**RTC Windows SDK接口文档**

1. 函数结构体说明

## 设备信息类

typedef struct {

char mDeviceName[MAX\_DEVICE\_ID\_LENGTH]; // 设备名称

char mDeviceId[MAX\_DEVICE\_ID\_LENGTH]; // 设备id

}tUCloudRtcDeviceInfo;

## 媒体启用配置类

最少启用一种媒体

typedef struct {

bool mVideoEnable; // 启用视频

bool mAudioEnable; // 启用音频

}tUCloudRtcMediaConfig;

## 媒体轨道类型类型描述

typedef enum {

UCLOUD\_RTC\_TRACKTYPE\_AUDIO = 0, // 音频轨道

UCLOUD\_RTC\_TRACKTYPE\_VIDEO = 1 // 视频轨道

} eUCloudRtcTrackType;

## 媒体流类型描述

typedef enum {

UCLOUD\_RTC\_MEDIATYPE\_NONE = 0, // 无效类型

UCLOUD\_RTC\_MEDIATYPE\_VIDEO = 1, // 摄像头

UCLOUD\_RTC\_MEDIATYPE\_SCREEN = 2 // 桌面流

} eUCloudRtcMeidaType;

## 流信息结构体

typedef struct {

const char\* mUserId; // 用户id

bool mEnableVideo; //启用视频

bool mEnableAudio; // 启用音频

bool mEnableData; // 启用数据通道（暂时无效）

eUCloudRtcMeidaType mStreamMtype;// 流类型

} tUCloudRtcStreamInfo;

## Mute操作结构体

typedef struct {

const char\* mUserId; // 用户id

bool mMute; //true 关闭媒体 false 打开媒体

eUCloudRtcMeidaType mStreamMtype; // 媒体流类型

} tUCloudRtcMuteSt;

## 录制输入等级

typedef enum {

UCLOUD\_RTC\_RECORDPROFILE\_SD = 1, //标清等级

UCLOUD\_RTC\_RECORDPROFILE\_HD, //高清等级

UCLOUD\_RTC\_RECORDPROFILE\_HDPLUS//超清等级

} eUCloudRtcRecordProfile;

## 录制水印位置

typedef enum {

UCLOUD\_RTC\_WATERMARKPOS\_LEFTTOP = 1, //左上

UCLOUD\_RTC\_WATERMARKPOS\_LEFTBOTTOM， // 左下

UCLOUD\_RTC\_WATERMARKPOS\_RIGHTTOP， // 右上

UCLOUD\_RTC\_WATERMARKPOS\_LEFTBOTTOM， // 右下

} eUCloudRtcWaterMarkPos;

## 录制媒体类型

typedef enum {

UCLOUD\_RTC\_RECORDTYPE\_AUDIOONLY = 1, //直录音频（混音）

UCLOUD\_RTC\_RECORDTYPE\_AUDIOVIDEO // 录制混音混流

} eUCloudRtcRecordType;

## 录制配置信息

typedef struct {

const char\* mMainviewuid; // 录制视频时主媒体

eUCloudRtcRecordProfile mProfile; // 输出等级

eUCloudRtcRecordType mRecordType; // 录制媒体类型

eUCloudRtcWaterMarkPos mWatermarkPos; // 水印位置

eUCloudRtcMeidaType mMainviewmediatype; // 主媒体类型

}tUCloudRtcRecordConfig;

## 渲染模式

typedef enum {

UCLOUD\_RTC\_RENDER\_MODE\_DEFAULT = 0, //默认平铺

UCLOUD\_RTC\_RENDER\_MODE\_FIT = 1, //保持比例

UCLOUD\_RTC\_RENDER\_MODE\_FILL = 2 //平铺

} eUCloudRtcRenderMode;

## 日志级别

typedef enum {

UCLOUD\_RTC\_LOG\_LEVEL\_DEBUG,

UCLOUD\_RTC\_LOG\_LEVEL\_INFO,

UCLOUD\_RTC\_LOG\_LEVEL\_WARN,

UCLOUD\_RTC\_LOG\_LEVEL\_ERROR,

UCLOUD\_RTC\_LOG\_LEVEL\_NONE,

} eUCloudRtcLogLevel;

## 视频质量级别

typedef enum {

UCLOUD\_RTC\_VIDEO\_PROFILE\_NONE = -1,

UCLOUD\_RTC\_VIDEO\_PROFILE\_320\_180 = 1,

UCLOUD\_RTC\_VIDEO\_PROFILE\_320\_240 = 2,

UCLOUD\_RTC\_VIDEO\_PROFILE\_640\_360 = 3,

UCLOUD\_RTC\_VIDEO\_PROFILE\_640\_480 = 4,

UCLOUD\_RTC\_VIDEO\_PROFILE\_1280\_720 = 5,

UCLOUD\_RTC\_VIDEO\_PROFILE\_1920\_1080 = 6

} eUCloudRtcVideoProfile;

