

# AnyChat for Linux SDK

## 开发手册

(版本: V2.1)



广州佰锐网络科技有限公司

**GuangZhou BaiRui Network Technology Co.,Ltd.**

<http://www.bairuitech.com> <http://www.anychat.cn>

2014 年 09 月

# 目 录

<b>1. 系统概述 .....</b>	<b>6</b>
1.1 系统介绍 .....	6
1.2 系统特性 .....	7
1.2.1 视频技术 .....	7
1.2.2 音频技术 .....	7
1.2.3 P2P技术 .....	8
1.3 关于佰锐科技 .....	8
1.4 技术支持 .....	8
1.5 版权申明 .....	9
<b>2. 编程指南 .....</b>	<b>10</b>
2.1. 客户端SDK概述 .....	10
2.2. 函数调用顺序 .....	11
2.3. 服务器SDK概述 .....	12
<b>3. 数据结构及常量定义 .....</b>	<b>13</b>
3.1. 视频图像格式 .....	13
3.2. 音频设备定义 .....	13
3.3. 功能模式定义 .....	13
3.4. SDK内核参数定义 .....	14
3.5. 用户状态标志定义 .....	18
3.6. 视频呼叫事件定义 .....	18
<b>4. 回调函数 .....</b>	<b>19</b>
4.1. 异步事件 .....	19
4.1.1. 事件预定义 .....	19
4.1.2. 网络连接事件 .....	20
4.1.3. 登录系统事件 .....	20
4.1.4. 自己进入房间事件 .....	20
4.1.5. 音频设备状态变化事件 .....	20
4.1.6. 用户进入（离开）房间事件 .....	21
4.1.7. 网络连接关闭事件 .....	21
4.1.8. 当前房间在线用户事件 .....	22
4.1.9. 用户摄像头状态变化事件 .....	22
4.1.10. 用户聊天模式变化事件 .....	22
4.1.11. 用户活动状态变化事件 .....	23
4.1.12. P2P连接状态变化事件 .....	23
4.1.13. 用户视频分辨率发生变化事件 .....	23
4.1.14. 用户信息更新通知消息 .....	24
4.1.15. 好友在线状态变化消息 .....	24
4.1.16. 用户发起私聊请求事件 .....	24
4.1.17. 用户回复私聊请求事件 .....	24

4.1.18.	用户退出私聊事件 .....	25
4.1.19.	SDK警告事件 .....	25
4.2.	回调函数定义 .....	25
4.2.1.	视频数据回调函数 .....	26
4.2.2.	视频数据扩展回调函数 .....	26
4.2.3.	音频数据回调函数 .....	27
4.2.4.	音频数据扩展回调函数 .....	27
4.2.5.	文字消息回调函数 .....	28
4.2.6.	透明通道数据回调函数 .....	28
4.2.7.	透明通道数据扩展回调函数 .....	29
4.2.8.	文件传输回调函数 .....	29
4.2.9.	音量变化回调函数 .....	30
4.2.10.	SDK Filter通信数据回调函数.....	30
4.2.11.	录像、快照任务完成回调函数 .....	30
4.2.12.	异步事件通知回调函数 .....	31
4.2.13.	视频呼叫事件回调函数 .....	31
4.2.14.	数据加密、解密回调函数 .....	32
5.	函数说明 .....	34
5.1.	初始化与资源释放 .....	34
5.1.1.	初始化SDK .....	34
5.1.2.	设置视频数据回调函数 .....	35
5.1.3.	设置视频数据扩展回调函数 .....	36
5.1.4.	设置音频数据回调函数 .....	36
5.1.5.	设置音频数据扩展回调函数 .....	37
5.1.6.	设置文字消息回调函数 .....	37
5.1.7.	设置透明通道数据回调函数 .....	38
5.1.8.	设置透明通道数据扩展回调函数 .....	38
5.1.9.	设置文件传输回调函数 .....	39
5.1.10.	设置音量变化回调函数 .....	39
5.1.11.	设置SDK Filter通信数据回调函数.....	39
5.1.12.	设置录像快照通知回调函数 .....	40
5.1.13.	设置异步消息通知回调函数 .....	40
5.1.14.	设置视频呼叫通知回调函数 .....	41
5.1.15.	设置数据加密、解密回调函数 .....	42
5.1.16.	设置回调函数 .....	42
5.1.17.	释放SDK资源 .....	43
5.2.	登录流程 .....	43
5.2.1.	设置服务器认证密码 .....	43
5.2.2.	连接服务器 .....	44
5.2.3.	登录系统 .....	44
5.2.4.	视频呼叫控制 .....	45
5.2.5.	进入房间（根据房间编号） .....	47
5.2.6.	进入房间（根据房间名称） .....	48
5.2.7.	离开房间 .....	48

5.2.8.	注销系统 .....	49
5.3.	音视频操作 .....	49
5.3.1.	用户视频控制 .....	49
5.3.2.	用户语音控制 .....	50
5.3.3.	设置视频显示位置 .....	50
5.3.4.	重绘指定用户的视频 .....	51
5.3.5.	设置外部输入视频格式 .....	51
5.3.6.	外部视频数据输入 .....	52
5.3.7.	设置外部输入音频格式 .....	52
5.3.8.	外部音频数据输入 .....	53
5.4.	查询状态 .....	54
5.4.1.	查询摄像头的状态 .....	54
5.4.2.	查询用户音频设备采集状态 .....	54
5.4.3.	查询用户级别 .....	55
5.4.4.	查询用户昵称 .....	55
5.4.5.	查询用户状态 .....	56
5.4.6.	查询房间名称 .....	58
5.5.	普通功能 .....	59
5.5.1.	获取SDK版本信息 .....	59
5.5.2.	获取当前房间在线用户列表 .....	59
5.5.3.	传送文本消息 .....	60
5.5.4.	透明通道传送缓冲区 .....	60
5.5.5.	透明通道传送缓冲区扩展 .....	61
5.5.6.	传送文件 .....	63
5.5.7.	查询传输任务相关信息 .....	64
5.5.8.	取消传输任务 .....	64
5.5.9.	激活（关闭）SDK调用日志 .....	65
5.5.10.	发送SDK Filter通信数据 .....	65
5.5.11.	音视频录制 .....	66
5.5.12.	图像抓拍（拍照） .....	67
5.5.13.	组播功能控制 .....	68
5.6.	私聊函数 .....	69
5.6.1.	更改当前的聊天模式 .....	69
5.6.2.	获取指定用户聊天模式 .....	70
5.6.3.	请求与对方私聊 .....	70
5.6.4.	回复对方的私聊请求 .....	71
5.6.5.	回复对方的私聊请求（扩展） .....	71
5.6.6.	退出与某用户的私聊 .....	72
5.7.	好友列表 .....	72
5.7.1.	获取好友列表 .....	73
5.7.2.	获取好友在线状态 .....	73
5.7.3.	获取好友分组列表 .....	74
5.7.4.	获取分组名称 .....	74
5.7.5.	获取分组所对应的用户列表 .....	75

5.7.6.	获取好友用户信息 .....	75
5.7.7.	用户信息控制 .....	76
5.8.	好友列表 .....	76
5.8.1.	获取好友列表 .....	77
5.8.2.	获取好友在线状态 .....	77
5.8.3.	获取好友分组列表 .....	78
5.8.4.	获取分组名称 .....	78
5.8.5.	获取分组所对应的用户列表 .....	79
5.8.6.	获取好友用户信息 .....	79
5.8.7.	用户信息控制 .....	80
5.9.	系统设置 .....	80
5.9.1.	显示本地视频调节对话框 .....	80
5.9.2.	枚举本地视频采集设备 .....	81
5.9.3.	选择指定的视频采集设备 .....	81
5.9.4.	获取当前视频采集设备 .....	82
5.9.5.	枚举本地音频采集设备 .....	82
5.9.6.	选择指定的音频采集设备 .....	83
5.9.7.	获取当前音频采集设备 .....	83
5.9.8.	枚举本地音频播放设备 .....	84
5.9.9.	选择指定的音频播放设备 .....	84
5.9.10.	获取当前音频播放设备 .....	84
5.9.11.	获取音频设备的当前音量 .....	85
5.9.12.	设置指定音频设备的音量 .....	85
5.9.13.	SDK内核参数设置 .....	86
5.9.14.	SDK内核参数状态查询 .....	90
6.	附录一：错误代码参考 .....	92

# 1. 系统概述

非常感谢您使用佰锐科技的产品，我们将为您提供最好的服务。

本手册可能包含技术上不准确的地方或排版错误。本手册的内容将做定期的更新，恕不另行通知；更新的内容将会在本手册的新版本中加入。我们随时会改进或更新本手册中描述的产品或程序。

## 1.1 系统介绍

AnyChat 音视频互动开发平台（SDK）是一套跨平台的即时通讯解决方案，基于先进的 H.264 视频编码标准、AAC 音频编码标准与 P2P 技术，整合了佰锐科技在音视频编码、多媒体通讯领域领先的开发技术和丰富的产品经验而设计的高质量、宽适应性、分布式、模块化的网络音视频互动平台。

AnyChat 音视频互动开发平台（SDK）包含了音视频处理模块（采集、编解码）、流媒体管理模块（丢包重传、抖动平滑、动态缓冲）、流媒体播放模块（多路混音、音视频同步）以及 P2P 网络模块（NAT 穿透、UPnP 支持）等多个子模块，封装了底层的硬件操作（音视频采集、播放）、封装了流媒体处理（编解码、网络传输）等非常专业和复杂的技术，为上层应用提供简单的 API 控制接口，可以在极短的开发周期，以及极少的人力资源投入下为客户的现有平台增加音视频即时通讯、多方会议的功能。

AnyChat SDK 分为客户端 SDK 和服务器 SDK 两大部分，其中客户端 SDK 用于实现语音、视频的交互以及其它客户端相关的功能，而服务器 SDK 主要实现业务层逻辑控制，以及与第三方平台的互联等。客户端 SDK 和服务器 SDK 均支持 C++、C#、VB、Java 以及 Delphi 等开发语言。

## 1.2 系统特性

“AnyChat for Linux SDK”采用增强的 H.264 视频编码算法和 AMR 语音编码算法，具有高画质、语音清晰、流畅的特点，采用 P2P 技术进行网络传输，服务器采用 EPOLL 模型的重叠 IO，具有极高的并发处理能力。

“AnyChat for Linux SDK”采用优化的音频处理算法，支持回音消除（AEC）、噪音抑制（NS）、自动增益控制（AGC）、静音检测（VAD）等特性，大幅度提升语音的质量与用户体验。

服务器支持“SDK Filter Plus”和“AnyChat Server SDK”两种可扩展编程接口，可方便实现与其它系统进行集成，增强 AnyChat 的可扩展性。上层应用也可利用服务器 SDK 来实现更复杂的业务逻辑处理。

AnyChat 支持跨平台的应用，服务器支持 Windows、Linux、Unix 等所有主流服务器操作系统，客户端目前支持 Windows、桌面 Linux、嵌入式 Linux 以及 Web 浏览器、Android 平台、iOS 平台等。

