- Added the eye enlarging, face slimming, chin slimming, and animated widget features to the Enterprise Edition.

Optimization

- Improved smoothness in poor network conditions.
- Optimized the volume callback algorithm, improving the accuracy of the values returned.
- Supported specifying data frame timestamps externally for the publishing of custom audio and video.
- Optimized the `setMixTranscodingConfig` API by adding the `roomID` parameter for stream mixing during cross-room co-anchoring.
- Optimized the `setMixTranscodingConfig` API by adding the `pureAudio` parameter for audio mixing and recording in audio-only call scenarios.
- Improved the 720p video decoding performance on low-end Android devices.

Bug fixing

- Fixed failure to switch to the hands-free mode.
- Fixed the occasional issue where live streaming (TXLivePlayer) latency increases and does not fall back.
- Fixed failure to use `setVideoEncoderRotation` in live streaming scenarios.
- Fixed the issue where no error callback is returned after the mic permission is denied on Android.
- Fixed the issue where a window pops up after the demo is opened on Android 9.0.
- Fixed failure by audience to adjust the volume using the volume buttons.

Version 6.3 Released on April 2, 2019

New features

- Supported 64-bit Android.
- Added a custom video capturing API: TRTCCloud > sendCustomVideoData.
- Added a custom audio capturing API: TRTCCloud > sendCustomAudioData.
- Added custom video rendering APIs: TRTCCloud > setLocalVideoRenderDelegate + setRemoteVideoRenderDelegate.
- Added a custom audio data callback API: TRTCCloud > setAudioFrameDelegate, which you can use to do the following:
  - Return the data captured by the mic: TRTCAudioFrameDelegate > onCapturedAudioFrame.
  - Return the audio data of each remote user: TRTCAudioFrameDelegate > onPlayAudioFrame.
  - Return the mixed audio data sent to the speaker for playback: TRTCAudioFrameDelegate > onMixedPlayAudioFrame.

Version 6.2 Released on March 8, 2019

New features

- Added the filter strength setting API `setFilterConcentration()` .
- Added the `sendSEIMsg()` API for sending custom messages through SEI headers in video frames. The feature is mainly used to insert timestamp information into video streams.
- Added the cross-room call feature `connectOtherRoom` , which allows two existing TRTC rooms to communicate with each other. This feature can be used to enable anchor competition across rooms.

Optimization

- Improved CPU utilization and stability.
- Enhanced video clarity in poor network conditions.
- Disabled the creation of multiple `TRTCCloud` instances and restricted instance creation to singletons. This can avoid cases where different instances of `TRTCCloud` compete for network resources, which compromise user experience.

Bug fixing

Fixed the problems with relayed push in audio-only call scenarios (such as Werewolf playing). You must specify the `bussInfo` field in `TRTCParam` to use the feature.

Version 6.1 Released on January 31, 2019

Optimization

- Supported watching screen sharing streams.
- Supported publishing custom video.
- Optimized CDN live streaming and stream mixing.
- Introduced two types of scenarios: live streaming and video calls, which are specified during room entry.
- Enhanced stability and fixed occasional crash.
- Optimized QoS control and improved performance in poor network conditions.

Version 6.0 Released on January 18, 2019

Optimization

- Updated the architecture to the LiteAV kernel.
- Adopted a new QoS algorithm, reducing stuttering and improving smoothness.

- Introduced a new audio module, enhancing audio quality in various network conditions.
- Supported dual-channel (primary stream and substream) encoding. We recommend you use this feature on Windows and macOS only.
- Supported CDN live streaming and stream mixing.

Release Notes (Desktop Browser)

Last updated : 2021-06-03 18:01:17

The forming rule of version number `major.minor.patch` is as follows:

- major: major version number. If there is major version refactoring, this field will be incremented. Generally, the APIs of different major versions are not compatible with each other.
- minor: minor version number. The APIs of different minor versions are compatible with each other. If there is a new or optimized API, this field will be incremented.
- patch: patch number. If there is a feature improvement or bug fix, this field will be incremented.

Note :

- We recommend you update to the latest version in time for better service stability and online support.
- For notes on version upgrade, please see Update Guide.

Version 4.8.6 Released on March 1, 2021

Improvements

Supported stereo playback during pull.

Note :
Unavailable on iOS yet.

Fixes

Fixed the issue where the web received the `stream-removed` event when the video was muted and frozen on a mobile device.

Version 4.8.5 Released on January 29, 2021

Improvements

- Supported setting multiple turn servers for Client.setTurnServer.

- Optimized the `userId` verification logic.

