# RTCS接口说明文档

目录

[RTCS接口说明文档 1](#_Toc5198932)

[一、回调函数定义 2](#_Toc5198933)

[异步消息通知回调函数定义 2](#_Toc5198934)

[透明通道数据回调函数 2](#_Toc5198935)

[二、API方法定义 3](#_Toc5198936)

[2.1 SDK相关 3](#_Toc5198937)

[初始化系统 3](#_Toc5198938)

[释放SDK资源 3](#_Toc5198939)

[SDK内核参数设置 3](#_Toc5198940)

[获取SDK内核参数设置 3](#_Toc5198941)

[2.2回调函数注册 4](#_Toc5198942)

[设置异步消息通知回调函数 4](#_Toc5198943)

[设置透明通道数据回调函数 4](#_Toc5198944)

[2.3业务流程 5](#_Toc5198945)

[连接服务器 5](#_Toc5198946)

[登录服务器 5](#_Toc5198947)

[注销系统 5](#_Toc5198948)

[进入房间 5](#_Toc5198949)

[离开房间 6](#_Toc5198950)

[获取指定房间在线用户列表 6](#_Toc5198952)

[查询用户状态 6](#_Toc5198954)

[透明通道传送缓冲区 7](#_Toc5198955)

[2.4多媒体设备相关 7](#_Toc5198956)

[枚举本地视频采集设备 7](#_Toc5198957)

[选择指定的视频采集设备 8](#_Toc5198958)

[获取当前使用的视频采集设备 8](#_Toc5198959)

[获取指定视频采集设备属性 8](#_Toc5198960)

[设置视频显示位置 8](#_Toc5198961)

[用户信息水印叠加 9](#_Toc5198962)

[操作用户视频 9](#_Toc5198963)

[用户全屏幕共享 9](#_Toc5198964)

[用户窗体共享 9](#_Toc5198965)

[改变某用户视频绘制效果 10](#_Toc5198966)

[枚举本地音频采集设备 10](#_Toc5198967)

[选择指定的音频采集设备 10](#_Toc5198968)

[获取当前使用的音频采集设备 10](#_Toc5198969)

[枚举本地音频播放设备 11](#_Toc5198970)

[选择指定的音频播放设备 11](#_Toc5198971)

[获取当前使用的音频播放设备 11](#_Toc5198972)

[获取指定音频设备的当前音量 12](#_Toc5198973)

[设置指定音频设备的音量 12](#_Toc5198974)

[操作用户语音 12](#_Toc5198975)

## **一、回调函数定义**

### 异步消息通知回调函数定义

typedef void (CALLBACK\* RTCS\_NotifyMessage\_CallBack)(int dwNotifyMsg, int wParam, int lParam, void\* lpUserValue);

**参数**：

dwNotifyMsg 事件类型，参考：“消息预定义”章节

wParam： 事件附带参数

lParam： 事件附带参数

lpUserValue： 用户自定义参数，在设置回调函数时传入

**备注**：

当注册该回调函数后，SDK 内部有消息需要通知上层应用时，将触发该回调函数。

### 透明通道数据回调函数

typedef void (CALLBACK \* RTCS\_TransBuffer\_CallBack)(int dwUserid, const char\* lpBuf, int dwLen, void\* lpUserValue);

**参数**：

dwUserid： 用户 ID，指示发送用户，如果为 0，则表示是服务器发送

lpBuf： 缓冲区地址

dwLen： 缓冲区大小

lpUserValue： 用户自定义参数，在设置回调函数时传入

**备注**：

当收到其它用户使用“BRAC\_TransBuffer”方法发送的数据时，接收方将会

触发该回调函数。

## 二、API方法定义

## 2.1 SDK相关

### 初始化系统

RTCS\_C\_API int RTCS\_InitSDK(unsigned int dwFuncMode);

**功能**： 初始化 SDK；

**返回值**：0 表示成功，否则为出错代码；

**参数**：

dwFuncMode： 功能模式组合待添加；

**备注**：

上层应用通过“异步事件通知”回调函数来获得相关事件通知。

### 释放SDK资源

RTCS\_C\_API int RTCS\_Release(VOID);

