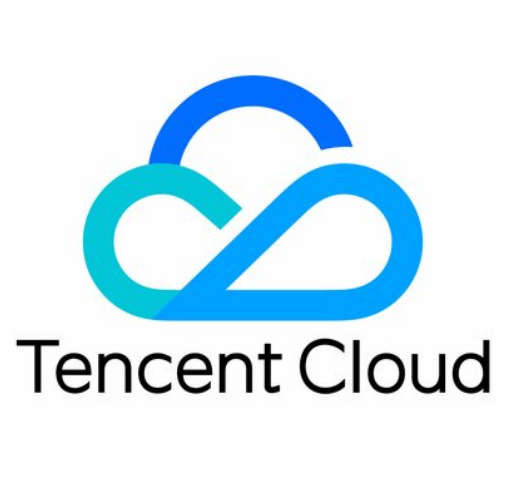
Mobile Live Video Broadcasting

Windows-based Integration

Product Introduction



©2013-2018 Tencent Cloud. All rights reserved.

Copyright Notice

©2013-2018 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice

Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

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

©2013-2018 Tencent Cloud. All rights reserved.

Windows-based Integration
C++
Project Setting

Last updated：2018-10-10 11:35:24

SDK Information

You can update the LVB SDK on Tencent Cloud's official website. The SDK for Windows only offers simplified features:

Version Type	Feature
LVB simplified version	Support push and LVB

The decompressed SDK is composed as follows:

File Name	Description
SDK	SDK include, lib, and dll files with detailed API description
Demo	The simplified Demo based on the implicitly linked dynamic library, including a simple demonstration of UI and main SDK features. Use VS2015 to quickly import the demo and try it out.

Visual Studio Project Settings

Environment requirements

OS: Windows 7 and later

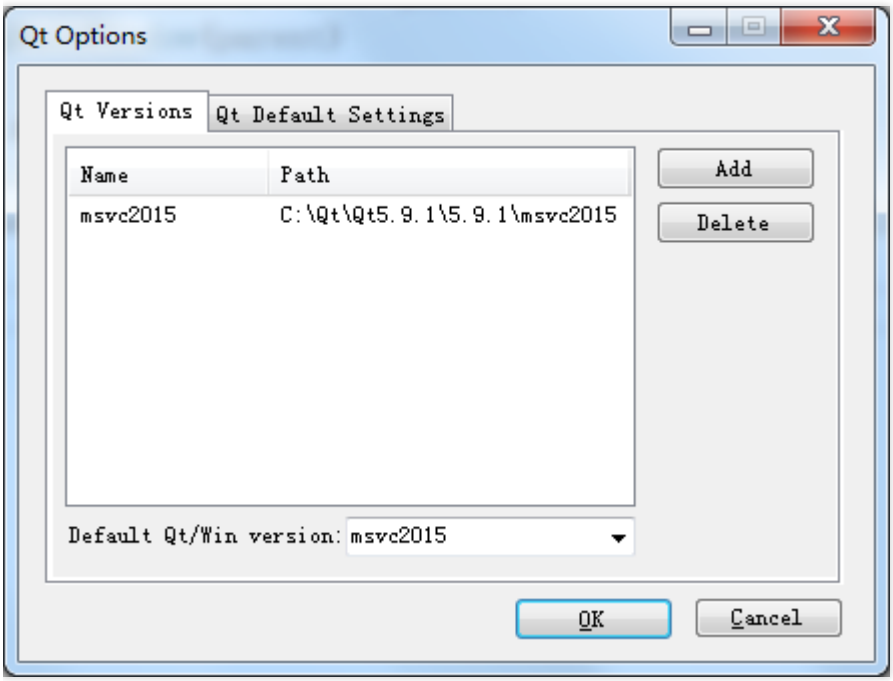
Visual Studio: 2008 and later

Note: For the project of CustomServiceDemo, Visual Studio 2015 and later as well as Qt 5.9 and later are supported.

Qt project settings

©2013-2018 Tencent Cloud. All rights reserved.

After installing Qt 5.9 32bit and Visual Studio Add-in For Qt plug-in, add Qt Versions to the Qt VS Tools-Qt Options in Visual Studio, as shown below:



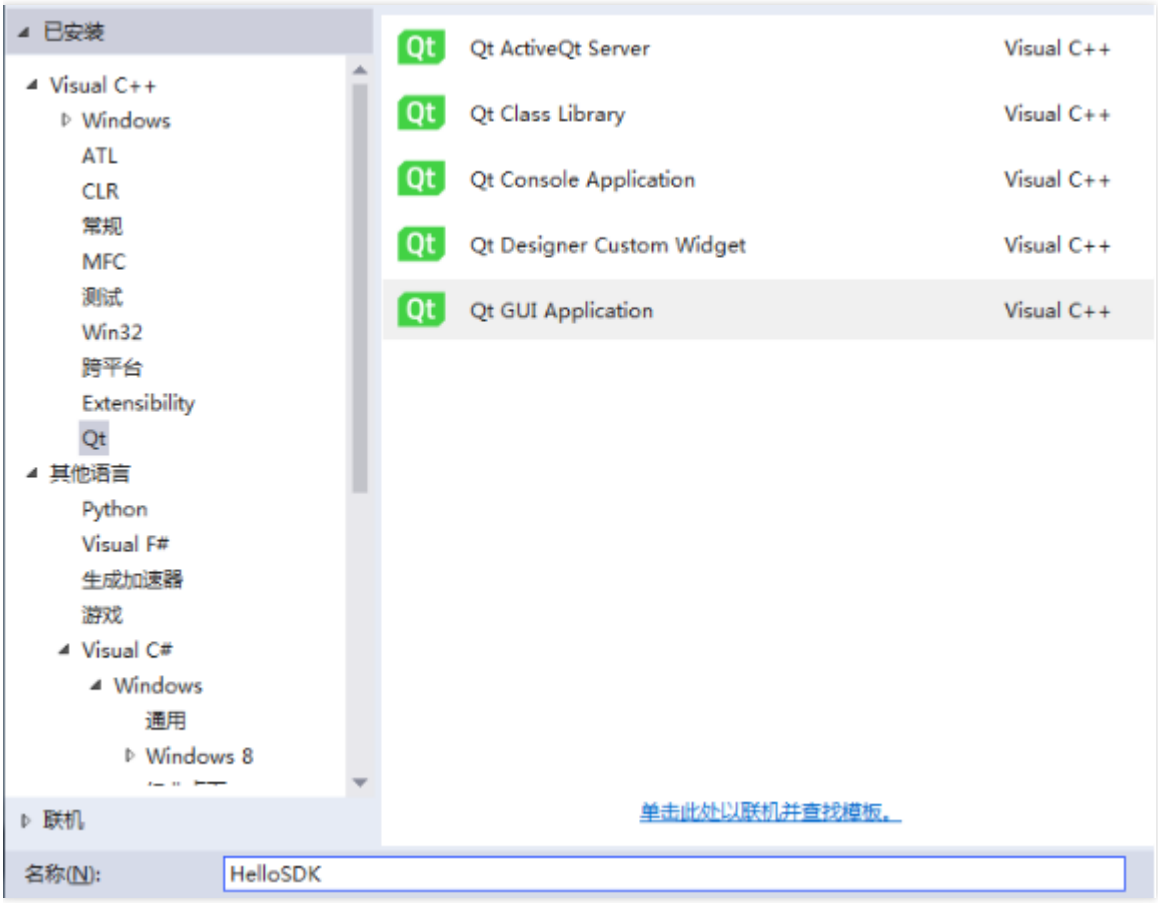
Project settings

The following describes how to configure the SDK in a Visual Studio project using a simple Qt GUI application project.

1. Create a Qt project

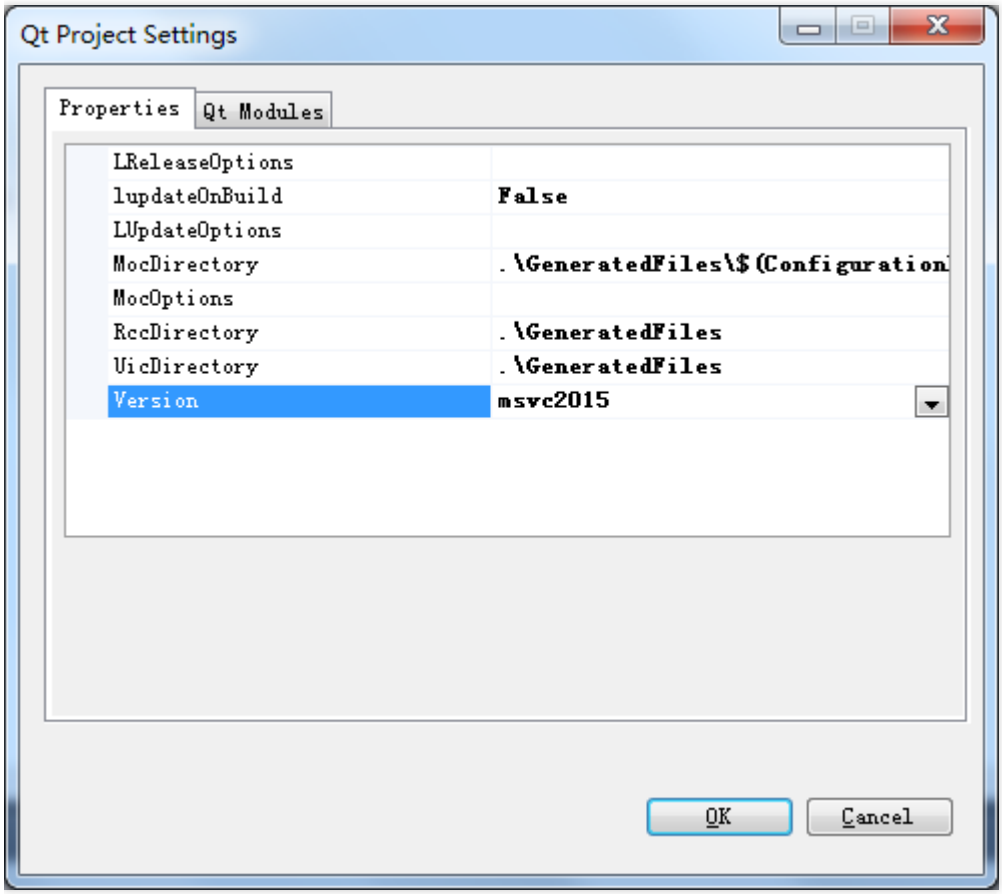
In this example, create a Qt GUI application named HelloSDK, as shown below:

©2013-2018 Tencent Cloud. All rights reserved.



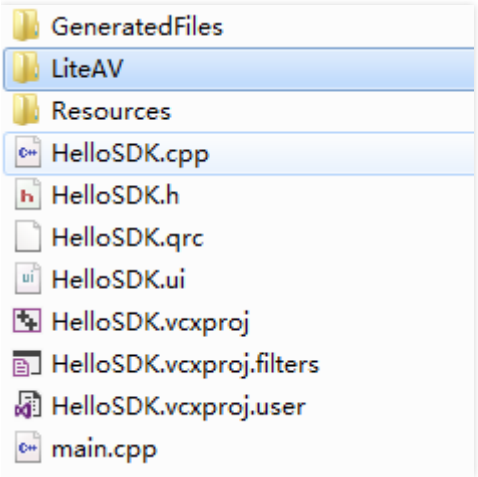
Right-click the project, and select Qt Project Settings to set Version. See the figure below:

©2013-2018 Tencent Cloud. All rights reserved.



2. Copy the SDK file

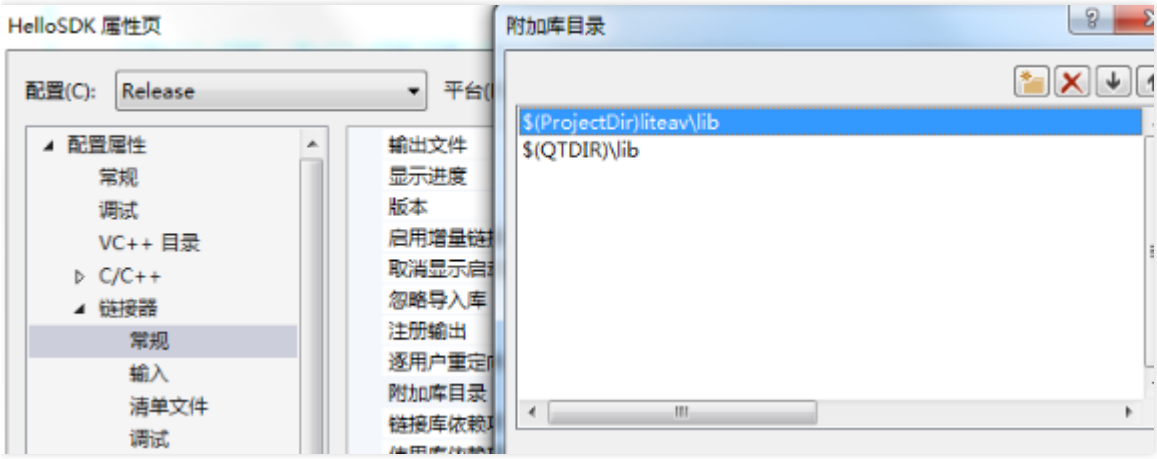
Create a LiteAV folder in the project directory, and copy the content of the downloaded SDK folder to the project directory. The figure below shows the directory structure:



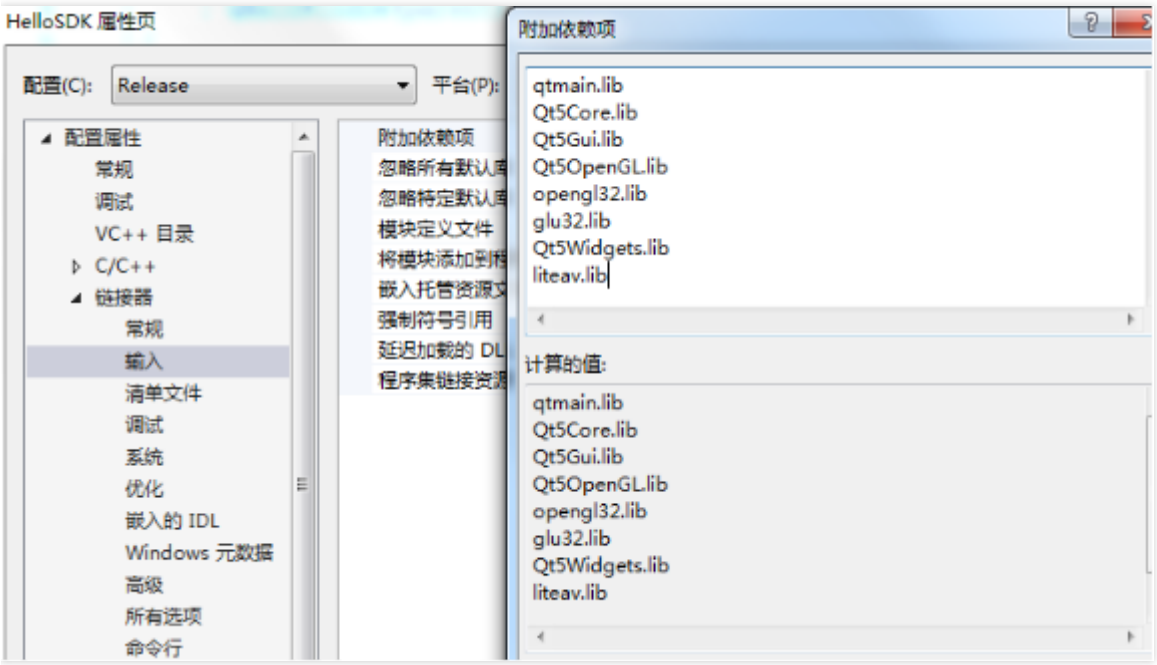
3. Add a library dependency

Add an additional library directory from **Project Attribute** -> **Linker** -> **General**, as shown below:

©2013-2018 Tencent Cloud. All rights reserved.



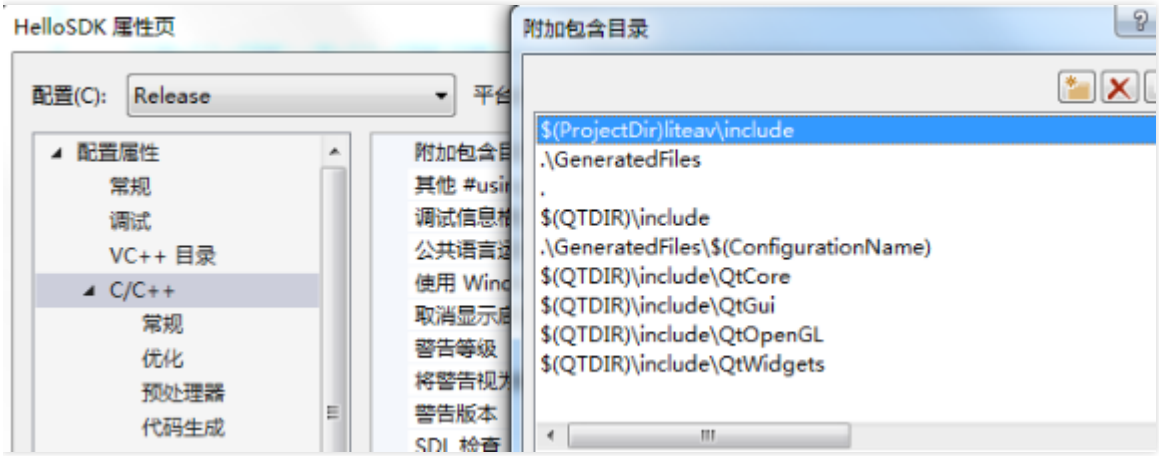
Add liteav.lib from **Project Attribute** -> **Linker** -> **Input** -> **Add a Dependency**, as shown below:



4. Add a header file

Add additional include directories from **Project Attribute** -> **C/C++** -> **General**, as shown below:

©2013-2018 Tencent Cloud. All rights reserved.



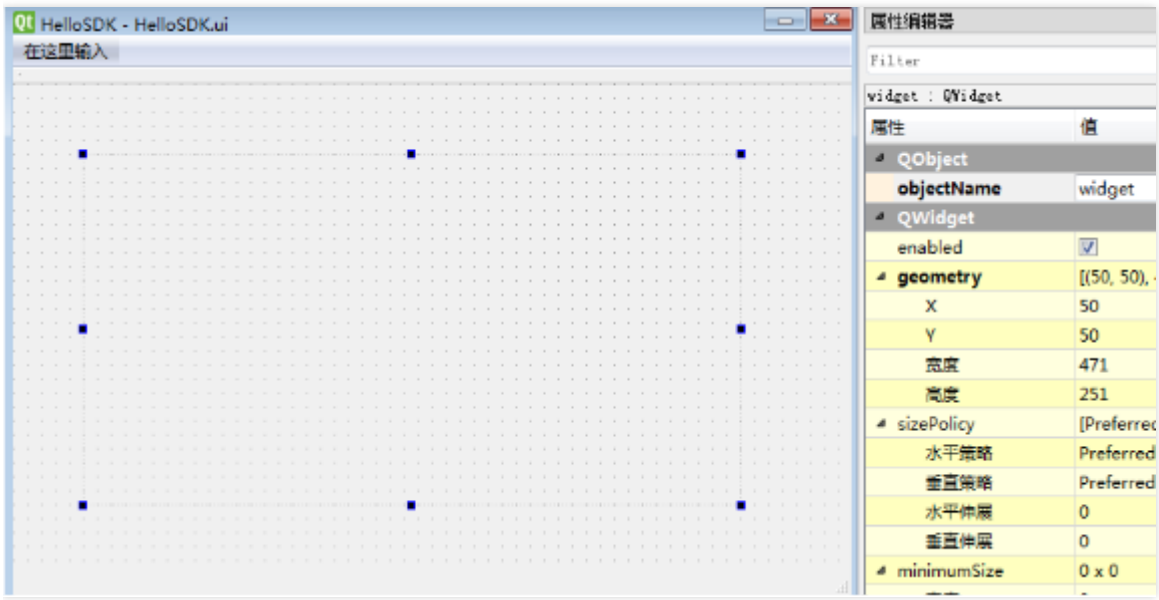
Note that this operation is not mandatory. If you do not add the header file search path for LiteAV/include, "LiteAV/include" needs to be added before the SDK-related header file when the header file is referenced, as shown below:

```
#include "LiteAV\include\TXLivePusher.h"
```

Verification

1. Add a rendering control

Open HelloSDK.ui from the project "From Files", and drag a Widget control to the form, as shown below:



2. Reference the header file

Reference the SDK header file at the beginning of HelloSDK.cpp:

©2013-2018 Tencent Cloud. All rights reserved.

```
#include "TXLivePusher.h"
```

3. Add calling code

Declaration

```
#pragma once

#include <QtWidgets/QMainWindow>
#include "ui_HelloSDK.h"
#include "TXLivePusher.h"

class HelloSDK : public QMainWindow
{
Q_OBJECT

public:
HelloSDK(QWidget *parent = Q_NULLPTR);

private:
Ui::HelloSDKClass ui;
TXLivePusher m_pusher;
};
```

Implementation

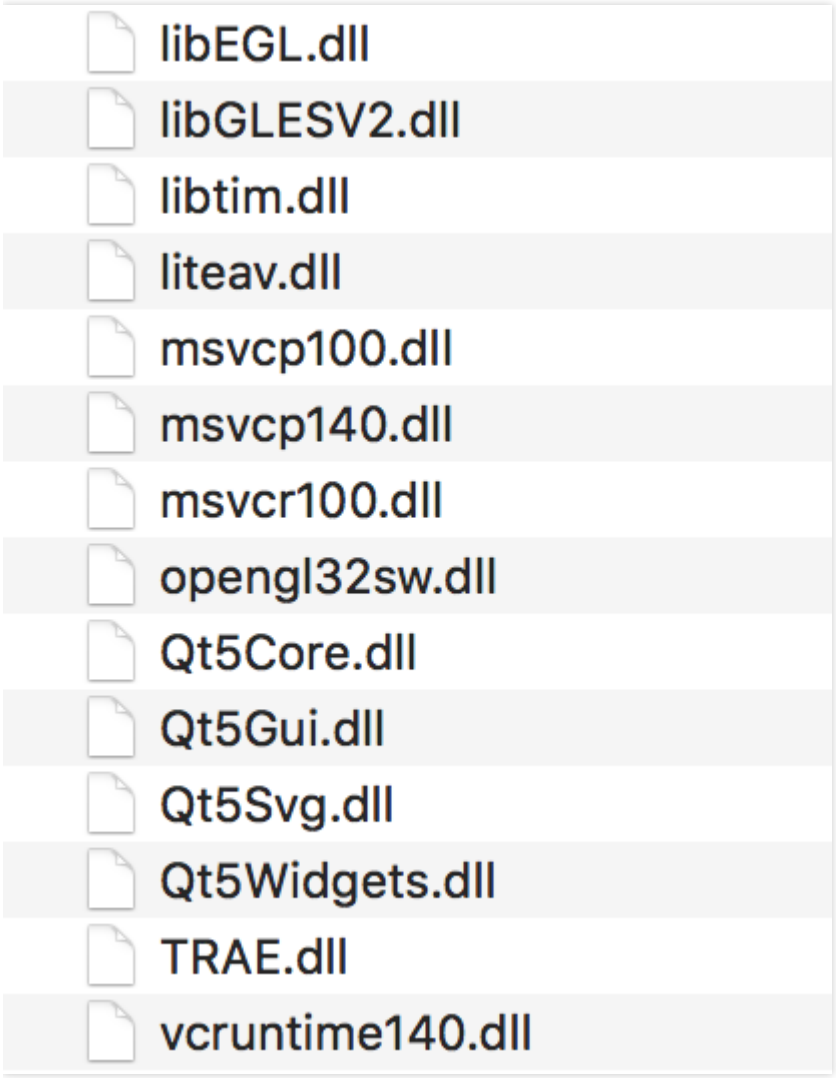
```
#include "HelloSDK.h"

HelloSDK::HelloSDK(QWidget *parent)
: QMainWindow(parent)
{
ui.setupUi(this);
m_pusher.startPreview((HWND)ui.widget->winId(), RECT{ 0, 0, ui.widget->width(), ui.widget->height()
}, 0);
}
```

4. Copy dll

Copy the dll required by the program from LiteAV to the release path, as shown below:

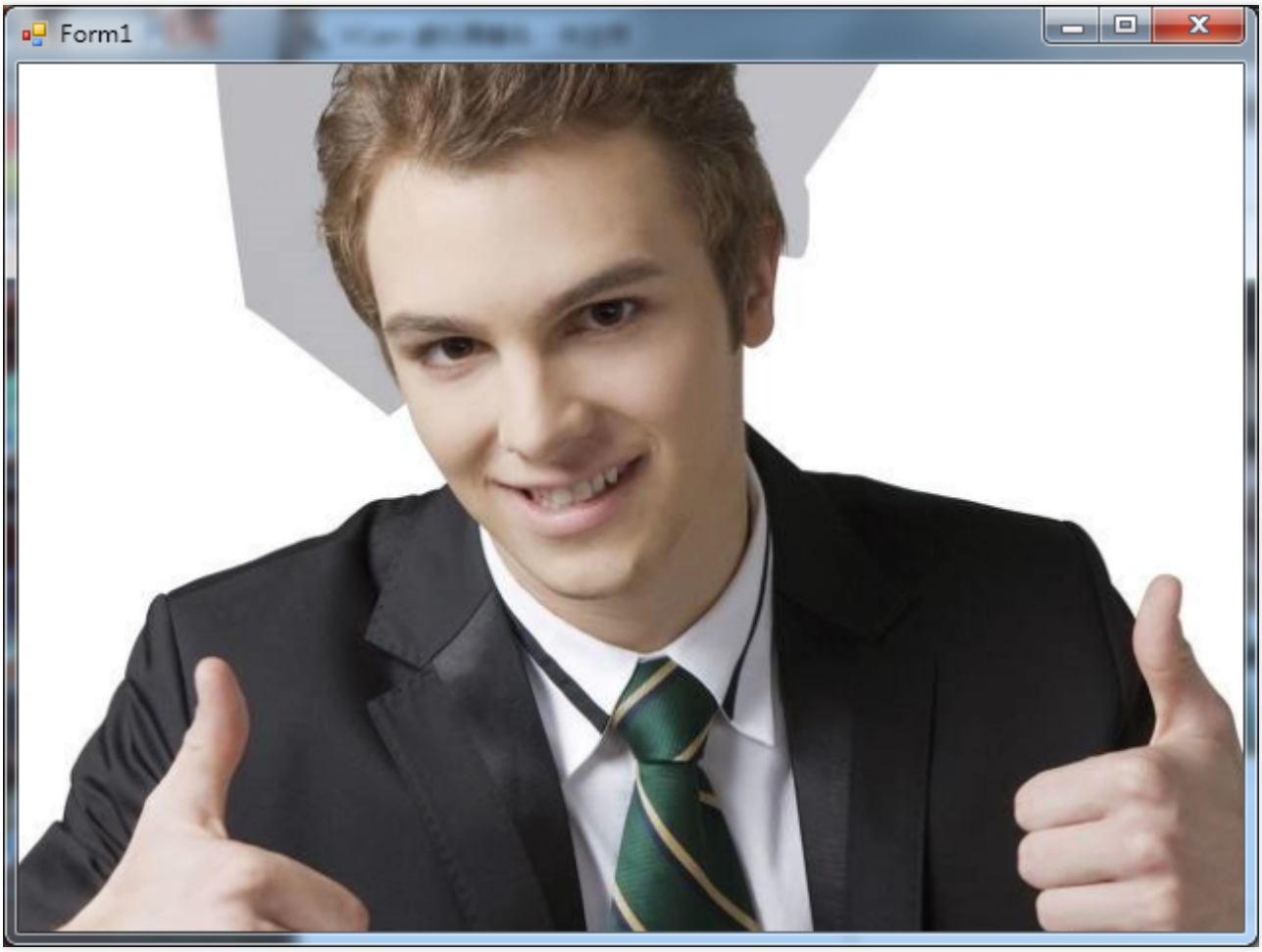
©2013-2018 Tencent Cloud. All rights reserved.



5. Compile and run the project

The HelloSDK project can be successfully compiled and run if you performed the previous steps correctly. Launch the project under Release, and then you can see the screen captured by the camera display on the program form:

©2013-2018 Tencent Cloud. All rights reserved.



©2013-2018 Tencent Cloud. All rights reserved.

Basic Features
API interface list

Last updated：2018-07-10 11:36:57

API List

API Document

Download SDK from the Tencent Cloud official website to view the API comments under SDK\include.

The following lists the APIs of Tencent Video Cloud Windows SDK (C++), which are categorized into TXLivePusher and TXLivePlayer APIs, as well as relevant enums and global constants.

TXLivePusher

APIs

Name	Description
setCallback(callback, pUserData)	Sets the callback proxy of TXLivePusher
enumCameras(camerasName, capacity)	Enumerates current cameras
micDeviceCount()	Queries the number of available microphones
micDeviceName(index, name[])	Queries the names of microphones
selectMicDevice(index)	Selects the specified microphone as the recording device
micVolume()	Queries the volume of the selected microphone
setMicVolume(volume)	Sets the volume of the selected microphone
enableMic(bEnable)	Indicates whether to enable microphone (used with audio mixing)
openSystemVoiceInput(szPlayerPath)	Enables system sound capture

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
closeSystemVoiceInput()	Disables system sound capture
setSystemVoiceInputVolume(value)	Sets the volume for system sound capture
startAudioCapture(srcType)	Enables audio capture
stopAudioCapture()	Disables audio capture
startPreview(rendHwnd, rect, cameraIndex)	Enables camera preview
updatePreview(rendHwnd, rect)	Resets camera preview area
stopPreview()	Disables camera preview
startPreview(srcType,rendHwnd,rect,userType)	Enables video preview, including screencap, camera and external data sources
setScreenCaptureParam(captureHwnd, rect)	Sets screen capture parameters
enumCaptureWindow(windowArray, capacity)	Enumerates windows available for capture
captureVideoSnapShot(filePath, length)	Captures screenshots of the pushed images and saves to local device
startPush(url)	Starts push
stopPush()	Stops push
switchCamera(cameraIndex)	Switches between cameras. Dynamic switching during push is supported.
setMute(mute)	Enables Mute
setRenderMode(mode)	Sets the rendering (fill) mode of image
setRotation(rotation)	Sets the clockwise rotation of image
setVideoQualityParamPreset(paramType)	Sets the preset options for pushed image quality
setVideoResolution(resolution)	Sets video resolution
setBeautyStyle(beautyStyle , beautyLevel, whitenessLevel)	Sets beauty filter and whitening effects
setRenderYMirror(bMirror)	Sets the mirroring effect for preview rendering

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
setOutputYMirror(bMirror)	Sets the mirroring effect of pushed images
setVideoBitRate(bitrate)	Sets video bitrate
setAutoAdjustStrategy(adjuststrategy)	Sets traffic control policy
setVideoBitRateMin(videoBitrateMin)	The minimum video bitrate of SDK output (used with setAutoAdjustStrategy)
setVideoBitRateMax(videoBitrateMax)	The maximum video bitrate of SDK output (used with setAutoAdjustStrategy)
setVideoFPS(fps)	Sets video frame rate
setPauseVideo(bPause)	Sets video pause

TXLivePlayer

APIs

Name	Description
setCallback(callback, pUserData)	Sets the callback proxy of TXLivePlayer
speakerDeviceCount()	Queries the number of available speakers
speakerDeviceName(index, name)	Queries the names of speakers
selectSpeakerDevice(index)	Selects the specified speaker as the audio playback device
speakerVolume()	Queries the volume for SDK playback
setSpeakerVolume(volume)	Sets the volume for SDK playback
setRenderFrame(hWnd, rect)	Sets video image rendering window and area
updateRenderFrame(hWnd, rect)	Resets image rendering area
closeRenderFrame()	Disables image rendering
startPlay(url, type)	Starts playback
stopPlay()	Stops playback
pause()	Pauses playback

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
resume()	Resumes playback
isPlaying()	Indicates whether the playback is in progress
setMute(mute)	Enables Mute
setRenderMode(mode)	Sets the rendering (fill) mode of image
setRotation(TXEVideoRotation)	Sets the clockwise rotation of image
setRenderYMirror(mirror)	Sets the mirroring effect for rendering
setOutputVideoFormat(format)	Sets the video encoding format
captureVideoSnapShot(filePath,length)	Captures screenshots of the pulled images and saves to local device

Details of TXLivePusher APIs

1.setCallback(callback, pUserData)

API details: void setCallback(callback, pUserData)
Sets the callback proxy of TXLivePusher for listening on push events.

- **Parameter description**

Parameter	Type	Description
callback	TXLivePusherCallback *	Proxy pointer
pUserData	void *	Transparently transfers user data to the callback function of TXLivePusherCallback

- **Sample code**:

```
pusher.setCallback(this, reinterpret_cast<void*>(index));
```

2.enumCameras(camerasName, capacity)

API details: int enumCameras(camerasName, capacity)

©2013-2018 Tencent Cloud. All rights reserved.

Enumerates current cameras. If multiple cameras are installed on a Windows machine, you can use this function to get the number and names of the available cameras.

- **Parameter description**

Parameter	Type	Description
camerasName	wchar_t **	Camera names
capacity	size_t	Size of camerasName array

- **Sample code**:

```
m_cameraCount = pusher.enumCameras();
wchar_t **camerasName = new wchar_t *[m_cameraCount];
for (int i = 0; i < m_cameraCount; ++i)
{
camerasName[i] = new wchar_t[256];
}
pusher.enumCameras(camerasName, m_cameraCount);
```

3. micDeviceCount()

API details: int micDeviceCount() const

Queries the number of available microphones.

- **Returned result**

Parameter	Type	Description
vRet	int	Number of microphones

- **Sample code**:

```
int ret = pusher.micDeviceCount();
```

4. micDeviceName(index,name)

API details: bool micDeviceName(unsigned int index, char name[SPEAKER_DEVICE_NAME_MAX_SIZE])

Queries the names of microphones.