Fixes

Fixed the issue where the mute status was occasionally inaccurate after push was started.

Version 4.8.4 Released on January 15, 2021

Improvements

- Supported dynamically calling the LocalStream.setVideoProfile API.
- Optimized the data reporting logic of the dashboard.
- Optimized the processing logic when autoplay was restricted. For more information, please see Recommendations for Restricted Autoplay.
- Optimized the processing logic when auto resumption failed upon device plug/unplug. For more information, please see DEVICE_AUTO_RECOVER_FAILED.

Fixes

Fixed the issue where the mute status was occasionally inaccurate after push was resumed.

Version 4.8.3 Released on January 7, 2021

Improvements

Optimized the parameter verification logic of the room entry API `roomId` .

Fixes

- Fixed the issue where the version 4.8.2 lacked third-party dependencies.
- Fixed the occasional issue of muted playback when only audio was subscribed to.
- Fixed the issue where `resume` was muted and `getAudioLevel` was 0 on iOS after autoplay was restricted.

Version 4.8.2 Released on December 31, 2020

Breaking changes

Deleted the disused API `setDefaultMuteRemoteStreams` . Please use the client.unsubscribe API instead.

Improvements

- Optimized the parameter verification logic of the room entry API `roomId`. For more information, please see the API documentation and Upgrade Guide.
- Optimized the notification timing of `peer-join` and `peer-leave` events.

Fixes

Fixed the occasional error of `Cannot read property 'isConnected' of null` during room exit.

Version 4.8.1 Released on December 25, 2020

Fixes

- Fixed the occasional issue where the remote user's voice couldn't be heard on Windows.
- Fixed the issue where the `client.getRemoteVideoStats()` API returned empty data.

Version 4.8.0 Released on December 18, 2020

New features

- Supported On-Cloud MixTranscoding.
- Supported string room IDs on all platforms. For more information, please see the `userDefineRecordId` parameter of `TRTC.createClient`.

Improvements

- Optimized the H.264 support check logic.
- Optimized the device switching logic.
- Optimized the status judgment logic of the `hasAudio` / `hasVideo` APIs.

Fixes

- Fixed the occasional issue where reconnection failed after network disconnection.
- Fixed the issue of black screen caused by frequently adding/removing tracks on Safari.

Version 4.7.1 Released on November 27, 2020

Improvements

- Optimized the auto capturing resumption logic when the media device was changed (for example, the interface was loose or the device was plugged/unplugged).
- Added the new error code DEVICE_AUTO_RECOVER_FAILED, which could be used to prompt when device recovery failed.

Fixes

- Fixed the occasional issue of black screen on Chrome 87.
- Fixed the issue where the screen sharing stream disappeared during camera push + screen sharing for Native and resubscription or unsubscription for web.

Version 4.7.0 Released on November 20, 2020

New features

Supported Firefox M56+ and Edge M80+ for the desktop.

Breaking changes

TRTC.checkSystemRequirements could return detailed check results. For more information, please see the API documentation and Upgrade Guide.

Improvements

- Optimized the upstream bitrate control logic.
- Optimized the retry logic for getting media devices.
- Optimized the WebSocket reconnection logic.
- Optimized the auto push resumption logic when the device was changed and supported auto push resumption when the mic was plugged/unplugged during audio mixing.

Version 4.6.7 Released on November 5, 2020

Fixes

- Fixed the occasional issue of blurred screen during pull for playback when hardware acceleration was enabled on Chrome.
- Fixed the issue where room entry couldn't be performed for pull in the browser built in WeChat on iOS.

Version 4.6.6 Released on October 23, 2020

Improvements

- Optimized the upstream `peerConnection` reconnection logic.
- Optimized the downstream `peerConnection` reconnection logic.
- Optimized the `TRTC.checkSystemRequirements` check logic.
- Supported screen sharing on Safari. For more information, please see Screen Sharing Guide.

Fixes

Fixed the issue where the `getAudioLevel` value was 0 after audio playback was manually resumed due to the restriction of the autoplay policy.

Version 4.6.5 Released on October 14, 2020

Improvements

- Optimized the WebSocket signaling channel reconnection logic to improve the connection stability.
- Optimized the log output logic.

Fixes

- Fixed the issue where the `getAudioLevel` API returned a value of 0 after resubscription on Chrome.
- Fixed the issue of muted playback after resubscription on Safari.
- Fixed the issue where the `getLocalVideoStats` API returned `undefined` after the upstream audio track was replaced with `replaceTrack` .
- Fixed the occasional issue of WebSocket disconnection upon network type switch during call on a mobile device.