### 1.2.1 视频技术

视频制式：PAL-B

分辨率：160×120 —1920×1080（可调节）

帧率：1~30（可调节）

视频编码器：H.264

视频流码率：10kbps ~ 2000kbps（VBR）

### 1.2.2 音频技术

采样率：16000 Hz、22050Hz、44100Hz、48000Hz（可设置）

量化值：16 bit

声道：Mono、Stereo

音频编码器：AMR\_WB、AAC、SPEEX

音频流码率：6kbps ~ 128kbps

音效处理：回音消除（AEC）、噪音抑制（NS）、自动增益控制（AGC）、静音检测（VAD）

### 1.2.3 P2P技术

传输方式：UDP、TCP

支持的 NAT 类型：

Cone NAPT  $\longleftrightarrow$  Cone NAPT

Cone NAPT  $\longleftrightarrow$  Symmetric NAT

支持 UPNP 协议

## 1.3 关于佰锐科技

广州佰锐网络科技有限公司（GuangZhou BaiRui Network Technology Co., Ltd.）是一家从事网络语音视频技术与系统开发的高新技术企业（广州市黄花岗科技园入园企业），企业注册资金 200 万元人民币，拥有独立知识产权。主要从事网络流媒体及相关平台的研发工作。专注于语音视频技术研发与创新。旗下有“中华视频网（<http://www.chinavideo.org>）”、“ffmpeg 工程组（<http://www.ffmpeg.com.cn>）”和“AnyChat 互动平台（<http://www.anychat.com.cn>）”和“Business Eyes（商务眼）（<http://www.shangwuyan.com>）”等多个网站。

佰锐科技始创于 2005 年 11 月，并于 2006 年启动国内第一家专注于视频技术的技术性网站：中华视频网，2006 年 3 月正式启动 AnyChat 项目。广州佰锐网络科技有限公司于 2006 年申请了两项发明专利（2006101239829、2006101241176）和两项软件著作权（软著登字第 066348 号、软著登字第 092142 号）。

## 1.4 技术支持

在您使用本 SDK 的过程中，遇到任何困难，请与我们联系，我们将热忱为您提供帮助。

您可以通过如下方式与我们取得联系：

- 1、在线论坛：<http://bbs.anychat.cn/>
- 2、官方网站：<http://www.anychat.cn>



- 3、知识中心: <http://www.anychat.cn/faq/>
- 4、电子邮件: [service@bairuitech.com](mailto:service@bairuitech.com)
- 5、24 小时客服电话: +86 (020) 85276986、38109065、38103410

## 1.5 版权申明

“AnyChat for Linux SDK”是由广州佰锐网络科技有限公司开发,拥有自主知识产权(软著登字第 066348 号)的系统平台,

广州佰锐网络科技有限公司拥有与本产品所用技术相关的知识产权。这些知识产权包括但不限于一项或多项发明专利或者正在进行申请的专利(2006101239829、2006101241176)。

本产品发行所依照的许可协议限制其使用、复制分发和反编译。未经广州佰锐网络科技有限公司事先书面授权,不得以任何形式或借助任何手段复制本产品的任何部分。

随本 SDK 一同发布的 Demo 演示程序源代码版权归广州佰锐网络科技有限公司所有。

AnyChat 是广州佰锐网络科技有限公司的商标。

*第三方组件: AnyChat 使用了开源的第三方组件, 包括: FFmpeg (LGPL 2.1)、libvpx (BSD)、libspeex (BSD)、WebRTC (BSD)。其中FFmpeg的版权所有者信息、源代码以及其它信息均可在其官方网站: <http://www.ffmpeg.org> 找到。*

## 2. 编程指南

### 2.1. 客户端SDK概述

“AnyChat for Linux SDK”的客户端组件（简称“客户端”）是由一系列的.so 动态库组合而成，这些组件分别完成不同的功能，最终所有的功能都集中在“libanychatcore.so”上，“libanychatcore.so”是一个动态连接库，可应用于 C++、Java 等开发环境。“AnyChat for Linux SDK”所提供的所有功能都由该插件集中处理。

系统采用模块化设计，每个模块都独立完成特定的任务，模块之间采用弱关联设计，今后系统某部分功能的升级，如音频、视频编码算法的改进，只需要替换相关的模块即可，不影响系统的接口。

AnyChat 客户端与服务器有一系列的交互过程，包括：连接服务器、登录系统、进入房间，交互过程的结果（如连接服务器是否成功）SDK 内部将会采用异步的机制通知上层应用。只有进入同一房间的两个用户之间才能进行语音、视频、文字的交互，当某用户打开了本地设备后，其它用户请求该用户的数据时，便能收到该用户的数据。

AnyChat 客户端在房间中，收到其它用户的流媒体数据后，可以由 SDK 来处理（如自动显示视频，自动播放声音），也可以由上层应用来处理，SDK 可以将解码之后的语音数据（PCM 类型）、视频数据（RGB 位图）等通过回调的方法传给上层应用，由上层应用来进行视频显示等。

## 2.2. 函数调用顺序

调用 顺序	函数	功能	备注
A	BRAC_SetSDKOption	设置 SDK 相关参数	初始化
	BRAC_InitSDK	初始化系统	
	BRAC_SetVideoDataCallBack	设置视频数据回调函数	
	BRAC_SetAudioDataCallBack	设置音频数据回调函数	
	BRAC_SetTextMessageCallBack	设置文字消息回调函数	
	BRAC_SetTransBufferCallBack	设置透明通道数据回调函数	
	BRAC_SetVolumeChangeCallBack	设置音量变化回调函数	
B	BRAC_Connect	连接服务器	进入系统
	BRAC_Login	用户登录系统	
	BRAC_EnterRoom	进入房间	
C	BRAC_GetOnlineUser	获取当前房间在线用户数	其它功能
	BRAC_GetCameraState	查询用户摄像头的状态	
	BRAC_GetSpeakState	查询用户发言状态	
	BRAC_ShowLVProperty	显示本地视频画面调节对话框	
	BRAC_EnumVideoCapture	枚举本地视频采集设备	
	BRAC_SelectVideoCapture	选择指定的视频采集设备	
	BRAC_GetCurVideoCapture	获取当前使用的视频采集设备	
	BRAC_EnumAudioCapture	枚举本地音频采集设备	
	BRAC_SelectAudioCapture	选择指定的音频采集设备	
	BRAC_GetCurAudioCapture	获取当前使用的音频采集设备	
	BRAC_AudioGetVolume	获取指定音频设备的当前音量	
	BRAC_AudioSetVolume	设置指定音频设备的音量	
	BRAC_SetVideoPos	设置视频显示位置	
	BRAC_UserCameraControl	操作用户视频	
	BRAC_UserSpeakControl	操作用户语音	
	BRAC_SendTextMessage	发送文本消息	
	BRAC_ChangeChatMode	更改当前的聊天模式	
	BRAC_GetUserChatMode	获取指定用户当前的聊天模式	
	BRAC_PrivateChatRequest	请求与对方私聊，发起私聊请求	
	BRAC_PrivateChatEcho	回复对方的私聊请求	
	BRAC_PrivateChatExit	退出与某用户的私聊	
	.....	.....	
D	BRAC_LeaveRoom	离开房间	退出系统
	BRAC_Logout	用户注销	
	BRAC_Release	释放资源	

## 2.3. 服务器SDK概述

“AnyChat Server SDK”与“SDK Filter Plus”均是服务器扩展编程接口，均为动态连接库（DLL）形式，两者的主要区别是：（1）、“SDK Filter Plus”的 DLL 被 AnyChat 核心服务器程序（AnyChatCoreServer.exe）调用，与 AnyChat 核心服务器程序属同一个进程；（2）、“AnyChat Server SDK”被业务层服务器程序（需要用户编写）调用，与 AnyChat 核心服务器程序属不同的进程，与 AnyChat 核心服务器采用 IPC 的方式进行通信。

“AnyChat Server SDK”与“SDK Filter Plus”两者可以实现相同的功能，通常来说，“SDK Filter Plus”适合业务逻辑较简单的应用，而“AnyChat Server SDK”则适合业务逻辑较复杂的应用，实现独立的业务层服务器，有对应的界面显示。

有关“SDK Filter Plus”的详细介绍可参考相关的开发文档（《AnyChat SDK Filter Plus 开发指南》）及 SDK 包中所附带的相关源代码。

有关“AnyChat Server SDK”的详细介绍可参考相关的开发文档（《AnyChat Server SDK 开发指南》）及 SDK 包中所附带的相关源代码。

有关AnyChat平台用户身份验证与第三方平台集成的问题可参考技术论坛相关介绍：<http://bbs.anychat.cn/forum.php?mod=viewthread&tid=12&extra=page%3D1>

服务器与客户端之间可以传输缓冲区、文件等数据，详情可参考在线文档：<http://www.anychat.cn/faq/index.php?action=artikel&cat=2&id=206&artlang=zh>

## 3. 数据结构及常量定义

### 3.1. 视频图像格式

```
enum BRAC_PixelFormat{
    BRAC_PIX_FMT_RGB24 = 0,    // Packed RGB 8:8:8, 24bpp, RGBRGB...
    BRAC_PIX_FMT_RGB32,        // Packed RGB 8:8:8, 32bpp, (msb)8A 8R 8G 8B(lsb), in cpu endianness
    BRAC_PIX_FMT_YV12,         // Planar YUV 4:2:0, 12bpp, (1 Cr & Cb sample per 2x2 Y samples)
    BRAC_PIX_FMT_YUY2,         // Packed YUV 4:2:2, 16bpp, Y0 Cb Y1 Cr
    BRAC_PIX_FMT_YUV420P,      ///< Planar YUV 4:2:0, 12bpp, (1 Cr & Cb sample per 2x2 Y samples)
};
```

该图像格式将会在请求视频数据回调函数中使用，表示上层需要的视频格式，如果从摄像头原始采集的格式与所请求的格式不同，则 SDK 将会自动进行转换，然后再回调。

### 3.2. 音频设备定义

```
enum BRAC_AudioDevice{
    BRAC_AD_WAVEIN = 0,        // 输入设备：Mic
    BRAC_AD_WAVEOUT,           // 输出设备：Wave
};
```

该设备定义将会在操作音频相关的方法中使用，表示操作的是音频输入设备（Mic），还是音频输出设备（Wave），另外在音量更改回调函数中也有一个这样的参数，表示是输入设备的音量发生了变化，还是输出设备的音量发生了变化。

### 3.3. 功能模式定义

API: BRAC\_InitSDK 传入参数，用于确定 SDK 的一些功能是否生效。

```
#define BRAC_FUNC_VIDEO_CBDATA      0x00000001L    ///< 通过回调函数输出视频数据
#define BRAC_FUNC_VIDEO_AUTODISP    0x00000002L    ///< 由SDK包处理视频,将视频显示在指定的窗口上
#define BRAC_FUNC_AUDIO_CBDATA      0x00000004L    ///< 通过回调函数输出音频数据
#define BRAC_FUNC_AUDIO_AUTOPLAY    0x00000008L    ///< 由SDK包处理音频,直接播放
#define BRAC_FUNC_CONFIG_LOCALINI    0x00000010L    ///< 生成本地配置文件（AnyChatSDK.ini）
#define BRAC_FUNC_FIREWALL_OPEN      0x00000020L    ///< 允许SDK操作Windows防火墙，将当前应用程序加入防火墙访问列表（避免Windows提示用户是否阻止当前应用程序）
```

```

#define BRAC_FUNC_CHKDEPENDMODULE 0x00000040L ///< 自动检查SDK所依赖的组件，并自动注册
#define BRAC_FUNC_AUDIO_VOLUME CALC 0x00000080L ///< 由SDK自动计算语音的音量
#define BRAC_FUNC_AUDIO_AUTOVOLUME 0x00000100L ///< 允许SDK自动控制Mic录音音量
#define BRAC_FUNC_NET_SUPPORTUPNP 0x00000200L ///< 允许SDK打开用户网络中的UPNP设备，如果
    用户的路由器或是防火墙支持UPNP协议，则可提高P2P打洞的成功率
#define BRAC_FUNC_DISABLEDECODE 0x00000400L ///< 禁止对收到的数据进行解码和播放，为了
    提高代理客户端的数据转发性能，可设置该标志，否则不能设置该标志
#define BRAC_FUNC_MAINTHREADCB 0x00000800L ///< 主线程进行回调操作，默认是多线程环境
    下的数据回调
#define BRAC_FUNC_AUDIO_FORBIDCFHW 0x00001000L ///< 禁止修改音频硬件配置
#define BRAC_FUNC_CORE_FORBIDWINMSG 0x00002000L ///< 禁止使用windows消息循环
#define BRAC_FUNC_AUDIO_LARGEBUFFER 0x00004000L ///< 音频大缓冲区模式，适合音乐播放类应用
#define BRAC_FUNC_NET_LARGEDELAY 0x00010000L ///< 网络高延迟模式，适用于卫星网络环境