©2013-2018 Tencent Cloud. All rights reserved.

- **Parameter description**

Parameter	Type	Description
index	int	Index of microphone
name	char[]	A string array of microphone names

- **Sample code**:

```
int index = 0;
char name[SPEAKER_DEVICE_NAME_MAX_SIZE];
pusher.micDeviceName(index, name);
```

5. selectMicDevice(index)

API details: void selectMicDevice(unsigned int index)

Selects the specified microphone as the audio recording device. When this API is not called, the microphone with index 0 is selected by default.

- **Parameter description**

Parameter	Type	Description
index	int	Index of microphone

- **Sample code**:

```
int index = 0;
pusher.selectMicDevice(index);
```

6. micVolume()

API details: unsigned int micVolume()

Queries the volume of the selected microphone.

- **Returned result**

Parameter	Type	Description

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
vRet	int	Volume value, which ranges from 0 to 65535.

- **Sample code**:

```
int ret = pusher.micVolume();
```

7. setMicVolume(volume)

API details: void setMicVolume(unsigned int volume)

Sets the volume of the selected microphone.

- **Parameter description**

Parameter	Type	Description
volume	int	The volume you set, which ranges from 0 to 65535.

- **Sample code**:

```
int volume = 100;
pusher.setMicVolume(volume);
```

8. enableMic(bEnable)

API details: void enableMic(bool bEnable)

Indicates whether to enable microphone.

- **Parameter description**

Parameter	Type	Description
bEnable	bool	false: Disable microphone capture; true: Enable microphone capture

- **Sample code**:

```
pusher.enableMic(true);
```

©2013-2018 Tencent Cloud. All rights reserved.

9. openSystemVoiceInput(szPlayerPath)

API details: void openSystemVoiceInput(const char* szPlayerPath)

Enables system or process sound input for mixing with the microphone sound input.

- **Parameter description**

Parameter	Type	Description
szPlayerPath	string	Player address. If this parameter is left empty or filled with null, all sounds in the system are captured. If the path of the exe program (e.g. Kugou Music or QQ Music) is entered, the program is started and only the sounds from the program are captured. \

- **Sample code**:

```
pusher.openSystemVoiceInput(null);
```

10. closeSystemVoiceInput()

API details: void closeSystemVoiceInput()

Disables system or process sound input for mixing.

- **Sample code**:

```
pusher.closeSystemVoiceInput();
```

11. setSystemVoiceInputVolume(value)

API details: void setSystemVoiceInputVolume(int value)

Sets the volume for system sound capture.

- **Parameter description**

Parameter	Type	Description
value	int	Sets the volume you desired. The value ranges from 0 to 100.

- **Sample code**:

©2013-2018 Tencent Cloud. All rights reserved.

```
pusher.setSystemVoiceInputVolume(50);
```

12. startAudioCapture(srcType)

API details: void startAudioCapture(TXEAudioCaptureSrcType srcType = TXE_AudioSrc_Sdk_Data)

Enables audio capture. The SDK uses a sampling rate of 48 KHz and 16-bit mono channel to deliver a real-time audio effect with a low delay.

- **Parameter description**

Parameter	Type	Description
srcType	enum	Audio data source: see the definition of TXEAudioCaptureSrcType.

- **Sample code**:

```
pusher.startAudioCapture(TXE_AudioSrc_Sdk_Data);
```

13. stopAudioCapture()

API details: void stopAudioCapture()

Disables audio capture.

- **Sample code**:

```
pusher.stopAudioCapture();
```

14.startPreview(rendHwnd, rect, cameraIndex)

API details: bool startPreview(rendHwnd, rect, cameraIndex)
Enables camera preview. true: API call is successful; false: API call failed.

- **Parameter description**

Parameter	Type	Description
rendHwnd	HWND	The HWND that identifies the preview window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
cameraIndex	int	Specifies the camera for which the preview is to be enabled.

- **Sample code**:

```
pusher.startPreview(m_pushRender, m_pushRect, 0);
```

15.updatePreview(rendHwnd, rect)

API details: void updatePreview(rendHwnd, rect)
Resets the preview area for camera. When the size of the window identified by the specified HWND changes, you can resize the video rendering area using this function.

- **Parameter description**

Parameter	Type	Description
rendHwnd	HWND	The HWND that identifies the preview window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

- **Sample code**:

```
pusher.updatePreview(m_pushRender, m_pushRect);
```

16.stopPreview()

API details: void stopPreview()

Disables camera preview. Calling this function before calling stopPush does not stop the push, but can cause the SDK to only push audio data.

- **Sample code**:

```
pusher.stopPreview();
```

17.startPreview(srcType,rendHwnd,rect,userType)

©2013-2018 Tencent Cloud. All rights reserved.

API details: bool startPreview(TXEVideoCaptureSrcType srcType, HWND rendHwnd, const RECT &rect, TXEVideoUserDataSrcType userType)
Enables the video source preview. The SDK supports various preview video sources, such as cameras and videos recorded by using screencap feature. true: API call is successful; false: API call failed.

- **Parameter description**

Parameter	Type	Description
srcType	enum	Video source type. See the definition of TXEVideoCaptureSrcType.
rendHwnd	HWND	The HWND that identifies the preview window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.
userType	enum	Reserved parameter

- **Sample code**:

```
pusher.startPreview(TXE_VideoSrc_Sdk_Camera, m_pushRender, m_pushRect, TXE_UserData_Undefined);
```

18.setScreenCaptureParam(captureHwnd,rect)

API details: void setScreenCaptureParam(HWND captureHwnd, const RECT &rect)

Sets the screen area capture parameters. This API is called before startPreview(srcType = TXE_VideoSrc_Sdk_Screen..). The entire desktop is captured by default.

- **Parameter description**

Parameter	Type	Description
captureHwnd	HWND	Specifies the window to be captured. If captureHwnd is not NULL, the window identified by it is captured as a whole, and the captureRect does not take effect.
rect	const RECT	Specifies the rectangle for capturing the area on main screen. It only takes effect when captureHwnd is NULL. When captureRect is {0}, the main screen is captured as a whole.

- **Sample code**:

©2013-2018 Tencent Cloud. All rights reserved.

```
pusher.setScreenCaptureParam(TXE_AudioSrc_Sdk_Data);
```

19.enumCaptureWindow(windowArray, capacity)

API details: static int enumCaptureWindow(HWND* windowArray,size_t capacity)

Enumerates the windows available for capture. If there are multiple windows on the desktop at the same time, this function obtains the handles and names of the windows that can be captured.

- **Parameter description**

Parameter	Type	Description
windowArray	HWND*	Array of handles and names of windows that can be captured
capacity	size_t	Size of the windowArray

- **Sample code**:

```
int iCntWnd = m_pusher.enumCaptureWindow();
HWND *hwndList = new HWND[iCntWnd];
pusher.enumCaptureWindow(hwndList, iCntWnd);
```

20. captureVideoSnapShot(filePath,length)

API details: int captureVideoSnapShot(wchar_t * filePath, unsigned int length)

Captures screenshots of the pushed images and saves to local device

- **Parameter description**

Parameter	Type	Description
filePath	wchar_t *	File path. Only .jpg format is supported.
length	unsigned int	Length of filePath

- **Sample code**:

```
std::wstring path = L"D:\\132.jpg";
pusher.captureVideoSnapShot(path.c_str(), path.size());
```

©2013-2018 Tencent Cloud. All rights reserved.

21.startPush(url)

API details: bool startPush(url)

Starts the push (you need to call startPreview to enable camera preview before calling startPush, otherwise only audio data is pushed)

Note: A push URL is exclusive, that is, a push URL can only be used by one pusher for pushing streams at a time.

true: API call is successful; false: API call failed. Memory allocation, resource request failure, and other reasons may result in the failure to return response.

- **Parameter description**

Parameter	Type	Description
url	const char *	A valid push URL that supports RTMP protocol. The URL starts with "rtmp://". For more information on how to obtain Tencent Cloud push URL, please see DOC.

- **Sample code**:

```
pusher.startPush(m_pushUrl.c_str());
```

22.stopPush()

API details: void stopPush()

Stops push.

- **Sample code**:

```
pusher.stopPush();
```

23.switchCamera(cameraIndex)

API details: void switchCamera(cameraIndex)

Switches between cameras. Dynamic switching during push is supported.

- **Parameter description**

Parameter	Type	Description

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
cameraIndex	int	Camera index. Returned result: 0-(number of cameras-1)

- **Sample code**:

```
pusher.switchCamera(0);
```

24.setMute(mute)

API details: void setMute(mute)

Enables Mute.

- **Parameter description**

Parameter	Type	Description
mute	bool	Indicates whether to enable Mute

- **Sample code**:

```
pusher.setMute(true);
```

25.setRenderMode(mode)

API details: void setRenderMode(mode)

Sets the rendering (fill) mode of image.

- **Parameter description**

Parameter	Type	Description
mode	TXERenderMode	See TXERenderMode's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
pusher.setRenderMode(TXE_RENDER_MODE_ADAPT);
```

©2013-2018 Tencent Cloud. All rights reserved.

26.setRotation(rotation)

API details: void setRotation(rotation)

Sets the clockwise rotation of image.

- **Parameter description**

Parameter	Type	Description
rotation	TXEVideoRotation	See TXEVideoRotation's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
pusher.setRotation(TXE_VIDEO_ROTATION_90);
```

27.setVideoQualityParamPreset(paramType)

API details: void setVideoQualityParamPreset(TXEVideoQualityParamPreset paramType)

Sets the preset options for pushed image quality.

- **Parameter description**

Parameter	Type	Description
paramType	TXEVideoQualityParamPreset	See TXEVideoQualityParamPreset's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
pusher.setVideoQualityParamPreset(TXE_VIDEO_QUALITY_STANDARD_DEFINITION);
```

28.setVideoResolution(resolution)

API details: void setVideoResolution(resolution)

Sets video resolution.

- **Parameter description**

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
resolution	TXEVideoResolution	See TXEVideoResolution's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
pusher.setVideoResolution(TXE_VIDEO_RESOLUTION_640x480);
```

29.setBeautyStyle(beautyStyle , beautyLevel, whitenessLevel)

API details: void setBeautyStyle(beautyStyle, beautyLevel, whitenessLevel)

Sets beauty filter and whitening effects.

- **Parameter description**

Parameter	Type	Description
beautyStyle	TXEBeautyStyle	See TXEBeautyStyle's enumerated values defined in TXLiveSDKTypeDef.h
beautyLevel	int	Beauty filter level: 1-9. 0 indicates disabling beauty filter. A greater value means a bigger effect.
whitenessLevel	Int	Whiteness level: 1-9. 0 indicates disabling whitening. A greater value means a bigger effect.

- **Sample code**:

```
pusher.setBeautyStyle(TXE_BEAUTY_STYLE_NATURE, 5, 5);
```

30.setRenderYMirror(mirror)

API details: void setRenderYMirror(mirror)

Sets the mirroring effect for preview rendering.

- **Parameter description**

Parameter	Type	Description

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
mirror	bool	true: The image is mirrored horizontally; false: The image remains as it is.

- **Sample code**:

```
pusher.setRenderYMirror(true);
```

31.setOutputYMirror(mirror)

API details: void setOutputYMirror(mirror)

Sets the mirroring effect of pushed image.

- **Parameter description**

Parameter	Type	Description
mirror	bool	true: the image is mirrored horizontally; false: the image remains as it is.

- **Sample code**:

```
pusher.setOutputYMirror(true);
```

32.setVideoBitRate(bitrate)

API details: void setVideoBitRate(bitrate)

Sets video bitrate. Note: The higher the bitrate, the clearer the image is. But a higher resolution does not necessarily mean a higher image clarity.

- **Parameter description**

Parameter	Type	Description
bitrate	unsigned long	Video bitrate (in Kbps). For example, a resolution of 640x360 is used with a video bitrate of 800 Kbps.

- **Sample code**:

©2013-2018 Tencent Cloud. All rights reserved.

```
pusher.setVideoBitRate(800);
```

33.setAutoAdjustStrategy(strategy)

API details: void setAutoAdjustStrategy(strategy)

Sets the traffic control policy, that is, whether to allow SDK to adjust the video bitrate to adapt to the network condition, so as to avoid the stutter caused by slow upload speed.

- **Parameter description**

Parameter	Type	Description
strategy	TXEAutoAdjustStrategy	See TXEAutoAdjustStrategy's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
pusher.setAutoAdjustStrategy(TXE_AUTO_ADJUST_REALTIME_VIDEOCHAT_STRATEGY);
```

34.setVideoBitRateMin(videoBitrateMin) && setVideoBitRateMax(videoBitrateMax)

API details: void setVideoBitRateMin(videoBitrateMin)

void setVideoBitRateMax(videoBitrateMax)

They are used with setAutoAdjustStrategy. If the AutoAdjust policy is specified as TXE_AUTO_ADJUST_NONE, calls of both of the functions are invalid.

- **Parameter description**

Parameter	Type	Description
videoBitrateMin	int	The minimum video bitrate of SDK output. For example, this parameter is set to 300 Kbps for a resolution of 640x360.
videoBitrateMax	int	The maximum video bitrate of SDK output. For example, this parameter is set to 1000 Kbps for a resolution of 640x360.

- **Sample code**:

©2013-2018 Tencent Cloud. All rights reserved.

```
pusher.setVideoBitRateMin(300);
pusher.setVideoBitRateMax(1000);
```

35.setVideoFPS(fps)

API details: void setVideoFPS(fps)

Sets video frame rate.

- **Parameter description**

Parameter	Type	Description
fps	unsigned long	Video frame rate, which takes effect after restart. Default: 15.

- **Sample code**:

```
pusher.setVideoFPS(15);
```

36.setPauseVideo(bPause)

API details: void setPauseVideo(bool bPause)

Sets video frame rate.

- **Parameter description**

Parameter	Type	Description
bPause	bool	Indicates whether to pause the video.

- **Sample code**:

```
pusher.setPauseVideo(true);
```

##

TXLivePlayer Object APIs

1.setCallback(callback, pUserData)

©2013-2018 Tencent Cloud. All rights reserved.

API details: void setCallback(callback, pUserData)

Sets the callback proxy of TXLivePlayer for listening on push events.

- **Parameter description**

Parameter	Type	Description
callback	TXLivePlayerCallback *	Proxy pointer
pUserData	void *	Transparently transfers user data to the callback function of TXLivePlayerCallback

- **Sample code**:

```
player.setCallback(this, reinterpret_cast<void*>(index));
```

2. speakerDeviceCount()

API details: int speakerDeviceCount() const

Queries the number of available speakers.

- **Returned result**

Parameter	Type	Description
vRet	int	Number of speakers

- **Sample code**:

```
int ret = player.speakerDeviceCount();
```

3. speakerDeviceName(index,name)

API details: bool speakerDeviceName(unsigned int index, char name[SPEAKER_DEVICE_NAME_MAX_SIZE])

Queries the names of speakers.

- **Parameter description**

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
index	int	Index of speaker
name	string	A string array of speaker names

- **Sample code**:

```
int index = 0;
char name[SPEAKER_DEVICE_NAME_MAX_SIZE];
player.speakerDeviceName(index, name);
```

4. selectSpeakerDevice(index)

API details: void selectSpeakerDevice(unsigned int index)

Selects the specified speaker as the audio playback device. When this API is not called, the speaker with index 0 is selected by default.

- **Parameter description**

Parameter	Type	Description
index	int	Index of speaker

- **Sample code**:

```
int index = 0;
player.selectSpeakerDevice(index);
```

5. speakerVolume()

API details: unsigned int speakerVolume()

Queries the playback volume of SDK. The value ranges from 0 to 65535.

- **Returned result**

Parameter	Type	Description
vRet	int	Volume value, which ranges from 0 to 65535.

©2013-2018 Tencent Cloud. All rights reserved.

- **Sample code**:

```
int ret = player.speakerVolume();
```

6. setSpeakerVolume(volume)

API details: void setSpeakerVolume(unsigned int volume);

Sets the playback volume of SDK. Note: This is not the volume of system speaker.

- **Parameter description**

Parameter	Type	Description
volume	int	The volume you set, which ranges from 0 to 65535.

- **Sample code**:

```
int volume = 100;
player.setSpeakerVolume(volume);
```

7.setRenderFrame(hWnd, rect)

API details: void setRenderFrame(hWnd, rect)

Sets video image rendering window and area.

- **Parameter description**

Parameter	Type	Description
hWnd	HWND	The HWND that identifies the video image window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

- **Sample code**:

```
player.setRenderFrame(rendHwnd, rect);
```

8.updateRenderFrame(hWnd, rect)

©2013-2018 Tencent Cloud. All rights reserved.

API details: void updateRenderFrame(hWnd, rect)

Resets the image rendering area. When the size of the window identified by the specified HWND changes, you can resize the video rendering area using this function.

- **Parameter description**

Parameter	Type	Description
hWnd	HWND	The HWND that identifies the video image window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

- **Sample code**:

```
player.updateRenderFrame(rendHwnd, rect);
```

9.closeRenderFrame()

API details: void closeRenderFrame()

Disables image rendering.

- **Sample code**:

```
player.closeRenderFrame();
```

10.startPlay(url, type)

API details: void startPlay(url, type)
Starts playback. You need to call setRenderFrame before calling startPlay.

- **Parameter description**

Parameter	Type	Description
url	const char *	A valid pull URL, which is the video playback URL. The URL starts with "rtmp://". For more information on how to obtain Tencent Cloud pull URL, please see DOC.
type	TXEPlayType	Playback type. See TXEPlayType's enumerated values defined in TXLiveSDKTypeDef.h

©2013-2018 Tencent Cloud. All rights reserved.

- **Sample code**:

```
player.startPlay(playUrl, PLAY_TYPE_LIVE_RTMP_ACC);
```

11.stopPlay()

API details: void stopPlay()

Stops playback.

- **Sample code**:

```
player.stopPlay();
```

12.pause()

API details: void pause()

Pauses playback.

- **Sample code**:

```
player.pause()
```

13.resume()

API details: void resume()

Resumes playback.

- **Sample code**:

```
player.resume()
```

14.isPlaying()

API details: bool isPlaying()

Indicates whether the playback is in progress.

true: in progress; false: not in progress.

- **Sample code**:

©2013-2018 Tencent Cloud. All rights reserved.

```
bool isPlaying = player.isPlaying();
```

15.setMute(mute)

API details: void setMute(mute)

- **Parameter description**

Parameter	Type	Description
mute	bool	Indicates whether to enable Mute

- **Sample code**:

```
player.setMute(true);
```

16.setRenderMode(mode)

API details: void setRenderMode(mode)

Sets the rendering (fill) mode of image.

- **Parameter description**

Parameter	Type	Description
mode	TXERenderMode	See TXERenderMode's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
player.setRenderMode(AX_TXE_RENDER_MODE_ADAPT);
```

17.setRotation(rotation)

API details: void setRotation(rotation)

Sets the clockwise rotation of image.

- **Parameter description**

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
rotation	TXEVideoRotation	See TXEVideoRotation's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
player.setRotation(TXE_VIDEO_ROTATION_90);
```

18.setRenderYMirror(mirror)

API details: void setRenderYMirror(mirror)

Sets the mirroring effect for preview rendering.

- **Parameter description**

Parameter	Type	Description
mirror	bool	true: the image is mirrored horizontally; false: the image remains as it is.

- **Sample code**:

```
player.setRenderYMirror(true);
```

19. setOutputVideoFormat(format)

API details: void setOutputVideoFormat(format)

Sets video encoding format. Default format: TXE_OUTPUT_VIDEO_WITHOUT_OUTPUT

- **Parameter description**

Parameter	Type	Description
format	TXEOutputVideoFormat	Specifies video encoding format. See TXEOutputVideoFormat's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

©2013-2018 Tencent Cloud. All rights reserved.

```
player.setOutputVideoFormat(TXE_OUTPUT_VIDEO_FORMAT_YUV420);
```

20. captureVideoSnapShot(filePath,length)

API details: int captureVideoSnapShot(wchar_t * filePath, unsigned int length)

Captures screenshots of pulled images and saves to local device.

- **Parameter description**

Parameter	Type	Description
filePath	wchar_t *	File path. Only .jpg format is supported.
length	unsigned int	Length of filePath

- **Sample code**:

```
std::wstring path = L"D:\\132.jpg";
player.captureVideoSnapShot(path.c_str(), path.size());
```

Enumeration Type Definition - API Parameters

TXEAudioCaptureSrcType

Audio data source types

TXE_AudioSrc_Sdk_Data = 1, Audio capture source is the sound data from LiteAvSDK, including **local** microphone, **local** player, **system**, etc.

TXEVideoCaptureSrcType

Video data source types

```
TXE_VideoSrc_Undefined = 0, No video capture source
TXE_VideoSrc_Sdk_Camera = 1, Video capture source is LiteAvSDK camera data.
TXE_VideoSrc_Sdk_Screen = 2, Video capture source is LiteAvSDK screencap data.
TXE_VidoeSrc_User_Data = 1001, Reserved parameter
```

TXEVideoUserDataSrcType

©2013-2018 Tencent Cloud. All rights reserved.

Reserved parameter has not been enabled.

```
TXE_UserData_Undefined = 0,
```

TXEVideoResolution

Push video resolutions

```
// Standard screen 4:3
TXE_VIDEO_RESOLUTION_320x240 = 1,
TXE_VIDEO_RESOLUTION_480x360 = 2,
TXE_VIDEO_RESOLUTION_640x480 = 3,
TXE_VIDEO_RESOLUTION_960x720 = 4,

// Wide screen 16:9
TXE_VIDEO_RESOLUTION_320x180 = 101,
TXE_VIDEO_RESOLUTION_480x272 = 102,
TXE_VIDEO_RESOLUTION_640x360 = 103,
TXE_VIDEO_RESOLUTION_960x540 = 104,

// Wide screen 9:16
TXE_VIDEO_RESOLUTION_180x320 = 201,
TXE_VIDEO_RESOLUTION_272x480 = 202,
TXE_VIDEO_RESOLUTION_360x640 = 203,
TXE_VIDEO_RESOLUTION_540x960 = 204,
```

TXERenderMode

Video rendering mode on window

```
TXE_RENDER_MODE_ADAPT = 1, // Adaption. The video image is displayed in full on the screen, poss
ibly with black edges around it.
TXE_RENDER_MODE_FILLSCREEN = 2, // Filling. No black edge exists, but the parts beyond the render
ing area are trimmed off, so that the image is centered on the screen.
```

TXEVideoRotation

Rendering video rotation

```
TXE_VIDEO_ROTATION_NONE = 1, // No rotation of the image.
TXE_VIDEO_ROTATION_90 = 2, // Rotates the image 90 degrees clockwise, with its width and height i
nterchanged.
```

©2013-2018 Tencent Cloud. All rights reserved.

```
TXE_VIDEO_ROTATION_180 = 3, // Rotates the image 180 degrees clockwise, with the image reversed upside down.
TXE_VIDEO_ROTATION_270 = 4, // Rotates the image 270 degrees clockwise, with its width and height interchanged.
```

TXEAutoAdjustStrategy

Traffic control policy for pushing videos

```
TXE_AUTO_ADJUST_NONE = -1, // No traffic control. The video bitrate specified by setVideoBitRate is always used.
TXE_AUTO_ADJUST_LIVEPUSH_STRATEGY = 0, // Suitable for common push scenarios in LVB. This policy has a lower sensitivity and adapts to the bandwidth change slowly. This is helpful to maintain the image clarity when the bandwidth fluctuates.
TXE_AUTO_ADJUST_LIVEPUSH_RESOLUTION_STRATEGY = 1, // Suitable for common push scenarios in LVB. Upgraded from LIVEPUSH_STRATEGY, this policy allows the SDK to adapt the resolution to the bitrate automatically.
TXE_AUTO_ADJUST_REALTIME_VIDEOCHAT_STRATEGY = 5, // Suitable for real-time video chats. It is used by VIDEO_QUALITY_REALTIME_VIDEOCHAT.
// This policy is highly sensitive to network condition and makes adaptive adjustment in case of any network fluctuation.
```

TXEVideoQualityParamPreset

Preset options for SDK push image quality

```
TXE_VIDEO_QUALITY_STANDARD_DEFINITION = 1, // SD: Suitable for customers who care more about smoothness.
TXE_VIDEO_QUALITY_HIGH_DEFINITION = 2, // HD: Suitable for customers who care more about image clarity.
TXE_VIDEO_QUALITY_SUPER_DEFINITION = 3, // SD: Not recommended unless you are watching videos on a large screen.
TXE_VIDEO_QUALITY_LINKMIC_MAIN_PUBLISHER = 4, // Joint broadcasting with the primary VJ
TXE_VIDEO_QUALITY_LINKMIC_SUB_PUBLISHER = 5, // Joint broadcasting with a secondary VJ
TXE_VIDEO_QUALITY_REALTIME_VIDEOCHAT = 6, // Real-time video chats
TXE_VIDEO_QUALITY_STILLIMAGE_DEFINITION = 7, // Suitable for scenarios featuring static image quality. Generally, the screenshots captured and pushed by users have few dynamic changes. With the algorithm available for adaptive bitrate and resolution, custom settings are not recommended.
```

TXEOutputVideoFormat

Sets the video output format

©2013-2018 Tencent Cloud. All rights reserved.

```
TXE_OUTPUT_VIDEO_WITHOUT_OUTPUT = 1, // No data output
TXE_OUTPUT_VIDEO_FORMAT_YUV420 = 2, // yuv420 format
TXE_OUTPUT_VIDEO_FORMAT_RGBA = 3, // RBGA format
```

TXEBeautyStyle

Beauty filter style

```
TXE_BEAUTY_STYLE_SMOOTH = 0, // Smooth
TXE_BEAUTY_STYLE_NATURE = 1, // Natural
TXE_BEAUTY_STYLE_BLUR = 2, // Hazy
```

TXEPlayType

Playback type

```
PLAY_TYPE_LIVE_RTMP = 0, RTMP live streaming
PLAY_TYPE_LIVE_RTMP_ACC, Accelerated RTMP live streaming
```

CallBack Event Definitions of callback parameters

PushEvent

Push events

```
TXE_STATUS_UPLOAD_EVENT : 200001, // Push-related data

PUSH_EVT_CONNECT_SUCC = 1001, // Connected to the push server
PUSH_EVT_PUSH_BEGIN = 1002, // Handshake with the server completed. Push starts.
PUSH_EVT_OPEN_CAMERA_SUCC = 1003, // Camera enabled successfully
PUSH_EVT_CHANGE_RESOLUTION = 1005, // Resolution is changed dynamically during push
PUSH_EVT_CHANGE_BITRATE = 1006, // Bitrate is changed dynamically during push
PUSH_EVT_FIRST_FRAME_AVAILABLE = 1007, // The first frame is captured
PUSH_EVT_START_VIDEO_ENCODER = 1008, // Encoder started
PUSH_EVT_CAMERA_REMOVED = 1009, // Camera removed (for PC SDK)
PUSH_EVT_CAMERA_AVAILABLE = 1010, // Camera is available again (for PC SDK)
PUSH_EVT_CAMERA_CLOSED = 1011, // Camera disabled (for PC SDK)
```

©2013-2018 Tencent Cloud. All rights reserved.

```
PUSH_EVT_SNAPSHOT_RESULT = 1012, // Error code returned for screencap

PUSH_ERR_OPEN_CAMERA_FAIL = -1301, // Failed to enable camera
PUSH_ERR_OPEN_MIC_FAIL = -1302, // Failed to enable microphone
PUSH_ERR_VIDEO_ENCODE_FAIL = -1303, // Video encoding failed
PUSH_ERR_AUDIO_ENCODE_FAIL = -1304, // Audio encoding failed
PUSH_ERR_UNSUPPORTED_RESOLUTION = -1305, // Unsupported video resolution
PUSH_ERR_UNSUPPORTED_SAMPLERATE = -1306, // Unsupported audio sampling rate
PUSH_ERR_NET_DISCONNECT = -1307, // Network disconnected. Too many failed reconnection attem
pts. Restart the push for more retries.
PUSH_ERR_CAMERA_OCCUPY = -1308, // The camera is in use. Enable another camera (for PC SDK)

PUSH_WARNING_NET_BUSY = 1101, // Bad network condition: data upload is blocked because upstre
am bandwidth is too small
PUSH_WARNING_RECONNECT = 1102, // Network disconnected. Auto reconnection has been initiate
d (no more attempts will be made after three failed attempts)
PUSH_WARNING_HW_ACCELERATION_FAIL = 1103, // Failed to start hard-encoding. Soft-encoding is
used instead.
PUSH_WARNING_VIDEO_ENCODE_FAIL = 1104, // Video encoding failed. Non-fatal error. Encoder will
be restarted internally.
PUSH_WARNING_BEAUTYSURFACE_VIEW_INIT_FAIL = 1105, // Video encoding bitrate exception
PUSH_WARNING_VIDEO_ENCODE_BITRATE_OVERFLOW = 1106, // Video encoding bitrate exception
PUSH_WARNING_DNS_FAIL = 3001, // RTMP - DNS resolution failed
PUSH_WARNING_SEVER_CONN_FAIL = 3002, // Failed to connect to RTMP server
PUSH_WARNING_SHAKE_FAIL = 3003, // Handshake with RTMP server failed
PUSH_WARNING_SERVER_DISCONNECT = 3004, // RTMP server disconnected automatically. Check th
e validity of push URL or the validity period of the hotlink protection.
PUSH_WARNING_SERVER_NO_DATA = 3005, // No data was sent within 30 seconds. Server disconnect
ed automatically.
```

PlayEvent

Playback events

```
TXE_STATUS_DOWNLOAD_EVENT : 200002, // Pull-related data

PLAY_EVT_CONNECT_SUCC : 2001, // Connected to the server
PLAY_EVT_RTMP_STREAM_BEGIN : 2002, // Connected to the server. Pull started.
PLAY_EVT_RCV_FIRST_I_FRAME : 2003, // The first video data packet (IDR) is rendered
PLAY_EVT_PLAY_BEGIN : 2004, // Video playback started
PLAY_EVT_PLAY_PROGRESS : 2005, // Video playback progress
PLAY_EVT_PLAY_END : 2006, // Video playback ended
PLAY_EVT_PLAY_LOADING : 2007, // Video playback loading
PLAY_EVT_START_VIDEO_DECODER : 2008, // Decoder started
PLAY_EVT_CHANGE_RESOLUTION : 2009, // Video resolution changed
```

©2013-2018 Tencent Cloud. All rights reserved.

```
PLAY_EVT_SNAPSHOT_RESULT : 2010, // Error code returned for screencap

PLAY_ERR_NET_DISCONNECT : -2301, // Network disconnected. Too many failed reconnection attemp
ts. Restart the playback for more retries.
PLAY_ERR_GET_RTMP_ACC_URL_FAIL : -2302, // Failed to get the accelerated pull address

PLAY_WARNING_VIDEO_DECODE_FAIL : 2101, // Failed to decode the current video frame
PLAY_WARNING_AUDIO_DECODE_FAIL : 2102, // Failed to decode the current audio frame
PLAY_WARNING_RECONNECT : 2103, // Network disconnected. Auto reconnection has been initiated
(no more attempts will be made after three failed attempts)
PLAY_WARNING_RECV_DATA_LAG : 2104, // Unstable inbound packet: This may be caused by insuffici
ent downstream bandwidth, or inconsistent outbound stream from the VJ end.
PLAY_WARNING_VIDEO_PLAY_LAG : 2105, // Stutter occurred during the video playback (user experie
nce)
PLAY_WARNING_HW_ACCELERATION_FAIL : 2106, // Failed to start hard decoding. Soft decoding is us
ed instead.
PLAY_WARNING_VIDEO_DISCONTINUITY : 2107, // Discontinuous sequence of video frames. Some fra
mes may be dropped.
PLAY_WARNING_FIRST_IDR_HW_DECODE_FAIL : 2108, // Hard decoding of the first I-frame of current
stream failed. Switched to soft decoding automatically.
PLAY_WARNING_DNS_FAIL : 3001, // RTMP - DNS resolution failed
PLAY_WARNING_SEVER_CONN_FAIL : 3002, // Failed to connect to RTMP server
PLAY_WARNING_SHAKE_FAIL : 3003, // Handshake with RTMP server failed
PLAY_WARNING_SERVER_DISCONNECT : 3004, // RTMP server disconnected automatically
```

TXE_STATUS_UPLOAD_EVENT && TXE_STATUS_DOWNLOAD_EVENT

Definitions of status keys

```
#define NET_STATUS_CPU_USAGE "CPU_USAGE" // CPU utilization
#define NET_STATUS_CPU_USAGE_D "CPU_USAGE_DEVICE" // Total CPU usage of device
#define NET_STATUS_VIDEO_BITRATE "VIDEO_BITRATE" // The output bitrate of the current vid
eo encoder - the video data bits produced by the encoder per sec (in Kbps).
#define NET_STATUS_AUDIO_BITRATE "AUDIO_BITRATE" // The output bitrate of the current au
dio encoder - the audio data bits produced by the encoder per sec (in Kbps).
#define NET_STATUS_VIDEO_FPS "VIDEO_FPS" // Current video frame rate - the number of frame
s produced by video encoder per sec.
#define NET_STATUS_VIDEO_GOP "VIDEO_GOP" // I-frame interval of current video: the interval
between I-frames of video encoder (in sec).
#define NET_STATUS_NET_SPEED "NET_SPEED" // Current network speed
#define NET_STATUS_NET_JITTER "NET_JITTER" // Network jitter. The greater the jitter, the mor
e unstable the network.
#define NET_STATUS_CACHE_SIZE "CACHE_SIZE" // Buffer size. A larger buffer indicates the curr
ent upstream bandwidth is not enough to consume the video data produced.
#define NET_STATUS_DROP_SIZE "DROP_SIZE"
```

©2013-2018 Tencent Cloud. All rights reserved.