**功能**：释放 SDK 占用的所有资源；

**返回值**：0 表示成功，否则为出错代码；

**参数**： 无。

**备注**：

该方法必须后一个被调用，调用该方法后，SDK 内部所占用的资源将被释

放，如果在其后面再调用其它的方法，将会返回没有初始化的错误。 该方法通常在上层应用退出系统时被调用。

### SDK内核参数设置

RTCS\_C\_API int RTCS\_SetSDKOption(int optname,const char\* optval,int optlen);

**功能**：SDK内核参数，设置编解码器参数，网络参数，SDK相关设置参数等。

**返回值**：0表示成功，否则为出错代码

**参数**：

Optname 内核参数名称代码(待确定参数代码)

Optval 参数内存数据

Optlen 参数内存数据所占字节

### 获取SDK内核参数设置

RTCS\_C\_API int RTCS\_GetSDKOption(int optname, char\* optval, int

optlen);

**功能**：SDK 内核参数状态查询

**返回值**：0 表示成功，否则为出错代码

**参数**：

optname 内核参数名称；

optval 设置的参数地址；

optlen 设置的参数所占内存大小。

## 2.2回调函数注册

### 设置异步消息通知回调函数

RTCS\_C\_API int RTCS\_SetNotifyMessageCallBack(RTCS\_NotifyMessage\_CallBack lpFunction, void\* lpUserValue = 0);

**功能**：设置异步消息通知回调函数，当 SDK 内部有消息需要通知上层应用时，

将会触发所注册的回调函数。

**返回值**：0 表示成功，否则为出错代码

**参数**：

lpFunction 回调函数地址，函数定义参考“回调函数”一节；

lpUserValue 用户自定义参数，该参数在回调函数中被返回，默认为 NULL，

通常传入一个对象的地址（指针）。

**备注**：

SDK 的状态变化默认是通过异步消息（PostMessage）传递给上层应

用（接收消息的窗口是在 RTCS\_InitSDK 接口中传入），如果上层应用不希望处

理异步消息（因为需要为每一个消息定义消息处理函数），则可以调用该方法，

注册一个异步消息回调函数，让 SDK 通过回调的形式将消息通知上层应用。

### 设置透明通道数据回调函数

RTCS\_C\_API int RTCS\_SetTransBufferCallBack(RTCS\_TransBuffer\_CallBack

lpFunction, LPVOID lpUserValue=NULL);

**功能**：设置透明通道数据回调函数，使得当有别的用户发送的透明通道数据时，

能通过回调函数，将透明通道的缓冲区数据回调给上层应用。

**返回值**：0 表示成功，否则为出错代码；

**参数**：

lpFunction 回调函数地址，函数定义参考“回调函数”一节；

lpUserValue 用户自定义参数，该参数在回调函数中被返回，默认为 NULL，

通常传入一个对象的地址（指针）。

## 2.3业务流程

### 连接服务器

RTCS\_C\_API int RTCS\_Connect(const char\* lpServerAddr, int port);

**功能**： 用于与核心服务器建立连接。

**返回值**：0 表示成功，否则为出错代码

103 域名解析失败

**参数**：

lpServerAddr：服务器IP地址，或是网站域名（URL）地址；

port： 服务端口号。

**备注**：

返回值为 0 并不表示连接服务器成功，仅表示 SDK 已成功收到连接服务器的指令，如果连接成功，或是失败，都将会通过相应的消息通知上层应用，这里是一个异步的过程。

### 登录服务器

RTCS\_C\_API int RTCS\_Login(const char\* username, const char\* passwd);

**功能**：登录服务器，请求身份认证。

**返回值**：0 表示成功，否则为出错代码；

201 表示当前已登录。

**参数**：

username：注册用户名

passwd：登录密码

### 注销系统

RTCS\_C\_API int RTCS\_Logout(VOID);