```

### 3.4. SDK内核参数定义

在使用 API “BRAC\_SetSDKOption”、“BRAC\_GetSDKOption” 来设置、获取 SDK 内核心参数时，可根据不同的参数类型设置、获取不同的值，目前可用参数类型定义如下：

```

#define BRAC_SO_AUDIO_VADCTRL 1 ///< 音频静音检测控制（参数为：int型：打开，关闭）
#define BRAC_SO_AUDIO_NSCTRL 2 ///< 音频噪音抑制控制（参数为：int型：打开，关闭）
#define BRAC_SO_AUDIO_ECHOCTRL 3 ///< 音频回音消除控制（参数为：int型：打开，关闭）
#define BRAC_SO_AUDIO_AGCCTRL 4 ///< 音频自动增益控制（参数为：int型：打开，关闭）
#define BRAC_SO_AUDIO_CAPTUREMODE 5 ///< 音频采集模式设置（参数为：int型：发言模式，放
    歌模式，卡拉OK模式，线路输入模式）
#define BRAC_SO_AUDIO_MICBOOST 6 ///< 音频采集Mic增强控制（参数为：int型：取消，选中，
    设备不存在[查询时返回值]）
#define BRAC_SO_AUDIO_AUTOPARAM 7 ///< 根据音频采集模式，自动选择合适的相关参数，包括
    编码器、采样参数、码率参数等（参数为int型：启用，关闭[默认]）
#define BRAC_SO_AUDIO_MONOBITRATE 8 ///< 设置单声道模式下音频编码目标码率（参数为：int
    型，单位：bps）
#define BRAC_SO_AUDIO_STEREOBITRATE 9 ///< 设置双声道模式下音频编码目标码率（参数为：int
    型，单位：bps）
#define BRAC_SO_AUDIO_PLAYDRVCTRL 70 ///< 音频播放驱动选择（参数为：int型，默认驱动，1
    DSound驱动，2 WaveOut驱动，3 Java播放[Android平台使用]）
#define BRAC_SO_AUDIO_CNGCTRL 71 ///< 舒适噪音生成控制（参数为：int型：打开，关闭）
#define BRAC_SO_AUDIO_CODECID 72 ///< 本地音频编码器ID设置（参数为int型，-1表示默认，
    如果设置的编码器ID不存在，则内核会采用默认的编码器）
#define BRAC_SO_AUDIO_SOFTVOLMODE 73 ///< 设置软件音量模式控制（参数为int型，打开[默认]，
    关闭），使用软件音量模式，将不会改变系统的音量设置
#define BRAC_SO_AUDIO_RECORDDRVCTRL 74 ///< 音频采集驱动控制（参数为int型，默认驱动，1
    DSound驱动，2 WaveIn驱动，3 Java采集[Android平台使用]）

```

```

#define BRAC_SO_AUDIO_ECHODELAY      75  ///< 音频回声消除延迟参数设置（参数为int型，单位：
ms）
#define BRAC_SO_RECORD_VIDEOBR       10  ///< 录像视频码率设置（参数为：int型，单位：bps）
#define BRAC_SO_RECORD_AUDIOBR       11  ///< 录像音频码率设置（参数为：int型，单位：bps）
#define BRAC_SO_RECORD_TMPDIR        12  ///< 录像文件临时目录设置（参数为字符串TCHAR类型，
必须是完整的绝对路径）
#define BRAC_SO_SNAPSHOT_TMPDIR       13  ///< 快照文件临时目录设置（参数为字符串TCHAR类型，
必须是完整的绝对路径）
#define BRAC_SO_CORESDK_TMPDIR        14  ///< 设置AnyChat Core SDK临时目录（参数为字符串TCHAR
类型，必须是完整的绝对路径）
#define BRAC_SO_CORESDK_LOADCODEC     16  ///< 加载外部编解码器（参数为字符串TCHAR类型，必须
是完整的绝对路径，包含文件名，或包含文件名的绝对路径）
#define BRAC_SO_CORESDK_USEARMV6LIB   17  ///< 强制使用ARMv6指令集的库，android平台使用（参数
为：int型，使用ARMv6指令集，0内核自动判断[默认]）
#define BRAC_SO_CORESDK_USEHWCODEC    18  ///< 使用平台内置硬件编解码器（参数为int型，不使用
硬件编解码器[默认] 1 使用内置硬件编解码器）
#define BRAC_SO_CORESDK_PATH          20  ///< 设置AnyChat Core SDK相关组件路径（参数为字符串
TCHAR类型，必须是完整的绝对路径）
#define BRAC_SO_CORESDK_DUMP_COREINFO 21  ///< 输出内核信息到日志文件中，便于分析故障原因（参
数为：int型：输出）
#define BRAC_SO_CORESDK_MAINVERSION   22  ///< 查询SDK主版本号（参数为int型）
#define BRAC_SO_CORESDK_SUBVERSION    23  ///< 查询SDK从版本号（参数为int型）
#define BRAC_SO_CORESDK_BUILDTIME     24  ///< 查询SDK编译时间（参数为字符串TCHAR类型）
#define BRAC_SO_CORESDK_ACTIVESTATE    25  ///< 应用程序活动状态控制（参数为int型，1 应用程序
处于活动状态，0 应用程序处于非活动状态），适用于iPhone等设备程序可后台运行的场合
#define BRAC_SO_CORESDK_EXTVIDEOINPUT 26  ///< 外部扩展视频输入控制（参数为int型，0 关闭外部
视频输入[默认]，1 启用外部视频输入）
#define BRAC_SO_CORESDK_EXTAUDIOINPUT 27  ///< 外部扩展音频输入控制（参数为int型，0 关闭外部
音频输入[默认]，1 启用外部音频输入）
#define BRAC_SO_CORESDK_LOWDELAYCTRL   28  ///< 低延迟模式控制（参数为int型，关闭低延迟模式[默
认]，1 启用低延迟模式）

#define BRAC_SO_LOCALVIDEO_BITRATECTRL 30  ///< 本地视频编码码率设置（参数为int型，单位bps，同
服务器配置：VideoBitrate）
#define BRAC_SO_LOCALVIDEO_QUALITYCTRL 31  ///< 本地视频编码质量因子控制（参数为int型，同服务
器配置：VideoQuality）
#define BRAC_SO_LOCALVIDEO_GOPCTRL     32  ///< 本地视频编码关键帧间隔控制（参数为int型，同服
务器配置：VideoGOPSize）
#define BRAC_SO_LOCALVIDEO_FPSCTRL     33  ///< 本地视频编码帧率控制（参数为int型，同服务器配
置：VideoFps）
#define BRAC_SO_LOCALVIDEO_PRESETCTRL  34  ///< 本地视频编码预设参数控制（参数为int型，-5）
#define BRAC_SO_LOCALVIDEO_APPLYPARAM  35  ///< 应用本地视频编码参数，使得前述修改即时生效（参
数为int型：使用新参数，使用默认参数）
#define BRAC_SO_LOCALVIDEO_VIDEOSIZEPOLITIC 36  ///< 本地视频采集分辨率控制策略（参数为int

```



型，自动向下逐级匹配[默认]；使用采集设备默认分辨率），当配置的分辨率不被采集设备支持时有效

`#define BRAC_SO_LOCALVIDEO_DEINTERLACE 37` ///< 本地视频反交织参数控制（参数为int型：0 不进行反交织处理[默认]；反交织处理），当输入视频源是隔行扫描源（如电视信号）时通过反交织处理可以提高画面质量

`#define BRAC_SO_LOCALVIDEO_WIDTHCTRL 38` ///< 本地视频采集分辨率宽度控制（参数为int型，同服务器配置：VideoWidth）

`#define BRAC_SO_LOCALVIDEO_HEIGHTCTRL 39` ///< 本地视频采集分辨率高度控制（参数为int型，同服务器配置：VideoHeight）

`#define BRAC_SO_LOCALVIDEO_FOCUSCTRL 90` ///< 本地视频摄像头对焦控制（参数为int型，表示自动对焦，0表示手动对焦）

`#define BRAC_SO_LOCALVIDEO_PIXFMTCTRL 91` ///< 本地视频采集优先格式控制（参数为int型，-1表示智能匹配，否则优先采用指定格式，参考：BRAC\_PixelFormat）

`#define BRAC_SO_LOCALVIDEO_OVERLAY 92` ///< 本地视频采用Overlay模式（参数为int型，表示采用Overlay模式，0表示普通模式[默认]）

`#define BRAC_SO_LOCALVIDEO_CODECID 93` ///< 本地视频编码器ID设置（参数为int型，-1表示默认，如果设置的编码器ID不存在，则内核会采用默认的编码器）

`#define BRAC_SO_LOCALVIDEO_ROTATECTRL 94` ///< 本地视频旋转控制（参数为int型，表示不进行旋转，表示垂直翻转）

`#define BRAC_SO_LOCALVIDEO_CAPDRIVER 95` ///< 本地视频采集驱动设置（参数为int型，表示自动选择[默认]，1 Video4Linux，2 DirectShow，3 Java采集[Android平台使用]）

`#define BRAC_SO_LOCALVIDEO_FIXCOLORDEVIA 96` ///< 修正视频采集颜色偏色（参数为int型，表示关闭[默认]，开启）

`#define BRAC_SO_LOCALVIDEO_ORIENTATION 97` ///< 本地视频设备方向（参数为：int型，定义为常量：ANYCHAT\_DEVICEORIENTATION\_XXXX）

`#define BRAC_SO_LOCALVIDEO_AUTOROTATION 98` ///< 本地视频自动旋转控制（参数为int型，0表示关闭，1 开启[默认]，视频旋转时需要参考本地视频设备方向参数）

`#define BRAC_SO_LOCALVIDEO_SURFACEROTATION 99` ///< 设置本地视频预览显示旋转角度（参数为int型，角度）

`#define BRAC_SO_LOCALVIDEO_CAMERAFADE 100` ///< 本地摄像头方向（前置、后置）

`#define BRAC_SO_LOCALVIDEO_CAMERAROTATION 101` ///< 本地摄像头旋转角度

`#define BRAC_SO_LOCALVIDEO_DEVICEROTATION 102` ///< 设备旋转角度

`#define BRAC_SO_LOCALVIDEO_DEVICEMODE 103` ///< 设备类型

`#define BRAC_SO_NETWORK_P2PPOLITIC 40` ///< 本地网络P2P策略控制（参数为：int型：禁止本地P2P，服务器控制P2P[默认]，上层应用控制P2P连接，按需建立P2P连接）

`#define BRAC_SO_NETWORK_P2PCONNECT 41` ///< 尝试与指定用户建立P2P连接（参数为int型，表示目标用户ID），连接建立成功后，会通过消息反馈给上层应用，P2P控制策略=2时有效