```
#define NET_STATUS_VIDEO_WIDTH "VIDEO_WIDTH"
#define NET_STATUS_VIDEO_HEIGHT "VIDEO_HEIGHT"
#define NET_STATUS_SERVER_IP "SERVER_IP"
#define NET_STATUS_CODEC_CACHE "CODEC_CACHE" // Buffer size for encoding/decoding
#define NET_STATUS_CODEC_DROP_CNT "CODEC_DROP_CNT" // Encoding/decoding queue DRO
PCNT
#define NET_STATUS_SET_VIDEO_BITRATE "SET_VIDEO_BITRATE"
```

TXLivePusherCallback

Listening on push events

API Definition	Description
onEventCallback(eventId, paramCount, paramKeys, paramValues, pUserData)	Push event notification of TXLivePusher

TXLivePlayerCallback

Listening on playback events

API Definition	Description
onEventCallback(eventId, paramCount, paramKeys, paramValues, pUserData)	Playback event notification of TXLivePlayer

©2013-2018 Tencent Cloud. All rights reserved.

Push and Play

Last updated：2018-07-11 11:18:25

Push and Playback

Tencent Video Cloud SDK is an audio/video SDK component that integrates **standard push and pull** and **real-time video call**. This document mainly introduces the **standard push and pull** integration solution of C++ version.

Product Description

Compared with Obs Studio (a commonly used push software) and Flash player (a common playback plug-in for PC browsers), the significant advantages of the Video Cloud SDK are:

- Low-delay optimization: Real-time audio/video calls.
- UDP protocol acceleration: Better push stability.

User Experience

Download the Windows SDK+Demo and zip package of C++ version. Decompress the package and open the LiteAVDemo(Qt).exe program. The home page contains such modes as **Push + Pull**, **LVB Experience Room**, **Two-person Audio & Video** and **Multi-person Audio & Video**.

After selecting **Push +Pull**, click the **New** button to get a new test push address. After enabling push and playback, Demo is shown as follows:

©2013-2018 Tencent Cloud. All rights reserved.

Interfacing

- Step 1: Go to the LVB Console to activate Tencent Cloud LVB service. For more information on how to activate the service, please see DOC.
- Step 2: Go to LVB Console -> LVB Code Access -> Access Configuration to configure push hotlink protection KEY and API authentication KEY. For configuration method, please see DOC. These URLs are automatically generated by your backend business server.
- Step 3: Go to LVB Console -> LVB Code Access -> Push Generator to generate a Push URL for test. Constructing a playback URL is as simple as constructing a push URL, except that the sub-domain name needs to be changed from **livepush** to **liveplay**.
- Step 4: Download SDK package, and embed the SDK in your APP development project according to Project Configuration. Then perform the related settings and access of push and playback according to the **code interfacing** section of this document.

©2013-2018 Tencent Cloud. All rights reserved.

- Step 5: You can also download source code of Demo in Windows SDK+Demo for reference.

Push

Code interfacing

Step 1: Create a TXLivePusher object

Create a **TXLivePusher** object, which will be used later to complete the push task.

```
TXLivePusher pusher;
```

After the creation is completed, you can call the relevant API of TXLivePusher to set the mirror effect, bitrate, resolution and fill mode before starting push. For more information on specific APIs, please see API List.

```
pusher.setRenderYMirror(true);
pusher.setOutputYMirror(true);
pusher.setVideoBitRate(900);
pusher.setVideoBitRateMin(300);
pusher.setVideoBitRateMax(1000);
pusher.setVideoResolution(TXE_VIDEO_RESOLUTION_640x480);
pusher.setRenderMode(TXE_RENDER_MODE_FILLSCREEN);
...
```

Step 2: Set the preview area

Next, we need to look for a place to display the images of camera. For Windows, HWND window handle is used as the basic rendering unit. In a Qt form application, each control can call its WinID() to get the HWND. Therefore, you only need to prepare and pass a HWND and the rendering area to the **startPreview** API of the TXLivePusher object.

```
// Set callback for getting push event notification
pusher.setCallback(this, reinterpret_cast<void*>(index));
// Enable the camera, and the camera index value can be obtained via the enumCameras API of the T
XLivePusher object.
// When only one camera is accessed, the camera index value is 0 by default.
pusher.startPreview((HWND)ui.widget_video_push->winId(),
RECT{ 0, 0, ui.widget_video_push->width(), ui.widget_video_push->height() }, m_cameraIndex);
```

Step 3: Start push

©2013-2018 Tencent Cloud. All rights reserved.

After the preparations in Step 1 and Step 2, you can use the following codes to start the push:

```
// rtmpURL is the push URL generated during the interfacing process. You can also use the New featu
re in Demo to generate a test URL to experience it.
Qstring rtmpURL = "rtmp://3891.livepush.myqcloud.com/live/xxxxxx";
pusher.startPush(rtmpURL.toLocal8Bit());
```

Step 4: Control the camera

If there are multiple cameras in a computer, you can switch between different cameras simply by calling the **switchCamera** API of TXLivePusher object and passing the index value of the camera.

```
// Pass the index value of camera to be switched to
int index = ui.cmb_cameras->currentIndex();
pusher.switchCamera(index);
```

The enumCameras function of TXLivePusher can get the index value of the camera. For specific examples, please see API List.

- Enabling a USB camera under Windows takes a long time for circuit and drive startup. The recommended practice is to specify cameraIndex as -1 when calling the startPreview of TXLivePusher to enable all cameras, and then use switchCamera to instantly switch the camera.
- SDK supports virtual cameras, whose enabling method is the same as that of normal cameras. If your PC does not have a physical camera, you can install virtual camera software such as VCam to assist debugging.

Step 5: End push

Ending a push is simple but very important (for example, we need to release the occupied camera hardware). Improper cleanup may cause problems in the next push.

```
// Set callback as null
pusher.setCallback(NULL, NULL);
// Stop camera preview
pusher.stopPreview();
// Stop push
pusher.stopPush();
```

Step 6: More features

For more information on such features of the local camera as screen mirroring, mute and sharpness settings, please see API List.

©2013-2018 Tencent Cloud. All rights reserved.

Event Handling

SDK monitors push-related events using the setCallback API of TXLivePusher object.

1. Normal events

Events that are always prompted during successful pushes. For example, receiving 1003 means that the system will start rendering the camera pictures.

Event ID	Value	Description
PUSH_EVT_CONNECT_SUCC	1001	Successfully connected to push server
PUSH_EVT_PUSH_BEGIN	1002	Handshake with the server completed, and ready to start push
PUSH_EVT_OPEN_CAMERA_SUCC	1003	Camera is enabled
PUSH_EVT_CHANGE_RESOLUTION	1005	Dynamic push resolution adjustment
PUSH_EVT_CHANGE_BITRATE	1006	Dynamic push bitrate adjustment
PUSH_EVT_FIRST_FRAME_AVAILABLE	1007	The first frame is captured
PUSH_EVT_START_VIDEO_ENCODER	1008	Encoder is started
PUSH_EVT_CAMERA_REMOVED	1009	Camera device has been removed
PUSH_EVT_CAMERA_AVAILABLE	1010	Camera device is available again
PUSH_EVT_CAMERA_CLOSED	1011	Camera is disabled
PUSH_EVT_SNAPSHOT_RESULT	1012	Error code of screenshot snapshot

2. Error notification

SDK has found some serious problems, and the push cannot be continued. For example, the camera cannot be enabled as Camera has been occupied by other programs.

Event ID	Value	Description
PUSH_ERR_OPEN_CAMERA_FAIL	-1301	Failed to start the camera
PUSH_ERR_OPEN_MIC_FAIL	-1302	Failed to start the microphone
PUSH_ERR_VIDEO_ENCODE_FAIL	-1303	Video encoding failed
PUSH_ERR_AUDIO_ENCODE_FAIL	-1304	Audio encoding failed

©2013-2018 Tencent Cloud. All rights reserved.

Event ID	Value	Description
PUSH_ERR_UNSUPPORTED_RESOLUTION	-1305	Unsupported video resolution
PUSH_ERR_UNSUPPORTED_SAMPLERATE	-1306	Unsupported audio sampling rate
PUSH_ERR_NET_DISCONNECT	-1307	Network disconnected. Reconnection attempts have failed for many times, thus no more retries will be performed. Please restart the push manually
PUSH_ERR_CAMERA_OCCUPY	-1308	The camera is being occupied. Enable another camera.

3. Warning events

Compared with the error message, issues represented by WARNING events usually do not interrupt the push process, and SDK generally initiates automatic repair logic internally. You don't need to care about most WARNING events, but the following two events are important and meaningful:

WARNING_NET_BUSY

Poor upstream network speed. It represents that the user's current audio/video data cannot be pushed smoothly to the server. In this case, generally the user can be given some UI prompts, for example, "Your network speed is not good" so as to allow the user to prepare for the lag on the other end.

WARNING_SERVER_DISCONNECT

The push request is rejected by the backend. This is usually because that txSecret in the push address is calculated incorrectly, or the push address is occupied by others (push URL is exclusive and a push URL can only be used by one user at a time).

Event ID	Value	Description
PUSH_WARNING_NET_BUSY	1101	Bad network connection: data upload is blocked because upstream bandwidth is too small
PUSH_WARNING_RECONNECT	1102	Network disconnected. Auto reconnection has been initiated (no more attempts will be made after three failed attempts)
PUSH_WARNING_HW_ACCELERATION_FAIL	1103	Failed to start hard-encoding. Soft-encoding is used.

©2013-2018 Tencent Cloud. All rights reserved.

Event ID	Value	Description
PUSH_WARNING_VIDEO_ENCODE_FAIL	1104	Video encoding failed. Non-fatal error. Encoder will be restarted internally.
PUSH_WARNING_BEAUTYSURFACE_VIEW_INIT_FAIL	1105	Warning of video encoding bitrate error
PUSH_WARNING_VIDEO_ENCODE_BITRATE_OVERFLOW	1106	Warning of video encoding bitrate error
PUSH_WARNING_DNS_FAIL	3001	RTMP - DNS resolution failed (this will trigger retry process)
PUSH_WARNING_SEVER_CONN_FAIL	3002	Failed to connect to the RTMP server (this will trigger retry process)
PUSH_WARNING_SHAKE_FAIL	3003	Handshake with RTMP server failed (this triggers a retry)
PUSH_WARNING_SERVER_DISCONNECT	3004	RTMP server disconnected automatically. Check the validity of push URL or the validity period of hotlink protection
PUSH_WARNING_SERVER_NO_DATA	3005	Actively disconnected for no data is sent for more than 30s.

Playback Feature

Code interfacing

Step 1: Create a TXLivePlayer object

Create a **TXLivePlayer** object, which will be used later to complete the push task.

```
TXLivePlayer player;
```

Step 2: Set the rendering area

Similar to push, we need to look for a place to display the images of camera. For Windows, HWND window handle is used as the basic rendering unit. In a Qt form application, each control can call its

©2013-2018 Tencent Cloud. All rights reserved.

WinID() to get the HWND. You only need to prepare and pass a HWND and the rendering area to the **setRenderFrame** API of the TXLivePusher object.

```
// Set callback to get playback event notification
player.setCallback(this, reinterpret_cast<void*>(index));
// Pass HWND window handle, rendering position and size
player.setRenderFrame((HWND)ui.widget_video_play->winId(),
RECT{ 0, 0, ui.widget_video_play->width(), ui.widget_video_play->height() });
```

Step 3: CDN playback

To start the playback feature, simply call the startPlay API of TXLivePlayer, where PLAY_TYPE_LIVE_RTMP is the RTMP address of the regular CDN. The advantage is that the bandwidth price is very low, but the delay is usually high.

```
// rtmpURL is the playback URL generated during the interfacing process.
string rtmpURL = "rtmp://3891.liveplay.myqcloud.com/live/xxxxxx";
player.startPlay(rtmpURL,PLAY_TYPE_LIVE_RTMP);
```

Step 3': Ultra-low delay playback

Ultra-low delay playback can pull the end-to-end delay down to about 400ms. In this mode, the internal mechanism of SDK is completely different from that in the normal mode. Meanwhile, the audio/video lines used by SDK are not regular CDN lines. Compared with CDN, the unit price is higher. It is generally applicable to audio/video calls and scenarios that highly requiring low delay.

```
// rtmpURL is the playback URL with hotlink protection signature generated during the interfacing pr
ocess. You can also use the New feature in Demo to generate a test URL to experience it.
string rtmpURL = "rtmp://3891.liveplay.myqcloud.com/live/xxxxxxbizid=2157&
txSecret=xxxxxx&txTime=xxxxxx";
player.startPlay(rtmpURL,PLAY_TYPE_LIVE_RTMP_ACC);
```

Notes

- This feature does not need to be enabled in advance, but it requires that LVB stream be located in Tencent Cloud. Implementing low-latency linkage across cloud providers is not just a technical difficulty.
- The playback URL cannot be a normal CDN URL. It must have a hotlink protection signature.
- When calling the startPlay API, specify type as **PLAY_TYPE_LIVE_RTMP_ACC** for SDK to pull an ultra-low delay LVB stream.
- It supports concurrence playback of 10 channels simultaneously at most. So you can only use it in interactive scenarios (such as the channel of joint broadcasting VJ or the operator in Prize Claw LVB).

©2013-2018 Tencent Cloud. All rights reserved.

- The RTMP stream launched by Obs Studio introduces a delay of more than 1s on the client. Therefore, it is recommended to use the video cloud SDK for push.

- **updateRenderFrame**
  When the window size specified by the HWND changes, the video rendering area can be rescaled via the **updateRenderFrame** function.
- **setRenderMode**

Step 4: End playback

When ending a playback, the stopPlay API of TXLivePlayer needs to be called to release the resources (such as network downloading).

```
// Set callback as null
player.setCallback(NULL, NULL);
// Stop pull playback
player.stopPlay();
```

Step 5: More features

For more information, please see API List.

Event Handling

Monitor playback-related events using the setCallback API of TXLivePlayer object.

1. Normal events

Events that may be notified for successful playback.

Event ID	Value	Description
PLAY_EVT_CONNECT_SUCC	2001	Connected to the server
PLAY_EVT_RTMP_STREAM_BEGIN	2002	Connected to the server and started to pull stream
PLAY_EVT_RCV_FIRST_I_FRAME	2003	Render the first video packet (IDR)
PLAY_EVT_PLAY_BEGIN	2004	Video playback starts
PLAY_EVT_PLAY_PROGRESS	2005	Video playback progress
PLAY_EVT_PLAY_END	2006	Video playback ends
PLAY_EVT_PLAY_LOADING	2007	Video playback loading

©2013-2018 Tencent Cloud. All rights reserved.

Event ID	Value	Description
PLAY_EVT_START_VIDEO_DECODER	2008	Decoder starts
PLAY_EVT_CHANGE_RESOLUTION	2009	Video resolution changes
PLAY_EVT_SNAPSHOT_RESULT	2010	Error code of screenshot snapshot

2. Error notification

Some fatal errors occurred with SDK, such as network disconnection, can cause the failure to continue playback. In this case, you need to deal with these errors accordingly.

Event ID	Value	Description
PLAY_ERR_NET_DISCONNECT	-2301	The network is disconnected and reconnection attempts failed, which will result in playback failure.
PLAY_ERR_GET_RTMP_ACC_URL_FAIL	-2302	Failed to acquire pull accelerating address, which will result in playback failure

3. Warning events

SDK has found some non-fatal errors that generally do not end playback. You don't need to care about most WARNING events, but the following two events are important and meaningful:

PLAY_WARNING_VIDEO_PLAY_LAG

The event signal for SDK to inform the playback stutter externally, which means that the stutter of video screen (the refresh time between two frames) exceeds 500ms.

PLAY_WARNING_SERVER_DISCONNECT

The playback request is rejected by the backend. In this case, you need to check whether the playback URL is valid.

Event ID	Value	Description
PLAY_WARNING_VIDEO_DECODE_FAIL	2101	Failed to decode the current video frame
PLAY_WARNING_AUDIO_DECODE_FAIL	2102	Failed to decode the current audio frame
PLAY_WARNING_RECONNECT	2103	Network disconnected. Auto reconnection has been initiated (no more attempts will be made after three failed attempts)

©2013-2018 Tencent Cloud. All rights reserved.

Event ID	Value	Description
PLAY_WARNING_RECV_DATA_LAG	2104	Unstable incoming packet from network: This may be caused by insufficient downstream bandwidth, or unstable outgoing stream at the VJ end
PLAY_WARNING_VIDEO_PLAY_LAG	2105	Stutter occurred during the current video playback (intuitive user experience)
PLAY_WARNING_HW_ACCELERATION_FAIL	2106	Failed to start hard-decoding; Soft-decoding is used (not supported for now)
PLAY_WARNING_VIDEO_DISCONTINUITY	2107	Current video frames are discontinuous and frame loss may occur
PLAY_WARNING_FIRST_IDR_HW_DECODE_FAIL	2108	Hard decoding of Frame I for the current stream fails, and SDK automatically switches to soft decoding
PLAY_WARNING_DNS_FAIL	3001	RTMP - DNS resolution failed
PLAY_WARNING_SEVER_CONN_FAIL	3002	Failed to connect to the RTMP server
PLAY_WARNING_SHAKE_FAIL	3003	Handshake with RTMP server failed
PLAY_WARNING_SERVER_DISCONNECT	3004	The RTMP server is actively disconnected

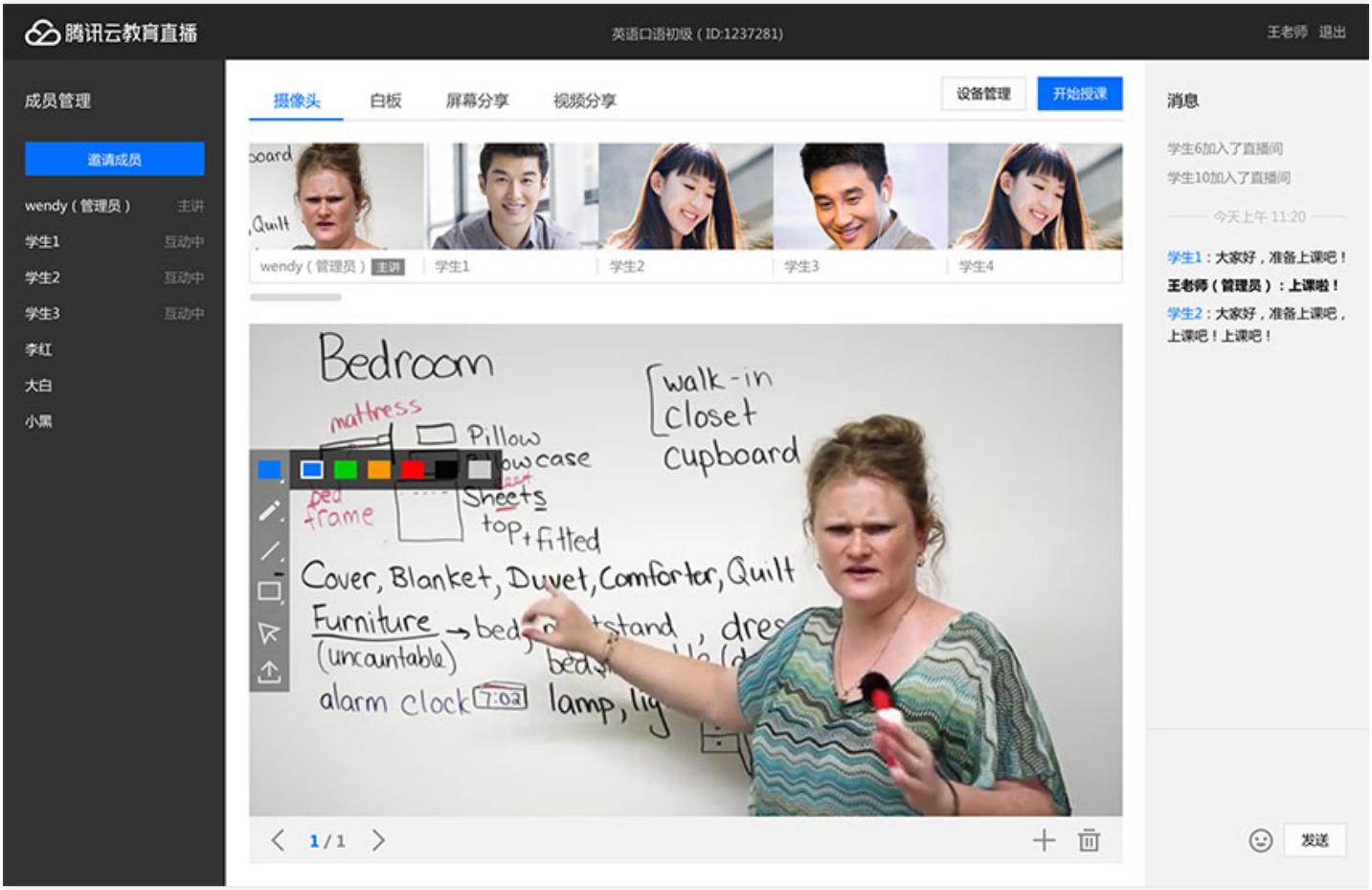
©2013-2018 Tencent Cloud. All rights reserved.

RealTime LiveRoom

Last updated：2018-07-11 11:18:50

LVB+Joint Broadcasting (LiveRoom)

LVB+Joint Broadcasting is an LVB mode commonly used in the **Live Show** and **Online Education** scenarios. With a good applicability to many scenarios, it supports online live broadcasting featuring both high concurrency and low cost, but also enables video chats between VJs and viewers via joint broadcasting.



Tencent Cloud provides "LVB+Joint Broadcasting" by using **LiveRoom**, a component consisting of Client and Server (both open source). For more information on how to interface with it, please see DOC. This document displays the API list for Client:

©2013-2018 Tencent Cloud. All rights reserved.

Download the SDK package on Tencent Cloud's official website. LiveRoom related header files and source code files are included in SDK\Rooms\LiveRoom.

LiveRoom

Name	Description
instance()	Gets a single instance of LiveRoom and calls the API of LiveRoom through the single instance.
setCallback(ILiveRoomCallback * callback)	Sets the callback proxy for LiveRoom callback, and listens to the internal status of LiveRoom and the execution results of the API.
login(const std::string & serverDomain, const LRAuthData & authData, ILoginLiveCallback* callback)	Logs in to the business server RoomService before you can normally use other APIs and the IM feature.
logout()	Logs out of the business server RoomService. Make sure to call "logout" after "leaveRoom" is called, otherwise the call of leaveRoom will fail.
getRoomList(int index, int count, IGetLiveRoomListCallback* callback)	Gets the room list. If there are many rooms, you can get the list in pages.
void getAudienceList(const std::string& roomID)	Gets the list of viewers and returns only the last 30 viewers who entered the room.
createRoom(const std::string& roomID, const std::string& roomInfo)	Creates a room, and the new room will be added to the room list in the backend, and then VJs can start pushing.
enterRoom(const std::string& roomID, HWND rendHwnd, const RECT & rect)	Enters a room
leaveRoom()	Leaves the room. If the primary VJ leaves, the room will be terminated by the backend; if secondary VJs or viewers leave, the remaining participants can continue watching.
sendRoomTextMsg(const char * msg)	Sends plain text messages, such as sending on-screen comments under LVB scenarios

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
sendRoomCustomMsg(const char *cmd, const char* msg)	Sends custom messages, such as giving a "Like" or flower under LVB scenarios.
startLocalPreview(HWND rendHwnd, const RECT & rect)	Enables the default camera preview
updateLocalPreview(HWND rendHwnd, const RECT &rect)	Resets the preview area for camera. When the size of the window identified by the specified local HWND changes, you can resize the video rendering area using this function.
stopLocalPreview()	Disables camera preview
startScreenPreview(HWND rendHwnd, HWND captureHwnd, const RECT & renderRect, const RECT & captureRect)	Enables screen sharing
stopScreenPreview()	Disables screen sharing
addRemoteView(HWND rendHwnd, const RECT & rect, const char * userID)	Plays videos of other VJs in the room
removeRemoteView(const char * userID)	Stops playing videos of other VJs
setMute(bool mute)	Enables Mute
setVideoQuality(LRVideoQuality quality, LRAspectRatio ratio)	Sets the preset options for image quality
setBeautyStyle(TXEBeautyStyle beautyStyle, int beautyLevel, int whitenessLevel)	Sets beauty filter and whitening effects
requestJoinPusher()	Joint broadcasting request from viewer
acceptJoinPusher(const std::string& userID)	The primary VJ accepts the joint broadcasting request and informs the request initiator
rejectJoinPusher(const std::string& userID, const std::string& reason)	The primary VJ rejects the joint broadcasting request and informs the request initiator
kickoutSubPusher(const std::string& userID)	The primary VJ kicks out a secondary VJ

©2013-2018 Tencent Cloud. All rights reserved.

ILoginLiveCallback

Name	Description
onLogin(const LRResult& res, const LRAuthData& authData)	Callback of login result

IGetLiveRoomListCallback

Name	Description
onGetRoomList(const LRResult& res, const std::vector& rooms)	Callback of obtaining a room list

ILiveRoomCallback

Name	Description
onCreateRoom(const LRResult& res, const std::string& roomID)	Callback of creating a room
onEnterRoom(const LRResult& res)	Callback of entering a room
onUpdateRoomData(const LRResult& res, const LRRoomData& roomData)	Callback of room information change
void onGetAudienceList(const LRResult& res, const std::vector& audiences)	Callback of viewer list query
onPusherJoin(const LRMemberData& member)	Callback of joint broadcasting viewer entering the room
void onPusherQuit(const LRMemberData& member)	Callback of joint broadcasting viewer exiting the room
onRoomClosed(const std::string& roomID)	Callback of room dissolution
onRecvRoomTextMsg(const char *roomID, const char* userID, const char *userName, const char* userAvatar, const char * msg)	Receives plain text messages
onRecvRoomCustomMsg(const char *roomID, const char* userID, const char *userName, const char* userAvatar, const char *cmd, const char* message)	Receives custom messages

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
onError(const LRResult& res, const std::string& userID)	Notification of internal LiveRoom error
onRecvJoinPusherRequest(const std::string& roomID, const std::string& userID, const std::string& userName, const std::string& userAvatar)	Receives a joint broadcasting request
onRecvAcceptJoinPusher(const std::string& roomID, const std::string& msg)	Receives the notification of accepting a joint broadcasting request
onRecvRejectJoinPusher(const std::string& roomID, const std::string& msg)	Receives the notification of rejecting a joint broadcasting request
onRecvKickoutSubPusher(const std::string& roomID)	Receives the notification of the primary VJ kicking out a secondary VJ

Details of LiveRoom Object APIs

1. instance

- Definition: static LiveRoom* instance()
- Description: Gets a single instance of LiveRoom and calls the API of LiveRoom through the single instance
- Example:

```
LiveRoom* m_liveRoom = NULL;
m_liveRoom = LiveRoom::instance();
```

2. setCallback

- Definition: void setCallback(ILiveRoomCallback * callback)

©2013-2018 Tencent Cloud. All rights reserved.

- Description: Sets the callback proxy for LiveRoom callback, and listens to the internal status of LiveRoom and the execution results of the API

- Parameter:

Parameter	Type	Description
callback	ILiveRoomCallback *	Proxy pointer of ILiveRoomCallback type

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
MainDialog();
virtual ~MainDialog();

virtual void onCreateRoom(const LRResult& res, const std::string& roomID);
virtual void onEnterRoom(const LRResult& res);
virtual void onUpdateRoomData(const LRResult& res, const LRRoomData& roomData);
...
};

MainDialog::MainDialog()
{
m_liveRoom->setCallback(this); //Set callback
}

...
```

3. login

- Definition: void login(const std::string & serverDomain, const LRAuthData & authData, ILoginLiveCallback* callback)

- Description: Logs in to the business server RoomService before you can normally use other APIs and the IM feature

©2013-2018 Tencent Cloud. All rights reserved.

- Parameters:

Parameter	Type	Description
serverDomain	const std::string &	URL of RoomService. In consideration of security, it is recommended to access the https encrypted link. For more information, please see DOC.
authData	const LRAuthData &	Login information provided by RoomService, including IM related configuration fields. The token field can be obtained after a success login. For more information, please see DOC.
callback	ILoginLiveCallback*	Proxy pointer of ILoginLiveCallback type, used to call back the login result

- Example:

```
std::string serverDomain = "https://roomtest.qcloud.com/weapp/live_room";

m_authData.sdkAppID = sdkAppID;
m_authData.accountType = accountType;
m_authData.userID = userID;
m_authData.userSig = userSig;
m_authData.userName = userName;
m_authData.userAvatar = userAvatar;
m_authData.token = token;

// This class implements the ILoginCallback API to obtain login results
m_liveRoom->login(serverDomain, m_authData, this);
```

4. logout

- Definition: void logout();

- Description: Logs out of the business server RoomService. Make sure to call "logout" after "leaveRoom" is called, otherwise the call of leaveRoom will fail.

- Example:

```
m_liveRoom->logout();
```

©2013-2018 Tencent Cloud. All rights reserved.

5. getRoomList

- Definition: void getRoomList(int index, int count, IGetLiveRoomListCallback* callback)
- Description: Gets the room list. If there are many rooms, you can get the list in pages.
- Parameters:

Parameter	Type	Description
index	int	Retrieval by page. The initial default can be set to 0, and the subsequent retrieval is the initial room index (for example, set index=0 and cnt=5 for the first retrieval, and set index=5 for the second page).
count	int	The maximum number of rooms returned per call; 0 indicates all rooms that meet the conditions
callback	IGetLiveRoomListCallback*	Proxy pointer of IGetLiveRoomListCallback type, used to call back the query result

- Example:

```
// Start from index=0. A maximum of 20 rooms can be queried. This class implements the IGetRoomListCallback API to obtain query results
m_liveRoom->getRoomList(0, 20, this);
```

6. getAudienceList

- Definition: void getAudienceList(const std::string& roomID)
- Description: Gets the list of viewers and returns only the last 30 viewers who entered the room
- Parameter:

Parameter	Type	Description
roomID	const std::string&	roomID, which is obtained in the room list returned by the getRoomList API

- Example:

©2013-2018 Tencent Cloud. All rights reserved.

```
m_liveRoom->getRoomList(roomID);
```

7. createRoom

- Definition: void createRoom(const std::string& roomID, const std::string& roomInfo)
- Description: Creates a room, and the new room will be added to the room list in the backend, and then VJs can start pushing.
- Parameters:

Parameter	Type	Description
roomID	const std::string&	Room ID. If an empty string is specified, a roomID is assigned to you. Otherwise, the specified roomID is used as the ID of this room.
roomInfo	const std::string&	Custom data. This field is included in the room information. It is recommended that you define roomInfo in json format, which is highly extensible.

- Example:

```
// A roomID is assigned to you
m_liveRoom->createRoom("", "{\"roomName\": \"My first room\"}");
```

8. enterRoom

- Definition: void enterRoom(const std::string& roomID, HWND rendHwnd, const RECT & rect)
- Description: Enters a room.
- Parameters:

Parameter	Type	Description
roomID	const std::string&	roomID - the ID of the room to be entered, which is obtained in the room list returned by the getRoomList API
rendHwnd	HWND	The HWND that identifies the preview window.

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

- Example:

```
RECT rect = { 0 };
::GetClientRect(hwnd, &rect); // Render in the entire HWND window
m_liveRoom->enterRoom(roomID, hwnd, rect);
```

9. leaveRoom

- Definition: void leaveRoom()
- Description: Leaves the room. If the primary VJ leaves, the room will be terminated by the backend; if secondary VJs or viewers leave, the remaining participants can continue watching.
- Example:

```
m_liveRoom->leaveRoom();
```

10. sendRoomTextMsg

- Definition: void sendRoomTextMsg(const char * msg)
- Description: Sends plain text messages, such as sending on-screen comments under LVB scenarios.
- Parameter:

Parameter	Type	Description
msg	const char *	Text message

- Example:

```
m_liveRoom->sendRoomTextMsg("Hello LiveRoom");
```

©2013-2018 Tencent Cloud. All rights reserved.

11. sendRoomCustomMsg

- Definition: void sendRoomCustomMsg(const char *cmd, const char* msg)
- Description: Sends custom messages, such as giving a "Like" or flower under LVB scenarios
- Parameter:

Parameter	Type	Description
cmd	const char *	Custom cmd, jointly determined by the sender and receiver
msg	const char *	Custom message

- Example:

```
// Assume that the cmd determined by the sender and receiver includes Smile and Anger
m_liveRoom->sendRoomCustomMsg("Smile", "I am hungry");
```

12. startLocalPreview

- Definition: void startLocalPreview(HWND rendHwnd, const RECT & rect)
- Description: Enables the default camera preview
- Parameters:

Parameter	Type	Description
rendHwnd	HWND	HWND that identifies the preview screen. OpenGL is used to draw images on HWND. If rendHwnd = null, there is no need to preview the video.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

- Example:

```
RECT rect = { 0 };
::GetClientRect(hwnd, &rect); // Render in the entire HWND window
```

©2013-2018 Tencent Cloud. All rights reserved.

```
m_liveRoom->startLocalPreview(hwnd, rect);
```

13. updateLocalPreview

- Definition: void updateLocalPreview(HWND rendHwnd, const RECT &rect)
- Description: Resets the preview area for camera. When the size of the window identified by the specified HWND changes, you can resize the video rendering area using this function.
- Parameters:

Parameter	Type	Description
rendHwnd	HWND	HWND that identifies the preview screen. OpenGL is used to draw images on HWND. If rendHwnd = NULL, there is no need to preview the video.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

- Example:

```
RECT rect = { 0 };
::GetClientRect(hwnd, &rect); // Render in the entire HWND window
m_liveRoom->updateLocalPreview(hwnd, rect);
```

14. stopLocalPreview

- Definition: void stopLocalPreview()
- Description: Disables camera preview
- Example:

```
m_liveRoom->stopLocalPreview();
```

15. startScreenPreview

©2013-2018 Tencent Cloud. All rights reserved.