Version 4.6.4 Released on September 24, 2020

Improvements

Supported stopping collecting network quality statistics after room exit.

Fixes

- Fixed the error of room entry on Chrome 56.
- Fixed the issue of screen rotation during relayed push on a mobile device.
- Fixed the issue of exceptional on-cloud recording when pure audio was pushed.
- Fixed the issue where auto push resumption failed due to inconsistent resolution after the camera was unplugged.

Version 4.6.3 Released on August 28, 2020

Improvements

- Optimized the compatibility check logic.
- Optimized the log reporting logic.
- Optimized the upstream bitrate control logic.

Version 4.6.2 Released on August 14, 2020

Improvements

- Optimized the upstream bitrate control logic.
- Optimized the `switchRole` parameter verification logic.
- Optimized the calculation logic of upstream network quality.
- Optimized the error messages.
- Supported automatically resuming push when the change of the current stream capturing device was detected.

Fixes

Fixed the issue where immediate republishing failed after `unpublish` succeeded.

Version 4.6.1 Released on July 28, 2020

Improvements

- Added TRTC.isScreenShareSupported. Safari didn't support screen sharing yet.
- Improved the parameter verification logic of the `subscribe` and `unsubscribe` APIs.
- Added the network quality logging feature.

Fixes

- Fixed the issue where the SDK reported `OverconstrainedError` when the media device was not authorized and the device ID passed in by the `TRTC.createStream` API was an empty string.
- Fixed the issue where no log was printed when the upstream `peerConnection` was interrupted.

Version 4.6.0 Released on July 16, 2020

New features

Added the `NETWORK_QUALITY` event.

Version 4.5.0 Released on July 2, 2020

New features

Added the `screenAudio` parameter for the `createStream` API.

Fixes

- Fixed the issue where echo cancellation did not work in a browser on Android.
- Fixed the issue where the RTT value returned by the `getTransportStats` API was `NAN` .

Version 4.4.0 Released on May 28, 2020

New features

Supported capturing audio of the system (Windows) or current tab (macOS) during screen sharing on Chrome 74 or above.

Version 4.3.13 Released on April 16, 2020

Improvements

Optimized the availability check logic.

Version 4.3.12 Released on April 13, 2020

Fixes

Fixed a potential exception of `RTCPeerConnection` status change.

Version 4.3.11 Released on March 28, 2020

Improvements

Added check for Mobile QQ Browser, which didn't support the TRTC SDK for Desktop Browser yet.

Fixes

Fixed the Boolean returned value type.

Version 4.3.10 Released on March 17, 2020

Improvements

- Optimized the environment check logic.
- Added `name code` for `RtcError` .

Version 4.3.9 Released on March 13, 2020

Improvements

- Added auto deployment environment check.
- Optimized the logging feature.

Version 4.3.8 Released on February 24, 2020

Improvements

Added the `streamId` and `userdefinerecordid` fields for `createClient` .

Version 4.3.7 Released on February 21, 2020

Improvements

Fixed the exception thrown by device switch during screen sharing.

Fixes

- Fixed the issue where devices were in use after `MediaStream` was released during device switch.
- Fixed the potential error of the subscription API.

Version 4.3.6 Released on February 5, 2020

Fixes

Adjusted the audio/video playback sequence of `Stream.resume()` and fixed the issue of exceptional playback in WeChat Browser on iOS.

Version 4.3.5 Released on February 5, 2020

Improvements

Added the `publish` timeout check to improve the success rate of signaling.

Version 4.3.4 Released on January 6, 2020

Improvements

Upgraded `core-js` to v3.6.1.

Fixes

- Fixed the issue where an exception was thrown after `unpublish` timed out.
- Fixed the v8 issue caused by third-party libraries.

Version 4.3.3 Released on December 25, 2019

Improvements

- Added active environment check for WebRTC support.
- Optimized the SDP response mechanism.
- Optimized the reporting logic.

Fixes

Fixed the TURN URL protocol format.

Version 4.3.2 Released on December 9, 2019

Improvements

- Added the automatic reconnection mechanism for downstream ICE disconnection.
- Removed the STUN hole punching operation to increase the success rate and speed of user connection over the private network.

- Uniformly used the UTC time calibrated with the server as the timestamp of log reporting.
- Optimized ICE error reporting.
- Added more key events to `avmonitor` reporting.

Fixes

- Fixed the reconnection exception and error of the WebSocket signaling channel 1005.
- Fixed the issue of downstream packet loss rate reporting.

Version 4.3.1 Released on November 23, 2019

Improvements

Added the automatic reconnection mechanism for upstream linkage ICE disconnection during call.