## 桌面输出等级

typedef enum {

UCLOUD\_RTC\_SCREEN\_PROFILE\_LOW = 1,

UCLOUD\_RTC\_SCREEN\_PROFILE\_MIDDLE = 2,

UCLOUD\_RTC\_SCREEN\_PROFILE\_HIGH = 3

} eUCloudRtcScreenProfile;

## 桌面采集参数

typedef struct

{

long mScreenindex; // 桌面索引

int mXpos; // 起始x坐标

int mYpos; // 起始y坐标

int mWidth;

int mHeight;

}tUCloudRtcScreenPargram;

## 流权限

typedef enum {

UCLOUD\_RTC\_USER\_STREAM\_ROLE\_PUB, // 上行

UCLOUD\_RTC\_USER\_STREAM\_ROLE\_SUB,// 下行

UCLOUD\_RTC\_USER\_STREAM\_ROLE\_BOTH //双向

} eUCloudRtcUserStreamRole;

## 渲染窗口

typedef struct

{

int mVideoView; // 渲染的窗口句柄

const char\* mUserId; // 用户id

eUCloudRtcMeidaType mStreamMtype; // 媒体流类型

eUCloudRtcRenderMode mRenderMode; // 渲染模式

}tUCloudRtcVideoCanvas;

## 登录信息类

typedef struct

{

const char\* mAppId; // 平台分配的appid

const char\* mRoomId; // room 标识

const char\* mUserId; // 用户标识

const char\* mUserToken; // 用户通过用户服务器获取的token

}tUCloudRtcAuth;

## 当前媒体状态统计

typedef struct {

const char\* mUserId;

int mStreamMtype;

int mTracktype; // 1 audio 2 video

int mAudioBitrate = 0; // audio bitrate,unit:bps

int mVideoBitrate = 0;

int mVideoWidth = 0; // video width

int mVideoHeight = 0; // video height

int mVideoFrameRate = 0; // video frames per secon

float mPacketLostRate = 0.0f;

}tUCloudRtcStreamStats;

## 消息回调事件接口类

下面所有code 为0 代表成功 其他代表失败

class UCloudRtcEventListener

{

public：

// 服务器断开

virtual void onServerDisconnect() {}

// 加入房间通知

virtual void onJoinRoom(int code, const char\* msg, const char\* uid, const char\* roomid) {}

// 离开房间通知

virtual void onLeaveRoom(int code, const char\* msg, const char\* uid, const char\* roomid) {}

//房间重连中

virtual void onRejoining(const char\* uid, const char\* roomid) {}

// 房间重连成功

virtual void onReJoinRoom(const char\* uid, const char\* roomid) {}

// 本地流发布结果回调

virtual void onLocalPublish(const int code, const char\* msg, tUCloudRtcStreamInfo & info) {}

// 本地流取消发布结果回调

virtual void onLocalUnPublish(const int code, const char\* msg, tUCloudRtcStreamInfo & info) {}

// 远端有用户加入房间

virtual void onRemoteUserJoin(const char\* uid) {}

// 远端有用户离开房间

virtual void onRemoteUserLeave(const char\* uid,int reson) {}

// 远端用户发布视频

virtual void onRemotePublish(tUCloudRtcStreamInfo & info) {}

// 远端用户取消发布视频

virtual void onRemoteUnPublish(tUCloudRtcStreamInfo & info) {}

// 订阅某条视频流回调

virtual void onSubscribeResult(const int code, const char\* msg, tUCloudRtcStreamInfo & info) {}

// 取消订阅某条视频流回调

virtual void onUnSubscribeResult(const int code, const char\* msg, tUCloudRtcStreamInfo & info) {}

// 操作本地媒体流结果回调

virtual void onLocalStreamMuteRsp(const int code, const char\* msg,

eUCloudRtcMeidaType mediatype, UCloudTracktype tracktype, bool mute) {}

// 操作远端媒体流结果回调

virtual void onRemoteStreamMuteRsp(const int code, const char\* msg, const char\* uid,

eUCloudRtcMeidaType mediatype, eUCloudRtcTrackType tracktype, bool mute) {}

// 远端媒体流状态回调

virtual void onRemoteTrackNotify(const char\* uid,

eUCloudRtcMeidaType mediatype, eUCloudRtcTrackType tracktype, bool mute) {}

//发送状态信息

virtual void onSendRTCStats(tUCloudRtcStreamStats & rtstats) {}

//接收状态信息

virtual void onRemoteRTCStats(tUCloudRtcStreamStats rtstats) {}

//开启录制回调

virtual void onStartRecord(const int code, const char\* msg, const char\* recordid) {}