- Definition: bool startScreenPreview(HWND rendHwnd, HWND captureHwnd, const RECT & renderRect, const RECT & captureRect)
- Description: Enables screen sharing
- Parameters:

Parameter	Type	Description
rendHwnd	HWND	HWND that identifies the preview screen. OpenGL is used to draw images on HWND. If rendHwnd = NULL, there is no need to preview the video.
captureHwnd	HWND	Specifies the capture window. If it is NULL, captureRect does not work and the entire screen is captured; if it is not NULL, the current window is captured.
renderRect	const RECT &	Specifies the rendering area of the video screen on rendHwnd
captureRect	const RECT &	Specifies the customer area of the capture window

- Returned value: Success or Failed
- Example:

```
RECT renderRect = { 0 };
::GetClientRect(rendHwnd, &renderRect); // Render in the entire HWND window

RECT captureRect = { 0 };
::GetClientRect(captureHwnd, &captureRect); // Capture the entire window

m_liveRoom->startScreenPreview(rendHwnd, captureHwnd, renderRect, captureRect);
```

16. stopScreenPreview

- Definition: void stopScreenPreview()
- Description: Disables screen sharing

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
m_liveRoom->stopScreenPreview();
```

17. addRemoteView

- Definition: void addRemoteView(HWND rendHwnd, const RECT & rect, const char * userID)
- Description: Plays videos of other VJs in the room
- Parameters:

Parameter	Type	Description
rendHwnd	HWND	The HWND that identifies the preview window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.
userID	const char *	User ID

- Example:

```
RECT rect = { 0 };
::GetClientRect(hwnd, &rect); // Render in the entire HWND window

// Enable playback of users' audios/videos
m_liveRoom->addRemoteView(hwnd, rect, userID);
```

18. removeRemoteView

- Definition: void removeRemoteView(const char * userID)
- Description: Stops playing videos of other VJs
- Parameter:

Parameter	Type	Description

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
userID	const char *	User ID

- Example:

```
m_liveRoom->removeRemoteView(userID);
```

19. setMute

- Definition: void setMute(bool mute)
- Description: Enables Mute
- Parameter:

Parameter	Type	Description
mute	bool	Indicates whether to enable Mute

- Example:

```
m_liveRoom->setMute(true); // Set to mute
```

20. setVideoQuality

- Definition: void setVideoQuality(LRVideoQuality quality, LRAspectRatio ratio)
- Description: Sets the pushed video quality
- Parameter:

Parameter	Type	Description
quality	LRVideoQuality	Image quality, see LRVideoQuality's enumerated values defined in LiveRoomUtil.h

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
ratio	LRAspectRatio	Aspect ratio, see LRAspectRatio's enumerated values defined in LiveRoomUtil.h

- Example:

```
// Set the image quality to HD and the aspect ratio to 4:3
m_liveRoom->setVideoQuality(LIVEROOM_VIDEO_QUALITY_HIGH_DEFINITION, LIVEROOM_ASPECT_RATIO_4_3);
```

21. setBeautyStyle

- Definition: void setBeautyStyle(TXEBeautyStyle beautyStyle, int beautyLevel, int whitenessLevel)
- Description: Sets beauty filter and whitening effects
- Parameter:

Parameter	Type	Description
beautyStyle	TXEBeautyStyle	See LRBeautyStyle's enumerated values defined in LiveRoomUtil.h
beautyLevel	int	Beauty filter level: 1-9. 0 indicates disabling beauty filter. A greater value means a bigger effect.
whitenessLevel	int	Whiteness level: 1-9. 0 indicates disabling whitening. A greater value means a bigger effect.

- Example:

```
// Natural; beauty filter level: 5; whitening level: 5
m_liveRoom->setBeautyStyle(TXE_BEAUTY_STYLE_NATURE, 5, 5);
```

22. requestJoinPusher

- Definition: void requestJoinPusher()

©2013-2018 Tencent Cloud. All rights reserved.

- Description: Joint broadcasting request from viewer
- Example:

```
m_liveRoom->requestJoinPusher();
```

23. acceptJoinPusher

- Definition: void acceptJoinPusher(const std::string& userID)
- Description: The primary VJ accepts the joint broadcasting request and informs the request initiator.
- Parameter:

Parameter	Type	Description
userID	const std::string&	User ID

- Example:

```
m_liveRoom->acceptJoinPusher(userID);
```

24. rejectJoinPusher

- Definition: void rejectJoinPusher(const std::string& userID, const std::string& reason)
- Description: The primary VJ rejects the joint broadcasting request and informs the request initiator.
- Parameters:

Parameter	Type	Description
userID	const std::string&	User ID
reason	const std::string&	Reason for rejection

- Example:

©2013-2018 Tencent Cloud. All rights reserved.

```
m_liveRoom->acceptJoinPusher(userID, reason);
```

25. kickoutSubPusher

- Definition: void kickoutSubPusher(const std::string& userID)
- Description: The primary VJ kicks out a secondary VJ.
- Parameter:

Parameter	Type	Description
userID	const std::string&	User ID

- Example:

```
m_liveRoom->kickoutSubPusher(userID);
```

Details of ILoginLiveCallback APIs

1. onLogin

- Definition: virtual void onLogin(const LRResult& res, const LRAuthData& authData)
- Description: Callback of login result
- Parameters:

Parameter	Type	Description
res	const LRResult&	Execution result. See LRResult structure defined in LiveRoomUtil.h
authData	const LRAuthData&	Login information provided by RoomService, including IM related configuration fields. The token field can be obtained after a successful login.

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
class LoginDialog : public ILoginLiveCallback
{
public:
virtual void onLogin(const LRResult& res, const LRAuthData& authData)
{
if (LIVEROOM_SUCCESS != res.ec)
{
// Login failed
}
else
{
// Login succeeded, and authData information is saved
}
}
};

// For more information on how to use callback, please see LiveRoom::login API
...
```

Details of IGetLiveRoomListCallback APIs

1. onGetRoomList

- Definition: virtual void onGetRoomList(const LRResult& res, const std::vector& rooms)
- Description: Callback of obtaining a room list
- Parameters:

Parameter	Type	Description
res	const LRResult&	Execution result. See LRResult structure defined in LiveRoomUtil.h
rooms	const std::vector&	List of room information

- Example:

©2013-2018 Tencent Cloud. All rights reserved.

```
class RoomListDialog : public IGetRoomListCallback
{
public:
virtual void onGetRoomList(const LRResult& res, const std::vector<LRRoomData>& rooms)
{
if (LIVEROOM_SUCCESS != res.ec)
{
// Query Failed
}
else
{
// Query succeeded, process rooms data
}
}
};

// For more information on how to use callback, please see LiveRoom::getRoomList API
...
```

Details of ILiveRoomCallback APIs

For more information on how to set ILiveRoomCallback, please see the API LiveRoom::setCallback.

1. onCreateRoom

- Definition: virtual void onCreateRoom(const LRResult& res, const std::string& roomID)
- Description: Callback of creating a room
- Parameters:

Parameter	Type	Description
res	const LRResult&	Execution result. See LRResult structure defined in LiveRoomUtil.h
roomID	const std::string&	Room ID

- Example:

©2013-2018 Tencent Cloud. All rights reserved.

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onCreateRoom(const LRResult& res, const std::string& roomID)
{
if (LIVEROOM_SUCCESS != res.ec)
{
// If the room fails to be created, a prompt pops up.
}
else
{
// If the room is created successfully, the viewers can watch the LVB.
// Process onUpdateRoomData callback immediately
}
}

...
};
```

2. onEnterRoom

- Definition: virtual void onEnterRoom(const LRResult& res)
- Description: Callback of entering a room
- Parameter:

Parameter	Type	Description
res	const LRResult&	Execution result. See LRResult structure defined in LiveRoomUtil.h

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onEnterRoom(const LRResult& res)
{
if (LIVEROOM_SUCCESS != res.ec)
{
// If the room fails to be entered, a prompt pops up.
```

©2013-2018 Tencent Cloud. All rights reserved.

```
}
else
{
// If viewers enter the room successfully, process onUpdateRoomData callback immediately
}
}

...
};
```

3. onUpdateRoomData

- Definition: virtual void onUpdateRoomData(const LRResult& res, const LRRoomData& roomData)
- Description: Callback of room information change
- Parameters:

Parameter	Type	Description
res	const LRResult&	Execution result. See LRResult structure defined in LiveRoomUtil.h
roomData	const LRRoomData&	Room information. See LRRoomData structure defined in LiveRoomUtil.h.

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onUpdateRoomData(const LRResult& res, const LRRoomData& roomData)
{
if (LIVEROOM_SUCCESS != res.ec)
{
// If the update fails, a prompt pops up.
}
else
{
// If the update succeeds, the appropriate processing is performed based on whether the API is trig
gered by createRoom or enterRoom.
}
```

©2013-2018 Tencent Cloud. All rights reserved.

```
}

...
};
```

4. onGetAudienceList

- Definition: virtual void onGetAudienceList(const LRResult& res, const std::vector& audiences)
- Description: Callback of viewer list query
- Parameters:

Parameter	Type	Description
res	const LRResult&	Execution result. See LRResult structure defined in LiveRoomUtil.h
audiences	const std::vector&	Viewer information list

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onGetAudienceList(const LRResult& res, const std::vector<LRAudienceData>& audience
s)
{
if (LIVEROOM_SUCCESS != res.ec)
{
// If the query fails, a prompt pops up.
}
else
{
// If the query succeeds, the list is updated to UI.
}
}

...
};
```

5. onPusherJoin

- Definition: virtual void onPusherJoin(const LRMemberData& member)

©2013-2018 Tencent Cloud. All rights reserved.

- Description: Callback of joint broadcasting viewer entering the room
- Parameter:

Parameter	Type	Description
member	const LRMemberData&	Member information. See LRMemberData structure defined in LiveRoomUtil.h.

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onPusherJoin(const LRMemberData& member)
{
// Usually, the video and audio of the joint broadcasting viewer are turned on after he or she requ
ests joint broadcasting.
m_liveRoom->addRemoteView(rendHwnd, rect, member.userID);
...
}

...
};
```

6. onPusherQuit

- Definition: virtual void onPusherQuit(const LRMemberData& member)
- Description: Callback of the joint broadcasting viewer exiting the room
- Parameter:

Parameter	Type	Description
member	const LRMemberData&	Member information. See LRMemberData structure defined in LiveRoomUtil.h.

- Example:

©2013-2018 Tencent Cloud. All rights reserved.

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onPusherQuit(const LRMemberData& member)
{
// Usually, the video and audio of the joint broadcasting viewer are turned off after he or she stops joint broadcasting.
m_liveRoom->removeRemoteView(member.userID);
...
}

...
};
```

7. onRoomClosed

- Definition: virtual void onRoomClosed(const std::string& roomID)
- Description: Callback of room dissolution
- Parameter:

Parameter	Type	Description
roomID	const std::string&	Room ID

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onRoomClosed(const std::string& roomID)
{
// It prompts that the room is dismissed. The opened audio/video is closed.
...
}

...
};
```

©2013-2018 Tencent Cloud. All rights reserved.

8. onRecvRoomTextMsg

- Definition: virtual void onRecvRoomTextMsg(const char *roomID, const char* userID, const char *userName, const char* userAvatar, const char * msg)
- Description: Receives plain text messages
- Parameters:

Parameter	Type	Description
roomID	const char *	Room ID
userID	const char *	Sender ID
userName	const char *	Sender nickname
userAvatar	const char *	Sender profile photo
message	const char *	Text message content

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onRecvRoomTextMsg(const char * roomID, const char * userID, const char * userNa
me, const char * userAvatar, const char * msg)
{
// Text messages are displayed on IM panel
}

...
};
```

9. onRecvRoomCustomMsg

- Definition: virtual void onRecvRoomCustomMsg(const char *roomID, const char* userID, const char *userName, const char* userAvatar, const char *cmd, const char* message)
- Description: Receives custom messages

©2013-2018 Tencent Cloud. All rights reserved.

- Parameters:

Parameter	Type	Description
roomID	const char *	Room ID
userID	const char *	Sender ID
userName	const char *	Sender nickname
userAvatar	const char *	Sender profile photo
cmd	const char *	Custom CMD
message	const char *	Custom message content

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onRecvRoomCustomMsg(const char * roomID, const char * userID, const char * user
Name, const char * userAvatar, const char * cmd, const char * message)
{
// Parse and process custom messages such as Likes and gifts
}

...
};
```

10. onError

- Definition: virtual void onError(const LRResult& res, const std::string& userID)
- Description: Notification of internal LiveRoom error
- Parameters:

Parameter	Type	Description
res	const LRResult&	Execution result. See LRResult structure defined in LiveRoomUtil.h

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
userID	const std::string&	User ID

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onError(const LRResult& res, const std::string& userID)
{
// An error message pops up. Decide whether to close LVB based on severity of errors
}

...
};
```

11. onRecvJoinPusherRequest

- Definition: virtual void onRecvJoinPusherRequest(const std::string& roomID, const std::string& userID, const std::string& userName, const std::string& userAvatar)

- Description: Receives a joint broadcasting request

- Parameters:

Parameter	Type	Description
roomID	const std::string&	Room ID
userID	const std::string&	Sender ID
userName	const std::string&	Sender nickname
userAvatar	const std::string&	Sender profile photo

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
```

©2013-2018 Tencent Cloud. All rights reserved.

```
virtual void onRecvJoinPusherRequest(const std::string& roomID, const std::string& userID, const std::string& userName, const std::string& userAvatar)
{
// The primary VJ receives a joint broadcasting request from a viewer.
// Verify the validity of roomID
// UI shows who initiates the request.
// The primary VJ selects "acceptJoinPusher" to accept or "rejectJoinPusher" to reject the request.
}

...
};
```

12. onRecvAcceptJoinPusher

- Definition: virtual void onRecvAcceptJoinPusher(const std::string& roomID, const std::string& msg)
- Description: Receives the notification of accepting a joint broadcasting request.
- Parameters:

Parameter	Type	Description
roomID	const std::string&	Room ID
msg	const std::string&	Message

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onRecvAcceptJoinPusher(const std::string& roomID, const std::string& msg)
{
// The viewer receives the notification of the primary VJ accepting the joint broadcasting request.
// Verify the validity of roomID
// Process msg
}

...
};
```

©2013-2018 Tencent Cloud. All rights reserved.

13. onRecvRejectJoinPusher

- Definition: virtual void onRecvRejectJoinPusher(const std::string& roomID, const std::string& msg)
- Description: Receives the notification of rejecting a joint broadcasting request
- Parameter:

Parameter	Type	Description
roomID	const std::string&	Room ID
msg	const std::string&	Message

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onRecvRejectJoinPusher(const std::string& roomID, const std::string& msg)
{
// The viewer receives the notification of the primary VJ rejecting the joint broadcasting request.
// Verify the validity of roomID
// Process msg
}

...
};
```

14. onRecvKickoutSubPusher

- Definition: virtual void onRecvKickoutSubPusher(const std::string& roomID)
- Description: Receives the notification of the primary VJ kicking out a secondary VJ.
- Parameter:

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
roomID	const std::string&	Room ID

- Example:

```
class MainDialog : public ILiveRoomCallback
{
public:
virtual void onRecvRejectJoinPusher(const std::string& roomID, const std::string& msg)
{
// The primary VJ kicks out a viewer invited for joint broadcasting
// Verify the validity of roomID
// Process msg
}

...
};
```

©2013-2018 Tencent Cloud. All rights reserved.

RTCRoom

Last updated：2018-07-11 11:19:35

Real-time Audio/Video (RTCRoom) APIs (C++)

RTCRoom

Name	Description
instance()	Gets a single instance for RTCRoom and calls RTCRoom APIs via the single instance
setCallback(IRTCRoomCallback * callback)	Sets the callback proxy for RTCRoom callback, and listens to the internal status of RTCRoom and the execution results of the API
login(const std::string & serverDomain, const RTCAuthData & authData, ILoginRTCCallback* callback)	Logs in to the business server RoomService before you can normally use other APIs and the IM feature
logout()	Logs out of the business server RoomService. Make sure to call "logout" after "leaveRoom" is called, otherwise the call of leaveRoom will fail.
getRoomList(int index, int count, IGetRTCRoomListCallback* callback)	Gets the room list. If there are many rooms, you can get the list in pages.
createRoom(const std::string& roomID, const std::string& roomInfo)	Creates a room, and the new room will be added to the room list in the backend, and then the meeting initiator can start pushing.
enterRoom(const std::string& roomID, HWND rendHwnd, const RECT & rect)	Enters a room
leaveRoom()	Leaves the room. If the meeting initiator leaves, the room will be terminated by the backend; if other meeting participants leave, the remaining participants can continue chatting.

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
sendRoomTextMsg(const char * msg)	Sends plain text messages
sendRoomCustomMsg(const char *cmd, const char* msg)	Sends custom messages
startLocalPreview(HWND rendHwnd, const RECT & rect)	Enables the default camera preview
updateLocalPreview(HWND rendHwnd, const RECT &rect)	Resets the preview area for camera. When the size of the window identified by the specified local HWND changes, you can resize the video rendering area using this function.
stopLocalPreview()	Disables camera preview
startScreenPreview(HWND rendHwnd, HWND captureHwnd, const RECT & renderRect, const RECT & captureRect)	Enables screen sharing
stopScreenPreview()	Disables screen sharing
addRemoteView(HWND rendHwnd, const RECT & rect, const char * userID)	Plays videos of other meeting participants in the room
updateRemotePreview(HWND rendHwnd, const RECT &rect, const char * userID)	Resets the video preview area of the specified userID. When the size of the window identified by the specified local HWND changes, you can resize the video rendering area using this function.
removeRemoteView(const char * userID)	Stops playing videos of other meeting participants
setMute(bool mute)	Enables Mute
setBeautyStyle(TXEBeautyStyle beautyStyle, int beautyLevel, int whitenessLevel)	Sets beauty filter and whitening effects

ILoginRTCCallback

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
onLogin(const RTCResult& res, const RTCAuthData& authData)	Callback of login result

IGetRTCRoomListCallback

Name	Description
onGetRoomList(const RTCResult& res, const std::vector& rooms)	Callback of obtaining a room list

IRTCRoomCallback

Name	Description
onCreateRoom(const RTCResult& res, const std::string& roomID)	Callback of creating a room
onEnterRoom(const RTCResult& res)	Callback of entering a room
onUpdateRoomData(const RTCResult& res, const RTCRoomData& roomData)	Callback of room information change
onPusherJoin(const RTCMemberData& member)	Callback of members entering a room
void onPusherQuit(const RTCMemberData& member)	Callback of members exiting a room
onRoomClosed(const std::string& roomID)	Callback of room dissolution
onRecvRoomTextMsg(const char *roomID, const char* userID, const char *userName, const char* userAvatar, const char * msg)	Receives plain text messages
onRecvRoomCustomMsg(const char *roomID, const char* userID, const char *userName, const char* userAvatar, const char *cmd, const char* message)	Receives custom messages

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
onError(const RTCResult& res, const std::string& userID)	Notification of internal RTCRoom error

Details of RTCRoom object APIs

1. instance

- Definition: static RTCRoom* instance()
- Description: Gets a single instance for RTCRoom and calls RTCRoom APIs via the single instance
- Example:

```
RTCRoom* m_RTCRoom = NULL;
m_RTCRoom = RTCRoom::instance();
```

2. setCallback

- Definition: void setCallback(IRTCRoomCallback * callback)
- Description: Sets the callback proxy for RTCRoom callback, and listens to the internal status of RTCRoom and the execution results of the API
- Parameter:

Parameter	Type	Description
callback	IRTCRoomCallback *	Proxy pointer of IRTCRoomCallback type

- Example:

```
class MainDialog : public IRTCRoomCallback
{
public:
MainDialog();
```

©2013-2018 Tencent Cloud. All rights reserved.

```
virtual ~MainDialog();

virtual void onCreateRoom(const RTCResult& res, const std::string& roomID);
virtual void onEnterRoom(const RTCResult& res);
virtual void onUpdateRoomData(const RTCResult& res, const LRRoomData& roomData);
...
};

MainDialog::MainDialog()
{
m_RTCRoom->setCallback(this); //Set callback
}

...
```

3. login

- Definition: void login(const std::string & serverDomain, const RTCAuthData & authData, ILoginRTCCallback* callback)
- Description: Logs in to the business server RoomService before you can normally use other APIs and the IM feature
- Parameters:

Parameter	Type	Description
serverDomain	const std::string &	URL of RoomService. In consideration of security, it is recommended to access the https encrypted link. For more information, please see DOC.
authData	const RTCAuthData &	Login information provided by RoomService, including IM related configuration fields. The token field can be obtained after a success login. For more information, please see DOC.
callback	ILoginRTCCallback*	Proxy pointer of ILoginRTCCallback type, used to call back the login result

- Example:

©2013-2018 Tencent Cloud. All rights reserved.

```
std::string serverDomain = "https://roomtest.qcloud.com/weapp/double_room";

m_authData.sdkAppID = sdkAppID;
m_authData.accountType = accountType;
m_authData.userID = userID;
m_authData.userSig = userSig;
m_authData.userName = userName;
m_authData.userAvatar = userAvatar;
m_authData.token = token;

// This class implements the ILoginCallback API to obtain login results
m_RTCRoom->login(serverDomain, m_authData, this);
```

4. logout

- Definition: void logout();
- Description: Logs out of the business server RoomService. Make sure to call "logout" after "leaveRoom" is called, otherwise the call of leaveRoom will fail.
- Example:

```
m_RTCRoom->logout();
```

5. getRoomList

- Definition: void getRoomList(int index, int count, IGetRTCRoomListCallback* callback)
- Description: Gets the room list. If there are many rooms, you can get the list in pages.
- Parameters:

Parameter	Type	Description
index	int	Gets the list in pages. The default value can be 0, and for the subsequent retrieval, the value can be set as the starting room index (for example, set index=0 and cnt=5 for the first retrieval, and set index=5 for the second page).

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
count	int	The maximum number of rooms returned per call; 0 indicates all rooms that meet the conditions
callback	IGetRTCRoomListCallback*	Proxy pointer of IGetRTCRoomListCallback type, used to call back the query result

- Example:

```
// Start from index=0. A maximum of 20 rooms can be queried. This class implements the IGetRoomListCallback API to obtain query results
m_RTCRoom->getRoomList(0, 20, this);
```

6. createRoom

- Definition: void createRoom(const std::string& roomID, const std::string& roomInfo)
- Description: Creates a room, and the new room will be added to the room list in the backend, and then the meeting initiator can start pushing.
- Parameters:

Parameter	Type	Description
roomID	const std::string&	Room ID. If an empty string is specified, a roomID is assigned to you. Otherwise, the specified roomID is used as the ID of this room.
roomInfo	const std::string&	Custom data. This field is included in the room information. It is recommended that you define roomInfo in json format, which is highly extensible.

- Example:

```
// A roomID is assigned to you
m_RTCRoom->createRoom("", "{\"roomName\": \"My first room\"}");
```

7. enterRoom

©2013-2018 Tencent Cloud. All rights reserved.

- Definition: void enterRoom(const std::string& roomID)
- Description: Enters a room.
- Parameter:

Parameter	Type	Description
roomID	const std::string&	roomID - the ID of the room to be entered, which is obtained in the room list returned by the getRoomList API

- Example:

```
m_RTCRoom->enterRoom(roomID);
```

8. leaveRoom

- Definition: void leaveRoom()
- Description: Leaves the room. If the meeting initiator leaves, the room will be terminated by the backend; if other meeting participants leave, the remaining participants can continue chatting.
- Example:

```
m_RTCRoom->leaveRoom();
```

9. sendRoomTextMsg

- Definition: void sendRoomTextMsg(const char * msg)
- Description: Sends plain text messages
- Parameter:

Parameter	Type	Description
msg	const char *	Text message

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
m_RTCRoom->sendRoomTextMsg("Hello RTCRoom");
```

10. sendRoomCustomMsg

- Definition: void sendRoomCustomMsg(const char *cmd, const char* msg)
- Description: Sends custom messages
- Parameters:

Parameter	Type	Description
cmd	const char *	Custom cmd, jointly determined by the sender and receiver
msg	const char *	Custom message

- Example:

```
// Assume that the cmd determined by the sender and receiver includes Smile and Anger
m_RTCRoom->sendRoomCustomMsg("Smile", "I am hungry");
```

11. startLocalPreview

- Definition: void startLocalPreview(HWND rendHwnd, const RECT & rect)
- Description: Enables the default camera preview
- Parameters:

Parameter	Type	Description
rendHwnd	HWND	HWND that identifies the preview screen. OpenGL is used to draw images on HWND. If rendHwnd = null, there is no need to preview the video.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
RECT rect = { 0 };
::GetClientRect(hwnd, &rect); // Render in the entire HWND window
m_RTCRoom->startLocalPreview(hwnd, rect);
```

12. updateLocalPreview

- Definition: void updateLocalPreview(HWND rendHwnd, const RECT &rect)
- Description: Resets the preview area for camera. When the size of the window identified by the specified HWND changes, you can resize the video rendering area using this function.
- Parameters:

Parameter	Type	Description
rendHwnd	HWND	HWND that identifies the preview screen. OpenGL is used to draw images on HWND. If rendHwnd = NULL, there is no need to preview the video.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

- Example:

```
RECT rect = { 0 };
::GetClientRect(hwnd, &rect); // Render in the entire HWND window
m_RTCRoom->updateLocalPreview(hwnd, rect);
```

13. stopLocalPreview

- Definition: void stopLocalPreview()
- Description: Disables camera preview
- Example:

```
m_RTCRoom->stopLocalPreview();
```

©2013-2018 Tencent Cloud. All rights reserved.

14. startScreenPreview

- Definition: bool startScreenPreview(HWND rendHwnd, HWND captureHwnd, const RECT & renderRect, const RECT & captureRect)
- Description: Enables screen sharing
- Parameters:

Parameter	Type	Description
rendHwnd	HWND	HWND that identifies the preview screen. OpenGL is used to draw images on HWND. If rendHwnd = NULL, there is no need to preview the video.
captureHwnd	HWND	Specifies the capture window. If it is NULL, captureRect does not work and the entire screen is captured; if it is not NULL, the current window is captured.
renderRect	const RECT &	Specifies the rendering area of the video screen on rendHwnd
captureRect	const RECT &	Specifies the customer area of the capture window

- Returned value: Success or Failed
- Example:

```
RECT renderRect = { 0 };
::GetClientRect(rendHwnd, &renderRect); // Render in the entire HWND window

RECT captureRect = { 0 };
::GetClientRect(captureHwnd, &captureRect); // Capture the entire window

m_RTCRoom->startScreenPreview(rendHwnd, captureHwnd, renderRect, captureRect);
```

15. stopScreenPreview

©2013-2018 Tencent Cloud. All rights reserved.

- Definition: void stopScreenPreview()
- Description: Disables screen sharing
- Example:

```
m_RTCRoom->stopScreenPreview();
```

16. addRemoteView

- Definition: void addRemoteView(HWND rendHwnd, const RECT & rect, const char * userID)
- Description: Plays videos of other meeting participants in the room
- Parameters:

Parameter	Type	Description
rendHwnd	HWND	The HWND that identifies the preview window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.
userID	const char *	User ID

- Example:

```
RECT rect = { 0 };
::GetClientRect(hwnd, &rect); // Render in the entire HWND window

// Enable playback of users' audios/videos
m_RTCRoom->addRemoteView(hwnd, rect, userID);
```

17. updateRemotePreview

- Definition: void updateRemotePreview(HWND rendHwnd, const RECT &rect, const char * userID)
- Description: Resets the video preview area of the specified userID. When the size of the window identified by the specified local HWND changes, you can resize the video rendering area using this function.

©2013-2018 Tencent Cloud. All rights reserved.

- Parameters:

Parameter	Type	Description
rendHwnd	HWND	The HWND that identifies the preview window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.
userID	const char *	User ID

- Example:

```
RECT rect = { 0 };
::GetClientRect(hwnd, &rect); // Render in the entire HWND window

// Update playback of users' audios/videos
m_RTCRoom->updateRemotePreview(hwnd, rect, userID);
```

18. removeRemoteView

- Definition: void removeRemoteView(const char * userID)
- Description: Stops playing videos of other meeting participants
- Parameter:

Parameter	Type	Description
userID	const char *	User ID

- Example:

```
m_RTCRoom->removeRemoteView(userID);
```

19. setMute

- Definition: void setMute(bool mute)

©2013-2018 Tencent Cloud. All rights reserved.

- Description: Enables Mute
- Parameter:

Parameter	Type	Description
mute	bool	Indicates whether to enable Mute

- Example:

```
m_RTCRoom->setMute(true); // Set to mute
```

20. setBeautyStyle

- Definition: void setBeautyStyle(TXEBeautyStyle beautyStyle, int beautyLevel, int whitenessLevel)
- Description: Sets beauty filter and whitening effects
- Parameters:

Parameter	Type	Description
beautyStyle	TXEBeautyStyle	See RTCBeautyStyle's enumerated values defined in RTCRoomUtil.h
beautyLevel	int	Beauty filter level: 1-9. 0 indicates disabling beauty filter. A greater value means a bigger effect.
whitenessLevel	int	Whiteness level: 1-9. 0 indicates disabling whitening. A greater value means a bigger effect.

- Example:

```
// Natural; beauty filter level: 5; whitening level: 5
m_RTCRoom->setBeautyStyle(TXE_BEAUTY_STYLE_NATURE, 5, 5);
```

Details of ILoginRTCCallback APIs

©2013-2018 Tencent Cloud. All rights reserved.

1. onLogin

- Definition: virtual void onLogin(const RTCResult& res, const RTCAuthData& authData)
- Description: Callback of login result
- Parameters:

Parameter	Type	Description
res	const RTCResult&	Execution result. See RTCResult structure defined in RTCRoomUtil.h
authData	const RTCAuthData&	Login information provided by RoomService, including IM related configuration fields. The token field can be obtained after a success login.

- Example:

```
class LoginDialog : public ILoginRTCCallback
{
public:
virtual void onLogin(const RTCResult& res, const RTCAuthData& authData)
{
if (RTCROOM_SUCCESS != res.ec)
{
// Login failed
}
else
{
// Login succeeded, and authData information is saved
}
}
};

// For more information on how to use callback, please see the API RTCRoom::login
...
```

Details of IGetRTCRoomListCallback APIs

1. onGetRoomList

©2013-2018 Tencent Cloud. All rights reserved.

- Definition: virtual void onGetRoomList(const RTCResult& res, const std::vector& rooms)
- Description: Callback of obtaining a room list
- Parameters:

Parameter	Type	Description
res	const RTCResult&	Execution result. See RTCResult structure defined in RTCRoomUtil.h
rooms	const std::vector&	List of room information

- Example:

```
class RoomListDialog : public IGetRTCRoomListCallback
{
public:
virtual void onGetRoomList(const RTCResult& res, const std::vector<RTCRoomData>& rooms)
{
if (RTCROOM_SUCCESS != res.ec)
{
// Query failed
}
else
{
// Query succeeded, process rooms data
}
}
};

// For more information on how to use callback, please see the API RTCRoom::getRoomList
...
```

Details of IRTCRoomCallback APIs

For more information on how to set IRTCRoomCallback, please see the API RTCRoom::setCallback.

1. onCreateRoom

©2013-2018 Tencent Cloud. All rights reserved.

- Definition: virtual void onCreateRoom(const RTCResult& res, const std::string& roomID)
- Description: Callback of creating a room
- Parameters:

Parameter	Type	Description
res	const RTCResult&	Execution result. See RTCResult structure defined in RTCRoomUtil.h
roomID	const std::string&	Room ID

- Example:

```
class MainDialog : public IRTCRoomCallback
{
public:
virtual void onCreateRoom(const RTCResult& res, const std::string& roomID)
{
if (RTCROOM_SUCCESS != res.ec)
{
// If the room fails to be created, a prompt pops up.
}
else
{
// If the room is created successfully, other audiences can participate in the meeting.
// Process onUpdateRoomData callback immediately
}
}

...
};
```

2. onEnterRoom

- Definition: virtual void onEnterRoom(const RTCResult& res)
- Description: Callback of entering a room
- Parameter:

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
res	const RTCResult&	Execution result. See RTCResult structure defined in RTCRoomUtil.h

- Example:

```
class MainDialog : public IRTCRoomCallback
{
public:
virtual void onEnterRoom(const RTCResult& res)
{
if (RTCROOM_SUCCESS != res.ec)
{
// If the room fails to be entered, a prompt pops up.
}
else
{
// If viewers enter the room successfully, process onUpdateRoomData callback immediately
}
}

...
};
```

3. onUpdateRoomData

- Definition: virtual void onUpdateRoomData(const RTCResult& res, const RTCRoomData& roomData)
- Description: Callback of room information change
- Parameters:

Parameter	Type	Description
res	const RTCResult&	Execution result. See RTCResult structure defined in RTCRoomUtil.h
roomData	const RTCRoomData&	Room information. See RTCRoomData structure defined in RTCRoomUtil.h

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
class MainDialog : public IRTCRoomCallback
{
public:
virtual void onUpdateRoomData(const RTCResult& res, const RTCRoomData& roomData)
{
if (RTCROOM_SUCCESS != res.ec)
{
// If the update fails, a prompt pops up.
}
else
{
// If the update succeeds, the appropriate processing is performed based on whether the API is trig
gered by createRoom or enterRoom.
}
}

...
};
```

4. onPusherJoin

- Definition: virtual void onPusherJoin(const RTCMemberData& member)
- Description: Callback of members entering a room
- Parameter:

Parameter	Type	Description
member	const RTCMemberData&	Member information. See the RTCMemberData structure defined in RTCRoomUtil.h.

- Example:

```
class MainDialog : public IRTCRoomCallback
{
public:
virtual void onPusherJoin(const RTCMemberData& member)
{
```

©2013-2018 Tencent Cloud. All rights reserved.

```
// A member enters a room and enables his/her camera and microphone.
m_RTCRoom->addRemoteView(rendHwnd, rect, member.userID);
...
}

...
};
```

5. onPusherQuit

- Definition: virtual void onPusherQuit(const RTCMemberData& member)
- Description: Callback of members exiting the room
- Parameter:

Parameter	Type	Description
member	const RTCMemberData&	Member information. See the RTCMemberData structure defined in RTCRoomUtil.h.