**功能**：将用户从系统中注销。

**返回值**：0 表示成功，否则为出错代码。

**参数**：无。

**备注**：

在切换用户（如用户换用其它的用户名登录系统）时需要先调用该方法，或

是在退出系统前需要调用该方法。

### 进入房间

RTCS\_C\_API int RTCS\_EnterRoom(int roomid, const char\* lpRoomPass = 0);

**功能**： 根据房间编号进入房间

**返回值**： 0 表示成功，否则为出错代码

308 当前已进入房间

**参数**：

roomid：房间编号，系统唯一；

lpRoomPass：房间密码（当房间需要密码时有效，如果没有可为空）；

**备注**：

该方法可以登录系统之后立即调用，而不用关心登录系统是否成功，当 SDK登录系统成功之后，如果之前调用过该方法，则 SDK 将会自动向服务器发出进入房间的申请。

返回值为 0 并不表示进入房间成功，仅表示 SDK 已成功收到进入房间的指令，不论成功，或是失败，都将会通过相应的消息通知上层应用，这里是一个异步的过程。

### 离开房间

RTCS\_C\_API int RTCS\_LeaveRoom(int roomid);

**功能**：离开房间。

**返回值**：0 表示成功，否则为出错代码

3 不在房间中

208 没有登录

**参数**：

roomid：房间编号，为-1 表示退出当前房间

**备注**：

在用户变换房间之前，需要调用该方法离开房间，然后才能进入新的房间。

### 获取指定房间在线用户列表

RTCS\_C\_API int RTCS\_GetRoomOnlineUsers(int roomid, int\* userIDArray, unsigned int& dwUserNum);

**功能**：获取指定房间在线用户列表

**参数**：

dwRoomID：整形值，房间编号，系统唯一；

lpUserIDArray：用户列表，可以为 NULL，表示只获取用户数量；

dwUserNum：用户数量。

**返回值**：返回在线用户 ID 数组。

**说明**：

获取指定房间当前在线用户列表，接口调用后会返回在线用户 ID 数组。

### 查询用户状态

RTCS\_C\_API int RTCS\_QueryUserState(int userid, int infoname, char\*

infoval, int infolen);

**功能**：查询指定用户状态；

**返回值**：0 表示成功，否则为出错代码；

**参数**：

dwUserId 用户编号，可用-1 代表本地用户（自己）；

infoname 需要查询的信息代码（见备注附表）；

infoval 查询信息值（结果）的保存地址；

infolen 保存查询信息值参数类型所占内存大小。

**备注**：

通过调用该方法，可以查询指定用户的相关状态值。

### 透明通道传送缓冲区

RTCS\_C\_API int RTCS\_TransBuffer(int userid, const char\*

lpBuf, int dwLen);

**功能**：传输消息；

**返回值**：0 表示成功，否则为出错代码；

**参数**：

userid：-1向当前房间广播，不支持频率很高的调用

-2向当前登录的用户广播，频率无所谓

=0向服务器发送

>0向指定用户发送；

lpBuf：发送的信息内容；

dwLen：发送信息长度。

**备注**：

通过调用该方法，可以发送相关信息。

## 2.4多媒体设备相关

### 枚举本地视频采集设备

RTCS\_C\_API int RTCS\_EnumVideoCapture(char\*\* lpDeviceName, unsigned int& dwDeviceNum);

**功能**： 枚举本地视频采集设备

**返回值**：0 表示成功，否则为出错代码

**参数**：

lpDeviceName 视频设备名称，可为空，表示只获取设备数量；

dwDeviceNum 视频设备数量;

**备注**：

该方法将会在内部分配缓冲区，外部使用完成之后，必需手工释放这些缓冲区，否则会造成内存泄露

### 选择指定的视频采集设备

RTCS\_C\_API int RTCS\_SelectVideoCapture(char\* szCaptureName);

**功能**： 选择指定的视频采集设备；

**返回值**：0 表示成功，否则为出错代码；

**参数**：

szCaptureName 所获取设备的名称；

**备注**：

当用户有多个视频采集设备（USB 摄像头、虚拟摄像头、采集卡等）时，可以通过该方法选用指定的视频采集设备。

### 获取当前使用的视频采集设备

RTCS\_C\_API int RTCS\_GetCurVideoCapture(char\* deviceName, int len);