Fixes

Fixed the issue where the host ICE candidate of public IP type did not take effect after STUN hole punching failed.

Version 4.3.0 Released on November 15, 2019

New features

Added the `Client.getTransportStats()` API.

Improvements

- Supported more detailed log reporting.
- Supported wildcard for event unbinding.
- Extended the connection timeout period to 5 seconds.
- Extended the publishing timeout period to 5 seconds.

Fixes

Fixed the issue of exceptional SDK judgment due to modification of the prototype chain by `zone.js` .

Version 4.2.0 Released on November 4, 2019

New features

Added the `Client.off()` API to unbind client events.

Improvements

- Optimized the call status statistics.
- Added permission check for `Client.publish()` .
- Added error prompt for autoplay for `Stream.play()/resume()` .

Fixes

Fixed the issue where the screen turned black during camera switch with `LocalStream.switchDevice()` .

Version 4.1.1 Released on October 24, 2019

Fixes

- Fixed the issue of lost logs.
- Fixed the issue of lost remote users upon reconnection.

Version 4.1.0 Released on October 17, 2019

New features

- Supported passing the `HTMLDivElement` object to the `Stream.play()` API.
- Added audio bitrate adjustment settings. You could set the audio attribute through `LocalStream.setAudioProfile()` . Two profiles were supported: standard and high.

Fixes

- Fixed the issue where the number of `WebAudio Context` values was limited on legacy Chrome.
- Fixed the issue where `replaceTrack()` did not restart the local audio/video player.
- Fixed the issue where the `LocalStream.setScreenProfile()` custom attribute configuration did not take effect.
- Fixed the issues of audio/video player restart and status reporting.

Version 4.0.0 Released on October 11, 2019

Added the rebuild version of the TRTC SDK for Desktop Browser to provide APIs in the `Client/Stream` model, where the responsibilities of each object were clearer, and the semantics were more concise. The rebuild version was not compatible with the legacy version. In addition to API changes, it also had the following features:

- Video attributes (resolution, frame rate, and bitrate) were completely controlled by the application through the SDK's `LocalStream.setVideoProfile()` API. The "image settings (Spear Role)" in the Tencent Cloud console for the legacy version were no longer supported.
- The SDK encapsulated an audio/video player in the `Stream` object, and audio/video playback was completely controlled by the SDK.
- The features of subscribing to and unsubscribing from remote streams were offered. You could flexibly control the reception of remote audio, video, or audio/video data streams through the `Client.subscribe()/unsubscribe()` APIs.

Release Notes (Electron)

Last updated : 2021-04-19 10:42:35

v8.4.1 Released on March 26, 2021

New features

- macOS: supported capturing system audio via startSystemAudioLoopback, i.e., the system loopback feature that is enabled on Windows. The feature allows the SDK to capture system audio so that anchors can stream local audio or video files to other users.
- macOS: supported callback of the system audio capturing result via onSystemAudioLoopbackError, which allows you to learn about the status of the system audio driver.
- macOS: supported local preview for screen sharing. You can now display screen sharing preview in a small window.
- All platforms: supported the beauty filter plugin architecture.

Quality improvement

- All platforms: improved audio quality in the TRTCAudioQualityMusic mode, which makes it more suitable for Clubhouse-like audio streaming scenarios.
- All platforms: improved the adaptability to poor network conditions across the audio-video link. Smooth audio and video can be delivered under 70% of extremely poor network conditions.
- Windows: improved audio quality in some streaming scenarios by significantly reducing audio damage.
- Windows: improved performance by 20-30% in some scenarios.

Bug fixing

- macOS: fixed the issue where after the screen sharing user switched to sharing the desktop and then back to a specific window on Mac mini (M1), remote users still saw the user's desktop.
- macOS: fixed the issue where the shared content was not highlighted (on macOS 11.1 and 10.14.5, the green border did not show around the shared content; on macOS 10.3.2, the green border showed, but the shared window flickered when maximized.)
- macOS: fixed the issue where the SDK crashed when users got the screen sharing list on Mac mini (M1) and "" was returned when `sourceName` was null.
- macOS: fixed the issue on Mac mini (M1) where the SDK crashed when `getCurrentMicDevice` was called because `sourceName` was empty.