- Example:

```
class MainDialog : public IRTCRoomCallback
{
public:
virtual void onPusherQuit(const RTCMemberData& member)
{
// A member exits the room and disables his/her camera and microphone.
m_RTCRoom->removeRemoteView(member.userID);
...
}

...
};
```

6. onRoomClosed

- Definition: virtual void onRoomClosed(const std::string& roomID)

©2013-2018 Tencent Cloud. All rights reserved.

- Description: Callback of room dissolution
- Parameter:

Parameter	Type	Description
roomID	const std::string&	Room ID

- Example:

```
class MainDialog : public IRTCRoomCallback
{
public:
virtual void onRoomClosed(const std::string& roomID)
{
// It prompts that the room is dismissed. The opened audio/video is closed.
...
}

...
};
```

7. onRecvRoomTextMsg

- Definition: virtual void onRecvRoomTextMsg(const char *roomID, const char* userID, const char *userName, const char* userAvatar, const char * msg)
- Description: Receives plain text messages
- Parameters:

Parameter	Type	Description
roomID	const char *	Room ID
userID	const char *	Sender ID
userName	const char *	Sender nickname
userAvatar	const char *	Sender profile photo

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
message	const char *	Text message content

- Example:

```
class MainDialog : public IRTCRoomCallback
{
public:
virtual void onRecvRoomTextMsg(const char * roomID, const char * userID, const char * userName, const char * userAvatar, const char * msg)
{
// Text messages are displayed on IM panel
}

...
};
```

8. onRecvRoomCustomMsg

- Definition: virtual void onRecvRoomCustomMsg(const char *roomID, const char* userID, const char *userName, const char* userAvatar, const char *cmd, const char* message)
- Description: Receives custom messages
- Parameters:

Parameter	Type	Description
roomID	const char *	Room ID
userID	const char *	Sender ID
userName	const char *	Sender nickname
userAvatar	const char *	Sender profile photo
cmd	const char *	Custom CMD
message	const char *	Custom message content

- Example:

©2013-2018 Tencent Cloud. All rights reserved.

```
class MainDialog : public IRTCRoomCallback
{
public:
virtual void onRecvRoomCustomMsg(const char * roomID, const char * userID, const char * user
Name, const char * userAvatar, const char * cmd, const char * message)
{
// Parse and process custom messages such as Likes and gifts
}

...
};
```

9. onError

- Definition: virtual void onError(const RTCResult& res, const std::string& userID)
- Description: Notification of internal RTCRoom error
- Parameters:

Parameter	Type	Description
res	const RTCResult&	Execution result. See RTCResult structure defined in RTCRoomUtil.h
userID	const std::string&	User ID

- Example:

```
class MainDialog : public IRTCRoomCallback
{
public:
virtual void onError(const RTCResult& res, const std::string& userID)
{
// An error message pops up. Decide whether to dismiss or exit the meeting based on severity of er
rors.
}

...
};
```

©2013-2018 Tencent Cloud. All rights reserved.

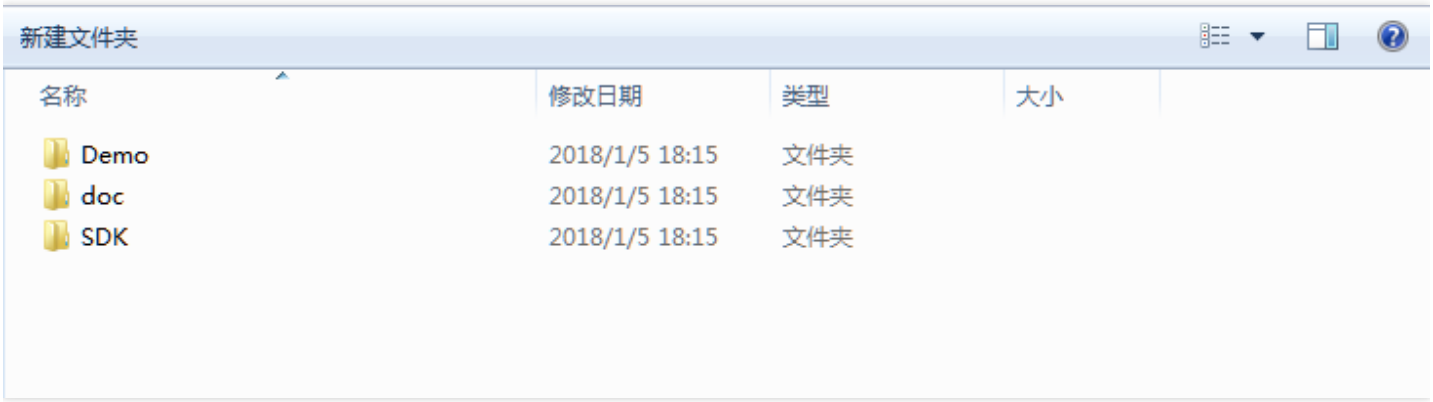
C#
Project Configuration

Last updated：2018-07-11 11:11:42

Configuring Project

SDK

You can obtain Tencent Video Cloud SDK from the Tencent Cloud official website.



File Name	Description
Demo	Contains the Demo installation package and source code. Upon the quick installation of the Demo package, the SDK features are ready to use. You can use Visual Studio to open the project of the source code.
doc	Contains project configuration and SDK connection documents.
SDK	".dll" file. The C# assembly "ManageLiteAV.dll" can be imported to the C# project.

Visual Studio Project Configuration

Development environment

©2013-2018 Tencent Cloud. All rights reserved.

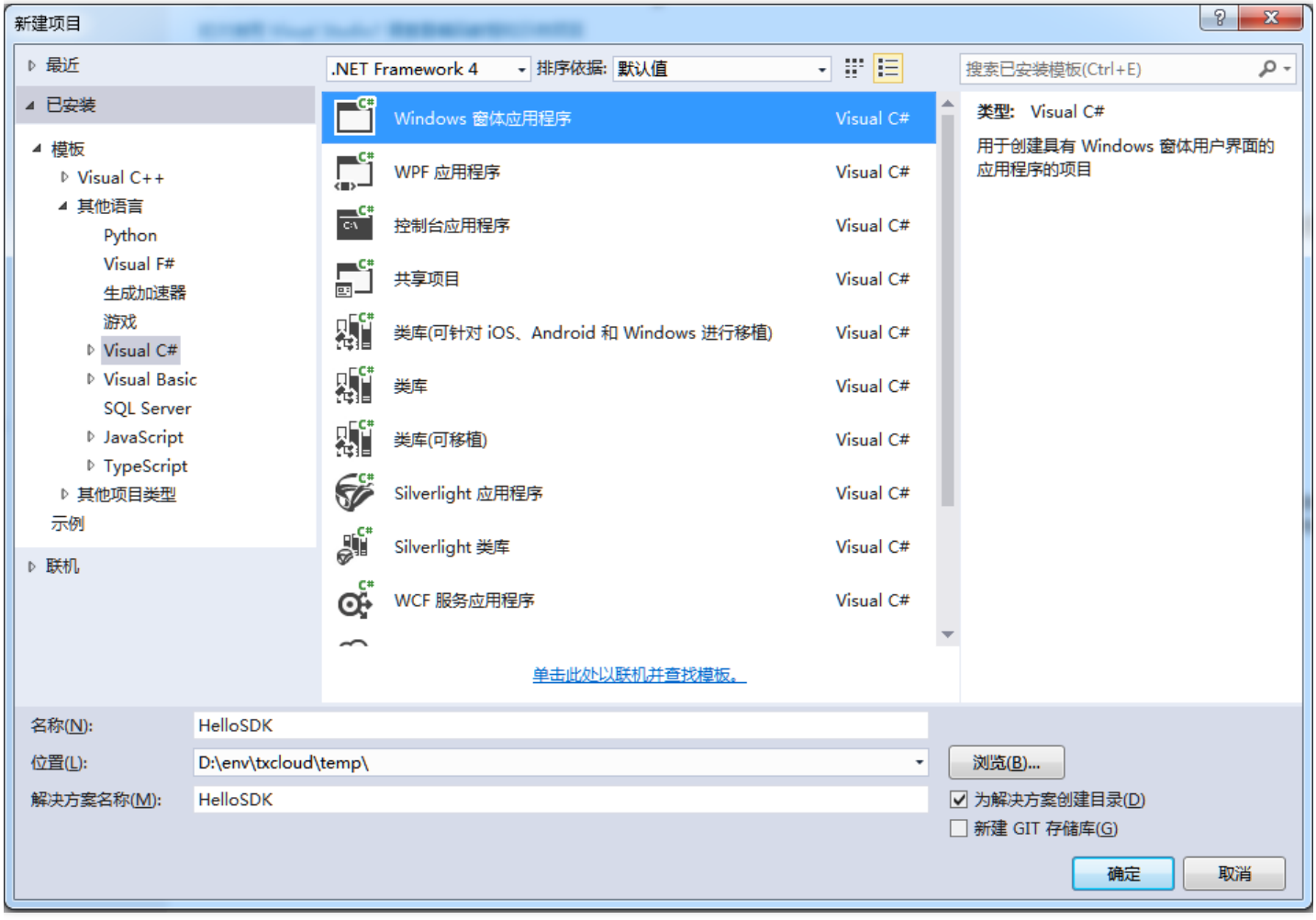
- Visual Studio: 2010 and later.
- .Net framework: 4.0 and later.
- Windows: Windows 7 and later.

C# Project Reference SDK

The following describes how to configure the SDK in a Visual Studio project using a simple C# WinForm project.

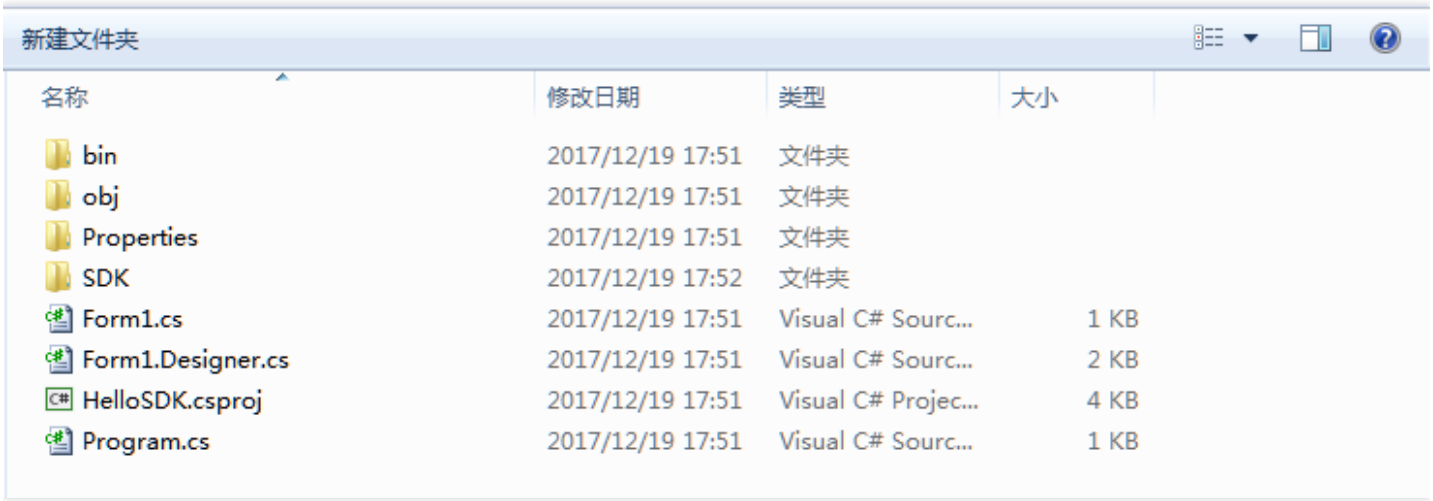
Step 1: Copy the SDK file

Create a C# WinForm application project and name it "HelloSDK". Make sure you use .NET Framework 4 or above.

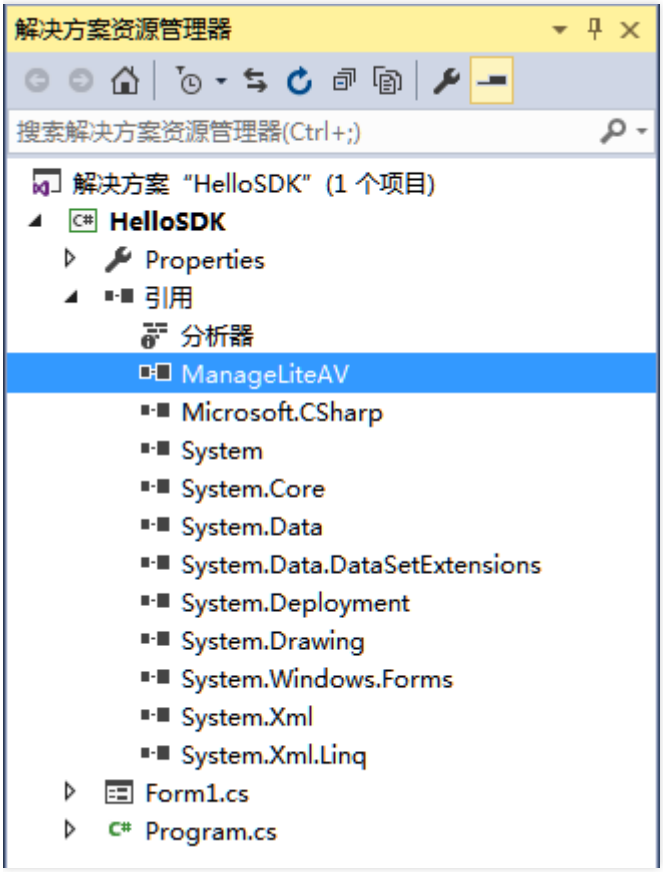


Copy the downloaded SDK file to the project directory.

©2013-2018 Tencent Cloud. All rights reserved.



Open the reference manager page in the HelloSDK project and click **Browse**. Add the reference "ManageLiteAV.dll", as shown below:



Step 2: Verify the configuration

Call SDK API in the HelloSDK code and display the camera preview to verify whether the project is correctly configured.

©2013-2018 Tencent Cloud. All rights reserved.

Add code

Add a "ManageLiteAV.TXLivePusher" instance to the Form1 class.

```
private ManageLiteAV.TXLivePusher pusher = new ManageLiteAV.TXLivePusher();
```

Add the "Load" event to the attribute panel. Add the code to enable the camera preview feature for "ManageLiteAV.TXLivePusher".

```
private void onLoaded(object sender, EventArgs e)
{
// At least one camera is available by default. So, enter 0 for cameraIndex.
pusher.startPreview(this.Handle, 0, 0, this.Width, this.Height, 0);
}
```

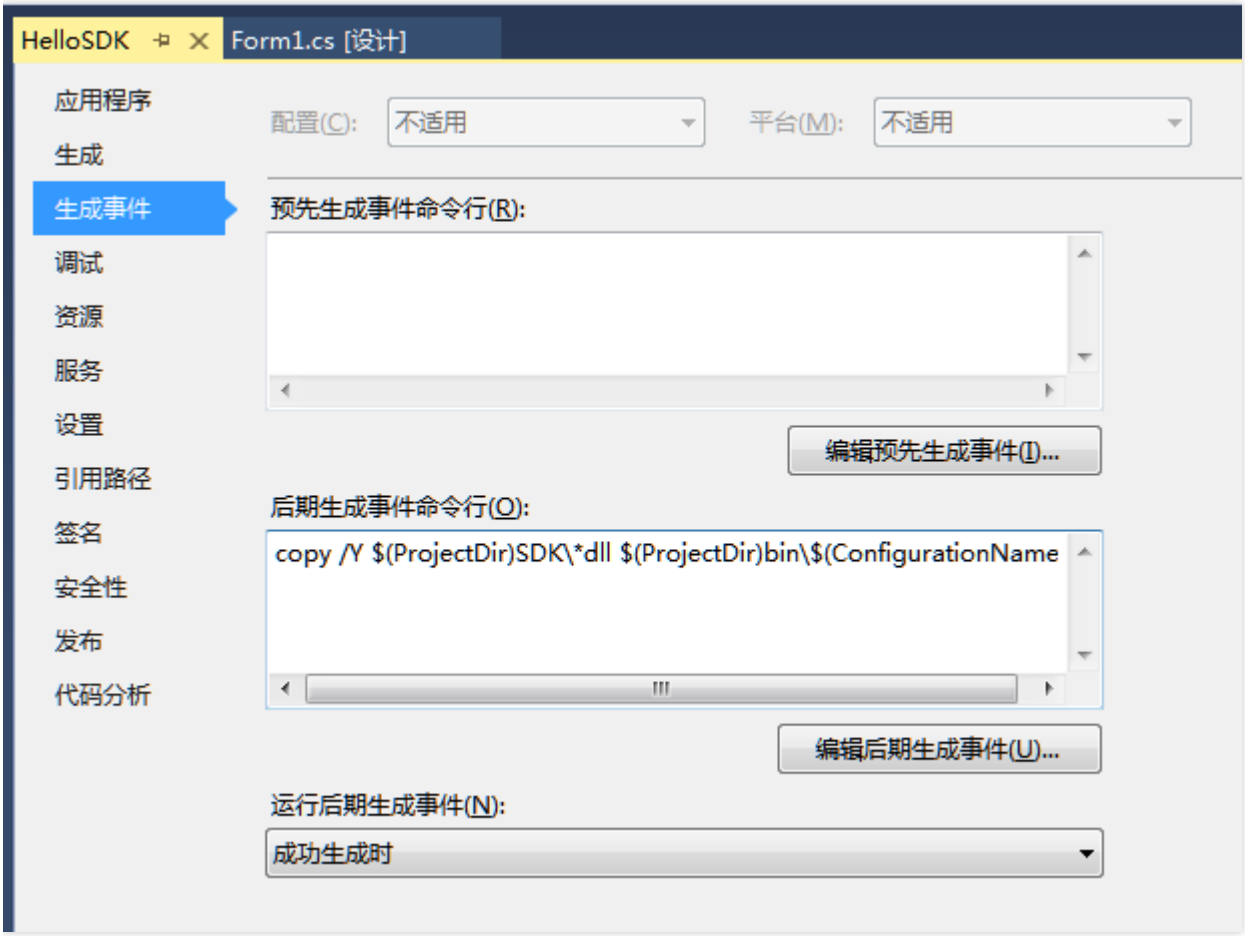
Compile code and run project

Open the attribute page of the HelloSDK project. Add the post-build command line, which is used to copy the .dll files to the directory where the exe file is located.

```
copy /Y $(ProjectDir)SDK\*dll $(ProjectDir)bin\$(ConfigurationName)
```

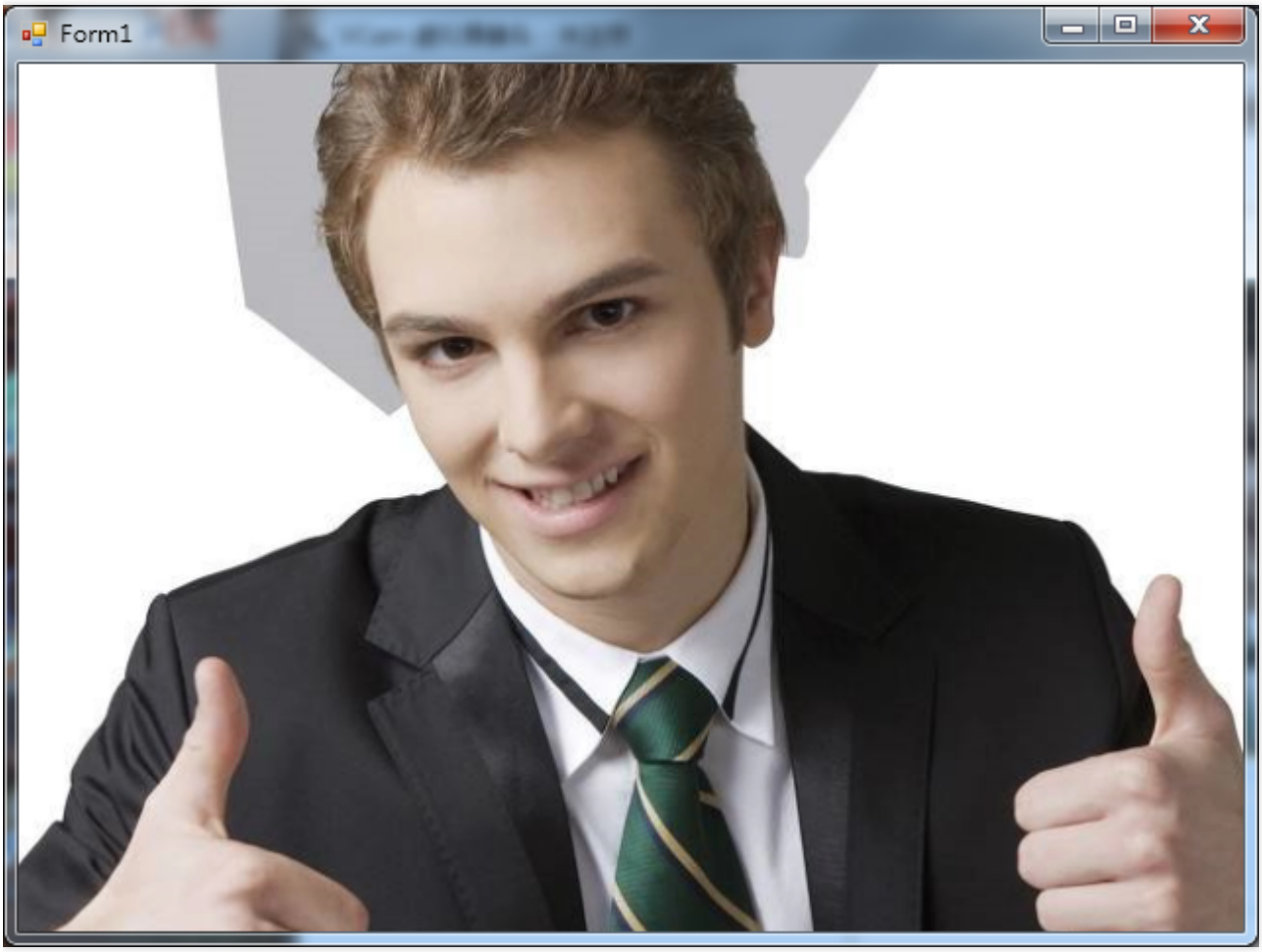
The command line is added as follows:

©2013-2018 Tencent Cloud. All rights reserved.



Compile (F7) the code, and then run and debug (F5) the project. You can see the camera image displayed in the window.

©2013-2018 Tencent Cloud. All rights reserved.



Now, the project configuration is completed.

©2013-2018 Tencent Cloud. All rights reserved.

Basic Features
API (C#)

Last updated：2018-07-11 11:17:28

API List

API Document

Download SDK from the Tencent Cloud official website to view the API comments under SDK\include.

The following lists the APIs of Tencent Video Cloud Windows SDK (C++), which are categorized into TXLivePusher and TXLivePlayer APIs, as well as relevant enums and global constants.

TXLivePusher

APIs

Name	Description
setCallback(callback, pUserData)	Sets the callback proxy of TXLivePusher
enumCameras(camerasName, capacity)	Enumerates current cameras
micDeviceCount()	Queries the number of available microphones
micDeviceName(index, name[])	Queries the names of microphones
selectMicDevice(index)	Selects the specified microphone as the recording device
micVolume()	Queries the volume of the selected microphone
setMicVolume(volume)	Sets the volume of the selected microphone
enableMic(enable)	Indicates whether to enable microphone (used with audio mixing)
openSystemVoiceInput(szPlayerPath)	Enables system sound capture

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
closeSystemVoiceInput()	Disables system sound capture
setSystemVoiceInputVolume(value)	Sets the volume for system sound capture
startAudioCapture(srcType)	Enables audio capture
stopAudioCapture()	Disables audio capture
startPreview(rendHwnd, rect, cameraIndex)	Enables camera preview
startPreview(srcType,rendHwnd,rect,dataFormat)	Enables video preview, including screencap, camera and external data sources
updatePreview(rendHwnd, rect)	Resets camera preview area
stopPreview()	Disables camera preview
enumCaptureWindow(windowArray, capacity)	Enumerates windows available for capture
setScreenCaptureParam(captureHwnd, rect)	Sets screen capture parameters
captureVideoSnapShot(filePath, length)	Captures screenshots of the pushed images and saves to local device
startPush(url)	Start push
stopPush()	Stops push
switchCamera(cameraIndex)	Switches between cameras. Dynamic switching during push is supported.
setMute(mute)	Enables Mute
setRenderMode(mode)	Sets the rendering (fill) mode of image
setRotation(rotation)	Sets the clockwise rotation of image
setVideoQualityParamPreset(paramType)	Sets the preset options for pushed image quality
setVideoResolution(resolution)	Sets video resolution
setBeautyStyle(beautyStyle , beautyLevel, whitenessLevel)	Sets beauty filter and whitening effects
setRenderYMirror(bMirror)	Sets the mirroring effect for preview rendering

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
setOutputYMirror(bMirror)	Sets the mirroring effect of pushed images
setVideoBitRate(bitrate)	Sets video bitrate
setAutoAdjustStrategy(adjuststrategy)	Sets traffic control policy
setVideoBitRateMin(videoBitrateMin)	The minimum video bitrate of SDK output (used with setAutoAdjustStrategy)
setVideoBitRateMax(videoBitrateMax)	The maximum video bitrate of SDK output (used with setAutoAdjustStrategy)
setVideoFPS(fps)	Sets video frame rate
setPauseVideo(bPause)	Sets video pause

TXLivePlayer

APIs

Name	Description
setCallback(callback, pUserData)	Sets the callback proxy of TXLivePlayer
speakerDeviceCount()	Queries the number of available speakers
speakerDeviceName(index, name)	Queries the names of speakers
selectSpeakerDevice(index)	Selects the specified speaker as the audio playback device
speakerVolume()	Queries the volume for SDK playback
setSpeakerVolume(volume)	Sets the volume for SDK playback
setRenderFrame(hWnd, rect)	Sets video image rendering attributes
updateRenderFrame(hWnd, rect)	Resets image rendering area
closeRenderFrame()	Disables image rendering
startPlay(url, type)	Starts playback
stopPlay()	Stops playback
pause()	Pauses playback

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
resume()	Resumes playback
isPlaying()	Indicates whether the playback is in progress
setMute(mute)	Enables Mute
setRenderMode(mode)	Sets the rendering (fill) mode of image
setRotation(TXEVideoRotation)	Sets the clockwise rotation of image
setRenderYMirror(mirror)	Sets the mirroring effect for rendering
setOutputVideoFormat(format)	Sets the video encoding format
captureVideoSnapShot(filePath,length)	Captures screenshots of the pulled images and saves to local device

Details of TXLivePusher APIs

1.setCallback(callback, pUserData)

API details: void setCallback(callback, pUserData)
Sets the callback proxy of TXLivePusher for listening on push events.

- **Parameter description**

Parameter	Type	Description
callback	TXLivePusherCallback *	Proxy pointer
pUserData	void *	Transparently transfers user data to the callback function of TXLivePusherCallback

- **Sample code**:

```
pusher.setCallback(this, reinterpret_cast<void*>(index));
```

2.enumCameras(camerasName, capacity)

API details: int enumCameras(camerasName, capacity)

©2013-2018 Tencent Cloud. All rights reserved.

Enumerates current cameras. If multiple cameras are installed on a Windows machine, you can use this function to get the number and names of the available cameras.

- **Parameter description**

Parameter	Type	Description
camerasName	wchar_t **	Camera names
capacity	size_t	Size of camerasName array

- **Sample code**:

```
m_cameraCount = pusher.enumCameras();
wchar_t **camerasName = new wchar_t *[m_cameraCount];
for (int i = 0; i < m_cameraCount; ++i)
{
camerasName[i] = new wchar_t[256];
}
pusher.enumCameras(camerasName, m_cameraCount);
```

3. micDeviceCount()

API details: int micDeviceCount() const

Queries the number of available microphones.

- **Returned result**

Parameter	Type	Description
vRet	int	Number of microphones

- **Sample code**:

```
int ret = pusher.micDeviceCount();
```

4. micDeviceName(index,name)

API details: bool micDeviceName(unsigned int index, char name[SPEAKER_DEVICE_NAME_MAX_SIZE])

Queries the names of microphones.

©2013-2018 Tencent Cloud. All rights reserved.

- **Parameter description**

Parameter	Type	Description
index	int	Index of microphone
name	char[]	A string array of microphone names

- **Sample code**:

```
int index = 0;
char name[SPEAKER_DEVICE_NAME_MAX_SIZE];
pusher.micDeviceName(index, name);
```

5. selectMicDevice(index)

API details: void selectMicDevice(unsigned int index)

Selects the specified microphone as the audio recording device. When this API is not called, the microphone with index 0 is selected by default.

- **Parameter description**

Parameter	Type	Description
index	int	Index of microphone

- **Sample code**:

```
int index = 0;
pusher.selectMicDevice(index);
```

6. micVolume()

API details: unsigned int micVolume()

Queries the volume of the selected microphone.

- **Returned result**

Parameter	Type	Description

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
vRet	int	Volume value, which ranges from 0 to 65535.

- **Sample code**:

```
int ret = pusher.micVolume();
```

7. setMicVolume(volume)

API details: void setMicVolume(unsigned int volume)

Sets the volume of the selected microphone.

- **Parameter description**

Parameter	Type	Description
volume	int	The volume you set, which ranges from 0 to 65535.

- **Sample code**:

```
int volume = 100;
pusher.setMicVolume(volume);
```

8. enableMic(enable)

API details: void enableMic(bool enable)

Indicates whether to enable microphone.

- **Parameter description**

Parameter	Type	Description
enable	bool	false: Disable microphone capture; true: Enable microphone capture

- **Sample code**:

```
pusher.enableMic(true);
```

©2013-2018 Tencent Cloud. All rights reserved.

9. openSystemVoiceInput(szPlayerPath)

API details: void openSystemVoiceInput(const char* szPlayerPath)

Enables system or process sound input for mixing with the microphone sound input.

- **Parameter description**

Parameter	Type	Description
szPlayerPath	string	Player address. If this parameter is left empty or filled with null, all sounds in the system are captured. If the path of the exe program (e.g. Kugou Music or QQ Music) is entered, the program is started and only the sounds from the program are captured. \

- **Sample code**:

```
pusher.openSystemVoiceInput(null);
```

10. closeSystemVoiceInput()

API details: void closeSystemVoiceInput()

Disables system or process sound input for mixing.

- **Sample code**:

```
pusher.closeSystemVoiceInput();
```

11. setSystemVoiceInputVolume(value)

API details: void setSystemVoiceInputVolume(int value)

Sets the volume for system sound capture.

- **Parameter description**

Parameter	Type	Description
value	int	Sets the volume you desired. The value ranges from 0 to 100.

- **Sample code**:

©2013-2018 Tencent Cloud. All rights reserved.

```
pusher.setSystemVoiceInputVolume(50);
```

12. startAudioCapture(srcType)

API details: void startAudioCapture(TXEAudioCaptureSrcType srcType = TXE_AUDIO_SRC_SDK_DATA)

Enables audio capture. The SDK uses a sampling rate of 48 KHz and 16-bit mono channel to deliver a real-time audio effect with a low delay.

- **Parameter description**

Parameter	Type	Description
srcType	enum	Audio data source: see the definition of TXEAudioCaptureSrcType.

- **Sample code**:

```
pusher.startAudioCapture(TXE_AudioSrc_Sdk_Data);
```

13. stopAudioCapture()

API details: void stopAudioCapture()

Disables audio capture.

- **Sample code**:

```
pusher.stopAudioCapture();
```

14.startPreview(rendHwnd, rect, cameraIndex)

API details: bool startPreview(rendHwnd, rect, cameraIndex)
Enables camera preview. true: API call is successful; false: API call failed.

- **Parameter description**

Parameter	Type	Description
rendHwnd	HWND	The HWND that identifies the preview window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
cameraIndex	int	Specifies the camera for which the preview is to be enabled.

- **Sample code**:

```
pusher.startPreview(m_pushRender, m_pushRect, 0);
```

15.startPreview(srcType,rendHwnd,rect,dataFormat)

API details: bool startPreview(TXEVideoCaptureSrcType srcType, HWND rendHwnd, const RECT &rect, TXEVideoUserDataFormat dataFormat)Enables the video source preview. The SDK supports various preview video sources, such as cameras and videos recorded by using screencap feature and videos specified by users. true: API call is successful; false: API call failed.

- **Parameter description**

Parameter	Type	Description
srcType	enum	Video source type. See the definition of TXEVideoCaptureSrcType.
rendHwnd	HWND	The HWND that identifies the preview window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.
dataFormat	enum	If srcType = TXE_VIDEO_SRC_USER_DATA, specifies the video format of the user data. See the definition of TXEVideoUserDataFormat.

- **Sample code**:

```
pusher.startPreview(TXE_VideoSrc_Sdk_Camera, m_pushRender, m_pushRect, TXE_UserData_Undefined);
```

16.updatePreview(rendHwnd, rect)

API details: void updatePreview(rendHwnd, rect)
Resets the preview area for camera. When the size of the window identified by the specified HWND changes, you can resize the video rendering area using this function.

©2013-2018 Tencent Cloud. All rights reserved.

- **Parameter description**

Parameter	Type	Description
rendHwnd	HWND	The HWND that identifies the preview window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

- **Sample code**:

```
pusher.updatePreview(m_pushRender, m_pushRect);
```

17.stopPreview()

API details: void stopPreview()

Disables camera preview. Calling this function before calling stopPush does not stop the push, but can cause the SDK to only push audio data.

- **Sample code**:

```
pusher.stopPreview();
```

18.enumCaptureWindow(windowArray, capacity)

API details: static int enumCaptureWindow(HWND* windowArray,size_t capacity)

Enumerates the windows available for capture. If there are multiple windows on the desktop at the same time, this function obtains the handles and names of the windows that can be captured.