// 结束录制回调

virtual void onStopRecord(const int code, const char\* msg, const char\* recordid) {}

// 麦克风能量回调

virtual void onMiceAudioLevel(int volume) {}

// 远端能量回调

virtual void onRemoteAudioLevel(const char\* uid, int volume) {}

// 踢下线

virtual void onKickoff(int code) {}

// 警告

virtual void onWarning(int warn) {}

// 错误

virtual void onError(int error) {}

};

## 音频测试回调

class UCloudRtcMediaListener

{

public:

// 音频能量回调 （0--255）

virtual void onMiceAudioLevel(int volume) {}

};

1. 相关接口

## 操作接口

class UcloudRtcEngine,引擎接口类，具体每个接口功能

### 获取引擎类

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取引擎类 |
| 接口 | static UCloudRtcEngine \*sharedInstance(UCloudRtcMediaListener \*listener); |
| 返回值 | UCloudRtcEngine\* 引擎类指针 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| listener[in] | 消息回调通知类 | UCloudRtcMediaListener \* | N |

#### 消息回调

无

### 销毁引擎

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 销毁引擎 |
| 接口 | static void destroy() |
| 返回值 | 无 |

#### 参数说明

无

#### 消息回调

无

### 获取SDK 版本

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取SDK 版本 |
| 接口 | static const char \*getSdkVersion() |
| 返回值 | 对应的SDK版本号 |

#### 参数说明

无

#### 消息回调

无

### 设置SDK模式

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置sdk 模式 |
| 接口 | virtual int setSdkMode (eUCloudRtcSdkMode mode) |
| 返回值 | 参见错误描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| mode [in] | Sdk 模式 测试模式 正式模式详见eUCloudRtcSdkMode | enum | N |

#### 消息回调

无

### 设置流操作权限

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置流操作权限 |
| 接口 | virtual int setStreamRole(eUCloudRtcUserStreamRole role) |
| 返回值 | 参见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| role[in] | 详见UCloudStreamRole说明 | enum | N |

#### 消息回调

无

### 设置日志等级

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置日志等级 |
| 接口 | virtual void setLogLevel(eUCloudRtcLogLevel level) |
| 返回值 | 无 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| level[in] | 日志等级 详见eUCloudRtcLogLevel说明 | enum | N |

#### 消息回调

无

### 设置自动发布和订阅

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置自动发布和订阅 |
| 接口 | virtual int setAutoPublishSubscribe(bool autoPub, bool autoSub) |
| 返回值 | 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| autoPub[in] | 是否自动发布流 | bool | N |
| autoSub[in] | 是否自动订阅流 | bool | N |

#### 消息回调

无

### 设置纯音频模式

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置纯音频模式 |
| 接口 | virtual int setAudioOnlyMode(bool audioOnly) |
| 返回值 | 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| audioOnly[in] | 纯音频模式 | bool | N |

#### 消息回调

无

### 加入房间

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 加入房间 |
| 接口 | virtual int joinChannel(tUCloudRtcAuth & auth) |
| 返回值 | 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| auth[in] | 加入房间参数 详见tUCloudRtcAuth说明 | struct | N |

#### 消息回调

参见 消息回调事件接口类对应接口

virtual void onJoinRoom(int code, const char\* msg, const char\* uid, const char\* roomid) {}

### 离开房间

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 离开房间 |
| 接口 | virtual int leaveChannel() |
| 返回值 | 详见错误码描述 |

#### 参数说明

无

#### 消息回调

参见 消息回调事件接口类对应接口

virtual void onLeaveRoom(int code, const char\* msg, const char\* uid, const char\* roomid) {}

### 开启本地预览

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 开启本地预览 |
| 接口 | virtual int startPreview(tUCloudRtcVideoCanvas& view) |
| 返回值 | 详见错误码 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| view [in] | 详见tUCloudRtcVideoCanvas参数说明 | struct | N |

#### 消息回调

无

### 停止本地预览

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 停止本地预览 |
| 接口 | virtual int stopPreview (tUCloudRtcVideoCanvas& view) |
| 返回值 | 详见错误码 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| view [in] | 详见tUCloudRtcVideoCanvas参数说明 | struct | N |

#### 消息回调

无

### 设置视频显示等级

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置视频显示等级 |
| 接口 | virtual void setVideoProfile(eUCloudRtcVideoProfile& profile) |
| 返回值 | 无 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| profile[in] | 详见eUCloudRtcVideoProfile参数说明 | enum | N |