`#define BRAC_SO_NETWORK_P2PBREAK 42` ///< 断开与指定用户的P2P连接（参数为int型，表示目标用户ID）[暂不支持，保留]

`#define BRAC_SO_NETWORK_TCPSERVICEPORT 43` ///< 设置本地TCP服务端口（参数为int型），连接服务器之前设置有效

`#define BRAC_SO_NETWORK_UDPSERVICEPORT 44` ///< 设置本地UDP服务端口（参数为int型），连接服务器之前设置有效

`#define BRAC_SO_NETWORK_MULTICASTPOLITIC 45` ///< 组播策略控制（参数为int型：执行服务器路由



策略，禁止组播发送[默认]，1 忽略服务器路由策略，只向组播组广播媒体流，2 执行服务器路由策略，同时组播)

```
#define BRAC_SO_NETWORK_TRANSBUFMAXBITRATE 46 ///< 传输缓冲区、文件最大码率控制(参数为int型,
不限制, 以最快速率传输[默认], 则表示限制码率, 单位为: bps)
#define BRAC_SO_PROXY_FUNCTIONCTRL          50 ///< 本地用户代理功能控制, (参数为: int型, 启
动代理, 关闭代理[默认])
#define BRAC_SO_PROXY_VIDEOCTRL             51 ///< 本地用户代理视频控制, 将本地视频变为指定
用户的视频对外发布(参数为int型, 表示其它用户的userid)
#define BRAC_SO_PROXY_AUDIOCTRL             52 ///< 本地用户代理音频控制, 将本地音频变为指定
用户的音频对外发布(参数同BRAC_SO_PROXY_VIDEOCTRL)
#define BRAC_SO_STREAM_MAXBUFFERTIME        60 ///< 最大流缓冲时间(参数为int型, 单位: 毫秒,
取值范围: ~ 5000, 默认: ), 发言模式设置值, 歌曲模式会自动增加一倍
#define BRAC_SO_STREAM_SMOOTHPLAYMODE       61 ///< 平滑播放模式(参数为int型, 关闭[默认], 1 打
开), 打开状态下遇到视频丢帧时会继续播放(可能出现马赛克), 不会卡住
#define BRAC_SO_VIDEOSHOW_DRIVERCTRL         83 ///< 视频显示驱动控制(参数为: int型, 默认驱动,
1 Windows DirectShow, Windows GDI, SDL, 4 Android2X, 5 Android Java)
#define BRAC_SO_VIDEOSHOW_GPUDIRECTRENDER 84 ///< 视频数据经过GPU直接渲染, 将解码后的视频数
据直接传输到GPU的物理地址(参数为: int型, 0 关闭[默认], 1 打开), 与硬件平台相关
#define BRAC_SO_CORESDK_TICKOUTUSER         110 ///< 从服务器上踢掉指定用户(参数为int型, 表示
目标用户ID)
#define BRAC_SO_CORESDK_DEVICEMODE          130 ///< 设备模式控制(局域网设备之间可以互相通信,
不依赖服务器; 参数为 int 型, 关闭[默认], 开启)
```

## 3.5. 用户状态标志定义

在使用 API “BRAC\_QueryUserState” 查询用户的相关状态时，可根据不同的状态标志来返回不同的值，目前支持的用户状态类型定义如下：

```
#define BRAC_USERSTATE_CAMERA      1    ///< 用户摄像头状态（参数为DWORD型）
#define BRAC_USERSTATE_HOLDMIC    2    ///< 用户音频设备状态（参数为DWORD型，返回值：音频采集关闭，1 音频采集开启）
#define BRAC_USERSTATE_SPEAKVOLUME 3    ///< 用户当前说话音量（参数为DOUBLE类型(.0 ~ 100.0)）
#define BRAC_USERSTATE_RECORDING  4    ///< 用户录像（音）状态（参数为DWORD型）
#define BRAC_USERSTATE_LEVEL      5    ///< 用户级别（参数为DWORD型）
#define BRAC_USERSTATE_NICKNAME   6    ///< 用户昵称（参数为字符串TCHAR类型）
#define BRAC_USERSTATE_LOCALIP    7    ///< 用户本地IP地址（内网，参数为字符串TCHAR类型）
#define BRAC_USERSTATE_INTERNETIP 8    ///< 用户互联网IP地址（参数为字符串TCHAR类型）
#define BRAC_USERSTATE_VIDEOBITRATE 9  ///< 用户当前的视频码率（参数为DWORD类型，Bps）
#define BRAC_USERSTATE_AUDIOBITRATE 10  ///< 用户当前的音频码率（参数为DWORD类型，Bps）
#define BRAC_USERSTATE_P2PCONNECT 11   ///< 查询本地用户与指定用户的当前P2P连接状态（参数为DWORD类型，返回值：P2P不通，1 P2P连接成功[TCP]，P2P连接成功[UDP]，P2P连接成功[TCP、UDP]）
#define BRAC_USERSTATE_NETWORKSTATUS 12 ///< 查询指定用户的网络状态（参数为DWORD类型，返回值：优良，较好，一般，较差，非常差），注：查询间隔需要>1s
#define BRAC_USERSTATE_VIDEOSIZE   13   ///< 查询指定用户的视频分辨率（参数为DWORD类型，返回值：低位表示宽度，高位表示高度）
#define BRAC_USERSTATE_PACKLOSSRATE 14   ///< 查询指定用户的网络流媒体数据丢包率（参数为DWORD类型，返回值：- 100，如：返回值为，表示丢包率为%）
#define BRAC_USERSTATE_DEVICETYPE  15   ///< 查询指定用户的终端类型（参数为DWORD类型，返回值：Unknow, Windows, Android, iOS, Web, Linux, Mac, Win Phone, WinCE）
#define BRAC_USERSTATE_SELFUSERSTATUS 16 ///< 查询本地用户的当前状态（参数为DWORD类型，返回值：Unknow, Connected, LoggedIn, In Room, Logouted, Link Closed）
#define BRAC_USERSTATE_SELFUSERID  17   ///< 查询本地用户的 ID（参数为 DWORD 类型，若用户登录成功，返回用户实际的 userid，否则返回-1）
```

## 3.6. 视频呼叫事件定义

API: BRAC\_VideoCallControl 传入参数、VideoCallEvent 回调参数：

```
#define BRAC_VIDEOCALL_EVENT_REQUEST 1    ///< 呼叫请求
#define BRAC_VIDEOCALL_EVENT_REPLY   2    ///< 呼叫请求回复
#define BRAC_VIDEOCALL_EVENT_START   3    ///< 视频呼叫会话开始事件
#define BRAC_VIDEOCALL_EVENT_FINISH  4    ///< 挂断（结束）呼叫会话
```

## 4. 回调函数

SDK 通过回调函数实现与上层应用的状态更新和数据交互。总体原则是：数据的交互通过回调函数返回给上层应用；

在实际的开发过程中，可根据具体情况来设置回调函数。

### 4.1. 异步事件

“AnyChat for Linux SDK”采用异步事件通知的方式来告知系统的各种状态变化。

#### 4.1.1. 事件预定义

#define WM_GV	WM_USER + 200
#define WM_GV_CONNECT	WM_GV + 1
#define WM_GV_LOGINSYSTEM	WM_GV + 2
#define WM_GV_ENTERROOM	WM_GV + 3
#define WM_GV_MICSTATECHANGE	WM_GV + 4
#define WM_GV_USERATROOM	WM_GV + 5
#define WM_GV_LINKCLOSE	WM_GV + 6
#define WM_GV_ONLINEUSER	WM_GV + 7
#define WM_GV_FORTUNEMENU	WM_GV + 8
#define WM_GV_CAMERASTATE	WM_GV + 11
#define WM_GV_CHATMODECHG	WM_GV + 12
#define WM_GV_ACTIVESTATE	WM_GV + 13
#define WM_GV_P2PCONNECTSTATE	WM_GV + 14
#define WM_GV_VIDEOSIZECHG	WM_GV + 15
#define WM_GV_USERINFOUPDATE	WM_GV + 16
#define WM_GV_FRIENDSTATUS	WM_GV + 17
#define WM_GV_PRIVATEREQUEST	WM_GV + 21

```
#define WM_GV_PRIVATEECHO          WM_GV + 22
#define WM_GV_PRIVATEEXIT          WM_GV + 23
#define WM_GV_SDKWARNING           WM_GV + 41
```

### 4.1.2. 网络连接事件

名称: WM\_GV\_CONNECT

参数 (WPARAM): 表示是否连接成功, BOOL 类型;

参数 (LPARAM): 出错代码, 0 表示连接成功;

说明: 当客户端连接服务器时被触发。

### 4.1.3. 登录系统事件

名称: WM\_GV\_LOGINSYSTEM

参数 (WPARAM): 表示自己的用户 ID 号, DWORD 类型;

参数 (LPARAM): 表示登录结果, DWORD 类型, 0 成功, 否则为出错代码,  
参考出错代码定义

说明: 当客户端登录系统时被触发。

### 4.1.4. 自己进入房间事件

名称: WM\_GV\_ENTERROOM

参数 (WPARAM): 表示所进入房间的 ID 号, DWORD 类型;

参数 (LPARAM): 表示是否进入房间: 0 成功进入, 否则为出错代码, 参考出错  
代码定义

说明: 当客户端进入房间时被触发。

### 4.1.5. 音频设备状态变化事件

名称: WM\_GV\_MICSTATECHANGE

参数 (WPARAM): 表示用户 ID 号, DWORD 类型;

参数 (LPARAM): 表示该用户的音频设备状态改变, BOOL 类型;

0 用户关闭音频采集设备

1 用户开启音频采集设备

**说明：**当客户端调用“BRAC\_UserSpeakControl”API 对本地的音频设备进行操作时，将触发该消息。

### 4.1.6. 用户进入（离开）房间事件

**名称：**WM\_GV\_USERATROOM

**参数（WPARAM）：**表示用户 ID 号，DWORD 类型；

**参数（LPARAM）：**进入状态标志，BOOL 类型，表示该用户是进入（TRUE）或离开（FALSE）房间；

**说明：**当其他用户进入或离开房间时被触发。

### 4.1.7. 网络连接关闭事件

**名称：**WM\_GV\_LINKCLOSE

**参数（WPARAM）：**表示连接断开的原因，DWORD 类型；

**参数（LPARAM）：**保留；

**说明：**该消息只有在客户端连接服务器成功之后，网络异常中断之时触发，如果是本地用户主动注销用户、释放资源，则不会触发该消息。

网络断开的原因可能是如下值：

- 0 正常断开，如网络中断，服务器被关闭等
- 1 该用户在其它计算机上登录，之前的连接被服务器断开
- 2 服务器功能受限制（演示模式下不允许长时间连接服务器）
- 3 接收到来自其它网络的连接，连接被主动关闭
- 4 客户端程序版本太旧，不允许连接
- 5 连接服务器认证失败（服务器设置了认证密码）
- 6 网络连接超时，服务器主动断开；
- 7 服务器不支持嵌入式设备连接，或连接设备超过授权设备数；

### 4.1.8. 当前房间在线用户事件

名称: WM\_GV\_ONLINEUSER

参数 (WPARAM): 表示在线用户数 (包含自己);

参数 (LPARAM): 表示房间 ID;

说明: 收到当前房间的在线用户信息, 进入房间后触发一次。需要注意的是, 在进入房间之前, 房间内的用户是通过该消息反馈给上层应用的, 上层应用收到该消息后, 可以通过调用相关的方法来获取当前房间在线用户列表, 而自己进入房间之后, 有新的用户进入, 或是有用户离开时, 是通过 WM\_GV\_USERATROOM 消息通知上层应用的。