- **Parameter description**

Parameter	Type	Description
windowArray	HWND*	Array of handles and names of windows that can be captured
capacity	size_t	Size of the windowArray

- **Sample code**:

©2013-2018 Tencent Cloud. All rights reserved.

```
int iCntWnd = m_pusher.enumCaptureWindow();

HWND *hwndList = new HWND[iCntWnd];
pusher.enumCaptureWindow(hwndList, iCntWnd);
```

19.setScreenCaptureParam(captureHwnd,rect)

API details: void setScreenCaptureParam(HWND captureHwnd, const RECT &rect)

Sets the screen area capture parameters. This API is called before startPreview(srcType = TXE_VIDEO_SRC_SDK_SCREEN..). The entire desktop is captured by default.

- **Parameter description**

Parameter	Type	Description
captureHwnd	HWND	Specifies the window to be captured. If captureHwnd is not NULL, the window identified by it is captured as a whole, and the captureRect does not take effect.
rect	const RECT	Specifies the rectangle for capturing the area on main screen. It only takes effect when captureHwnd is NULL. When captureRect is {0}, the main screen is captured as a whole.

- **Sample code**:

```
pusher.setScreenCaptureParam(TXE_AudioSrc_Sdk_Data);
```

20. captureVideoSnapShot(filePath,length)

API details: int captureVideoSnapShot(wchar_t * filePath, unsigned int length)

Captures screenshots of the pushed images and saves to local device

- **Parameter description**

Parameter	Type	Description
filePath	wchar_t *	File path. Only .jpg format is supported.

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
length	unsigned int	Length of filePath

- **Sample code**:

```
std::wstring path = L"D:\\132.jpg";
pusher.captureVideoSnapShot(path.c_str(), path.size());
```

21.startPush(url)

API details: bool startPush(url)

Starts the push (you need to call startPreview to enable camera preview before calling startPush, otherwise only audio data is pushed)

Note: A push URL is exclusive, that is, a push URL can only be used by one pusher for pushing streams at a time.

true: API call is successful; false: API call failed. Memory allocation, resource request failure, and other reasons may result in the failure to return response.

- **Parameter description**

Parameter	Type	Description
url	const char *	A valid push URL that supports RTMP protocol. The URL starts with "rtmp://". For more information on how to obtain Tencent Cloud push URL, please see DOC.

- **Sample code**:

```
pusher.startPush(m_pushUrl.c_str());
```

22.stopPush()

API details: void stopPush()

Stops push.

- **Sample code**:

©2013-2018 Tencent Cloud. All rights reserved.

```
pusher.stopPush();
```

23.switchCamera(cameraIndex)

API details: void switchCamera(cameraIndex)

Switches between cameras. Dynamic switching during push is supported.

- **Parameter description**

Parameter	Type	Description
cameraIndex	int	Camera index. Returned result: 0-(number of cameras-1)

- **Sample code**:

```
pusher.switchCamera(0);
```

24.setMute(mute)

API details: void setMute(mute)

Enables Mute.

- **Parameter description**

Parameter	Type	Description
mute	bool	Indicates whether to enable Mute

- **Sample code**:

```
pusher.setMute(true);
```

25.setRenderMode(mode)

API details: void setRenderMode(mode)

Sets the rendering (fill) mode of image.

- **Parameter description**

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
mode	TXERenderMode	See TXERenderMode's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
pusher.setRenderMode(TXE_RENDER_MODE_ADAPT);
```

26.setRotation(rotation)

API details: void setRotation(rotation)

Sets the clockwise rotation of image.

- **Parameter description**

Parameter	Type	Description
rotation	TXEVideoRotation	See TXEVideoRotation's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
pusher.setRotation(TXE_VIDEO_ROTATION_90);
```

27.setVideoQualityParamPreset(paramType)

API details: void setVideoQualityParamPreset(TXEVideoQualityParamPreset paramType)

Sets the preset options for pushed image quality.

- **Parameter description**

Parameter	Type	Description
paramType	TXEVideoQualityParamPreset	See TXEVideoQualityParamPreset's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

©2013-2018 Tencent Cloud. All rights reserved.

```
pusher.setVideoQualityParamPreset(TXE_VIDEO_QUALITY_STANDARD_DEFINITION);
```

28.setVideoResolution(resolution)

API details: void setVideoResolution(resolution)

Sets video resolution.

- **Parameter description**

Parameter	Type	Description
resolution	TXEVideoResolution	See TXEVideoResolution's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
pusher.setVideoResolution(TXE_VIDEO_RESOLUTION_640x480);
```

29.setBeautyStyle(beautyStyle , beautyLevel, whitenessLevel)

API details: void setBeautyStyle(beautyStyle, beautyLevel, whitenessLevel)

Sets beauty filter and whitening effects.

- **Parameter description**

Parameter	Type	Description
beautyStyle	TXEBeautyStyle	See TXEBeautyStyle's enumerated values defined in TXLiveSDKTypeDef.h
beautyLevel	int	Beauty filter level: 1-9. 0 indicates disabling beauty filter. A greater value means a bigger effect.
whitenessLevel	Int	Whiteness level: 1-9. 0 indicates disabling whitening. A greater value means a bigger effect.

- **Sample code**:

```
pusher.setBeautyStyle(TXE_BEAUTY_STYLE_NATURE, 5, 5);
```

©2013-2018 Tencent Cloud. All rights reserved.

30.setRenderYMirror(mirror)

API details: void setRenderYMirror(mirror)

Sets the mirroring effect for preview rendering.

- **Parameter description**

Parameter	Type	Description
mirror	bool	true: the image is mirrored horizontally; false: the image remains as it is.

- **Sample code**:

```
pusher.setRenderYMirror(true);
```

31.setOutputYMirror(mirror)

API details: void setOutputYMirror(mirror)

Sets the mirroring effect of pushed image.

- **Parameter description**

Parameter	Type	Description
mirror	bool	true: the image is mirrored horizontally; false: the image remains as it is.

- **Sample code**:

```
pusher.setOutputYMirror(true);
```

32.setVideoBitRate(bitrate)

API details: void setVideoBitRate(bitrate)

Sets video bitrate. Note: The higher the bitrate, the clearer the image is. But a higher resolution does not necessarily mean a higher image clarity.

- **Parameter description**

Parameter	Type	Description

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
bitrate	unsigned long	Video bitrate (in Kbps). For example, a resolution of 640x360 is used with a video bitrate of 800 Kbps.

- **Sample code**:

```
pusher.setVideoBitRate(800);
```

33.setAutoAdjustStrategy(strategy)

API details: void setAutoAdjustStrategy(strategy)

Sets the traffic control policy, that is, whether to allow SDK to adjust the video bitrate to adapt to the network condition, so as to avoid the stutter caused by slow upload speed.

- **Parameter description**

Parameter	Type	Description
strategy	TXEAutoAdjustStrategy	See TXEAutoAdjustStrategy's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
pusher.setAutoAdjustStrategy(TXE_AUTO_ADJUST_REALTIME_VIDEOCHAT_STRATEGY);
```

34.setVideoBitRateMin(videoBitrateMin) && setVideoBitRateMax(videoBitrateMax)

API details: void setVideoBitRateMin(videoBitrateMin)

void setVideoBitRateMax(videoBitrateMax)

They are used with setAutoAdjustStrategy. If the AutoAdjust policy is specified as TXE_AUTO_ADJUST_NONE, calls of both of the functions are invalid.

- **Parameter description**

Parameter	Type	Description
videoBitrateMin	int	The minimum video bitrate of SDK output. For example, this parameter is set to 300 Kbps for a resolution of 640x360.

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
videoBitrateMax	int	The maximum video bitrate of SDK output. For example, this parameter is set to 1000 Kbps for a resolution of 640x360.

- **Sample code**:

```
pusher.setVideoBitRateMin(300);
pusher.setVideoBitRateMax(1000);
```

35.setVideoFPS(fps)

API details: void setVideoFPS(fps)

Sets video frame rate.

- **Parameter description**

Parameter	Type	Description
fps	unsigned long	Video frame rate, which takes effect after restart. Default: 15.

- **Sample code**:

```
pusher.setVideoFPS(15);
```

36.setPauseVideo(bPause)

API details: void setPauseVideo(bool bPause)

Sets video frame rate.

- **Parameter description**

Parameter	Type	Description
bPause	bool	Indicates whether to pause the video.

- **Sample code**:

©2013-2018 Tencent Cloud. All rights reserved.

```
pusher.setPauseVideo(true);
```

##

TXLivePlayer Object APIs

1.setCallback(callback, pUserData)

API details: void setCallback(callback, pUserData)
Sets the callback proxy of TXLivePlayer for listening on push events.

- **Parameter description**

Parameter	Type	Description
callback	TXLivePlayerCallback *	Proxy pointer
pUserData	void *	Transparently transfers user data to the callback function of TXLivePlayerCallback

- **Sample code**:

```
player.setCallback(this, reinterpret_cast<void*>(index));
```

2. speakerDeviceCount()

API details: int speakerDeviceCount() const

Queries the number of available speakers.

- **Returned result**

Parameter	Type	Description
vRet	int	Number of speakers

- **Sample code**:

```
int ret = player.speakerDeviceCount();
```

©2013-2018 Tencent Cloud. All rights reserved.

3. speakerDeviceName(index,name)

API details: bool speakerDeviceName(unsigned int index, char name[SPEAKER_DEVICE_NAME_MAX_SIZE])

Queries the names of speakers.

- **Parameter description**

Parameter	Type	Description
index	int	Index of speaker
name	string	A string array of speaker names

- **Sample code**:

```
int index = 0;
char name[SPEAKER_DEVICE_NAME_MAX_SIZE];
player.speakerDeviceName(index, name);
```

4. selectSpeakerDevice(index)

API details: void selectSpeakerDevice(unsigned int index)

Selects the specified speaker as the audio playback device. When this API is not called, the speaker with index 0 is selected by default.

- **Parameter description**

Parameter	Type	Description
index	int	Index of speaker

- **Sample code**:

```
int index = 0;
player.selectSpeakerDevice(index);
```

5. speakerVolume()

API details: unsigned int speakerVolume()

Queries the playback volume of SDK. The value ranges from 0 to 65535.

©2013-2018 Tencent Cloud. All rights reserved.

- **Returned result**

Parameter	Type	Description
vRet	int	Volume value, which ranges from 0 to 65535.

- **Sample code**:

```
int ret = player.speakerVolume();
```

6. setSpeakerVolume(volume)

API details: void setSpeakerVolume(unsigned int volume);

Sets the playback volume of SDK. Note: This is not the volume of system speaker.

- **Parameter description**

Parameter	Type	Description
volume	int	The volume you set, which ranges from 0 to 65535.

- **Sample code**:

```
int volume = 100;
player.setSpeakerVolume(volume);
```

7.setRenderFrame(hWnd, rect)

API details: void setRenderFrame(hWnd, rect)

Sets video image rendering window and area.

- **Parameter description**

Parameter	Type	Description
hWnd	HWND	The HWND that identifies the video image window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

©2013-2018 Tencent Cloud. All rights reserved.

- **Sample code**:

```
player.setRenderFrame(rendHwnd, rect);
```

8.updateRenderFrame(hWnd, rect)

API details: void updateRenderFrame(hWnd, rect)

Resets the image rendering area. When the size of the window identified by the specified HWND changes, you can resize the video rendering area using this function.

- **Parameter description**

Parameter	Type	Description
hWnd	HWND	The HWND that identifies the video image window.
rect	const RECT &	Specifies the rendering area of the video image on the window identified by HWND.

- **Sample code**:

```
player.updateRenderFrame(rendHwnd, rect);
```

9.closeRenderFrame()

API details: void closeRenderFrame()

Disables image rendering.

- **Sample code**:

```
player.closeRenderFrame();
```

10.startPlay(url, type)

API details: void startPlay(url, type)
Starts playback. You need to call setRenderFrame before calling startPlay.

- **Parameter description**

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
url	const char *	A valid pull URL, which is the video playback URL. The URL starts with "rtmp://". For more information on how to obtain Tencent Cloud pull URL, please see DOC.
type	TXEPlayType	Playback type. See TXEPlayType's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
player.startPlay(playUrl, PLAY_TYPE_LIVE_RTMP_ACC);
```

11.stopPlay()

API details: void stopPlay()

Stops playback.

- **Sample code**:

```
player.stopPlay();
```

12.pause()

API details: void pause()

Pauses playback.

- **Sample code**:

```
player.pause()
```

13.resume()

API details: void resume()

Resumes playback.

- **Sample code**:

```
player.resume()
```

©2013-2018 Tencent Cloud. All rights reserved.

14.isPlaying()

API details: bool isPlaying()

Indicates whether the playback is in progress.

true: in progress; false: not in progress.

- **Sample code**:

```
bool isPlaying = player.isPlaying();
```

15.setMute(mute)

API details: void setMute(mute)

- **Parameter description**

Parameter	Type	Description
mute	bool	Indicates whether to enable Mute

- **Sample code**:

```
player.setMute(true);
```

16.setRenderMode(mode)

API details: void setRenderMode(mode)

Sets the rendering (fill) mode of image.

- **Parameter description**

Parameter	Type	Description
mode	TXERenderMode	See TXERenderMode's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
player.setRenderMode(AX_TXE_RENDER_MODE_ADAPT);
```

©2013-2018 Tencent Cloud. All rights reserved.

17.setRotation(rotation)

API details: void setRotation(rotation)

Sets the clockwise rotation of image.

- **Parameter description**

Parameter	Type	Description
rotation	TXEVideoRotation	See TXEVideoRotation's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
player.setRotation(TXE_VIDEO_ROTATION_90);
```

18.setRenderYMirror(mirror)

API details: void setRenderYMirror(mirror)

Sets the mirroring effect for preview rendering.

- **Parameter description**

Parameter	Type	Description
mirror	bool	true: the image is mirrored horizontally; false: the image remains as it is.

- **Sample code**:

```
player.setRenderYMirror(true);
```

19. setOutputVideoFormat(format)

API details: void setOutputVideoFormat(format)

Sets video encoding format. Default format: TXE_OUTPUT_VIDEO_WITHOUT_OUTPUT

- **Parameter description**

Parameter	Type	Description

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
format	TXEOutputVideoFormat	Specifies video encoding format. See TXEOutputVideoFormat's enumerated values defined in TXLiveSDKTypeDef.h

- **Sample code**:

```
player.setOutputVideoFormat(TXE_OUTPUT_VIDEO_FORMAT_YUV420);
```

20. captureVideoSnapShot(filePath,length)

API details: int captureVideoSnapShot(wchar_t * filePath, unsigned int length)

Captures screenshots of pulled images and saves to local device.

- **Parameter description**

Parameter	Type	Description
filePath	wchar_t *	File path. Only .jpg format is supported.
length	unsigned int	Length of filePath

- **Sample code**:

```
std::wstring path = L"D:\\132.jpg";
player.captureVideoSnapShot(path.c_str(), path.size());
```

Enumeration Type Definition - API Parameters

TXEAudioCaptureSrcType

Audio data source types

```
TXE_AUDIO_SRC_SDK_DATA = 1, Audio capture source is the sound data from LiteAvSDK, including l
ocal microphone, local player, system, etc.
TXE_AUDIO_SRC_USER_PCM = 1001, Audio capture source is the PCM raw audio data uploaded by us
ers.
TXE_AUDIO_SRC_USER_AAC = 1002, Audio capture source is the AAC audio data uploaded by users.
```

©2013-2018 Tencent Cloud. All rights reserved.

TXEVideoCaptureSrcType

Video data source types

```
TXE_VIDEO_SRC_UNDEFINED = 0, No video capture source
TXE_VIDEO_SRC_SDK_CAMERA = 1, Video capture source is LiteAvSDK camera data.
TXE_VIDEO_SRC_SDK_SCREEN = 2, Video capture source is LiteAvSDK screencap data.
TXE_VIDEO_SRC_USER_DATA = 1001, Video capture source is the video data uploaded by users.
```

TXEVideoUserDataFormat

The formats of videos uploaded by users in several common scenarios

```
TXE_UserData_Undefined = 0, Block all video data uploaded by users
TXE_UserData_Yuv420P, Yuv420P video data
TXE_UserData_H264Nal, H264Nal-encoded video data
```

TXEVideoResolution

Push video resolutions

```
// Standard screen 4:3
TXE_VIDEO_RESOLUTION_320x240 = 1,
TXE_VIDEO_RESOLUTION_480x360 = 2,
TXE_VIDEO_RESOLUTION_640x480 = 3,
TXE_VIDEO_RESOLUTION_960x720 = 4,

// Wide screen 16:9
TXE_VIDEO_RESOLUTION_320x180 = 101,
TXE_VIDEO_RESOLUTION_480x272 = 102,
TXE_VIDEO_RESOLUTION_640x360 = 103,
TXE_VIDEO_RESOLUTION_960x540 = 104,

// Wide screen 9:16
TXE_VIDEO_RESOLUTION_180x320 = 201,
TXE_VIDEO_RESOLUTION_272x480 = 202,
TXE_VIDEO_RESOLUTION_360x640 = 203,
TXE_VIDEO_RESOLUTION_540x960 = 204,
```

TXERenderMode

Video rendering mode on window

©2013-2018 Tencent Cloud. All rights reserved.

```
TXE_RENDER_MODE_ADAPT = 1, // Adaption. The video image is displayed in full on the screen, possibly with black edges around it.
TXE_RENDER_MODE_FILLSCREEN = 2, // Filling. No black edge exists, but the parts beyond the rendering area are trimmed off, so that the image is centered on the screen.
```

TXEVideoRotation

Rendering video rotation

```
TXE_VIDEO_ROTATION_NONE = 1, // No rotation of the image.
TXE_VIDEO_ROTATION_90 = 2, // Rotates the image 90 degrees clockwise, with its width and height interchanged.
TXE_VIDEO_ROTATION_180 = 3, // Rotates the image 180 degrees clockwise, with the image reversed upside down.
TXE_VIDEO_ROTATION_270 = 4, // Rotates the image 270 degrees clockwise, with its width and height interchanged.
```

TXEAutoAdjustStrategy

Traffic control policy for pushing videos

```
TXE_AUTO_ADJUST_NONE = -1, // No traffic control. The video bitrate specified by setVideoBitRate is always used.
TXE_AUTO_ADJUST_LIVEPUSH_STRATEGY = 0, // Suitable for common push scenarios in LVB. This policy has a lower sensitivity and adapts to the bandwidth change slowly. This is helpful to maintain the image clarity when the bandwidth fluctuates.
TXE_AUTO_ADJUST_LIVEPUSH_RESOLUTION_STRATEGY = 1, // Suitable for common push scenarios in LVB. Upgraded from LIVEPUSH_STRATEGY, this policy allows the SDK to adapt the resolution to the bitrate automatically.
TXE_AUTO_ADJUST_REALTIME_VIDEOCHAT_STRATEGY = 5, // Suitable for real-time video chats. It is used by VIDEO_QUALITY_REALTIME_VIDEOCHAT.
// This policy is highly sensitive to network condition and makes adaptive adjustment in case of any network fluctuation.
```

TXEVideoQualityParamPreset

Preset options for SDK push image quality

```
TXE_VIDEO_QUALITY_STANDARD_DEFINITION = 1, // SD: Suitable for customers who care more about smoothness.
TXE_VIDEO_QUALITY_HIGH_DEFINITION = 2, // HD: Suitable for customers who care more about imag
```

©2013-2018 Tencent Cloud. All rights reserved.

```
e clarity.
TXE_VIDEO_QUALITY_SUPER_DEFINITION = 3, // SD: Not recommended unless you are watching vide
os on a large screen.
TXE_VIDEO_QUALITY_LINKMIC_MAIN_PUBLISHER = 4, // Joint broadcasting with the primary VJ
TXE_VIDEO_QUALITY_LINKMIC_SUB_PUBLISHER = 5, // Joint broadcasting with a secondary VJ
TXE_VIDEO_QUALITY_REALTIME_VIDEOCHAT = 6, // Real-time video chats
TXE_VIDEO_QUALITY_STILLIMAGE_DEFINITION = 7, // Suitable for scenarios featuring static image qu
ality. Generally, the screenshots captured and pushed by users have few dynamic changes. With the al
gorithm available for adaptive bitrate and resolution, custom settings are not recommended.
```

TXEOutputVideoFormat

Sets the video output format

```
TXE_OUTPUT_VIDEO_WITHOUT_OUTPUT = 1, // No data output
TXE_OUTPUT_VIDEO_FORMAT_YUV420 = 2, // yuv420 format
TXE_OUTPUT_VIDEO_FORMAT_RGBA = 3, // RBGA format
```

TXEBeautyStyle

Beauty filter style

```
TXE_BEAUTY_STYLE_SMOOTH = 0, // Smooth
TXE_BEAUTY_STYLE_NATURE = 1, // Natural
TXE_BEAUTY_STYLE_BLUR = 2, // Hazy
```

TXEPlayType

Playback type

```
PLAY_TYPE_LIVE_RTMP = 0, RTMP LVB
PLAY_TYPE_LIVE_RTMP_ACC, Accelerated playback in RTMP LVB
```

Definitions of CallBack Event Parameters

PushEvent

©2013-2018 Tencent Cloud. All rights reserved.

Push events

```
TXE_STATUS_UPLOAD_EVENT : 200001, // Push-related data

PUSH_EVT_CONNECT_SUCC = 1001, // Connected to the push server
PUSH_EVT_PUSH_BEGIN = 1002, // Handshake with the server completed. Push starts.
PUSH_EVT_OPEN_CAMERA_SUCC = 1003, // Camera enabled successfully
PUSH_EVT_CHANGE_RESOLUTION = 1005, // Resolution is changed dynamically during push
PUSH_EVT_CHANGE_BITRATE = 1006, // Bitrate is changed dynamically during push
PUSH_EVT_FIRST_FRAME_AVAILABLE = 1007, // The first frame is captured
PUSH_EVT_START_VIDEO_ENCODER = 1008, // Encoder started
PUSH_EVT_CAMERA_REMOVED = 1009, // Camera removed (for PC SDK)
PUSH_EVT_CAMERA_AVAILABLE = 1010, // Camera is available again (for PC SDK)
PUSH_EVT_CAMERA_CLOSED = 1011, // Camera disabled (for PC SDK)
PUSH_EVT_SNAPSHOT_RESULT = 1012, // Error code returned for screencap

PUSH_ERR_OPEN_CAMERA_FAIL = -1301, // Failed to enable camera
PUSH_ERR_OPEN_MIC_FAIL = -1302, // Failed to enable microphone
PUSH_ERR_VIDEO_ENCODE_FAIL = -1303, // Video encoding failed
PUSH_ERR_AUDIO_ENCODE_FAIL = -1304, // Audio encoding failed
PUSH_ERR_UNSUPPORTED_RESOLUTION = -1305, // Unsupported video resolution
PUSH_ERR_UNSUPPORTED_SAMPLERATE = -1306, // Unsupported audio sampling rate
PUSH_ERR_NET_DISCONNECT = -1307, // Network disconnected. Too many failed reconnection attem
pts. Restart the push for more retries.
PUSH_ERR_CAMERA_OCCUPY = -1308, // The camera is in use. Enable another camera (for PC SDK)

PUSH_WARNING_NET_BUSY = 1101, // Bad network condition: data upload is blocked because upstre
am bandwidth is too small
PUSH_WARNING_RECONNECT = 1102, // Network disconnected. Auto reconnection has been initiate
d (no more attempts will be made after three failed attempts)
PUSH_WARNING_HW_ACCELERATION_FAIL = 1103, // Failed to start hard-encoding. Soft-encoding is
used instead.
PUSH_WARNING_VIDEO_ENCODE_FAIL = 1104, // Video encoding failed. Non-fatal error. Encoder will
be restarted internally.
PUSH_WARNING_BEAUTYSURFACE_VIEW_INIT_FAIL = 1105, // Video encoding bitrate exception
PUSH_WARNING_VIDEO_ENCODE_BITRATE_OVERFLOW = 1106, // Video encoding bitrate exception
PUSH_WARNING_DNS_FAIL = 3001, // RTMP - DNS resolution failed
PUSH_WARNING_SEVER_CONN_FAIL = 3002, // Failed to connect to RTMP server
PUSH_WARNING_SHAKE_FAIL = 3003, // Handshake with RTMP server failed
PUSH_WARNING_SERVER_DISCONNECT = 3004, // RTMP server disconnected automatically. Check th
e validity of push URL or the validity period of the hotlink protection.
PUSH_WARNING_SERVER_NO_DATA = 3005, // No data was sent within 30 seconds. Server disconnect
ed automatically.
```

©2013-2018 Tencent Cloud. All rights reserved.

PlayEvent

Playback events

```
TXE_STATUS_DOWNLOAD_EVENT : 200002, // Pull-related data

PLAY_EVT_CONNECT_SUCC : 2001, // Connected to the server
PLAY_EVT_RTMP_STREAM_BEGIN : 2002, // Connected to the server. Pull started.
PLAY_EVT_RCV_FIRST_I_FRAME : 2003, // The first video data packet (IDR) is rendered
PLAY_EVT_PLAY_BEGIN : 2004, // Video playback started
PLAY_EVT_PLAY_PROGRESS : 2005, // Video playback progress
PLAY_EVT_PLAY_END : 2006, // Video playback ended
PLAY_EVT_PLAY_LOADING : 2007, // Video playback loading
PLAY_EVT_START_VIDEO_DECODER : 2008, // Decoder started
PLAY_EVT_CHANGE_RESOLUTION : 2009, // Video resolution changed
PLAY_EVT_SNAPSHOT_RESULT : 2010, // Error code returned for screencap

PLAY_ERR_NET_DISCONNECT : -2301, // Network disconnected. Too many failed reconnection attemp
ts. Restart the playback for more retries.
PLAY_ERR_GET_RTMP_ACC_URL_FAIL : -2302, // Failed to get the accelerated pull address

PLAY_WARNING_VIDEO_DECODE_FAIL : 2101, // Failed to decode the current video frame
PLAY_WARNING_AUDIO_DECODE_FAIL : 2102, // Failed to decode the current audio frame
PLAY_WARNING_RECONNECT : 2103, // Network disconnected. Auto reconnection has been initiated
(no more attempts will be made after three failed attempts)
PLAY_WARNING_RECV_DATA_LAG : 2104, // Unstable inbound packet: This may be caused by insuffici
ent downstream bandwidth, or inconsistent outbound stream from the VJ end.
PLAY_WARNING_VIDEO_PLAY_LAG : 2105, // Stutter occurred during the video playback (user experie
nce)
PLAY_WARNING_HW_ACCELERATION_FAIL : 2106, // Failed to start hard decoding. Soft decoding is us
ed instead.
PLAY_WARNING_VIDEO_DISCONTINUITY : 2107, // Discontinuous sequence of video frames. Some fra
mes may be dropped.
PLAY_WARNING_FIRST_IDR_HW_DECODE_FAIL : 2108, // Hard decoding of the first I-frame of current
stream failed. Switched to soft decoding automatically.
PLAY_WARNING_DNS_FAIL : 3001, // RTMP - DNS resolution failed
PLAY_WARNING_SEVER_CONN_FAIL : 3002, // Failed to connect to RTMP server
PLAY_WARNING_SHAKE_FAIL : 3003, // Handshake with RTMP server failed
PLAY_WARNING_SERVER_DISCONNECT : 3004, // RTMP server disconnected automatically
```

TXE_STATUS_UPLOAD_EVENT && TXE_STATUS_DOWNLOAD_EVENT

Definitions of status keys

©2013-2018 Tencent Cloud. All rights reserved.

```
#define NET_STATUS_CPU_USAGE "CPU_USAGE" // CPU utilization
#define NET_STATUS_CPU_USAGE_D "CPU_USAGE_DEVICE" // Total CPU usage of device
#define NET_STATUS_VIDEO_BITRATE "VIDEO_BITRATE" // The output bitrate of the current video encoder - the video data bits produced by the encoder per sec (in Kbps).
#define NET_STATUS_AUDIO_BITRATE "AUDIO_BITRATE" // The output bitrate of the current audio encoder - the audio data bits produced by the encoder per sec (in Kbps).
#define NET_STATUS_VIDEO_FPS "VIDEO_FPS" // Current video frame rate - the number of frames produced by video encoder per sec.
#define NET_STATUS_VIDEO_GOP "VIDEO_GOP" // I-frame interval of current video: the interval between I-frames of video encoder (in sec).
#define NET_STATUS_NET_SPEED "NET_SPEED" // Current network speed
#define NET_STATUS_NET_JITTER "NET_JITTER" // Network jitter. The greater the jitter, the more unstable the network.
#define NET_STATUS_CACHE_SIZE "CACHE_SIZE" // Buffer size. A larger buffer indicates the current upstream bandwidth is not enough to consume the video data produced.
#define NET_STATUS_DROP_SIZE "DROP_SIZE"
#define NET_STATUS_VIDEO_WIDTH "VIDEO_WIDTH"
#define NET_STATUS_VIDEO_HEIGHT "VIDEO_HEIGHT"
#define NET_STATUS_SERVER_IP "SERVER_IP"
#define NET_STATUS_CODEC_CACHE "CODEC_CACHE" // Buffer size for encoding/decoding
#define NET_STATUS_CODEC_DROP_CNT "CODEC_DROP_CNT" // Encoding/decoding queue DROPCNT
#define NET_STATUS_SET_VIDEO_BITRATE "SET_VIDEO_BITRATE"
```

TXLivePusherCallback

Listening on push events

API Definition	Description
onEventCallback(eventId, paramCount, paramKeys, paramValues, pUserData)	Push event notification of TXLivePusher

TXLivePlayerCallback

Listening on playback events

API Definition	Description

©2013-2018 Tencent Cloud. All rights reserved.

API Definition	Description
onEventCallback(eventId, paramCount, paramKeys, paramValues, pUserData)	Playback event notification of TXLivePlayer

©2013-2018 Tencent Cloud. All rights reserved.

Push and Play

Last updated：2018-07-11 11:17:43

Push and Playback

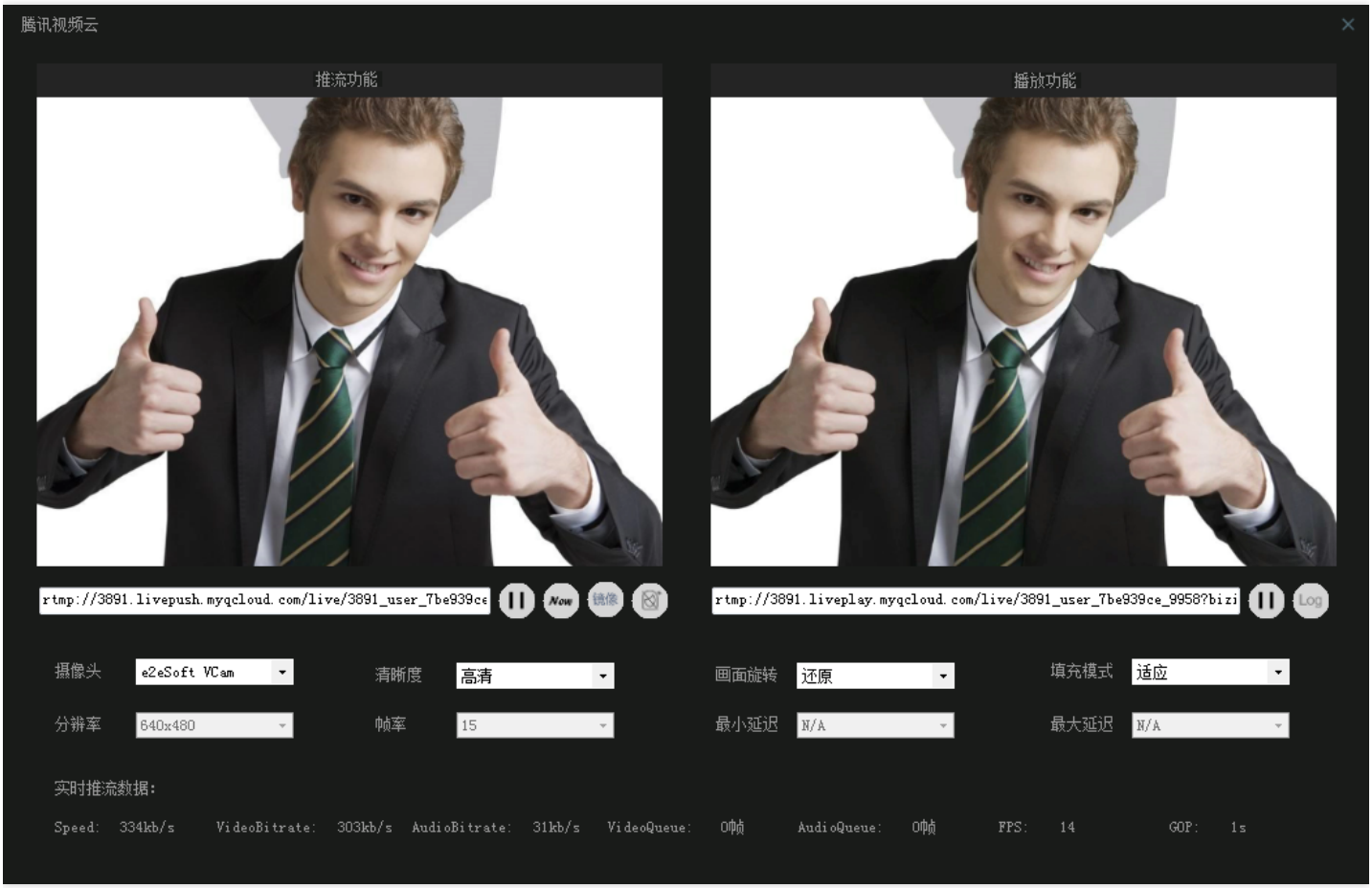
Product Description

Tencent Video Cloud SDK is an audio/video SDK component that integrates **standard push and pull** and **real-time video call**. This document mainly introduces the integration solution of C# version.

Compared with Obs Studio (a commonly used push software) and Flash player (a common playback plug-in for PC browsers), the significant advantages of the Video Cloud SDK are:

- Low-delay optimization: Real-time audio/video calls.
- UDP protocol acceleration: Better push stability.

©2013-2018 Tencent Cloud. All rights reserved.



Preparations

- **Acquiring SDK**
  Download SDK and follow the instructions in Project Configuration to add the SDK into your application development project.

- **Push URL for testing**
  The Demo supports getting the push URL for testing automatically from the backend. Click the **New** button in the above figure to get the push URL.

  You can also manually generate a push URL for testing. After activating the LVB service, use the LVB Console -> LVB Code Access -> Push Generator to generate a push URL. For more information, please see Acquiring Push/Playback URL.

- **Playback URL for testing**
  Push URL `rtmp://8888.livepush.myqcloud.com/live/8888_teststream?`

©2013-2018 Tencent Cloud. All rights reserved.

`bizid=8888&txSecret=6e18e8db0ff2070a339ab739ff46b957&txTime=5A3E7D7F`
The playback URL is: `rtmp://8888.liveplay.myqcloud.com/live/8888_teststream`

Push

Code interfacing

Step 1: Create a Pusher object

Create a **ManageLiteAV.TXLivePusher** object, which will be used later for pushing.

After the creation, you can call the configuration API of ManageLiteAV.TXLivePusher to set the mirror effect, bitrate, resolution and fill mode before pushing.