#### 消息回调

无

### 设置桌面输出等级

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置桌面输出等级 |
| 接口 | virtual void setDesktopProfile (eUCloudRtcVideoProfile& profile) |
| 返回值 | 无 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| profile[in] | 详见eUCloudRtcVideoProfile参数说明 | enum | N |

#### 消息回调

无

### 设置桌面采集参数

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置桌面采集参数 |
| 接口 | virtual void setCaptureScreenPagrams (tUCloudRtcScreenPargram& par) |
| 返回值 | 无 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| par[in] | 详见tUCloudRtcScreenPargram参数说明 | struct | N |

#### 消息回调

无

### 获取屏幕个数

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取屏幕个数 |
| 接口 | virtual void getDesktopNums () |
| 返回值 | 屏幕个数 |

#### 参数说明

无

#### 消息回调

无

### 发布媒体流

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 发布媒体流 |
| 接口 | virtual int publish(eUCloudRtcMeidaTypetype, bool hasvideo, bool hasaudio) |
| 返回值 | 详见错误码 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| type[in] | 发布流类型 详见eUCloudRtcMeidaType | enum | N |
| hasvideo[in] | 发布流是否带视频 | bool | N |
| has audio[in] | 发布流是否带音频 | bool | N |

#### 消息回调

参见 消息回调事件接口类对应接口

virtual void onLocalPublish (const int code, const char\* msg, tUCloudRtcStreamInfo& info) {}

### 停止媒体流发布

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 停止媒体流发布 |
| 接口 | virtual int unPublish(eUCloudRtcMeidaType type) |
| 返回值 | 详见错误码 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| type[in] | 发布流类型 详见eUCloudRtcMeidaType说明 | enum | N |

#### 消息回调

参见 消息回调事件接口类对应接口

virtual void onLocalUnPublish (const int code, const char\* msg, tUCloudRtcStreamInfo& info) {}

### 打开关闭本地音频

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 打开关闭本地音频（音频静音包发送） |
| 接口 | virtual int muteLocalMic(bool mute) |
| 返回值 | 详见错误码 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| mute[in] | 是否关闭音频 | bool | N |

#### 消息回调

参见 消息回调事件接口类对应接口

virtual void onLocalStreamMuteRsp(const int code, const char\* msg,

eUCloudRtcMeidaType mediatype, eUCloudRtcTrackType tracktype, bool mute) {}

### 打开关闭本地视频

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 打开关闭本地视频（黑屏帧发送） |
| 接口 | virtual int muteLocalVideo(bool mute, eUCloudRtcMeidaType mtype) |
| 返回值 | 详见错误码 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| mute[in] | 是否关闭视频 | bool | N |
| mtype[in] | 流类型 详见eUCloudRtcMeidaType说明 | enum | N |

#### 消息回调

参见 消息回调事件接口类对应接口

virtual void onLocalStreamMuteRsp(const int code, const char\* msg,

eUCloudRtcMeidaType mediatype, eUCloudRtcTrackType tracktype, bool mute) {}

### 开启远端渲染

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 开启远端渲染 函数只在自己订阅成功后才能调用 |
| 接口 | virtual int startRemoteView (tUCloudRtcVideoCanvas & view) |
| 返回值 | 详见错误码 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| view [in] | 详见tUCloudRtcVideoCanvas参数说明 | struct | N |

#### 消息回调

无

### 停止远端视频渲染

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 开启远端渲染 函数只在自己订阅成功后才能调用 |
| 接口 | virtual int stopRemoteView (tUCloudRtcVideoCanvas & view) |
| 返回值 | 详见错误码 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| view [in] | 详见tUCloudRtcVideoCanvas参数说明 | struct | N |

#### 消息回调

无

### 订阅远端媒体流

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 订阅远端媒体流 |
| 接口 | virtual int subscribe(tUCloudRtcStreamInfo & info) |
| 返回值 | 详见错误码 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| info[in] | 详见tUCloudRtcStreamInfo说明 | struct | N |

#### 消息回调

参见 消息回调事件接口类对应接口

virtual void onSubscribeResult(const int code, const char\* msg, tUCloudRtcStreamInfo& info) {}

### 取消订阅远端媒体流

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 取消订阅远端媒体流 |
| 接口 | virtual void unSubscribe(tUCloudRtcStreamInfo& info) |
| 返回值 | 0 成功 其他失败 详见错误码 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| info[in] | 详见tUCloudRtcStreamInfo | struct | N |

#### 消息回调

参见 消息回调事件接口类对应接口

virtual void onUnSubscribeResult(const int code, const char\* msg, tUCloudRtcStreamInfo& info) {}