### 4.1.9. 用户摄像头状态变化事件

名称: WM\_GV\_CAMERASTATE

参数 (WPARAM): 表示用户 ID 号;

参数 (LPARAM): 表示摄像头的当前状态;

说明: 当用户摄像头状态变化时被触发, 摄像头的状态定义为:

GV_CAMERA_STATE_NULL:	0	没有摄像头,
GV_CAMERA_STATE_NORMAL:	1	有摄像头但是没有打开,
GV_CAMERA_STATE_OPEN :	2	摄像头已打开。

### 4.1.10. 用户聊天模式变化事件

名称: WM\_GV\_CHATMODECHG

参数 (WPARAM): 表示用户 ID 号;

参数 (LPARAM): 表示用户的当前聊天模式;

说明: 当用户聊天模式发生变化时被触发, 聊天模式定义为:

GV_CHATMODE_PUBLIC:	0	公聊模式,
GV_CHATMODE_PRIVATE:	1	私聊模式。

### 4.1.11. 用户活动状态变化事件

名称: WM\_GV\_ACTIVESTATE

参数 (WPARAM): 表示用户 ID 号;

参数 (LPARAM): 表示用户的当前活动状态;

说明: 当用户活动状态发生变化时被触发, 活动状态定义为:

GV_ACTIVE_STATE_NORMAL	0	正常
GV_ACTIVE_STATE_BUSY	1	繁忙
GV_ACTIVE_STATE_AWAY	2	离开

### 4.1.12. P2P连接状态变化事件

名称: WM\_GV\_P2PCONNECTSTATE

参数 (WPARAM): 表示其它用户 ID 号;

参数 (LPARAM): 表示本地用户与其它用户的当前 P2P 网络连接状态;

说明: 当本地用户与其它用户的 P2P 网络连接状态发生变化时被触发, P2P 连接状态定义为:

0	没有任何连接
1	P2P 连接成功, TCP 连接
2	P2P 连接成功, UDP 连接
3	P2P 连接成功, TCP 与 UDP 连接

### 4.1.13. 用户视频分辨率发生变化事件

名称: WM\_GV\_VIDEOSIZECHG

参数 (WPARAM): 用户 ID 号;

参数 (LPARAM): 表示用户当前的视频分辨率;

说明: 当用户进入房间, 或是改变视频采集的分辨率时, 该用户自身以及房间内的其它用户都将被触发该消息, 上层应用程序收到该消息后, 可调节视频显示区域的大小, 适得显示区域的大小与原始视频分辨率成正比, 防止视频显示变形;

参数 LPARAM 是一个组合值，低 16 位表示宽度，高 16 位表示高度。

上层应用程序可通过“BRAC\_QueryUserState”接口随时查询指定用户的视频分辨率。

#### 4.1.14. 用户信息更新通知消息

名称: WM\_GV\_USERINFOUPDATE

参数 (WPARAM): 用户 ID 号;

参数 (LPARAM): 表示用户信息更新类别;

说明: 当服务器更新用户信息之后, 将会触发该消息, 通知客户端某个用户的信息已被更新, 客户端可以通过 API 接口获取更新后的信息显示在界面上:

#### 4.1.15. 好友在线状态变化消息

名称: WM\_GV\_FRIENDSTATUS

参数 (WPARAM): 用户 ID 号;

参数 (LPARAM): 表示用户的当前活动状态: 0 离线, 1 上线;

说明: 由服务器通知客户端, 当好友上线, 或是离线时将会收到通知。

#### 4.1.16. 用户发起私聊请求事件

名称: WM\_GV\_PRIVATEREQUEST

参数 (WPARAM): 表示发起者的用户 ID 号;

参数 (LPARAM): 表示私聊请求编号, 标识该请求;

说明: 当用户发起私聊请求时, 在被请求者端触发。

#### 4.1.17. 用户回复私聊请求事件

名称: WM\_GV\_PRIVATEECHO

参数 (WPARAM): 表示回复者的用户 ID 号;

参数 (LPARAM): 表示返回代码;

说明: 当用户回复私聊请求时被触发, 如果返回代码为 GV\_ERR\_SUCCESS (0)



时，表示对方已同意私聊请求，否则表示拒绝请求。

### 4.1.18. 用户退出私聊事件

名称：WM\_GV\_PRIVATEEXIT

参数（WPARAM）：表示退出者的用户 ID 号；

参数（LPARAM）：表示返回代码；

说明：当用户退出私聊时被触发，该用户退出之后，不会影响当前与其它用户的私聊。

### 4.1.19. SDK警告事件

名称：WM\_GV\_SDKWARNING

参数（WPARAM）：表示警告代码，定义为：GV\_ERR\_WARNING\_XXXX；

参数（LPARAM）：备用；

说明：当 SDK 在运行过程中自检发现异常状态时，将向上层发送该消息，警告代码可参考：GVErrorCodeDefine.h 中的相关定义。常见警告代码定义如下：

```
800    ///< 与服务器的 UDP 通信异常，流媒体服务将不能正常工作
801    ///< SDK 加载 brMiscUtil.dll 动态库失败，部分功能将失效
802    ///< SDK 加载 brMediaUtil.dll 动态库失败，部分功能将失效
```

## 4.2. 回调函数定义

通过回调函数实现 SDK 底层的数据实时的反馈给上层应用。回调函数的最后一个参数均为：LPVOID lpUserValue，可供用户自定义，默认为 NULL，通常上层应用可传递一个对象的句柄（指针），在回调函数体里面再通过强制类型转换，变为上层应用可用的对象句柄（指针），详细用法可参考示例程序中回调函数相关部分。

### 4.2.1. 视频数据回调函数

```
typedef void (CALLBACK * BRAC_VideoData_CallBack)(DWORD dwUserId, LPVOID lpBuf, DWORD dwLen, BITMAPINFOHEADER bmiHeader, LPVOID lpUserValue);
```

**参数：**

**dwUserId:** 用户 ID，指示该视频数据属于哪一个用户

**lpBuf:** 视频数据缓冲区

**dwLen** 缓冲区大小

**bmiHeader:** 视频缓冲区相关参数，包含了该视频数据相关的信息，如视频的大小、颜色深度等

**lpUserValue:** 用户自定义参数，在设置回调函数时传入

**备注：**

当收到其它用户发送的视频数据时，将会触发该回调函数。同时收到本地的视频采集数据时，也将会触发该回调函数。

收到视频数据并触发回调函数的前提是：

- 1、在 SDK 初始化时，设置视频数据回调标志，参考：BRAC\_InitSDK
- 2、调用 API 打开自己或请求对方的视频数据，参考：BRAC\_UserCameraControl

### 4.2.2. 视频数据扩展回调函数

```
typedef void (CALLBACK * BRAC_VideoDataEx_CallBack)(DWORD dwUserId, LPVOID lpBuf, DWORD dwLen, BITMAPINFOHEADER bmiHeader, DWORD dwTimeStamp, LPVOID lpUserValue);
```

**参数：**

**dwUserId:** 用户 ID，指示该视频数据属于哪一个用户

**lpBuf:** 视频数据缓冲区

**dwLen** 缓冲区大小

**bmiHeader:** 视频缓冲区相关参数，包含了该视频数据相关的信息，如视频的大小、颜色深度等

**dwTimeStamp** 视频帧时间戳，单位：ms，用于音视频同步

**lpUserValue:** 用户自定义参数，在设置回调函数时传入

**备注：**

V4.2 版本新增，与“视频数据回调函数”相比，增加了视频时间戳参数。

### 4.2.3. 音频数据回调函数

```
typedef void (CALLBACK * BRAC_AudioData_Callback)(DWORD dwUserId, LPVOID lpBuf, DWORD dwLen, WAVEFORMATEX waveFormatEx, LPVOID lpUserValue);
```

#### 参数：

**dwUserId:** 用户 ID，指示该音频数据属于哪一个用户

**lpBuf:** 音频数据缓冲区

**dwLen:** 缓冲区大小

**waveFormatEx:** 音频数据相关信息，如音频的采样频率、量化位数通道数等

**lpUserValue:** 用户自定义参数，在设置回调函数时传入

#### 备注：

当收到其它用户发送的音频数据时，将会触发该回调函数。同时收到本地的音频采集数据时，也将会触发该回调函数。

收到音频数据并触发回调函数的前提是：

- 1、在 SDK 初始化时，设置音频数据回调标志，参考：BRAC\_InitSDK
- 2、调用 API 打开自己或请求对方的音频数据，参考：BRAC\_UserSpeakControl

### 4.2.4. 音频数据扩展回调函数

```
typedef void (CALLBACK * BRAC_AudioDataEx_Callback)(DWORD dwUserId, LPVOID lpBuf, DWORD dwLen, WAVEFORMATEX waveFormatEx, DWORD dwTimeStamp, LPVOID lpUserValue);
```

#### 参数：

**dwUserId:** 用户 ID，指示该音频数据属于哪一个用户

**lpBuf:** 音频数据缓冲区

**dwLen:** 缓冲区大小

**waveFormatEx:** 音频数据相关信息，如音频的采样频率、量化位数通道数等

**dwTimeStamp** 音频帧时间戳，单位：ms，用于音视频同步

**lpUserValue:** 用户自定义参数，在设置回调函数时传入

#### 备注：

V4.2 版本新增，与“音频数据回调函数”相比，增加了音频时间戳参数。

## 4.2.5. 文字消息回调函数

```
typedef void (CALLBACK * BRAC_TextMessage_Callback)( DWORD dwFromUserid, DWORD dwToUserid,  
BOOL bSecret, LPCTSTR lpMsgBuf, DWORD dwLen, LPVOID lpUserValue);
```

参数:

**dwFromUserid:** 源用户 ID, 指示发送用户 (文字消息的发送者)

**dwToUserid:** 目标用户 ID, -1 表示对所有人说

**bSecret:** 是否为密语, 当目标用户 ID 不等-1 时, 表示悄悄话

**lpMsgBuf:** 文字消息缓冲区

**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\_SendTextMessage”方法发送的数据时, 将会触发该回调函数。

## 4.2.6. 透明通道数据回调函数

```
typedef void (CALLBACK * BRAC_TransBuffer_Callback)(DWORD dwUserid, LPBYTE lpBuf, DWORD  
dwLen, LPVOID 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。

由于该函数传递的数据是一个与本 SDK 无关的缓冲区 (由上层应用自己填充内容), 相对于本 SDK 来说是透明的, 故称为透明通道, 利用该通道, 可以向当前房间内的任何用户传输上层应用自定义的数据。

## 4.2.7. 透明通道数据扩展回调函数

```
typedef void (CALLBACK * BRAC_TransBufferEx_CallBack)(DWORD dwUserId, LPBYTE lpBuf, DWORD dwLen, DWORD wParam, DWORD lParam, DWORD dwTaskId, LPVOID 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
wParam:	缓冲区附带参数 (由发送者设置, 上层应用可自定义用途)
lParam:	缓冲区附带参数 2
dwTaskId:	该缓冲区所对应的传输任务编号
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\_TransBufferEx”方法发送的数据时, 将会触发该回调函数。