```
pusher.setRenderYMirror(true);
pusher.setOutputYMirror(true);
pusher.setVideoBitRate(900);
pusher.setVideoBitRateMin(300);
pusher.setVideoBitRateMax(1200);
pusher.setVideoResolution(ManageLiteAV.TXEVideoResolution.TXE_VIDEO_RESOLUTION_640x480);
pusher.setRenderMode(ManageLiteAV.TXERenderMode.TXE_RENDER_MODE_FILLSCREEN);
...
```

Step 2: Set the preview area

We need a place to display the camera images. For Windows, HWND window handle is used as the basic rendering unit. In a C# form application, each control can call its Handle to get the HWND. Therefore, you simply need to prepare a **startPreview** API passed by HWND to the ManageLiteAV.TXLivePusher object.

```
// Set the callback for audio/video data and internal SDK events.
pusher.setCallback(this, (IntPtr)UserDataFlag.PusherFlag);
// Enable the camera, and the index value can be obtained via the enumCameras API of the Manage LiteAV.TXLivePusher object.
pusher.startPreview(pushRenderPanel.Handle, 0, 0, pushRenderPanel.Width, pushRenderPanel.Heigh t, cameraIndex);
```

Step 3: Start push

After the preparations in Step 1 and Step 2, you can use the following codes to start the push:

```
string rtmpURL = "rtmp://2157.livepush.myqcloud.com/live/xxxxxx";
pusher.startPush(rtmpURL);
```

©2013-2018 Tencent Cloud. All rights reserved.

Step 4: Control the camera

If there are multiple cameras in a computer, you can switch between different cameras simply by calling the **switchCamera** API of ManageLiteAV.TXLivePusher object and passing the index value of the camera.

```
// Pass the index value of camera to be switched to
pusher.switchCamera(cameraComboBox.SelectedIndex);
```

- The enumCameras function of TXLivePusher can get the index value of the camera.
- Enabling a USB camera under Windows takes a long time for circuit and drive startup. You can set the last parameter of the startPreview of TXLivePusher the last parameter of the startPreview of TXLivePusher to -1, which means to enable all cameras. At this time, the response time for switching cameras will be much shorter.
- SDK supports virtual cameras, whose enabling method is the same as that of normal cameras. If your PC does not have a physical camera, you can install virtual camera software such as VCam to assist debugging.

Step 5: End push

Ending a push is simple, but proper cleanup work is required. Since only one ManageLiteAV.TXLivePusher object can run at a time, improper cleanup may adversely affect the next LVB.

```
// Set callback as null
pusher.setCallback(null, (IntPtr)null);
// Stop camera preview
pusher.stopPreview();
// Stop push
pusher.stopPush();
```

Step 6: More features

For more information on such features of the local camera as screen mirroring, mute and sharpness settings, please see API List.

Event handling

The SDK listens for push-related events using the setListener API of the ManageLiteAV.TXLivePusher object.

1. Normal events

A notification event is prompted after each successful push. For example, receiving 1003 means that the system will start rendering the camera pictures.

©2013-2018 Tencent Cloud. All rights reserved.

Event ID	Value	Description
PUSH_EVT_CONNECT_SUCC	1001	Successfully connected to push server
PUSH_EVT_PUSH_BEGIN	1002	Handshake with the server completed, and ready to start push
PUSH_EVT_OPEN_CAMERA_SUCC	1003	Camera is enabled
PUSH_EVT_CHANGE_RESOLUTION	1005	Dynamic push resolution adjustment
PUSH_EVT_CHANGE_BITRATE	1006	Dynamic push bitrate adjustment
PUSH_EVT_FIRST_FRAME_AVAILABLE	1007	The first frame is captured
PUSH_EVT_START_VIDEO_ENCODER	1008	Encoder is started
PUSH_EVT_CAMERA_REMOVED	1009	Camera device has been removed
PUSH_EVT_CAMERA_AVAILABLE	1010	Camera device is available again
PUSH_EVT_CAMERA_CLOSED	1011	Camera is disabled

2. Error notification

The push cannot continue as the SDK detected critical problems. For example, the camera has been occupied by other programs so the camera cannot be started.

Event ID	Value	Description
PUSH_ERR_OPEN_CAMERA_FAIL	-1301	Failed to start the camera
PUSH_ERR_OPEN_MIC_FAIL	-1302	Failed to start the microphone
PUSH_ERR_VIDEO_ENCODE_FAIL	-1303	Video encoding failed
PUSH_ERR_AUDIO_ENCODE_FAIL	-1304	Audio encoding failed
PUSH_ERR_UNSUPPORTED_RESOLUTION	-1305	Unsupported video resolution
PUSH_ERR_UNSUPPORTED_SAMPLERATE	-1306	Unsupported audio sampling rate
PUSH_ERR_NET_DISCONNECT	-1307	Network disconnected. Reconnection attempts have failed for many times, thus no more retries will be performed. Please restart the push manually

©2013-2018 Tencent Cloud. All rights reserved.

Event ID	Value	Description
PUSH_ERR_CAMERA_OCCUPY	-1308	The camera is being occupied. Enable another camera.

3. Warning events

Compared with the error message, issues represented by WARNING events usually do not interrupt the push process, and SDK generally initiates automatic repair logic internally. You don't need to care about most WARNING events, but the following two events are important and meaningful:

WARNING_NET_BUSY

Poor upstream network speed. It represents that the user's current audio/video data cannot be pushed smoothly to the server. In this case, generally the user can be given some UI prompts, for example, "Your network speed is not good" so as to allow the user to prepare for the lag on the other end.

WARNING_SERVER_DISCONNECT

The push request is rejected by the backend. This is usually because that txSecret in the push address is calculated incorrectly, or the push address is occupied by others (push URL is exclusive and a push URL can only be used by one user at a time).

Event ID	Value	Description
PUSH_WARNING_NET_BUSY	1101	Bad network connection: data upload is blocked because upstream bandwidth is too small
PUSH_WARNING_RECONNECT	1102	Network disconnected. Auto reconnection has been initiated (no more attempts will be made after three failed attempts)
PUSH_WARNING_HW_ACCELERATION_FAIL	1103	Failed to start hard-encoding. Soft-encoding is used.
PUSH_WARNING_VIDEO_ENCODE_FAIL	1104	Video encoding failed. Non-fatal error. Encoder will be restarted internally.
PUSH_WARNING_BEAUTYSURFACE_VIEW_INIT_FAIL	1105	Warning of video encoding bitrate error
PUSH_WARNING_VIDEO_ENCODE_BITRATE_OVERFLOW	1106	Warning of video encoding bitrate error

©2013-2018 Tencent Cloud. All rights reserved.

Event ID	Value	Description
PUSH_WARNING_DNS_FAIL	3001	RTMP - DNS resolution failed (this will trigger retry process)
PUSH_WARNING_SEVER_CONN_FAIL	3002	Failed to connect to the RTMP server (this will trigger retry process)
PUSH_WARNING_SHAKE_FAIL	3003	Handshake with RTMP server failed (this triggers a retry)
PUSH_WARNING_SERVER_DISCONNECT	3004	RTMP server disconnected automatically. Check the validity of push URL or the validity period of hotlink protection
PUSH_WARNING_SERVER_NO_DATA	3005	Actively disconnected for no data is sent for more than 30s.

Playback Feature

Code interfacing

Step 1: Create a Player object

Create a **ManageLiteAV.TXLivePlayer** object, which will be used later for pushing.

```
player = new ManageLiteAV.TXLivePlayer();
```

Step 2: Set the rendering area

We need a place to display the player's video image. For Windows, HWND window handle is used as the basic rendering unit. In a C# form application, each control can call its Handle to get the HWND.

```
//Set callback
player.setListener(this, (IntPtr)UserDataFlag.PlayerFlag);
// Pass HWND window handle, rendering position and size
player.setRenderFrame(pullRenderPanel.Handle, 0, 0, pullRenderPanel.Width, pullRenderPanel.Height);
```

Step 3: CDN playback

©2013-2018 Tencent Cloud. All rights reserved.

To start the playback feature, simply call the startPlay API of ManageLiteAV.TXLivePlayer, where PLAY_TYPE_LIVE_RTMP is the RTMP URL of the regular CDN. The advantage is that the bandwidth price is very low, but the delay is usually high.

```
string rtmpURL = "rtmp://2157.liveplay.myqcloud.com/live/xxxxxx";
player.startPlay(rtmpURL,ManageLiteAV.TXEPlayType.PLAY_TYPE_LIVE_RTMP);
```

Step 3': Ultra-low delay playback

Ultra-low delay playback can pull the end-to-end delay down to about 400ms. In this mode, the internal mechanism of SDK is completely different from that in the normal mode. Meanwhile, the audio/video lines used by SDK are not regular CDN lines. Compared with CDN, the unit price is higher. It is generally applicable to audio/video calls and scenarios that highly requiring low delay.

```
string rtmpURL = "rtmp://2157.liveplay.myqcloud.com/live/xxxxxxbizid=2157&
txSecret=6e18e8db0ff2070a339ab739ff46b957&txTime=5A3E7D7F";
player.startPlay(rtmpURL,ManageLiteAV.TXEPlayType.PLAY_TYPE_LIVE_RTMP_ACC);
```

- This feature does not need to be enabled in advance, but it requires that LVB stream be located in Tencent Cloud. Implementing low-latency linkage across cloud providers is not just a technical difficulty.
- The playback URL cannot be a normal CDN URL. It must have a hotlink protection signature. For more information about the computing method of hotlink protection signature, please see **txTime&txSecret**.
- When calling the startPlay API, specify type as **PLAY_TYPE_LIVE_RTMP_ACC** for SDK to pull an ultra-low delay LVB stream.
- It supports concurrence playback of 10 channels simultaneously at most. So you can only use it in interactive scenarios (such as the channel of joint broadcasting VJ or the operator in Prize Claw LVB).
- The RTMP stream launched by Obs Studio introduces a delay of more than 1s on the client. Therefore, use the video cloud SDK for pushing.

Step 4: Adjust the image

- **updateRenderFrame**
  When the window size specified by the HWND changes, the video rendering area can be rescaled via the **updateRenderFrame** function.
- **setRenderMode**

Option	Description

©2013-2018 Tencent Cloud. All rights reserved.

Option	Description
TXE_RENDER_MODE_ADAPT	Adaption. The video image is displayed in full on the screen, possibly with black edges around it.
TXE_RENDER_MODE_FILLSCREEN	Filling. No black edge exists, but the parts beyond the rendering area are trimmed off, so that the image is centered on the screen.

Step 5: End playback

To exit the current UI at the end of playback, call the stopPlay API of ManageLiteAV.TXLivePlayer.

```
// Set callback as null
player.setCallback(null, (IntPtr)null);
// Stop pull playback
player.stopPlay();
```

Step 6: More features

For more information, please see API List.

Event handling

Listen on push-related events using the setListener API of the ManageLiteAV.TXLivePlayer object.

1. Normal events

Event ID	Value	Description
PLAY_EVT_CONNECT_SUCC	2001	Connected to the server
PLAY_EVT_RTMP_STREAM_BEGIN	2002	Connected to the server and started to pull stream
PLAY_EVT_RCV_FIRST_I_FRAME	2003	Render the first video packet (IDR)
PLAY_EVT_PLAY_BEGIN	2004	Video playback starts
PLAY_EVT_PLAY_PROGRESS	2005	Video playback progress
PLAY_EVT_PLAY_END	2006	Video playback ends
PLAY_EVT_PLAY_LOADING	2007	Video playback loading
PLAY_EVT_START_VIDEO_DECODER	2008	Decoder starts
PLAY_EVT_CHANGE_RESOLUTION	2009	Video resolution changes

©2013-2018 Tencent Cloud. All rights reserved.

2. Error notification

Some fatal errors occurred with SDK, such as network disconnection, can cause the failure to continue playback. In this case, you need to deal with these errors accordingly.

Event ID	Value	Description
PLAY_ERR_NET_DISCONNECT	-2301	The network is disconnected and reconnection attempts failed, which will result in playback failure.
PLAY_ERR_GET_RTMP_ACC_URL_FAIL	-2302	Failed to acquire pull accelerating address, which will result in playback failure

3. Warning events

Non-fatal errors with SDK generally do not cause the stop of playback, so you can ignore the following events.

PLAY_WARNING_VIDEO_PLAY_LAG is an event that SDK throws to indicate the playback stutter. It means the lag between video images (the refresh time between two frames) exceeds 500 ms.

Event ID	Value	Description
PLAY_WARNING_VIDEO_DECODE_FAIL	2101	Failed to decode the current video frame
PLAY_WARNING_AUDIO_DECODE_FAIL	2102	Failed to decode the current audio frame
PLAY_WARNING_RECONNECT	2103	Network disconnected. Auto reconnection has been initiated (no more attempts will be made after three failed attempts)
PLAY_WARNING_RECV_DATA_LAG	2104	Unstable incoming packet from network: This may be caused by insufficient downstream bandwidth, or unstable outgoing stream at the VJ end
PLAY_WARNING_VIDEO_PLAY_LAG	2105	Stutter occurred during the current video playback (intuitive user experience)
PLAY_WARNING_HW_ACCELERATION_FAIL	2106	Failed to start hard-decoding; Soft-decoding is used (not supported for now)
PLAY_WARNING_VIDEO_DISCONTINUITY	2107	Current video frames are discontinuous and frame loss may occur

©2013-2018 Tencent Cloud. All rights reserved.

Event ID	Value	Description
PLAY_WARNING_FIRST_IDR_HW_DECODE_FAIL	2108	Hard decoding of Frame I for the current stream fails, and SDK automatically switches to soft decoding
PLAY_WARNING_DNS_FAIL	3001	RTMP - DNS resolution failed
PLAY_WARNING_SEVER_CONN_FAIL	3002	Failed to connect to the RTMP server
PLAY_WARNING_SHAKE_FAIL	3003	Handshake with RTMP server failed
PLAY_WARNING_SERVER_DISCONNECT	3004	The RTMP server is actively disconnected

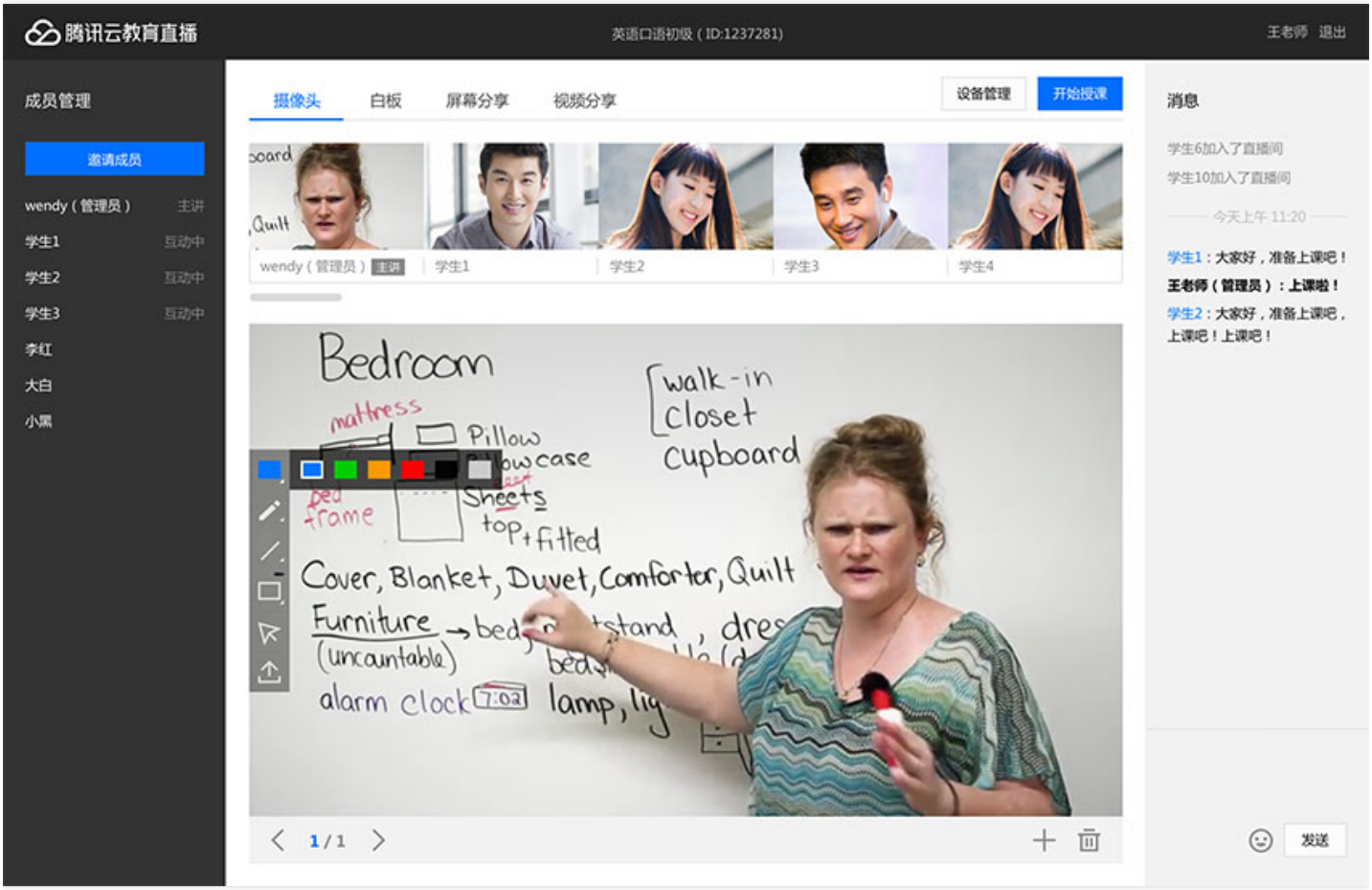
©2013-2018 Tencent Cloud. All rights reserved.

RealTime LiveRoom

Last updated：2018-07-11 11:08:05

LVB+Joint Broadcasting (LiveRoom)

LVB+Joint Broadcasting is an LVB mode commonly used in the **Live Show** and **Online Education** scenarios. With a good applicability to many scenarios, it supports online live broadcasting featuring both high concurrency and low cost, but also enables video chats between VJs and viewers via joint broadcasting.



Tencent Cloud provides "LVB+Joint Broadcasting" by using **LiveRoom**, a component consisting of Client and Server (both open source). For more information on how to interface with it, please see DOC. This document displays the API list for Client:

©2013-2018 Tencent Cloud. All rights reserved.

Download the SDK package on Tencent Cloud's official website. LiveRoom related header files and source code files are included in SDK\Rooms\LiveRoom.

LiveRoom

Name	Description
instance()	Gets a single instance of LiveRoom and calls the API of LiveRoom through the single instance.
setCallback(ILiveRoomCallback callback)	Sets the callback proxy for LiveRoom callback, and listens to the internal status of LiveRoom and the execution results of the API.
login(string serverDomain, LiveAuthData authData, ILoginLiveCallback callback)	Logs in to the business server RoomService before you can normally use other APIs and the IM feature.
logout()	Logs out of the business server RoomService. Make sure to call "logout" after "leaveRoom" is called, otherwise the call of leaveRoom will fail.
getRoomList(int index, int cnt, IGetLiveRoomListCallback callback)	Gets the room list. If there are many rooms, you can get the list in pages.
getAudienceList(string roomID)	Gets the list of viewers and returns only the last 30 viewers who entered the room.
createRoom(string roomID, string roomInfo)	Creates a room. The new room will be added to the room list in the backend, and then VJs can start pushing.
enterRoom(string roomID, IntPtr rendHwnd, Rectangle rect)	Enters a room.
leaveRoom()	Leaves the room. If the primary VJ leaves, the room will be terminated by the backend; if secondary VJs or viewers leave, the remaining participants can continue watching.
sendRoomTextMsg(string msg)	Sends plain text messages, such as sending on-screen comments under LVB scenarios.
sendRoomCustomMsg(string cmd, string msg)	Sends custom messages, such as giving a "Like" or flower under LVB scenarios.

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
startLocalPreview(IntPtr rendHwnd, Rectangle rect)	Enables the default camera preview
updateLocalPreview(IntPtr rendHwnd, Rectangle rect)	Resets the camera preview area. When the size of the specified local HWND window changes, the video rendering area can be rescaled via this function.
stopLocalPreview()	Disables camera preview
startScreenPreview(IntPtr rendHwnd, IntPtr captureHwnd, Rectangle renderRect, Rectangle captureRect)	Enables screen sharing
stopScreenPreview()	Disables screen sharing
addRemoteView(IntPtr rendHwnd, Rectangle rect, string userID)	Plays videos of other VJs in the room.
updateRemotePreview(IntPtr rendHwnd, Rectangle rect, string userID)	Resets the video preview area of the specified userID. When the size of the specified local HWND window changes, the video rendering area can be rescaled via this function.
removeRemoteView(string userID)	Stops playing videos of other VJs.
setMute(bool mute)	Enables Mute
setVideoQuality(LiveVideoQuality quality, LiveAspectRatio ratio)	Sets the preset options for image quality
setBeautyStyle(LiveBeautyStyle beautyStyle, int beautyLevel, int whitenessLevel)	Sets beauty filter and whitening effects
requestJoinPusher()	Joint broadcasting request from viewer
acceptJoinPusher(string userID)	The primary VJ accepts the joint broadcasting request and informs the request initiator
rejectJoinPusher(string userID, string reason)	The primary VJ rejects the joint broadcasting request and informs the request initiator
kickoutSubPusher(string userID)	The primary VJ kicks out a secondary VJ

©2013-2018 Tencent Cloud. All rights reserved.

ILoginLiveCallback

Name	Description
onLogin(LiveResult res, LiveAuthData authData)	Callback of the login result

IGetLiveRoomListCallback

Name	Description
onGetRoomList(LiveResult res, List rooms)	Callback of obtaining a room list

ILiveRoomCallback

Name	Description
onCreateRoom(LiveResult res, string roomID)	Callback of creating a room
onEnterRoom(LiveResult res)	Callback of entering a room
onUpdateRoomData(LiveResult res, LiveRoomData roomData)	Callback of room information change
void onGetAudienceList(LiveResult res, List audiences)	Callback of viewer list query
onPusherJoin(LiveMemberData member)	Callback of joint broadcasting viewer entering the room
void onPusherQuit(LiveMemberData member)	Callback of joint broadcasting viewers exiting the room
onRoomClosed(string roomID)	Callback of room dissolution
onRecvRoomTextMsg(string roomID, string userID, string userName, string userAvatar, string msg)	Receives plain text messages
onRecvRoomCustomMsg(string roomID, string userID, string userName, string userAvatar, string cmd, string message)	Receives custom messages
onError(LiveResult res, string userID)	Notification of internal LiveRoom error

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
onRecvJoinPusherRequest(string roomID, string userID, string userName, string userAvatar)	Receives a joint broadcasting request
onRecvAcceptJoinPusher(string roomID, string msg)	Receives the notification of accepting a joint broadcasting request
onRecvRejectJoinPusher(string roomID, string msg)	Receives the notification of rejecting a joint broadcasting request
onRecvKickoutSubPusher(string roomID)	Receives the notification of the primary VJ kicking out a secondary VJ

Details of LiveRoom Object APIs

1. instance

- Definition: static LiveRoom instance()
- Description: Gets a single instance of LiveRoom and calls the API of LiveRoom through the single instance
- Example:

```
LiveRoom m_liveRoom = null;
m_liveRoom = LiveRoom.instance();
```

2. setCallback

- Definition: void setCallback(ILiveRoomCallback callback)
- Description: Sets the callback proxy for LiveRoom callback, and listens to the internal status of LiveRoom and the execution results of the API
- Parameter:

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
callback	ILiveRoomCallback	Proxy API of ILiveRoomCallback type

- Example:

```
public class MainDialog : ILiveRoomCallback
{
MainDialog()
{
m_liveRoom.setCallback(this); //Set callback
}

void ILiveRoomCallback.onCreateRoom(LiveResult res, string roomID)
{

}

void ILiveRoomCallback.onEnterRoom(LiveResult res)
{

}

void ILiveRoomCallback.onUpdateRoomData(LiveResult res, LiveRoomData roomData)
{

}

...
}
```

3. login

- Definition: void login(string serverDomain, LiveAuthData authData, ILoginLiveCallback callback)
- Description: Logs in to the business server RoomService before you can normally use other APIs and the IM feature
- Parameters:

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
serverDomain	string	URL of RoomService. In consideration of security, it is recommended to access the https encrypted link. For more information, please see DOC.
authData	LiveAuthData	Login information provided by RoomService, including IM related configuration fields. The token field can be obtained after a success login. For more information, please see DOC.
callback	ILoginLiveCallback	Proxy API of ILoginLiveCallback type, used to call back the login result

- Example:

```
string serverDomain = "https://roomtest.qcloud.com/weapp/live_room";

m_authData.sdkAppID = sdkAppID;
m_authData.accountType = accountType;
m_authData.userID = userID;
m_authData.userSig = userSig;
m_authData.userName = userName;
m_authData.userAvatar = userAvatar;
m_authData.token = token;

// This class implements the ILoginCallback API to obtain login results
m_liveRoom.login(serverDomain, m_authData, this);
```

4. logout

- Definition: void logout();
- Description: Logs out of the business server RoomService. Make sure to call "logout" after "leaveRoom" is called, otherwise the call of leaveRoom will fail.
- Example:

```
m_liveRoom.logout();
```

5. getRoomList

©2013-2018 Tencent Cloud. All rights reserved.

- Definition: void getRoomList(int index, int cnt, IGetLiveRoomListCallback* callback)
- Description: Gets the room list. If there are many rooms, you can get the list in pages.
- Parameters:

Parameter	Type	Description
index	int	Gets the list in pages. The default value can be 0, and for the subsequent retrieval, the value can be set as the starting room index (for example, set index=0 and cnt=5 for the first retrieval, and set index=5 for the second page).
count	int	The maximum number of rooms returned per call; 0 indicates all rooms that meet the conditions
callback	IGetLiveRoomListCallback	Proxy API of IGetLiveRoomListCallback type, used to call back the query result

- Example:

```
// Start from index=0. A maximum of 20 rooms can be queried. This class implements the IGetRoomListCallback API to obtain query results
m_liveRoom.getRoomList(0, 20, this);
```

6. getAudienceList

- Definition: void getAudienceList(string roomID)
- Description: Gets the list of viewers and returns only the last 30 viewers who entered the room
- Parameter:

Parameter	Type	Description
roomID	string	roomID, which is obtained in the room list returned by the getRoomList API

- Example:

```
m_liveRoom.getRoomList(roomID);
```

©2013-2018 Tencent Cloud. All rights reserved.

7. createRoom

- Definition: void createRoom(string roomID, string roomInfo)
- Description: Creates a room, and the new room will be added to the room list in the backend, and then VJs can start pushing.
- Parameters:

Parameter	Type	Description
roomID	string	Room ID. If an empty string is specified, a roomID is assigned to you. Otherwise, the specified roomID is used as the ID of this room.
roomInfo	string	Custom data. This field is included in the room information. It is recommended that you define roomInfo in json format, which is highly extensible.

- Example:

```
// A roomID is assigned to you
m_liveRoom.createRoom("", "{\"roomName\": \"My first room\"}");
```

8. enterRoom

- Definition: void enterRoom(string roomID, IntPtr rendHwnd, Rectangle rect)
- Description: Enters a room.
- Parameters:

Parameter	Type	Description
roomID	string	roomID - the ID of the room to be entered, which is obtained in the room list returned by the getRoomList API
rendHwnd	IntPtr	HWND that identifies the preview window.
rect	Rectangle	Specifies the rendering area of the video image on the window identified by HWND.

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
// Render in the entire form window
Rectangle rect = new Rectangle(0, 0, form.Width, form.Height);
m_liveRoom.enterRoom(roomID, form.Handle, rect);
```

9. leaveRoom

- Definition: void leaveRoom()
- Description: Leaves the room. If the primary VJ leaves, the room will be terminated by the backend; if secondary VJs or viewers leave, the remaining participants can continue watching.
- Example:

```
m_liveRoom.leaveRoom();
```

10. sendRoomTextMsg

- Definition: void sendRoomTextMsg(string msg)
- Description: Sends plain text messages, such as sending on-screen comments under LVB scenarios.
- Parameter:

Parameter	Type	Description
msg	string	Text messages

- Example:

```
m_liveRoom.sendRoomTextMsg("Hello LiveRoom");
```

11. sendRoomCustomMsg

- Definition: void sendRoomCustomMsg(string cmd, string msg)

©2013-2018 Tencent Cloud. All rights reserved.

- Description: Sends custom messages, such as giving a "Like" or flower under LVB scenarios
- Parameters:

Parameter	Type	Description
cmd	string	Custom CMD, jointly determined by the sender and receiver
msg	string	Custom messages

- Example:

```
// Assume that the CMD determined by the sender and receiver includes Smile and Anger
m_liveRoom.sendRoomCustomMsg("Smile", "I am hungry");
```

12. startLocalPreview

- Definition: void startLocalPreview(IntPtr rendHwnd, Rectangle rect)
- Description: Enables the default camera preview
- Parameters:

Parameter	Type	Description
rendHwnd	IntPtr	HWND that identifies the preview screen. OpenGL is used to draw images on HWND. If rendHwnd = NULL, there is no need to preview the video.
rect	Rectangle	Specifies the rendering area of the video image on the window identified by HWND.

- Example:

```
// Render in the entire form window
Rectangle rect = new Rectangle(0, 0, form.Width, form.Height);
m_liveRoom.startLocalPreview(form.Handle, rect);
```

13. updateLocalPreview

©2013-2018 Tencent Cloud. All rights reserved.

- Definition: void updateLocalPreview(IntPtr rendHwnd, Rectangle rect)
- Description: Resets the preview area for camera. When the size of the window identified by the specified HWND changes, you can resize the video rendering area using this function.
- Parameters:

Parameter	Type	Description
rendHwnd	IntPtr	HWND that identifies the preview screen. OpenGL is used to draw images on HWND. If rendHwnd = NULL, there is no need to preview the video.
rect	Rectangle	Specifies the rendering area of the video image on the window identified by HWND.

- Example:

```
// Render in the entire form window
Rectangle rect = new Rectangle(0, 0, form.Width, form.Height);
m_liveRoom.updateLocalPreview(form.Handle, rect);
```

14. stopLocalPreview

- Definition: void stopLocalPreview()
- Description: Disables camera preview
- Example:

```
m_liveRoom.stopLocalPreview();
```

15. startScreenPreview

- Definition: bool startScreenPreview(IntPtr rendHwnd, IntPtr captureHwnd, Rectangle renderRect, Rectangle captureRect)
- Description: Enables screen sharing
- Parameters:

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
rendHwnd	IntPtr	HWND that identifies the preview screen. OpenGL is used to draw images on HWND. If rendHwnd = NULL, there is no need to preview the video.
captureHwnd	IntPtr	Specifies the capture window. If it is NULL, captureRect does not work and the entire screen is captured; if it is not NULL, the current window is captured.
renderRect	Rectangle	Specifies the rendering area of the video screen on rendHwnd
captureRect	Rectangle	Specifies the customer area of the capture window

- Returned value: Success or Failed
- Example:

```
// Render in the entire form window
Rectangle renderRect = new Rectangle(0, 0, rendForm.Width, rendForm.Height);

// Capture the entire window
Rectangle captureRect = new Rectangle(0, 0, captureForm.Width, captureForm.Height);

m_liveRoom.startScreenPreview(rendForm.Handle, captureForm.Handle, renderRect, captureRect);
```

16. stopScreenPreview

- Definition: void stopScreenPreview()
- Description: Disables screen sharing
- Example:

```
m_liveRoom.stopScreenPreview();
```

17. addRemoteView

- Definition: void addRemoteView(IntPtr rendHwnd, Rectangle rect, string userID)

©2013-2018 Tencent Cloud. All rights reserved.

- Description: Plays videos of other VJs in the room
- Parameters:

Parameter	Type	Description
rendHwnd	IntPtr	HWND that identifies the preview window.
rect	Rectangle	Specifies the rendering area of the video image on the window identified by HWND.
userID	string	User ID

- Example:

```
// Render in the entire form window
Rectangle renderRect = new Rectangle(0, 0, form.Width, form.Height);

// Enable playback of users' audios/videos
m_liveRoom.addRemoteView(form.Handle,rect, userID);
```

18. updateRemotePreview

- Definition: void updateRemotePreview(IntPtr rendHwnd, Rectangle rect, string userID)
- Description: Resets the video preview area of the specified userID. When the size of the window identified by the specified local HWND changes, you can resize the video rendering area using this function.
- Parameters:

Parameter	Type	Description
rendHwnd	IntPtr	HWND that identifies the preview window.
rect	Rectangle	Specifies the rendering area of the video image on the window identified by HWND.
userID	string	User ID

- Example:

©2013-2018 Tencent Cloud. All rights reserved.

```
// Render in the entire form window
Rectangle renderRect = new Rectangle(0, 0, form.Width, form.Height);

// Reset the video preview area of the specified userID
m_liveRoom.updateRemotePreview(form.Handle, rect, userID);
```

18. removeRemoteView

- Definition: void removeRemoteView(string userID)
- Description: Stops playing videos of other VJs
- Parameters:

Parameter	Type	Description
userID	string	User ID

- Example:

```
m_liveRoom.removeRemoteView(userID);
```

19. setMute

- Definition: void setMute(bool mute)
- Description: Enables Mute
- Parameter:

Parameter	Type	Description
mute	bool	Indicates whether to enable Mute

- Example:

```
m_liveRoom.setMute(true); // Set to mute
```

©2013-2018 Tencent Cloud. All rights reserved.