### 打开关闭远端音频

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 静音开启远端音频 |
| 接口 | virtual int muteRemoteAudio (tUCloudRtcStreamInfo& info,bool mute) |
| 返回值 | 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| info[in] | 远端流信息 详见tUCloudRtcStreamInfo | struct | N |
| mute[in] | 是否关闭远端音频 | bool | N |

#### 消息回调

参见 消息回调事件接口类对应接口

virtual void onRemoteStreamMuteRsp(const int code, const char\* msg, const char\* uid,

eUCloudRtcMeidaType mediatype, eUCloudRtcTrackType tracktype, bool mute) {}

### 打开关闭远端视频

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 打开关闭远端视频 |
| 接口 | virtual int muteRemoteVideo(tUCloudRtcStreamInfo& info, bool mute) |
| 返回值 | 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| info[in] | 远端流信息 详见tUCloudRtcStreamInfo | struct | N |
| mute[in] | 是否关闭远端音频 | bool | N |

#### 消息回调

参见 消息回调事件接口类对应接口

virtual void onRemoteStreamMuteRsp(const int code, const char\* msg, const char\* uid,

eUCloudRtcMeidaType mediatype, eUCloudRtcTrackType tracktype, bool mute) {}

### 开启录制

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 开启录制 |
| 接口 | virtual int startRecord(tUCloudRtcRecordConfig& recordconfig) |
| 返回值 | 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| recordconfig [in] | 远端流信息 详见tUCloudRtcRecordConfig | struct | N |

#### 消息回调

参见 消息回调事件接口类对应接口

virtual void onStartRecord(const int code, const char\* msg, const char\* recordid) {}

### 停止录制

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 开启录制 |
| 接口 | virtual int stopRecord () |
| 返回值 | 详见错误码描述 |

#### 参数说明

无

#### 消息回调

参见 消息回调事件接口类对应接口

virtual void onStopRecord(const int code, const char\* msg, const char\* recordid) {}

### 配置本地摄像头是否发布

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 配置本地摄像头是否发布 |
| 接口 | virtual int configLocalCameraPublish(bool enable) |
| 返回值 | 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| enable[in] | 是否可以发布视频 | bool | N |

#### 消息回调

无

### 摄像头是否发布

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 摄像头是否发布 |
| 接口 | virtual bool isLocalCameraPublishEnabled() |
| 返回值 | true 有 false 没有 |

#### 参数说明

无

#### 消息回调

无

### 配置本地桌面是否发布

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 配置本地桌面是否发布 |
| 接口 | virtual int configLocalScreenPublish (bool enable) |
| 返回值 | 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| enable[in] | 是否可以发布本地桌面 | bool | N |

#### 消息回调

无

### 桌面是否发布

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 桌面是否发布 |
| 接口 | virtual bool isLocalScreenPublishEnabled() |
| 返回值 | true 有 false 没有 |

#### 参数说明

无

#### 消息回调

无

### 配置本地音频是否发布

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 配置本地音频是否发布 |
| 接口 | virtual int configLocalAudioPublish(bool enable) |
| 返回值 | 无 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| enable[in] | 是否可以发布本地音频 | bool | N |

#### 消息回调

无

### 音频是否发布

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 音频是否发布 |
| 接口 | virtual bool isLocalAudioPublishEnabled() |
| 返回值 | true 有 false 没有 |

#### 参数说明

无

#### 消息回调

无

### 是否为自动发布模式

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 是否为自动发布模式 |
| 接口 | virtual bool isAutoPublish() |
| 返回值 | true 是 false 不是 |

#### 参数说明

无

#### 消息回调

无

### 是否为自动订阅模式

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 是否为自动订阅模式 |
| 接口 | virtual bool isAutoSubscribe() |
| 返回值 | true 是 false 不是 |

#### 参数说明

无

#### 消息回调

无

### 是否为纯音频模式

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 是否为纯音频模式 |
| 接口 | virtual bool isAudioOnly() |
| 返回值 | true 是 false 不是 |

#### 参数说明

无

#### 消息回调

无

## 设备接口

设备模式通过class UcloudMediaDevice接口类提供功能，具体每个接口功能如下

### 初始化设备模块

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 初始化设备模块 |
| 接口 | static UCloudRtcMediaDevice\*sharedInstance(UCloudMediaListener \*listener) |
| 返回值 | UCloudRtcMediaDevice\* 设备类指针 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| listener[in] | 麦克风能量回调 | UCloudRtcMediaListener指针 | N |

#### 消息回调

无

### 销毁设备模块

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 销毁设备模块 |
| 接口 | static void destory() |
| 返回值 | 无 |