## 4.2.8. 文件传输回调函数

```
typedef void (CALLBACK * BRAC_TransFile_CallBack)(DWORD dwUserId, LPCTSTR lpFileName, LPCTSTR lpTempFilePath, DWORD dwFileLength, DWORD wParam, DWORD lParam, DWORD dwTaskId, LPVOID 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
lpFileName:	文件名 (含扩展名, 不含路径)
lpTempFilePath:	接收完成后, SDK 保存在本地的临时文件 (包含完整路径)
dwFileLength:	文件总长度
wParam:	附带参数 1
lParam:	附带参数 2
dwTaskId:	该文件所对应的任务编号
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\_TransFile”方法发送的文件时, 将会触发该回

调函数。

特别提示：本 SDK 不会删除 “lpTempFilePath” 所指示的临时文件，上层应用在处理完毕后，需要主动删除该临时文件。

## 4.2.9. 音量变化回调函数

```
typedef void (CALLBACK * BRAC_VolumeChange_CallBack)(BRAC_AudioDevice device, DWORD dwCurrentVolume, LPVOID lpUserValue);
```

参数：

**device:** 表示音量更改的设备类型，是音频输入设备，或是音频输出设备

**dwCurrentVolume:** 表示该设备当前的音量水平，取值范围为 0~100

**lpUserValue:** 用户自定义参数，在设置回调函数时传入

备注：

当系统的音量变化时，将会触发该回调函数，通过响应该函数，可以让上层应用中的音量滑动条与系统实际音量设置保持一致。

## 4.2.10. SDK Filter通信数据回调函数

```
typedef void (CALLBACK * BRAC_SDKFilterData_CallBack)(LPBYTE lpBuf, DWORD dwLen, LPVOID lpUserValue);
```

参数：

**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：

当服务器 SDK Filter 发送数据给客户端时，将触发该回调函数。

## 4.2.11. 录像、快照任务完成回调函数

```
typedef void (CALLBACK * BRAC_RecordSnapShotEx_CallBack)(DWORD dwUserId, LPCTSTR lpFileName, DWORD dwElapse, DWORD dwFlags, DWORD dwParam, LPCTSTR lpUserStr, LPVOID lpUserValue);
```

参数：

**dwUserId:** 被录制用户 ID

lpFileName:	文件保存路径
dwElapse:	录像时长，单位：秒
dwFlags:	录像标志
dwParam:	用户自定义参数，整型
lpUserStr:	用户自定义参数，字符串类型

**备注：**

客户端调用 API: BRAC\_StreamRecordCtrlEx 录像完成之后，将会触发该回调事件，其中 lpFileName 为本地文件路径：

用户自定义参数包括整型（dwParam）、字符串类型（lpUserStr）与 API: StreamRecordCtrlEx 的传入参数对应；

进行中心服务器录像时，也可以触发客户端的本地回调函数，更多信息可参考：[中心服务器录像支持触发客户端回调事件](#)

## 4.2.12. 异步事件通知回调函数

```
typedef void (CALLBACK* BRAC_NotifyMessage_CallBack)(DWORD dwNotifyMsg, DWORD wParam,
DWORD lParam, LPVOID lpUserValue);
```

**参数：**

dwNotifyMsg	事件类型，参考：“事件预定义” 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。

## 4.2.13. 视频呼叫事件回调函数

```
typedef void (CALLBACK * BRAC_VideoCallEvent_CallBack)(DWORD dwEventType, DWORD dwUserId,
DWORD dwErrorCode, DWORD dwFlags, DWORD dwParam, LPCTSTR lpUserStr, LPVOID lpUserValue);
```

参数:

<b>dwEventType</b>	呼叫事件类型，详见函数 <b>BRAC_VideoCallControl</b> 中的定义
<b>dwUserId:</b>	视频呼叫事件发起方用户 ID
<b>dwErrorCode:</b>	错误代码，当事件类型为“Reply”和“Finish”时有效
<b>dwFlags:</b>	视频呼叫标志
<b>dwParam:</b>	事件附带参数（整型）
<b>lpUserStr:</b>	事件附带参数（字符串）
<b>lpUserValue:</b>	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，其它用户通过 API: **BRAC\_VideoCallControl** 发起视频呼叫时，将触发该回调函数。

用户 A 向用户 B 发送（Request）请求，用户 B 回复（Reply）同意通话之后，服务器会自动向 A、B 同时发送（Start）指令，表示会话开始，当客户端在回调函数中收到 **dwEventType= BRAC\_VIDEOCALL\_EVENT\_START** 事件时，**dwParam** 表示 **RoomId**，由服务器自动分配，这时用户 A、B 均需要主动进入分配的房间，打开本地音频、视频，同时请求对方的音频、视频才能完成整个视频呼叫过程。

更多关于视频呼叫事件的信息可参考技术论坛相关内容：

<http://bbs.anychat.cn/forum.php?mod=viewthread&tid=150&extra=page%3D1>

#### 4.2.14. 数据加密、解密回调函数

```
typedef DWORD (CALLBACK * BRAC_DataEncDec_CallBack)(DWORD dwUserId, DWORD dwFlags, LPBYTE lpInBuf, DWORD dwInSize, LPBYTE lpOutBuf, LPDWORD lpOutSize, LPVOID lpUserValue);
```

参数:

<b>dwUserId:</b>	数据来源用户 ID
<b>dwFlags:</b>	数据标志，包括加密、解密以及数据类型等信息
<b>lpInBuf:</b>	输入缓冲区（字节数组）
<b>dwInSize:</b>	输入缓冲区数据大小
<b>lpOutBuf:</b>	输出缓冲区（内核已经分配好内存）



**lpOutSize:** 输出缓冲区大小（返回实际输出缓冲区大小）

**lpUserValue:** 用户自定义参数，在设置回调函数时传入

备注：

内核默认情况下没有开始数据加密、解密功能，若需要触发该回调，则需要调用如下 API 接口开启：