20. setVideoQuality

- Definition: void setVideoQuality(LiveVideoQuality quality, LiveAspectRatio ratio)
- Description: Sets the pushed video quality
- Parameters:

Parameter	Type	Description
quality	LiveVideoQuality	Image quality. See LiveVideoQuality's enumerated values defined in LiveRoomUtil.h
ratio	LiveAspectRatio	Aspect ratio. See LiveAspectRatio's enumerated values defined in LiveRoomUtil.h

- Example:

```
// Set the image quality to HD and the aspect ratio to 4:3
m_liveRoom.setVideoQuality(LIVEROOM_VIDEO_QUALITY_HIGH_DEFINITION, LIVEROOM_ASPECT_RATIO_4_3);
```

21. setBeautyStyle

- Definition: void setBeautyStyle(LiveBeautyStyle beautyStyle, int beautyLevel, int whitenessLevel)
- Description: Sets beauty filter and whitening effects
- Parameters:

Parameter	Type	Description
beautyStyle	TXEBeautyStyle	See LiveBeautyStyle's enumerated values defined in LiveRoomUtil.h
beautyLevel	int	Beauty filter level: 1-9. 0 indicates disabling beauty filter. A greater value means a bigger effect.
whitenessLevel	int	Whiteness level: 1-9. 0 indicates disabling whitening. A greater value means a bigger effect.

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
// Natural; beauty filter level: 5; whitening level: 5
m_liveRoom.setBeautyStyle(TXE_BEAUTY_STYLE_NATURE, 5, 5);
```

22. requestJoinPusher

- Definition: void requestJoinPusher()
- Description: Joint broadcasting request from viewer
- Example:

```
m_liveRoom.requestJoinPusher();
```

23. acceptJoinPusher

- Definition: void acceptJoinPusher(string userID)
- Description: The primary VJ accepts the joint broadcasting request and informs the request initiator.
- Parameter:

Parameter	Type	Description
userID	string	User ID

- Example:

```
m_liveRoom.acceptJoinPusher(userID);
```

24. rejectJoinPusher

- Definition: void rejectJoinPusher(string userID, string reason)
- Description: The primary VJ rejects the joint broadcasting request and informs the request initiator.

©2013-2018 Tencent Cloud. All rights reserved.

- Parameters:

Parameter	Type	Description
userID	string	User ID
reason	string	Reason for rejection

- Example:

```
m_liveRoom.acceptJoinPusher(userID, reason);
```

25. kickoutSubPusher

- Definition: void kickoutSubPusher(string userID)
- Description: The primary VJ kicks out a secondary VJ.
- Parameter:

Parameter	Type	Description
userID	string	User ID

- Example:

```
m_liveRoom.kickoutSubPusher(userID);
```

Details of ILoginLiveCallback APIs

1. onLogin

- Definition: void onLogin(LiveResult res, LiveAuthData authData)
- Description: Callback of login result
- Parameters:

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
res	LiveResult	Execution result. See LiveResult class defined in LiveRoomUtil.h
authData	LiveAuthData	Login information provided by RoomService, including IM related configuration fields. The token field can be obtained after a success login.

- Example:

```
class LoginDialog : ILoginLiveCallback
{
void ILoginLiveCallback.onLogin(LiveResult res, LiveAuthData authData)
{
if (LIVEROOM_SUCCESS != res.ec)
{
// Login failed
}
else
{
// Login succeeded, and authData information is saved
}
}
}

// For more information on how to use callback, please see LiveRoom::login API
...
```

Details of IGetLiveRoomListCallback APIs

1. onGetRoomList

- Definition: void onGetRoomList(LiveResult res, List rooms)
- Description: Callback of obtaining a room list
- Parameters:

Parameter	Type	Description

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
res	LiveResult	Execution result. See LiveResult class defined in LiveRoomUtil.h
rooms	List	List of room information

- Example:

```
class RoomListDialog : IGetLiveRoomListCallback
{
void IGetRoomListCallback.onGetRoomList(LiveResult res, List<LiveRoomData> rooms)
{
if (LIVEROOM_SUCCESS != res.ec)
{
// Query failed
}
else
{
// Query succeeded, process rooms data
}
}
}

// For more information on how to use callback, please see LiveRoom::getRoomList API
...
```

Details of ILiveRoomCallback APIs

For more information on how to set ILiveRoomCallback, please see the API LiveRoom::setCallback.

1. onCreateRoom

- Definition: void onCreateRoom(LiveResult res, string roomID)
- Description: Callback of creating a room
- Parameters:

Parameter	Type	Description

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
res	LiveResult	Execution result. See LiveResult class defined in LiveRoomUtil.h
roomID	string	Room ID

- Example:

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onCreateRoom(LiveResult res, string roomID)
{
if (LIVEROOM_SUCCESS != res.ec)
{
// If the room fails to be created, a prompt pops up.
}
else
{
// If the room is created successfully, the viewers can watch the LVB.
// Process onUpdateRoomData callback immediately
}
}

...
}
```

2. onEnterRoom

- Definition: void onEnterRoom(LiveResult res)
- Description: Callback of entering a room
- Parameter:

Parameter	Type	Description
res	LiveResult	Execution result. See LiveResult class defined in LiveRoomUtil.h

- Example:

©2013-2018 Tencent Cloud. All rights reserved.

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onEnterRoom(LiveResult res)
{
if (LIVEROOM_SUCCESS != res.ec)
{
// If the room fails to be entered, a prompt pops up.
}
else
{
// If viewers enter the room successfully, process onUpdateRoomData callback immediately
}
}

...
}
```

3. onUpdateRoomData

- Definition: void onUpdateRoomData(LiveResult res, LiveRoomData roomData)
- Description: Callback of room information change
- Parameters:

Parameter	Type	Description
res	LiveResult	Execution result. See LiveResult class defined in LiveRoomUtil.h
roomData	LiveRoomData	Room information. See LiveRoomData class defined in LiveRoomUtil.h.

- Example:

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onUpdateRoomData(LiveResult res, LiveRoomData roomData)
{
if (LIVEROOM_SUCCESS != res.ec)
{
// If the update fails, a prompt pops up.
```

©2013-2018 Tencent Cloud. All rights reserved.

```
}
else
{
// If the update succeeds, the appropriate processing is performed based on whether the API is trig
gered by createRoom or enterRoom.
}
}

...
}
```

4. onGetAudienceList

- Definition: void onGetAudienceList(LiveResult res, List audiences)
- Description: Callback of viewer list query
- Parameters:

Parameter	Type	Description
res	LiveResult	Execution result. See LiveResult class defined in LiveRoomUtil.h
audiences	List	Viewer information list

- Example:

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onGetAudienceList(LiveResult res, List<LiveAudienceData> audiences)
{
if (LIVEROOM_SUCCESS != res.ec)
{
// If the query fails, a prompt pops up.
}
else
{
// If the query succeeds, the list is updated to UI.
}
}

...
}
```

©2013-2018 Tencent Cloud. All rights reserved.

5. onPusherJoin

- Definition: void onPusherJoin(LiveMemberData member)
- Description: Callback of joint broadcasting viewer entering the room
- Parameter:

Parameter	Type	Description
member	LiveMemberData	Member information. See LiveMemberData class defined in LiveRoomUtil.h.

- Example:

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onPusherJoin(LiveMemberData member)
{
// Usually, the video and audio of the joint broadcasting viewer are turned on after he or she requ
ests joint broadcasting.
m_liveRoom.addRemoteView(rendHwnd, rect, member.userID);
...
}

...
}
```

6. onPusherQuit

- Definition: void onPusherQuit(LiveMemberData member)
- Description: Callback of the joint broadcasting viewer exiting the room
- Parameter:

Parameter	Type	Description
member	LiveMemberData	Member information. See LiveMemberData class defined in LiveRoomUtil.h.

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onPusherQuit(LiveMemberData member)
{
// Usually, the video and audio of the joint broadcasting viewer are turned off after he or she stops joint broadcasting.
m_liveRoom.removeRemoteView(member.userID);
...
}

...
}
```

7. onRoomClosed

- Definition: void onRoomClosed(string roomID)
- Description: Callback of room dissolution
- Parameter:

Parameter	Type	Description
roomID	string	Room ID

- Example:

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onRoomClosed(string roomID)
{
// It prompts that the room is dismissed. The opened audio/video is closed.
...
}

...
}
```

©2013-2018 Tencent Cloud. All rights reserved.

8. onRecvRoomTextMsg

- Definition: void onRecvRoomTextMsg(string roomID, string userID, string userName, string userAvatar, string msg)
- Description: Receives plain text messages
- Parameters:

Parameter	Type	Description
roomID	string	Room ID
userID	string	Sender ID
userName	string	Sender nickname
userAvatar	string	Sender profile photo
message	string	Text message

- Example:

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onRecvRoomTextMsg(string roomID, string userID, string userName, string userAvatar, string msg)
{
// Text messages are displayed on IM panel
}

...
}
```

9. onRecvRoomCustomMsg

- Definition: void onRecvRoomCustomMsg(string roomID, string userID, string userName, string userAvatar, string cmd, string message)
- Description: Receives custom messages

©2013-2018 Tencent Cloud. All rights reserved.

- Parameters:

Parameter	Type	Description
roomID	string	Room ID
userID	string	Sender ID
userName	string	Sender nickname
userAvatar	string	Sender profile photo
cmd	string	Custom CMD
message	string	Custom message

- Example:

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onRecvRoomCustomMsg(string roomID, string userID, string userName, str
ing userAvatar, string cmd, string message)
{
// Parse and process custom messages such as Likes and gifts
}

...
}
```

10. onError

- Definition: void onError(LiveResult res, string userID)
- Description: Notification of internal LiveRoom error
- Parameters:

Parameter	Type	Description
res	LiveResult	Execution result. See LiveResult class defined in LiveRoomUtil.h
userID	string	User ID

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onError(LiveResult res, string userID)
{
// An error message pops up. Decide whether to close LVB based on severity of errors
}

...
}
```

11. onRecvJoinPusherRequest

- Definition: void onRecvJoinPusherRequest(string roomID, string userID, string userName, string userAvatar)
- Description: Receives a joint broadcasting request
- Parameters:

Parameter	Type	Description
roomID	string	Room ID
userID	string	Sender ID
userName	string	Sender nickname
userAvatar	string	Sender profile photo

- Example:

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onRecvJoinPusherRequest(string roomID, string userID, string userName, string userAvatar)
{
// The primary VJ receives a joint broadcasting request from a viewer.
// Verify the validity of roomID
// UI shows who initiates the request.
// The primary VJ selects "acceptJoinPusher" to accept or "rejectJoinPusher" to reject the request.
```

©2013-2018 Tencent Cloud. All rights reserved.

```
}

...
}
```

12. onRecvAcceptJoinPusher

- Definition: void onRecvAcceptJoinPusher(string roomID, string msg)
- Description: Receives the notification of accepting a joint broadcasting request.
- Parameters:

Parameter	Type	Description
roomID	string	Room ID
msg	string	Message

- Example:

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onRecvAcceptJoinPusher(string roomID, string msg)
{
// The viewer receives the notification of the primary VJ accepting the joint broadcasting request.
// Verify the validity of roomID
// Process msg
}

...
}
```

13. onRecvRejectJoinPusher

- Definition: void onRecvRejectJoinPusher(string roomID, string msg)
- Description: Receives the notification of rejecting a joint broadcasting request

©2013-2018 Tencent Cloud. All rights reserved.

- Parameters:

Parameter	Type	Description
roomID	string	Room ID
msg	string	Message

- Example:

```
class MainDialog : ILiveRoomCallback
{
void ILiveRoomCallback.onRecvRejectJoinPusher(string roomID, string msg)
{
// The viewer receives the notification of the primary VJ rejecting the joint broadcasting request.
// Verify the validity of roomID
// Process msg
}

...
}
```

14. onRecvKickoutSubPusher

- Definition: void onRecvKickoutSubPusher(string roomID)
- Description: Receives the notification of the primary VJ kicking out a secondary VJ.
- Parameter:

Parameter	Type	Description
roomID	string	Room ID

- Example:

```
class MainDialog : ILiveRoomCallback
{
```

©2013-2018 Tencent Cloud. All rights reserved.

```
void ILiveRoomCallback.onRecvRejectJoinPusher(string roomID, string msg)
{
// The primary VJ kicks out a viewer invited for joint broadcasting
// Verify the validity of roomID
// Process msg
}

...
}
```

©2013-2018 Tencent Cloud. All rights reserved.

Real time video comminucation

Last updated：2018-07-10 11:25:15

Real-time Audio/Video (RTCRoom) APIs (CSharp)

RTCRoom

Name	Description
instance()	Gets a single instance for RTCRoom and calls RTCRoom APIs via the single instance
setCallback(IRTCRoomCallback callback)	Sets the callback proxy for RTCRoom callback, and listens to the internal status of RTCRoom and the execution results of the API
login(string serverDomain, RTCAuthData authData, ILoginRTCCallback callback)	Logs in to the business server RoomService before you can normally use other APIs and the IM feature.
logout()	Logs out of the business server RoomService. Make sure to call "logout" after "leaveRoom" is called, otherwise the call of leaveRoom will fail.
getRoomList(int index, int cnt, IGetRTCRoomListCallback callback)	Gets the room list. If there are many rooms, you can get the list in pages.
createRoom(string roomID, string roomInfo)	Creates a room, and the new room will be added to the room list in the backend, and then the meeting initiator can start pushing.
enterRoom(string roomID)	Enters a room
leaveRoom()	Leaves the room. If the meeting initiator leaves, the room will be terminated by the backend; if other meeting participants leave, the remaining participants can continue chatting.

©2013-2018 Tencent Cloud. All rights reserved.

Name	Description
sendRoomTextMsg(string msg)	Sends plain text messages, such as sending on-screen comments in the room.
sendRoomCustomMsg(string cmd, string msg)	Sends custom messages, such as giving a "Like", or flower in the room.
startLocalPreview(IntPtr rendHwnd, Rectangle rect)	Enables the default camera preview
updateLocalPreview(IntPtr rendHwnd, Rectangle rect)	Resets the camera preview area. When the size of the specified local HWND window changes, the video rendering area can be rescaled via this function.
stopLocalPreview()	Disables camera preview
startScreenPreview(IntPtr rendHwnd, IntPtr captureHwnd, Rectangle renderRect, Rectangle captureRect)	Enables screen sharing
stopScreenPreview()	Disables screen sharing
addRemoteView(IntPtr rendHwnd, Rectangle rect, string userID)	Plays videos of other meeting participants in the room
updateRemotePreview(IntPtr rendHwnd, Rectangle rect, string userID)	Resets the video preview area of the specified userID. When the size of the specified local HWND window changes, the video rendering area can be rescaled via this function.
removeRemoteView(string userID)	Stops playing videos of other meeting participants
setMute(bool mute)	Enables Mute
setBeautyStyle(RTCBeautyStyle beautyStyle, int beautyLevel, int whitenessLevel)	Sets beauty filter and whitening effects

ILoginRTCCallback

Name	Description
onLogin(RTCResult res, RTCAuthData authData)	Callback of the login result

©2013-2018 Tencent Cloud. All rights reserved.

IGetRTCRoomListCallback

Name	Description
onGetRoomList(RTCResult res, List rooms)	Callback of obtaining a room list

IRTCRoomCallback

Name	Description
onCreateRoom(RTCResult res, string roomID)	Callback of creating a room
onEnterRoom(RTCResult res)	Callback of entering a room
onUpdateRoomData(RTCResult res, RTCRoomData roomData)	Callback of room information change
onPusherJoin(RTCMemberData member)	Callback of members entering a room
onPusherQuit(RTCMemberData member)	Callback of members exiting a room
onRoomClosed(string roomID)	Callback of room dissolution
onRecvRoomTextMsg(string roomID, string userID, string userName, string userAvatar, string msg)	Receives plain text messages
onRecvRoomCustomMsg(string roomID, string userID, string userName, string userAvatar, string cmd, string message)	Receives custom messages
onError(RTCResult res, string userID	Notification of internal RTCRoom error

Details of RTCRoom object APIs

1. instance

©2013-2018 Tencent Cloud. All rights reserved.

- Definition: static RTCRoom instance()
- Description: Gets a single instance for RTCRoom and calls RTCRoom APIs via the single instance
- Example:

```
RTCRoom mRTCRoom = RTCRoom.instance();
```

2. setCallback

- Definition: void setCallback(IRTCRoomCallback callback)
- Description: Sets the callback proxy for RTCRoom callback, and listens to the internal status of RTCRoom and the execution results of the API
- Parameter:

Parameter	Type	Description
callback	IRTCRoomCallback	API of IRTCRoomCallback type

- Example:

```
class MainDialog : IRTCRoomCallback
{
void IRTCRoomCallback.onCreateRoom(RTCResult res, string roomID){}
void IRTCRoomCallback.onEnterRoom(RTCResult res){}
void IRTCRoomCallback.onUpdateRoomData(RTCResult res, RTCRoomData roomData){}
...
};

MainDialog::MainDialog()
{
mRTCRoom.setCallback(this); //Set callback
}

...
```

©2013-2018 Tencent Cloud. All rights reserved.

3. login

- Definition: void login(string serverDomain, RTCAuthData authData, ILoginRTCCallback callback)
- Description: Logs in to the business server RoomService before you can normally use other APIs and the IM feature
- Parameters:

Parameter	Type	Description
serverDomain	string	URL of RoomService. In consideration of security, it is recommended to access the https encrypted link. For more information, please see DOC.
authData	RTCAuthData	Login information provided by RoomService, including IM related configuration fields. The token field can be obtained after a success login. For more information, please see DOC.
callback	ILoginRTCCallback	API of ILoginRTCCallback type, used to call back the login result

- Example:

```
string serverDomain = "https://roomtest.qcloud.com/weapp/double_room";

m_authData.sdkAppID = sdkAppID;
m_authData.accountType = accountType;
m_authData.userID = userID;
m_authData.userSig = userSig;
m_authData.userName = userName;
m_authData.userAvatar = userAvatar;
m_authData.token = token;

// This class implements the ILoginCallback API to obtain login results
mRTCRoom.login(serverDomain, m_authData, this);
```

4. logout

- Definition: void logout()

©2013-2018 Tencent Cloud. All rights reserved.

- Description: Logs out of the business server RoomService. Make sure to call "logout" after "leaveRoom" is called, otherwise the call of leaveRoom will fail.

- Example:

```
mRTCRoom.logout();
```

5. getRoomList

- Definition: void getRoomList(int index, int count, IGetRTCRoomListCallback callback)

- Description: Gets the room list. If there are many rooms, you can get the list in pages.

- Parameters:

Parameter	Type	Description
index	int	Gets the list in pages. The default value can be 0, and for the subsequent retrieval, the value can be set as the starting room index (for example, set index=0 and cnt=5 for the first retrieval, and set index=5 for the second page).
count	int	The maximum number of rooms returned per call; 0 indicates all rooms that meet the conditions
callback	IGetRTCRoomListCallback	API of IGetRTCRoomListCallback type, used to call back the query result

- Example:

```
// Start from index=0. A maximum of 20 rooms can be queried. This class implements the IGetRoomListCallback API to obtain query results
mRTCRoom.getRoomList(0, 20, this);
```

6. createRoom

- Definition: void createRoom(string roomID, string roomInfo)

©2013-2018 Tencent Cloud. All rights reserved.

- Description: Creates a room, and the new room will be added to the room list in the backend, and then the meeting initiator can start pushing.

- Parameters:

Parameter	Type	Description
roomID	string	Room ID. If an empty string is specified, a roomID is assigned to you. Otherwise, the specified roomID is used as the ID of this room.
roomInfo	string	Custom data. This field is included in the room information. It is recommended that you define roomInfo in json format, which is highly extensible.

- Example:

```
// A roomID is assigned to you
mRTCRoom.createRoom("", "{\"roomName\": \"My first room\"}");
```

7. enterRoom

- Definition: void enterRoom(string roomID)

- Description: Enters a room.

- Parameter:

Parameter	Type	Description
roomID	string	roomID - the ID of the room to be entered, which is obtained in the room list returned by the getRoomList API

- Example:

```
mRTCRoom.enterRoom(roomID);
```

8. leaveRoom

- Definition: void leaveRoom()

©2013-2018 Tencent Cloud. All rights reserved.

- Description: Leaves the room. If the meeting initiator leaves, the room will be terminated by the backend; if other meeting participants leave, the remaining participants can continue chatting.
- Example:

```
mRTCRoom.leaveRoom();
```

9. sendRoomTextMsg

- Definition: void sendRoomTextMsg(string msg)
- Description: Sends plain text messages
- Parameters:

Parameter	Type	Description
msg	string	Text messages

- Example:

```
mRTCRoom.sendRoomTextMsg("Hello RTCRoom");
```

10. sendRoomCustomMsg

- Definition: void sendRoomCustomMsg(string cmd, string msg)
- Description: Sends custom messages
- Parameters:

Parameter	Type	Description
cmd	string	Custom CMD, jointly determined by the sender and receiver
msg	string	Custom messages

- Example:

©2013-2018 Tencent Cloud. All rights reserved.

```
// Assume that the CMD determined by the sender and receiver includes Smile and Anger
mRTCRoom.sendRoomCustomMsg("Anger", "I am hungry");
```

11. startLocalPreview

- Definition: void startLocalPreview(IntPtr rendHwnd, Rectangle rect)
- Description: Enables the default camera preview
- Parameters:

Parameter	Type	Description
rendHwnd	IntPtr	HWND that identifies the preview screen. OpenGL is used to draw images on HWND. If rendHwnd = NULL, there is no need to preview the video.
rect	Rectangle	Specifies the rendering area of the video image on the window identified by HWND.

- Example:

```
mRTCRoom.startLocalPreview(panel_main.Handle, new Rectangle(0, 0, panel_main.Width, panel_main.Height));
```

12. updateLocalPreview

- Definition: void updateLocalPreview(IntPtr rendHwnd, Rectangle rect)
- Description: Resets the preview area for camera. When the size of the window identified by the specified HWND changes, you can resize the video rendering area using this function.
- Parameters:

Parameter	Type	Description
rendHwnd	IntPtr	HWND that identifies the preview screen. OpenGL is used to draw images on HWND. If rendHwnd = NULL, there is no need to preview the video.

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
rect	Rectangle	Specifies the rendering area of the video image on the window identified by HWND.

- Example:

```
mRTCRoom.updateLocalPreview(panel_main.Handle, new Rectangle(0, 0, panel_main.Width, panel_main.Height));
```

13. stopLocalPreview

- Definition: void stopLocalPreview()
- Description: Disables camera preview
- Example:

```
mRTCRoom.stopLocalPreview();
```

14. startScreenPreview

- Definition: bool startScreenPreview(IntPtr rendHwnd, IntPtr captureHwnd, Rectangle renderRect, Rectangle captureRect)
- Description: Enables screen sharing
- Parameters:

Parameter	Type	Description
rendHwnd	IntPtr	HWND that identifies the preview screen. OpenGL is used to draw images on HWND. If rendHwnd = NULL, there is no need to preview the video.
captureHwnd	IntPtr	Specifies the capture window. If it is NULL, captureRect does not work and the entire screen is captured; if it is not NULL, the current window is captured.

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
renderRect	Rectangle	Specifies the rendering area of the video screen on rendHwnd
captureRect	Rectangle	Specifies the customer area of the capture window

- Returned value: Success or Failed
- Example:

```
mRTCRoom.startScreenPreview(panel_main.Handle, null, new Rectangle(0, 0, panel_main.Width, panel_main.Height), null);
```

15. stopScreenPreview

- Definition: void stopScreenPreview()
- Description: Disables screen sharing
- Example:

```
mRTCRoom.stopScreenPreview();
```

16. addRemoteView

- Definition: void addRemoteView(IntPtr rendHwnd, Rectangle rect, string userID)
- Description: Plays videos of other meeting participants in the room
- Parameters:

Parameter	Type	Description
rendHwnd	IntPtr	HWND that identifies the preview window.
rect	Rectangle	Specifies the rendering area of the video image on the window identified by HWND.
userID	string	User ID

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
// Enable playback of users' audios/videos
mRTCRoom.addRemoteView(panel_display.Handle, new Rectangle(0, 0, panel_display.Width, panel_display.Height), userID);
```

17. updateRemotePreview

- Definition: void updateRemotePreview(IntPtr rendHwnd, Rectangle rect, string userID)
- Description: Resets the video preview area of the specified userID. When the size of the window identified by the specified local HWND changes, you can resize the video rendering area using this function.
- Parameters:

Parameter	Type	Description
rendHwnd	IntPtr	HWND that identifies the preview window.
rect	Rectangle	Specifies the rendering area of the video image on the window identified by HWND.
userID	string	User ID

- Example:

```
// Update playback of users' audios/videos
mRTCRoom.updateRemotePreview(panel_display.Handle, new Rectangle(0, 0, panel_display.Width, panel_display.Height), userID);
```

18. removeRemoteView

- Definition: void removeRemoteView(string userID)
- Description: Stops playing videos of other meeting participants
- Parameter:

Parameter	Type	Description
userID	string	User ID

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
mRTCRoom.removeRemoteView(userID);
```

19. setMute

- Definition: void setMute(bool mute)
- Description: Enables Mute
- Parameter:

Parameter	Type	Description
mute	bool	Indicates whether to enable Mute

- Example:

```
mRTCRoom.setMute(true); // Set to mute
```

20. setBeautyStyle

- Definition: void setBeautyStyle(RTCBeautyStyle beautyStyle, int beautyLevel, int whitenessLevel)
- Description: Sets beauty filter and whitening effects
- Parameters:

Parameter	Type	Description
beautyStyle	RTCBeautyStyle	See RTCBeautyStyle's enumerated values defined in RTCRoomUtil.cs
beautyLevel	int	Beauty filter level: 1-9. 0 indicates disabling beauty filter. A greater value means a bigger effect.
whitenessLevel	int	Whiteness level: 1-9. 0 indicates disabling whitening. A greater value means a bigger effect.

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
// Natural; beauty filter level: 5; whitening level: 5
mRTCRoom.setBeautyStyle(RTCBeautyStyle.RTCROOM_BEAUTY_STYLE_NATURE, 5, 5);
```

Details of ILoginRTCCallback APIs

1. onLogin

- Definition: void onLogin(RTCResult res, RTCAuthData authData)
- Description: Callback of login result
- Parameters:

Parameter	Type	Description
res	RTCResult	Execution result. See RTCResult class defined in RTCRoomUtil.cs
authData	RTCAuthData	Login information provided by RoomService, including IM related configuration fields. The token field can be obtained after a success login.

- Example:

```
class LoginDialog : ILoginRTCCallback
{
void onLogin(RTCResult res, RTCAuthData authData)
{
if (RTCErrorCode.RTCROOM_SUCCESS != res.ec)
{
// Login failed
}
else
{
// Login succeeded, and authData information is saved
}
}
}
```

©2013-2018 Tencent Cloud. All rights reserved.

```
// For more information on how to use callback, please see the API RTCRoom.login.
...
```

Details of IGetRTCRoomListCallback APIs

1. onGetRoomList

- Definition: void onGetRoomList(RTCResult res, List rooms)
- Description: Callback of obtaining a room list
- Parameters:

Parameter	Type	Description
res	RTCResult	Execution result. See RTCResult class defined in RTCRoomUtil.cs
rooms	List	List of room information

- Example:

```
class RoomListDialog : IGetRTCRoomListCallback
{
void onGetRoomList(RTCResult res, List<RTCRoomData> rooms)
{
if (RTCErrorCode.RTCROOM_SUCCESS != res.ec)
{
// Query failed
}
else
{
// Query succeeded, process rooms data
}
}
}

// For more information on how to use callback, please see the API RTCRoom.getRoomList.
...
```

©2013-2018 Tencent Cloud. All rights reserved.

Details of IRTCRoomCallback APIs

For more information on how to set IRTCRoomCallback, please see the API RTCRoom.setCallback.

1. onCreateRoom

- Definition: void onCreateRoom(RTCResult res, string roomID)
- Description: Callback of creating a room
- Parameters:

Parameter	Type	Description
res	RTCResult	Execution result. See RTCResult class defined in RTCRoomUtil.cs
roomID	string	Room ID

- Example:

```
class MainDialog : IRTCRoomCallback
{
void onCreateRoom(RTCResult res, string roomID)
{
if (RTCErrorCode.RTCROOM_SUCCESS != res.ec)
{
// If the room fails to be created, a prompt pops up.
}
else
{
// If the room is created successfully, other audiences can participate in the meeting.
// Process onUpdateRoomData callback immediately
}
}

...
}
```

2. onEnterRoom

- Definition: void onEnterRoom(RTCResult res)

©2013-2018 Tencent Cloud. All rights reserved.

- Description: Callback of entering a room
- Parameter:

Parameter	Type	Description
res	RTCResult	Execution result. See RTCResult class defined in RTCRoomUtil.cs

- Example:

```
class MainDialog : IRTCRoomCallback
{
void onEnterRoom(RTCResult res)
{
if (RTCErrorCode.RTCROOM_SUCCESS != res.ec)
{
// If the room fails to be entered, a prompt pops up.
}
else
{
// If viewers enter the room successfully, process onUpdateRoomData callback immediately
}
}

...
}
```

3. onUpdateRoomData

- Definition: void onUpdateRoomData(RTCResult res, RTCRoomData roomData)
- Description: Callback of room information change
- Parameters:

Parameter	Type	Description
res	RTCResult	Execution result. See RTCResult class defined in RTCRoomUtil.cs
roomData	RTCRoomData	Room information. See RTCRoomData class defined in RTCRoomUtil.cs

©2013-2018 Tencent Cloud. All rights reserved.

- Example:

```
class MainDialog : IRTCRoomCallback
{
void onUpdateRoomData(RTCResult res, RTCRoomData roomData)
{
if (RTCErrorCode.RTCROOM_SUCCESS != res.ec)
{
// If the update fails, a prompt pops up.
}
else
{
// If the update succeeds, the appropriate processing is performed based on whether the API is trig
gered by createRoom or enterRoom.
}
}

...
}
```

4. onPusherJoin

- Definition: void onPusherJoin(RTCMemberData member)
- Description: Callback of members entering a room
- Parameter:

Parameter	Type	Description
member	RTCMemberData	Member information. See the RTCMemberData class defined in RTCRoomUtil.cs.

- Example:

```
class MainDialog : IRTCRoomCallback
{
void onPusherJoin(RTCMemberData member)
{
// A member enters a room and enables his/her camera and microphone.
this.BeginInvoke(new Action(() =>
```

©2013-2018 Tencent Cloud. All rights reserved.

```
{
mRTCRoom.addRemoteView(panel_display.Handle, new Rectangle(0, 0, panel_display.Width, panel
_display.Height), userID);
...
}));
}

...
}
```

5. onPusherQuit

- Definition: void onPusherQuit(RTCMemberData member)
- Description: Callback of members exiting the room
- Parameter:

Parameter	Type	Description
member	RTCMemberData	Member information. See the RTCMemberData class defined in RTCRoomUtil.cs.

- Example:

```
class MainDialog : IRTCRoomCallback
{
void onPusherQuit(RTCMemberData member)
{
// A member exits the room and disables his/her camera and microphone.
this.BeginInvoke(new Action(() =>
{
mRTCRoom.removeRemoteView(member.userID);
...
}));
...
}

...
}
```

©2013-2018 Tencent Cloud. All rights reserved.

6. onRoomClosed

- Definition: void onRoomClosed(string roomID)
- Description: Callback of room dissolution
- Parameter:

Parameter	Type	Description
roomID	string	Room ID

- Example:

```
class MainDialog : IRTCRoomCallback
{
void onRoomClosed(string roomID)
{
// It prompts that the room is dismissed. The opened audio/video is closed.
...
}

...
}
```

7. onRecvRoomTextMsg

- Definition: void onRecvRoomTextMsg(string roomID, string userID, string userName, string userAvatar, string msg)
- Description: Receives plain text messages
- Parameters:

Parameter	Type	Description
roomID	string	Room ID
userID	string	Sender ID

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
userName	string	Sender nickname
userAvatar	string	Sender profile photo
message	string	Text message

- Example:

```
class MainDialog : IRTCRoomCallback
{
void onRecvRoomTextMsg(string roomID, string userID, string userName, string userAvatar, string msg)
{
// Text messages are displayed on IM panel
}

...
}
```

8. onRecvRoomCustomMsg

- Definition: void onRecvRoomCustomMsg(string roomID, string userID, string userName, string userAvatar, string cmd, string message)
- Description: Receives custom messages
- Parameters:

Parameter	Type	Description
roomID	string	Room ID
userID	string	Sender ID
userName	string	Sender nickname
userAvatar	string	Sender profile photo
cmd	string	Custom CMD

©2013-2018 Tencent Cloud. All rights reserved.

Parameter	Type	Description
message	string	Custom message

- Example:

```
class MainDialog : IRTCRoomCallback
{
void onRecvRoomCustomMsg(string roomID, string userID, string userName, string userAvatar, str
ing cmd, string message)
{
// Parse and process custom messages such as Likes and gifts
}

...
}
```

9. onError

- Definition: void onError(RTCResult res, string userID)
- Description: Notification of internal RTCRoom error
- Parameters:

Parameter	Type	Description
res	RTCResult	Execution result. See RTCResult class defined in RTCRoomUtil.cs
userID	string	User ID

- Example:

```
class MainDialog : IRTCRoomCallback
{
void onError(RTCResult res, string userID)
{
// An error message pops up. Decide whether to dismiss or exit the meeting based on severity of er
rors.
}
```

©2013-2018 Tencent Cloud. All rights reserved.

```
...
}
```

©2013-2018 Tencent Cloud. All rights reserved.