#### 参数说明

无

#### 消息回调

无

### 获取摄像头数量

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取摄像头数量 |
| 接口 | virtual int getCamNums() |
| 返回值 | 摄像头的数量 |

#### 参数说明

无

#### 消息回调

无

### 获取麦克风数量

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取麦克风数量 |
| 接口 | virtual int getRecordDevNums() |
| 返回值 | 获取的麦克风数量 |

#### 参数说明

无

#### 消息回调

无

### 获取播放设备数量

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取播放设备数量 |
| 接口 | virtual int getPlayoutDevNums() |
| 返回值 | 获取的播放设备数量 |

#### 参数说明

无

#### 消息回调

无

### 获取摄像头设备信息

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取摄像头设备信息 |
| 接口 | virtual int getVideoDevInfo(int index, char devname[MAX\_DEVICE\_NAME\_LEN], char devid[MAX\_DEVICE\_NAME\_LEN]) |
| 返回值 | 0 成功 其他错误 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| index[in] | 对应的数组下标 | int | N |
| devname[in] | 设备名称 | char数组 | N |
| devid[in] | 设备id 号 | char数组 | N |

#### 消息回调

无

### 获取麦克风设备信息

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取麦克风设备信息 |
| 接口 | virtual int getRecordDevInfo(int index, char devname[MAX\_DEVICE\_NAME\_LEN], char devid[MAX\_DEVICE\_NAME\_LEN]) |
| 返回值 | 0 成功 其他错误 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| index[in] | 对应的数组下标 | int | N |
| devname[in] | 设备名称 | char数组 | N |
| devid[in] | 设备id 号 | char数组 | N |

#### 消息回调

无

### 获取播放设备信息

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取播放设备信息 |
| 接口 | virtual int getPlayoutDevInfo(int index, char devname[MAX\_DEVICE\_NAME\_LEN], char devid[MAX\_DEVICE\_NAME\_LEN]) |
| 返回值 | 0 成功 其他错误 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| index[in] | 对应的数组下标 | int | N |
| devname[in] | 设备名称 | char数组 | N |
| devid[in] | 设备id 号 | char数组 | N |

#### 消息回调

无

### 获取当前使用的摄像头信息

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取当前使用的摄像头信息 |
| 接口 | virtual int getCurCamDev (char devname[MAX\_DEVICE\_NAME\_LEN], char devid[MAX\_DEVICE\_NAME\_LEN]) |
| 返回值 | 0 成功 其他错误 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| devname[in] | 设备名称 | char数组 | N |
| devid[in] | 设备id 号 | char数组 | N |

#### 消息回调

无

### 获取当前使用的麦克风设备信息

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取当前使用的麦克风设备信息 |
| 接口 | virtual int getCurRecordDev (char devname[MAX\_DEVICE\_NAME\_LEN], char devid[MAX\_DEVICE\_NAME\_LEN]) |
| 返回值 | 0 成功 其他错误 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| devname[in] | 设备名称 | char数组 | N |
| devid[in] | 设备id 号 | char数组 | N |

#### 消息回调

无

### 获取当前使用的播放设备信息

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取当前使用的播放设备信息 |
| 接口 | virtual int getCurPlayoutDev (char devname[MAX\_DEVICE\_NAME\_LEN], char devid[MAX\_DEVICE\_NAME\_LEN]) |
| 返回值 | 0 成功 其他错误 详见错误码描述 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| devname[in] | 设备名称 | char数组 | N |
| devid[in] | 设备id 号 | char数组 | N |

#### 消息回调

无

### 设置视频设备

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置视频设备 |
| 接口 | virtual int setVideoDevice(const char deviceId[MAX\_DEVICE\_NAME\_LEN]) |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| deviceid[in] | 设备id 号 | char数组 | N |

#### 消息回调

无

### 设置麦克风设备

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置麦克风设备 |
| 接口 | virtual int setRecordDevice(const char deviceId[MAX\_DEVICE\_NAME\_LEN]) |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| deviceid[in] | 设备id 号 | char数组 | N |

#### 消息回调

无

### 设置播放设备

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置播放设备 |
| 接口 | virtual int setPlayoutDevice (const char deviceId[MAX\_DEVICE\_NAME\_LEN]) |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| deviceid[in] | 设备id 号 | char数组 | N |

#### 消息回调

无

### 获取系统播放音量

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取系统播放音量 |
| 接口 | virtual int getPlaybackDeviceVolume (int \*volume) |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| volume[out] | 系统播放音量（0-255） | int 指针 | N |