**功能**： 获取当前使用的视频采集设备；

**返回值**：0 表示成功，否则为出错代码；

**参数**：

deviceName：所获取设备的名称；

len：设备名称长度。

**备注**：

当用户有多个视频采集设备（USB 摄像头、虚拟摄像头、采集卡等）时，可以通过该方法获取当前使用的视频采集设备。

### 获取指定视频采集设备属性

RTCS\_C\_API int RTCS\_GetVideoCaptureAttributes(char\* szCaptureName, char\* szAttributes);

**功能**： 获取指定的视频采集设备属性；

**返回值**：0 表示成功，否则为出错代码；

**参数**：

szCaptureName：所获取设备的名称；

szAttributes：设备属性。

**备注**：

当用户有多个视频采集设备（USB 摄像头、虚拟摄像头、采集卡等）时，可以通过该方法获取指定的视频采集设备属性。

### 设置视频显示位置

RTCS\_C\_API int RTCS\_SetVideoPos(int dwUserid, void\* hWnd, unsigned int dwLeft, unsigned int dwTop, unsigned int dwRight, unsigned int dwBottom);

**功能**：设置视频显示位置，或是刷新视频显示

**返回值**：0 表示成功，否则为出错代码

**参数**：

dwUserid：用户编号，为-1 表示操作自己的视频显示位置

hWnd：视频显示父窗口句柄

dwLeft、dwTop、dwRight、dwBottom：位置信息，相对于父窗口（hWnd）

**备注**：

该方法只有当初始化系统时，设置了 RTCS\_FUNC\_VIDEO\_AUTODISP（由

SDK 包来处理视频）标志，才必须调用，如果视频显示是由上层应用自己来渲

染，则不需要调用该方法。

### 用户信息水印叠加

RTCS\_C\_API int RTCS\_PaintUserValue(int dwuserid，const char\* data,const char\* font = "msyh", int size = 40);

**功能**：用户视频上显示信息

**返回值**：0标识成功，否则为出错代码

**参数**：

dwuserid：用户id

data：参数及显示信息

font：字体

size：字体大小。

### 操作用户视频

RTCS\_C\_API int RTCS\_UserCameraControl(int userid，bool isopen);

**功能**：操作用户视频，打开或关闭；

**返回值**：0标识成功，否则为出错代码

**参数**：

userid：用户id

isopen：true打开，false关闭

### 用户全屏幕共享

RTCS\_C\_API int RTCS\_UserDesktopShare(int desktop\_id);

**功能**：操作用户视频，打开或关闭；

**返回值**：0标识成功，否则为出错代码

**参数**：

desktop：用户id

### 用户窗体共享

RTCS\_C\_API int RTCS\_UserDesktopShare(void\* hWnd);

**功能**：用户窗体共享；

**返回值**：0标识成功，否则为出错代码

**参数**：

hWnd：

### 改变某用户视频绘制效果

RTCS\_C\_API int RTCS\_SetRenderFlag (int userid, int render\_flag);

**功能**：改变某用户视频绘制效果；

**返回值**：0标识成功，否则为出错代码

**参数**：

userid：用户id；

render\_flag：0 原视频根据窗口尺寸截取，全窗口展示；1 保留原视频所有信息，适应窗口比例展示。

### 枚举本地音频采集设备

RTCS\_C\_API int RTCS\_EnumAudioCapture(char\*\* lpDeviceName, unsigned int& dwDeviceNum);

**功能**： 枚举本地音频采集设备；

**返回值**：0 表示成功，否则为出错代码；

**参数**：

lpDeviceName 音频设备名称；

dwDeviceNum 音频设备数量；

**备注**：

该方法将会在内部分配缓冲区，外部使用完成之后，必需手工释放这些缓冲区，否则会造成内存泄露。

### 选择指定的音频采集设备

RTCS\_C\_API int RTCS\_SelectAudioCapture(char\* szCaptureName);