- Windows: fixed the issue where the SDK crashed when the desktop was shared on Windows Server 2019 Datacenter x64.
- Windows: fixed the issue where screen sharing sometimes ended unexpectedly when the target window was resized during screen sharing.
- Windows: fixed the issue of image capturing failure with some cameras.

v8.2.7 Released on January 6, 2021

New APIs

- Windows & macOS: added switchRoom to switch rooms.
- Windows & macOS: added setLocalRenderParams to set rendering parameters for the local image (primary stream).
- Windows & macOS: added setRemoteRenderParams to set rendering parameters for a remote image.
- Windows & macOS: added startPlayMusic to play background music.
- Windows & macOS: added stopPlayMusic to stop background music.
- Windows & macOS: added pausePlayMusic to pause background music.
- Windows & macOS: added resumePlayMusic to resume background music.
- Windows & macOS: added getMusicDurationInMS to get the total length of the background music file, in milliseconds.
- Windows & macOS: added seekMusicToPosInTime to set the playback progress of background music.
- Windows & macOS: added setAllMusicVolume to set background music volume. This API is used to control the volume of background music when background music is mixed into played audio.
- Windows & macOS: added setMusicPlayoutVolume to set the local playback volume of background music.
- Windows & macOS: added setMusicPublishVolume to set the remote playback volume of background music.
- Windows & macOS: added onSwitchRoom for the callback of room switching.
- Windows & macOS: added setRemoteAudioVolume to set the playback volume of a remote user.
- Windows & macOS: added snapshotVideo to take a video screenshot.
- Windows & macOS: added onSnapshotComplete for the callback of the completion of a screenshot.

Improvements

- Windows & macOS: supported the string type parameter `strRoomId` for `enterRoom` and `switchRoom` .
- Windows & macOS: fixed other bugs.

v7.9.348 Released on November 12, 2020

Improvements

- Windows: supported the use of Chinese paths to save recording files.
- Windows: supported the anti-covering feature in the window capturing area.

v7.8.342 Released on October 10, 2020

New APIs

- Windows & macOS: added onAudioDeviceCaptureVolumeChanged for volume change callback on the current audio capturing device.
- Windows & macOS: added onAudioDevicePlayoutVolumeChanged for volume change callback on the current audio playback device.

v7.7.330 Released on September 11, 2020

New APIs

Windows & macOS: added setAudioQuality to adjust audio quality.

Improvements

- Windows: fixed the issue where CPU utilization was too high when some low-end cameras were used.
- Windows: optimized the compatibility with multiple USB cameras and mics to make it easier to turn on such devices.
- Windows: optimized the selection policy of cameras and mics to avoid audio/video capturing exceptions caused by plugging in and unplugging cameras or mics.
- Windows & macOS: fixed other bugs.

v7.6.300 Released on August 26, 2020

New file

Windows & macOS: added setCurrentMicDeviceMute, getCurrentMicDeviceMute, setCurrentSpeakerDeviceMute, and getCurrentSpeakerDeviceMute to control mics and speakers on PC.

v7.5.210 Released on August 11, 2020

Improvements

- Windows & macOS: fixed the issue where SDK callbacks were not in sequence.
- Windows & macOS: fixed the issue where switching rendering modes caused crashes.
- Windows & macOS: fixed the issue where rendering failed for certain resolutions.
- Windows & macOS: fixed other bugs.

v7.4.204 Released on July 01, 2020

Improvements

- Windows: optimized the acoustic echo cancellation (AEC) effect on Windows.
- Windows: improved the compatibility with cameras on Windows.
- Windows: improved the compatibility with audio devices (mics and speakers) on Windows.
- Windows: fixed the issue where the `UserID` called back by `onPlayAudioFrame` on Windows was incorrect.
- Windows: supported system audio mixing on 64-bit Windows.

v7.2.174 Released on April 20, 2020

Improvements

- macOS: fixed the local custom rendering resolution inconsistency issue on macOS.
- Windows: optimized the `getCurrentCameraDevice` logic on Windows to return the first device as the default device when the camera is not used.
- Windows: fixed the issue where the highlighted window was displayed as a gray screen during screen sharing.
- Windows: fixed the issue where the system occasionally froze when users got screen share thumbnails on Windows 10.
- Windows & macOS: fixed the issue where the custom stream ID occasionally failed to take effect immediately after role switching.

- Windows & macOS: fixed the issue where the encoding parameters of screen sharing did not take effect.
- Windows: fixed the issue where it took a long time for a screen shared by a Windows user to be displayed to a WebRTC user.

v7.1.157 Released on April 02, 2020

New Features

Supported screen sharing via the primary stream.

Improvements

- improved the usability of preset stream mixing templates.
- increased the success rate of stream mixing.
- optimized screen sharing on Windows.

v7.0.149 Released on March 19, 2020

New file

Added the trtc.d.ts file for TypeScript developers.