```
DWORD dwDataEncryption = 1;  
BRAC_SetSDKOptionInt(BRAC_SO_CORESDK_DATAENCRYPTION, (CHAR*)&dwDataEncryption,  
sizeof(DWORD));
```

需要对数据进行加密或是解密处理，可通过 **dwFlags** 中的标志位区分，同时还可以区分数据类型，是视频数据，或是音频数据等；

输出缓冲区（**lpOutBuf**）已由内核分配好，在数据加密（或解密）完成之后，可以直接将处理后的数据拷贝到输出缓冲区，并设置实际输出的缓冲区大小（修改 **lpOutSize** 的值）；

更多关于数据加密、解密的信息可参考技术论坛相关内容：

<http://bbs.anychat.cn/forum.php?mod=viewthread&tid=674&extra=page%3D1>

## 5. 函数说明

### 5.1. 初始化与资源释放

#### 5.1.1. 初始化SDK

**DWORD BRAC\_InitSDK(void\* hWnd, DWORD dwFuncMode)**

功能：初始化 SDK

返回值：0 表示成功，否则为出错代码

参数：

hWnd 默认为 NULL

dwFuncMode 功能模式组合，见备注附表

备注：

功能模式组合可根据实际的需求灵活定义，如果在后续的方法调用中失败，则很有可能是某一项功能没有被定义。

该方法必须第一个被调用（BRAC\_SetSDKOption 方法除外），否则后续的其他方法调用将会返回没有初始化错误。

目前 SDK 支持如下功能参数组合：

参数定义	功能解释
BRAC_FUNC_VIDEO_CBDATA	通过回调函数输出视频数据，设置该标志后，还需要调用 BRAC_VideoData_CallBack 方法注册回调函数，上层应用才能收到视频数据
BRAC_FUNC_VIDEO_AUTODISP	由 SDK 处理视频，将视频显示在指定的窗口上，设置该标志后，必须在打开用户视频（BRAC_UserCameraControl）之前，调用方法 BRAC_SetVideoPos 设置视频显示的窗口及位置
BRAC_FUNC_AUDIO_CBDATA	通过回调函数输出音频数据，设置该标志后，还需要调用 BRAC_AudioData_CallBack 方法注册回调函数，上层应用才能收到音频数据

BRAC_FUNC_AUDIO_AUTOPLAY	由 SDK 处理音频，自动播放
BRAC_FUNC_CONFIG_LOCALINI	生成本地配置文件（AnyChatSDK.ini），便于 SDK 记录前一次所打开的音频、视频设备等信息，当 SDK 下次启动时，将自动选择 INI 文件所保存的配置信息
BRAC_FUNC_CHKDEPENDMODULE	自动检查 SDK 所依赖的组件，当组件没有注册时，SDK 会自动注册。设置该标志后，还必须调用 BRAC_SetSDKOption 方法设置 SDK 组件所在位置
BRAC_FUNC_AUDIO_VOLUMEALC	由 SDK 自动计算语音的音量，设置该标志后，可以通过调用 BRAC_QueryUserState 方法获取某一用户当前说话的音量大小
BRAC_FUNC_AUDIO_AUTOVOLUME	允许 SDK 自动控制 Mic 录音音量
BRAC_FUNC_NET_SUPPORTUPNP	允许 SDK 打开用户网络中的 UPNP 设备，如果用户的路由器或是防火墙支持 UPNP 协议，则可提高 P2P 打洞的成功率
BRAC_FUNC_DISABLEDECODE	禁止对收到的数据进行解码和播放，为了提高代理客户端的数据转发性能，可设置该标志，否则不能设置该标志（启用客户端的代理功能时建议设置）

注：通常情况下，BRAC\_FUNC\_VIDEO\_CBDATA 和 BRAC\_FUNC\_VIDEO\_AUTODISP 只需要选择其中一项，如果没有特殊的用途，也可不设置 BRAC\_FUNC\_AUDIO\_CBDATA 标志，只需要设置 BRAC\_FUNC\_AUDIO\_AUTOPLAY 即可。

### 5.1.2. 设置视频数据回调函数

**DWORD      BRAC\_SetVideoDataCallBack(BRAC\_PixelFormat      pixFmt, BRAC\_VideoData\_CallBack lpFunction, LPVOID lpUserValue=NULL);**

**功能：**设置视频数据回调函数，使得当有视频数据时，能通过回调函数，将原始的视频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：**

pixFmt 所请求的视频数据格式；

**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：**

如果需要 SDK 回调视频数据，则初始化 SDK 时，调用 BRAC\_InitSDK 方法的 dwFuncMode 参数必须包含 BRAC\_FUNC\_VIDEO\_CBDATA 值。

### 5.1.3. 设置视频数据扩展回调函数

**DWORD BRAC\_SetVideoDataExCallBack(BRAC\_PixelFormat pixFmt, BRAC\_VideoDataEx\_CallBack lpFunction, LPVOID lpUserValue=NULL);**

**功能：**设置视频数据扩展回调函数，使得当有视频数据时，能通过回调函数，将原始的视频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：**

**pixFmt** 所请求的视频数据格式；

**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：**

如果需要 SDK 回调视频数据，则初始化 SDK 时，调用 BRAC\_InitSDK 方法的 dwFuncMode 参数必须包含 BRAC\_FUNC\_VIDEO\_CBDATA 值。

V4.2 版本新增接口，相比于“视频数据回调函数”，增加了时间戳参数。

### 5.1.4. 设置音频数据回调函数

**DWORD BRAC\_SetAudioDataCallBack(BRAC\_AudioData\_CallBack lpFunction, LPVOID lpUserValue=NULL);**

**功能：**设置音频数据回调函数，使得当有音频数据时，能通过回调函数，将原始的音频数据回调给上层应用，便于上层进行音量的计算以及其它相关的任务。

**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）。

### 5.1.5. 设置音频数据扩展回调函数

**DWORD BRAC\_SetAudioDataExCallBack(BRAC\_AudioDataEx\_CallBack lpFunction, LPVOID lpUserValue=NULL);**

**功能：**设置音频数据扩展回调函数，使得当有音频数据时，能通过回调函数，将原始的音频数据回调给上层应用，便于上层进行相关的处理。

**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：**

V4.2 版本新增接口，相比于“音频数据回调函数”，增加了时间戳参数。

### 5.1.6. 设置文字消息回调函数

**DWORD BRAC\_SetTextMessageCallBack(BRAC\_TextMessage\_CallBack lpFunction, LPVOID lpUserValue=NULL);**

**功能：**设置文字消息回调函数，使得当有别的用户发送的文字消息时，能通过回调函数，将文字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）。

### 5.1.7. 设置透明通道数据回调函数

**DWORD BRAC\_SetTransBufferCallBack(BRAC\_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）。

### 5.1.8. 设置透明通道数据扩展回调函数

**DWORD BRAC\_SetTransBufferExCallBack (BRAC\_TransBufferEx\_Callback lpFunction, LPVOID lpUserValue=NULL);**

**功能：**设置透明通道数据扩展回调函数，使得当有别的用户通过透明通道扩展函数发送数据时，能通过指定的回调函数，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）。

### 5.1.9. 设置文件传输回调函数

**DWORD BRAC\_SetTransFileCallBack(BRAC\_TransFile\_CallBack lpFunction, LPVOID lpUserValue=NULL);**

**功能：**设置文件传输回调函数，使得当有别的用户通过文件传输函数发送文件，本地接收成功后，能通过指定的回调函数，将文件名、本地临时文件路径、文件大小等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）。

### 5.1.10. 设置音量变化回调函数

**DWORD BRAC\_SetVolumeChangeCallBack(BRAC\_VolumeChange\_CallBack lpFunction, LPVOID lpUserValue=NULL);**

**功能：**设置音量变化回调函数，使得当系统的音量变化时，能通过回调函数，将变化后的音量回调给上层应用，使得上层应用能同步更新界面上的音量值。

**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）。

### 5.1.11. 设置SDK Filter通信数据回调函数

**DWORD BRAC\_SetSDKFilterDataCallBack(BRAC\_SDKFilterData\_CallBack lpFunction, LPVOID lpUserValue=NULL);**

**功能：**设置 SDK Filter 通信数据回调函数，当服务器 SDK Filter 发送数据给客户端时，将触发该回调函数，将服务器 SDK Filter 传递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）。

### 5.1.12. 设置录像快照通知回调函数

**DWORD BRAC\_SetRecordSnapShotCallBack(BRAC\_RecordSnapShot\_CallBack lpFunction, LPVOID lpUserValue=NULL);**

**功能：**设置录像快照任务完成通知回调函数，当 SDK 录制到音视频，或是抓拍到图片时，将触发该回调函数，通知上层应用录制（抓拍）所生成的临时文件位置。

**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）。

### 5.1.13. 设置异步消息通知回调函数

**DWORD BRAC\_SetNotifyMessageCallBack(BRAC\_NotifyMessage\_CallBack lpFunction,, LPVOID lpUserValue=NULL);**

**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:**

AnyChat 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 BRAC\_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。

注册该回调函数后，SDK 仍然会向上层应用投递（PostMessage）异步消息，上层应用可根据实际情况处理异步消息，或是回调函数，通常只需要选择其中的一项即可。

## 5.1.14. 设置视频呼叫通知回调函数

**DWORD BRAC\_SetVideoCallEventCallBack(BRAC\_VideoCallEvent\_CallBack lpFunction, LPVOID lpUserValue=NULL);**

**功能:** 设置视频呼叫通知回调函数，当其它用户发起视频呼叫请求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:**

更多关于视频呼叫事件的信息可参考技术论坛相关内容：  
<http://bbs.anychat.cn/forum.php?mod=viewthread&tid=150&extra=page%3D1>

### 5.1.15. 设置数据加密、解密回调函数

**DWORD BRAC\_SetDataEncDecCallBack(BRAC\_DataEncDec\_Callback lpFunction, LPVOID lpUserValue=NULL);**

**功能：**设置数据加密、解密回调函数，当本地开启数据加密功能，或是收到其它用户加密的音视频数据时将触发该回调函数。

**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：**

更多关于数据加密、解密的信息可参考技术论坛相关内容：

<http://bbs.anychat.cn/forum.php?mod=viewthread&tid=674&extra=page%3D1>

### 5.1.16. 设置回调函数

**DWORD BRAC\_SetCallBack(DWORD dwCBType, LPVOID lpFunction, LPVOID lpUserValue);**

**功能：**设置回调函数。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwCBType 回调函数类型，定义为常量：BRAC\_CBTYPE\_XXXX

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：**

该方法可以设置所有回调函数。

### 5.1.17. 释放SDK资源

**DWORD BRAC\_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：**

该方法必须最后一个被调用，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。

## 5.2. 登录流程

### 5.2.1. 设置服务器认证密码

**DWORD BRAC\_SetServerAuthPass(LPCTSTR lpPassword);**

**功能：**设置服务器连接认证密码，确保 SDK 能正常连接到服务器。

**返回值：**0 表示成功，否则为出错代码

**参数：**

lpPassword 认证密码（大小写敏感）；

**备注：**

为了防止未授权 SDK 连接服务器，在服务器配置文件（AnyChatCoreServer.ini）中可设置“SDKAuthPass”，如果该配置项被设置，当 SDK 连接服务器时，会将该方法所传入的密码加密后传输到服务器，服务器再比较是否合法，如果密码不正确，则连接将被断开。如果该配置项未被设置（配置文件默认），则无论该方法是否被调用，SDK 均可正常连接到服务器。

## 5.2.2. 连接服务器

**DWORD BRAC\_Connect(LPCTSTR lpServerAddr, DWORD dwPort);**

**功能：**用于与服务器建立连接。

**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；  
dwPort           服务端口号（默认为 8902）

**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。

传入网站域名（URL）地址，需要 V2.71 版本之后的 SDK 才能支持。

## 5.2.3. 登录系统

**DWORD BRAC\_Login(LPCTSTR lpUserName, LPCTSTR lpPassword, DWORD dwPassEncType);**

**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        当前已登录

**参数：**

lpUserName      注册用户名；  
lpPassword      登录密码（为空表示游客）；  
dwPassEncType   保留，等于 0

**备注：**

该方法可以连接系统之后立即调用，而不用关心连接系统是否成功，当 SDK 连接系统成功之后，如果之前调用过该方法，则 SDK 将会自动向服务器发出登

录系统的申请。

返回值为 0 并不表示登录服务器成功，仅表示 SDK 已成功收到登录服务器的指令，如果登录成功，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。

如果服务器配置了“SDK Filter Plus”插件，则客户端调用该方法后，将会触发其 API 接口：BRFP\_VerifyUser，用户名、密码参数将会作为参数传递给该 API 函数，由“SDK Filter Plus”完成用户的身份验证工作，服务器根据该 API 接口的返回值来判定是否通过身份验证，详细信息可参考文档《AnyChat SDK Filter Plus 开发指南》。

如果在服务器端使用“AnyChat Server SDK”开发了业务层服务器，则客户端调用该方法后，将会触发业务层服务器的回调函数“BRAS\_VerifyUser\_CallBack”，由业务层服务器完成用户的身份验证工作，服务器根据回调函数的返回值来判定是否通过身份验证，详细信息可参考文档《AnyChat Server SDK 开发指南》。

## 5.2.4. 视频呼叫控制

**DWORD BRAC\_VideoCallControl(DWORD dwEventType, DWORD dwUserId, DWORD dwErrorCode, DWORD dwFlags, DWORD dwParam, LPCTSTR lpUserStr);**

**功能：**对视频呼叫业务流程进行控制，发起视频呼叫，或是结束视频通话等。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwEventType    事件类型，定义为：