#### 消息回调

无

### 设置系统播放音量

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置系统播放音量 |
| 接口 | virtual int setPlaybackDeviceVolume (int volume) |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| volume[in] | 系统播放音量（0-255） | int | N |

#### 消息回调

无

### 获取系统麦克风音量

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 获取系统麦克风音量 |
| 接口 | virtual int getRecordingDeviceVolume (int \*volume) |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| volume[out] | 系统麦克风音量（0-255） | int 指针 | N |

#### 消息回调

无

### 设置系统麦克风音量

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 设置系统麦克风音量 |
| 接口 | virtual int getRecordingDeviceVolume (int volume) |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| volume[in] | 系统麦克风音量（0-255） | int | N |

#### 消息回调

无

### 开始摄像头测试

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 开始摄像头测试 |
| 接口 | virtual int startCamTest(const char deviceId[MAX\_DEVICE\_NAME\_LEN]，UCloudRtcVideoProfile& profile , void\* videoview) |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| deviceid[in] | 摄像头的id | char 数组 | N |
| profile[in] | 摄像头的质量等级 | UCloudRtcVideoProfile | N |
| videoview[in] | 显示的渲染窗口句柄 | void\* | N |

#### 消息回调

无

### 停止摄像头测试

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 停止摄像头测试 |
| 接口 | virtual int stopCamTest() |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

无

#### 消息回调

无

### 开始麦克风测试

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 开始麦克风测试 |
| 接口 | virtual int startRecordingDeviceTest(const char deviceId[MAX\_DEVICE\_NAME\_LEN]) |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| deviceid[in] | 麦克风的id | char 数组 | N |

#### 消息回调

virtual void onMiceAudioLevel(int volume) {} volume 音频能量 （0--255）

### 停止麦克风测试

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 停止麦克风测试 |
| 接口 | virtual int stopRecordingDeviceTest() |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

无

#### 消息回调

无

### 开始播放设备测试

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 开始播放设备测试 |
| 接口 | virtual int startPlaybackDeviceTest(const char deviceId[MAX\_DEVICE\_NAME\_LEN],const char\* filePath) |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据类型** | **可空** |
| deviceid[in] | 播放设备的id | char 数组 | N |
| filePath[in] | 播放文件的地址 | const char\* | N |

#### 消息回调

无

### 停止播放设备测试

#### 接口定义

|  |  |
| --- | --- |
| 接口描述 | 停止播放设备测试 |
| 接口 | virtual int stopPlaybackDeviceTest() |
| 返回值 | 0 成功 其他错误 详见错误说明 |

#### 参数说明

无

#### 消息回调

无

1. 接口错误表

## 事件回调错误码

typedef enum \_tUCloudRtcCallbackErrCode{

UCLOUD\_RTC\_CALLBACK\_ERR\_CODE\_OK=0,

UCLOUD\_RTC\_CALLBACK\_ERR\_SAME\_CMD = 3000,

UCLOUD\_RTC\_CALLBACK\_ERR\_NOT\_IN\_ROOM,

UCLOUD\_RTC\_CALLBACK\_ERR\_ROOM\_JOINED,

UCLOUD\_RTC\_CALLBACK\_ERR\_NETWORK,

UCLOUD\_RTC\_CALLBACK\_ERR\_ROOM\_RECONNECTING,

UCLOUD\_RTC\_CALLBACK\_ERR\_PUB\_TOO\_MORE,

UCLOUD\_RTC\_CALLBACK\_ERR\_STREAM\_PUBED,

UCLOUD\_RTC\_CALLBACK\_ERR\_STREAM\_NOT\_PUB,

UCLOUD\_RTC\_CALLBACK\_ERR\_STREAM\_SUBED,

UCLOUD\_RTC\_CALLBACK\_ERR\_STREAM\_NO\_SUB,

UCLOUD\_RTC\_CALLBACK\_ERR\_SUB\_NO\_USER,

UCLOUD\_RTC\_CALLBACK\_ERR\_SUB\_NO\_MEDIA,

UCLOUD\_RTC\_CALLBACK\_ERR\_USER\_LEAVING,

UCLOUD\_RTC\_CALLBACK\_ERR\_SERVER\_CON\_FAIL,

UCLOUD\_RTC\_CALLBACK\_ERR\_INVAILD\_ST,

UCLOUD\_RTC\_CALLBACK\_ERR\_SERVER\_DIS,

UCLOUD\_RTC\_CALLBACK\_ERR\_NO\_HAS\_TRACK,

UCLOUD\_RTC\_CALLBACK\_ERR\_MSG\_TIMEOUT,

}tUCloudRtcCallbackErrCode;