**功能**： 选择指定的音频采集设备；

**返回值**：0 表示成功，否则为出错代码；

**参数**：

szCaputureName 所选择的设备名称；

**备注**：

当用户有多个音频采集设备（板载声卡、USB 声卡等）时，可以通过该方法选用指定的音频采集设备。

### 获取当前使用的音频采集设备

RTCS\_C\_API int RTCS\_GetCurAudioCapture(char\* deviceName, int len);

**功能**： 获取当前使用的音频采集设备；

**返回值**：0 表示成功，否则为出错代码；

**参数**：

deviceName：当前使用的设备名称；

len：设备名称长度。

**备注**：

当用户有多个音频采集设备（板载声卡、USB 声卡等）时，可以通过该方法获取当前使用的音频采集设备。

### 枚举本地音频播放设备

RTCS\_C\_API int RTCS\_EnumAudioPlayback(char\*\* lpDeviceName, unsigned int& dwDeviceNum);

**功能**： 枚举本地音频播放设备；

**返回值**：0 表示成功，否则为出错代码；

**参数**：

lpDeviceName 音频设备名称；

dwDeviceNum 音频设备数量；

**备注**：

该方法将会在内部分配缓冲区，外部使用完成之后，必需手工释放这些缓冲区，否则会造成内存泄露，由于内部采用了“GlobalAlloc”来分配高端内存，故外部需要调用“GlobalFree”来释放，而不能是 delete 或 free 方法。

### 选择指定的音频播放设备

RTCS\_C\_API int RTCS\_SelectAudioPlayback(char\* szDeviceName);

**功能**： 选择指定的音频播放设备；

**返回值**：0 表示成功，否则为出错代码；

**参数**：

szDeviceName 所选择的设备名称；

**备注**：

当用户有多个音频播放设备（板载声卡、USB 声卡、蓝牙耳机等）时，可以通过该方法选用指定的音频播放设备。

### 获取当前使用的音频播放设备

RTCS\_C\_API int RTCS\_GetCurAudioPlayback(char\* deviceName, int len);

**功能**： 获取当前使用的音频播放设备；

**返回值**：0 表示成功，否则为出错代码；

**参数**：

deviceName：当前使用的设备名称；

len：设备名称长度。

**备注**：

当用户有多个音频采集设备（板载声卡、USB 声卡等）时，可以通过该方法获取当前使用的音频播放设备。

### 获取指定音频设备的当前音量

RTCS\_C\_API int RTCS\_AudioGetVolume(RTCS\_AudioDevice device, unsigned int& dwVolume);

**功能**： 获取指定音频设备的当前音量

**返回值**：0 表示成功，否则为出错代码

**参数**：

device：设备类型，定义为：

BRAC\_AD\_WAVEIN = 0, ///< 输入设备：Mic

BRAC\_AD\_WAVEOUT = 1, ///< 输出设备：Wave

dwVolume： 保存该设备的当前音量，取值范围：0~100；

**备注**：

根据设备类型（device）参数的不同，可以获取放音设备（WaveOut）和录音设备（WaveIn）的当前音量大小。

### 设置指定音频设备的音量

RTCS\_C\_API int RTCS\_AudioSetVolume(RTCS\_AudioDevice device, unsigned int dwVolume);

**功能**：设置指定音频设备的音量

**返回值**：0 表示成功，否则为出错代码

**参数**：

device：设备类型，定义为：

BRAC\_AD\_WAVEIN = 0, ///< 输入设备：Mic BRAC\_AD\_WAVEOUT = 1, ///< 输出设备：Wave

dwVolume： 需要设置的音量，取值范围：0~100，值越大，音量越大；

**备注**：

根据设备类型（device）参数的不同，可以调节放音设备（WaveOut）和录音设备（WaveIn）的音量大小。

### 操作用户语音

RTCS\_C\_API int RTCS\_UserSpeakControl(int userid, bool isopen);

**功能**：操作用户语音，打开或关闭；

**返回值**：0 表示成功，否则为出错代码

**参数**：

userid：用户id，-1表示自己；

isopen：true打开，false关闭；