## 函数值错误码

typedef enum \_tUCloudRtcReturnErrCode {

UCLOUD\_RTC\_RETURN\_ERR\_CODE\_OK = 0,

UCLOUD\_RTC\_RETURN\_ERR\_AUTO\_PUB = 1000, //自动发布

UCLOUD\_RTC\_RETURN\_ERR\_AUTO\_SUB, //自动订阅

UCLOUD\_RTC\_RETURN\_ERR\_NOT\_INIT, //引擎没有初始化

UCLOUD\_RTC\_RETURN\_ERR\_IN\_ROOM,

UCLOUD\_RTC\_RETURN\_ERR\_NOT\_IN\_ROOM,

UCLOUD\_RTC\_RETURN\_ERR\_NOT\_PUB\_TRACK,

UCLOUD\_RTC\_RETURN\_ERR\_INVAILED\_PARGRAM,// 无效参数

UCLOUD\_RTC\_RETURN\_ERR\_INVAILED\_WND\_HANDLE,// 无效窗口句柄

UCLOUD\_RTC\_RETURN\_ERR\_INVAILED\_MEDIA\_TYPE,// 无效媒体类型

UCLOUD\_RTC\_RETURN\_ERR\_SUB\_ONEMORE,// 最少订阅一种媒体

UCLOUD\_RTC\_RETURN\_ERR\_NO\_PUB\_ROLE, //无发布权限

UCLOUD\_RTC\_RETURN\_ERR\_NO\_SUB\_ROLE, //无订阅权限

UCLOUD\_RTC\_RETURN\_ERR\_CAM\_NOT\_ENABLE, //没有配置本地cam 发送

UCLOUD\_RTC\_RETURN\_ERR\_SCREEN\_NOT\_ENABLE, //没有配置本地screen 发送

UCLOUD\_RTC\_RETURN\_ERR\_AUDIO\_MODE,

UCLOUD\_RTC\_RETURN\_ERR\_SECKEY\_INVALID

}tUCloudRtcReturnErrCode;

# **SDK接入指导**

## 开发语言以及系统要求

* 开发语言：C++
* 系统要求：Windows 7 及以上版本的 Windows 系统

## 开发环境

* [Visual Studio 2015](https://www.visualstudio.com/zh-hans/vs/older-downloads/) 及其它c++ 开发环境
* Win32 Platform

## 导入 SDK

1. 将 sdk/include 目录添加到项目的 INCLUDE 目录下。
2. 将 sdk/lib 目录放入项目的 LIB 目录下。
3. 将 sdk/dll 下的 dll 文件复制到你的可执行文件所在的目录下。

## 初始化

1. 继承实现UCloudRtcEventListener，用作事件处理
2. m\_rtcengine = UCloudRtcEngine::sharedInstance(UCloudRtcEventListenerImpl);

m\_rtcengine->setSdkMode (1); // 1 testmode 0 normal

m\_rtcengine->setTokenSecKey(TEST\_SECKEY);//测试模式下设置自己的秘钥

m\_rtcengine->setStreamRole(STREAM\_BOTH);

m\_rtcengine->setAudioOnlyMode(false);

m\_rtcengine->setAutoPublishSubscribe(false, true);

m\_rtcengine->configLocalAudioPublish(false);

m\_rtcengine->configLocalCameraPublish(true);

m\_rtcengine->configLocalScreenPublish(false);

m\_rtcengine->setVideoProfile(UCLOUD\_RTC\_VIDEO\_PROFILE\_640\_360);

## 加入房间

tUCloudRtcAuth auth;

auth.mAppId = appid;

auth.mRoomId = roomid;

auth.mUserId = userid;

auth.mUserToken = "1223222";

m\_rtcengine->joinChannel(auth);

## 发布 取消发布

* 发布流：

tUCloudRtcMediaConfig config;

config.mAudioEnable = true;

config.mVideoEnable = true;

m\_rtcengine->publish(UCLOUD\_RTC\_MEDIATYPE\_VIDEO, config.mVideoEnable,

config.mAudioEnable)

* 取消发布

tUCloudRtcVideoCanvas view;

view.mVideoView = (int)m\_localWnd->GetVideoHwnd();

view.mStreamMtype = UCLOUD\_RTC\_MEDIATYPE\_VIDEO; m\_rtcengine->stopPreview(view);

m\_rtcengine->unPublish(UCLOUD\_RTC\_MEDIATYPE\_VIDEO);

## 订阅流 取消订阅

* 订阅流：

m\_rtcengine->subscribe(tUCloudRtcStreamInfo & info)

* 取消订阅：

m\_rtcengine->unSubscribe(tUCloudRtcStreamInfo& info)

## 离开房间

m\_rtcengine->leaveChannel ()

## 编译运行

开始体验吧