```
#define BRAC_VIDEOCALL_EVENT_REQUEST 1    ///< 呼叫请求
#define BRAC_VIDEOCALL_EVENT_REPLY 2    ///< 呼叫请求回复
#define BRAC_VIDEOCALL_EVENT_START 3    ///< 视频呼叫会话开始事件
#define BRAC_VIDEOCALL_EVENT_FINISH 4    ///< 挂断（结束）呼叫会话
```

dwUserId        目标用户 ID

dwErrorCode    出错代码，与事件类型相关，默认为 0

dwFlags	视频呼叫标志，默认为 0
dwParam	用户自定义参数（整型），默认为 0
lpUserStr	用户自定义参数（字符串），默认为空

**备注：**

视频呼叫业务逻辑主要实现两个终端（PC、手机、Pad 等）之间的通话请求流程控制，包括请求（Request）、回复（Reply）、开始（Start）以及结束（Finish）等过程，可以形象理解为打电话的流程：拨号、等待、通话、挂断。为 V4.9 版本新增接口。

该 API 接口和回调函数（BRAC\_VideoCallEvent\_CallBack）中的 dwUserId 均为对方（被呼叫方）的用户 ID；

被呼叫方拒绝通话时，可通过发送发送回复（Reply）指令，其中 dwErrorCode=100104 来拒绝；

被呼叫方同意通话时，发送回复（Reply）指令，其中 dwErrorCode=0，然后服务器会向双方发送通话开始（Start）指令；

用户 A 向用户 B 发送（Request）请求，用户 B 回复（Reply）同意通话之后，服务器会自动向 A、B 同时发送（Start）指令，表示会话开始，当客户端在回调函数中收到 dwEventType= BRAC\_VIDEOCALL\_EVENT\_START 事件时，dwParam 表示 RoomId，由服务器自动分配，这时用户 A、B 均需要主动进入分配的房间，打开本地音频、视频，同时请求对方的音频、视频才能完成整个视频呼叫过程。

结束通话时，任何一方（包括业务服务器）均可以发送结束（Finish）指令，然后服务器会向双方发送通话结束（Finish）指令；

业务服务器可干预呼叫流程：在业务服务器端回调函数（BRAS\_OnVideoCallEvent\_CallBack）收到呼叫请求指令后，返回 0 表示允许呼叫，否则为出错代码，不允许呼叫；在会话过程中业务服务器可以发送结束（Finish）指令，强制挂断指定用户的通话；

API 接口中的 dwParam（整型）、lpUserStr（字符串）均为用户自定义用途；一个用户同时只能发起一路呼叫请求，也同时只能被一个用户呼叫；

更多关于视频呼叫事件的信息可参考技术论坛相关内容：

<http://bbs.anychat.cn/forum.php?mod=viewthread&tid=150&extra=page%3D1>

### 5.2.5. 进入房间（根据房间编号）

**DWORD BRAC\_EnterRoom(DWORD dwRoomid, LPCTSTR lpRoomPass, DWORD dwPassEncType);**

**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：**

dwRoomid	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）；
dwPassEncType	保留，等于 0

**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。

用户必须进入一个房间，否则无法进行相关的操作，后续用户所有的操作都是在房间内操作，针对游戏，房间可以理解为游戏桌（一桌游戏对应一个房间），针对视频会议，房间可以理解为会议室。

如果服务器配置了“SDK Filter Plus”插件，则客户端调用该方法后，将会触发其 API 接口：BRFP\_PrepareEnterRoom，用户 ID、房间 ID、房间密码将会作为参数传递给该 API 函数，由“SDK Filter Plus”完成用户进入房间的验证工作，服务器根据该 API 接口的返回值来判定是否允许进入房间，详细信息可参考文档《AnyChat SDK Filter Plus 开发指南》。

如果在服务器端使用“AnyChat Server SDK”开发了业务层服务器，则客户端调用该方法后，将会触发业务层服务器的回调函数

“BRAS\_PrepareEnterRoom\_CallBack”，由业务层服务器完成用户进入房间的验证工作，服务器根据该 API 接口的返回值来判定是否允许进入房间，详细信息可参考文档《AnyChat Server SDK 开发指南》。

### 5.2.6. 进入房间（根据房间名称）

**DWORD BRAC\_EnterRoomEx(LPCTSTR lpRoomName, LPCTSTR lpRoomPass)**

**功能：**根据房间名称进入房间

**返回值：**0 表示成功，否则为出错代码

**参数：**

lpRoomName	房间名称；
lpRoomPass	房间密码（当房间需要密码时有效，如果没有可为空）；
dwPassEncType	保留，等于 0；

**备注：**

该方法与“BRAC\_EnterRoom”功能相同，区别在于房间的标识不同，其中“BRAC\_EnterRoom”是用房间 ID 进入房间，而该方法是用房间名称进入房间，如果房间不存在，而且系统配置为自动创建房间时，将会由系统分配一个唯一的房间编号，通过进入房间消息（WM\_GV\_ENTERROOM）返回给上层应用，上层应用可以通过方法“BRAC\_GetRoomName”来获取房间名称。

### 5.2.7. 离开房间

**DWORD BRAC\_LeaveRoom(DWORD dwRoomid);**

**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：**



dwRoomid 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。

## 5.2.8. 注销系统

**DWORD BRAC\_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

## 5.3. 音视频操作

### 5.3.1. 用户视频控制

**DWORD BRAC\_UserCameraControl(DWORD dwUserId, BOOL bOpen);**

**功能：**用户视频控制，打开或关闭本地摄像头，或请求对方的视频

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwUserId: 用户编号，为-1 表示对本地视频进行控制

bOpen 是否打开视频

**备注：**

对于本地用户，该方法是直接操作用户的摄像头，而对于其它用户，该方法只是向对方发送一个请求（取消）视频流的申请，并不会直接操作对方的摄像头。

### 5.3.2. 用户语音控制

**DWORD BRAC\_UserSpeakControl(DWORD dwUserId, BOOL bOpen);**

**功能：**用户发言控制

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwUserId            用户编号，为-1 表示对本地发言进行控制

bOpen              是否允许用户发言，当 dwUserId=-1 时，1 表示请求发言（拿 Mic），0 表示停止发言（放 Mic）

**备注：**

对于本地用户，该方法是直接操作用户的 Mic，而对于其它用户，该方法只是向对方发送一个请求（取消）音频流的申请，并不会直接操作对方的 Mic。

### 5.3.3. 设置视频显示位置

**DWORD BRAC\_SetVideoPos(DWORD dwUserId, HWND hWnd, DWORD dwLeft, DWORD dwTop, DWORD dwRight, DWORD 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 BRAC\_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。

### 5.3.4. 重绘指定用户的视频

**DWORD BRAC\_RepaintVideo(DWORD dwUserId, HDC hDC);**

**功能：**重绘指定用户的视频，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
hDC	所需要刷新的画布句柄，可为 NULL

**备注：**

该方法只有当初始化系统时，设置了 BRAC\_FUNC\_VIDEO\_AUTODISP（由 SDK 包来处理视频）标志，而且采用 DirectShow 显示模式时，才必须调用，通常是应用程序响应 WM\_PAINT 消息，在该消息中调用该方法重绘视频。

如果不调用该方法，当 DirectShow 显示模式后，视频可能被背景图片所遮挡，而不会自动刷新，主要原因是 DirectShow 无法获得界面刷新的消息，所以必须要手工进行处理，详情可参考：[微软官方文档](#)。

### 5.3.5. 设置外部输入视频格式

**DWORD BRAC\_SetInputVideoFormat(BRAC\_PixelFormat pixFmt, DWORD dwWidth, DWORD dwHeight, DWORD dwFps, DWORD dwFlags);**

**功能：**设置外部输入视频数据的格式，初始化外部视频数据输入环境

**返回值：**0 表示成功，否则为出错代码

**参数：**

pixFmt	输入视频数据的格式，如 RGB24、YUV 等
dwWidth	输入视频数据的宽度
dwHeight	输入视频数据的高度
dwFps	输入视频数据的帧率
dwFlags	输入视频数据的标志，备用，必须为 0

**备注：**

要实现外部视频输入（禁止 AnyChat 内核对摄像头进行视频的采集操作），

则需要启动外部视频输入模式，参考如下代码：

```
BOOL bExtVideoInput = 1;

BRAC_SetSDKOption(BRAC_SO_CORESDK_EXTVIDEOINPUT, (CHAR*)&bExtVideoInput,
sizeof(DWORD));
```

V4.2 版本新增接口，进入房间成功之后调用，而且只需要调用一次即可。

### 5.3.6. 外部视频数据输入

**DWORD BRAC\_InputVideoData(LPBYTE lpVideoFrame, DWORD dwSize, DWORD dwTimeStamp);**

**功能：**将产生的视频帧输入 AnyChat，每次调用输入一帧数据

**返回值：**0 表示成功，否则为出错代码

**参数：**

lpVideoFrame 视频数据缓冲区地址（一帧完整的视频数据）  
dwSize 视频数据缓冲区大小  
dwTimeStamp 输入视频的时间戳，默认为 0，表示当前时间，单位：ms

**备注：**

V4.2 版本新增接口，进入房间成功之后调用，上层应用程序采集一帧调用一次，退出房间后不能再调用，而且必须在设置输入视频格式之后再调用。

最后一个参数为时间戳，可以为 0，或是上层设置的时间戳，毫秒间隔

### 5.3.7. 设置外部输入音频格式

**DWORD BRAC\_SetInputAudioFormat(DWORD dwChannels, DWORD dwSamplesPerSec, DWORD dwBitsPerSample, DWORD dwFlags);**

**功能：**设置外部输入音频数据的格式，初始化外部音频数据输入环境

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwChannels 输入音频数据通道数  
dwSamplesPerSec 输入音频数据的采样率  
dwBitsPerSample 输入音频数据的量化位数，通常为 16bits，即 2 个字节

**dwFlags**            输入音频数据的标志，备用，必须为 0

**备注：**

要实现外部音频输入（禁止 AnyChat 内核对声卡进行音频的采集操作），则需要启动外部音频输入模式，参考如下代码：

```
BOOL bExtAudioInput = 1;
```

```
BRAC_SetSDKOption(BRAC_SO_CORESDK_EXTAUDIOINPUT, (CHAR*)&bExtAudioInput, sizeof(DWORD));
```

V4.2 版本新增接口，进入房间成功之后调用，而且只需要调用一次即可。

### 5.3.8. 外部音频数据输入

**DWORD BRAC\_InputAudioData(LPBYTE lpSamples, DWORD dwSize, DWORD dwTimeStamp);**

**功能：**将采集的音频帧输入 AnyChat

**返回值：**0 表示成功，否则为出错代码

**参数：**

**lpSamples**            音频采样数据缓冲区地址

**dwSize**                音频数据缓冲区大小

**dwTimeStamp**        输入音频的时间戳，默认为 0，表示当前时间，单位：ms

**备注：**

V4.2 版本新增接口，进入房间成功之后调用，上层应用程序采集一段时间（如 100ms）调用一次，退出房间后不能再调用，而且必须在设置输入音频格式之后再调用。

最后一个参数为时间戳，可以为 0，或是上层设置的时间戳，毫秒间隔

## 5.4. 查询状态

### 5.4.1. 查询摄像头的状态

**DWORD BRAC\_GetCameraState(DWORD dwUserId, DWORD& dwState);**

**功能：**查询用户摄像头的状态

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwUserId 用户编号，为-1 时表示获取自己的摄像头状态；

dwState 摄像头状态，定义为：

GV_CAMERA_STATE_NULL	0	没有摄像头
GV_CAMERA_STATE_NORMAL	1	有摄像头但没有打开
GV_CAMERA_STATE_OPEN	2	摄像头已打开

**备注：**

该方法必须在登录系统之后调用方才有效，根据返回参数的不同，可以判别用户当前摄像头的状态，以及判断用户是否有摄像头。

推荐使用新的 API 接口：“**BRAC\_QueryUserState**”来替代该功能。

### 5.4.2. 查询用户音频设备采集状态

**DWORD BRAC\_GetSpeakState(DWORD dwUserId, DWORD& dwState);**

**功能：**查询用户音频设备采集状态

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwUserId 用户编号，为-1 时表示获取自己的音频设备状态；

dwState 音频设备采集状态，开启音频采集返回 1，否则返回 0

**备注：**

这里所说的“音频设备采集状态”是指在 SDK 内部是否已开始音频采集，当返回值为 1 时，表示 SDK 已经开始采集，当有其它用户请求时，才对外传输。

SDK 并不能获知用户是否已将麦克风接入声卡，估当用户没有接入麦克风时，就算 SDK 进行音频采集，也是无法采集到用户说话的声音的。

关于实际应用中的“公麦”、“麦序”等属于业务逻辑范畴，具体的实现方式可参考《AnyChat Server SDK 开发指南》中“常用业务处理逻辑”的章节。

推荐使用新的 API 接口：“**BRAC\_QueryUserState**”来替代该功能。

### 5.4.3. 查询用户级别

**DWORD BRAC\_GetUserLevel(DWORD dwUserId, DWORD& dwLevel);**

**功能：**查询用户级别

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwUserId 用户编号，为-1 时表示获取自己的级别；

dwLevel 用户级别返回值

**备注：**

这里所查询到的用户级别，是用户在身份验证时，服务器端调用 SDK Filter 的“BRGS\_VerifyUser”方法时，由 SDK Filter 返回给服务器的 lpUserLevel 参数值。

推荐使用新的 API 接口：“**BRAC\_QueryUserState**”来替代该功能。

### 5.4.4. 查询用户昵称

**DWORD BRAC\_GetUserName(DWORD dwUserId, CHAR\* lpUserName, DWORD dwLen);**

**功能：**查询用户昵称

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwUserId 用户编号，为-1 时表示获取自己的昵称；

lpUserName 返回用户昵称的缓冲区地址

dwLen 用户昵称缓冲区大小

**备注：**

这里所查询到的用户昵称，是用户在身份验证时，服务器端调用 SDK Filter 的“BRGS\_VerifyUser”方法时，由 SDK Filter 返回给服务器的 lpNickName 参数值，如果 lpNickName 为空，则默认采用登录用户名替代用户昵称。

当用户离开房间之后（包括在 WM\_GV\_USERATROOM 消息中，状态为用户离开时）将会查询失败。

推荐使用新的 API 接口：“BRAC\_QueryUserState”来替代该功能。

### 5.4.5. 查询用户状态

**DWORD BRAC\_QueryUserState(DWORD dwUserId, int infoname, char FAR\* 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。

目前提供的查询的信息代码见下表：

信息代码定义	参数类型	用途	备注
BRAC_USERSTATE_CAMERA	DWORD	获取用户摄像头的状态	0 没有摄像头 1 有摄像头，但没有打开 2 摄像头已启用
BRAC_USERSTATE_HOLDMIC	DWORD	获取用户的音频设备状态	0 关闭音频采集 1 开启音频采集



BRAC_USERSTATE_SPEAKVOLUME	DOUBLE	获取用户当前说话的音量	范围：0.0 ~ 100.0 见提示（1）
BRAC_USERSTATE_RECORDING	DWORD	获取用户的录像（音）状态	0 没有录像（音） 1 正在录像（音）
BRAC_USERSTATE_LEVEL	DWORD	用户级别	见提示（2）
BRAC_USERSTATE_NICKNAME	PCHAR	用户昵称	见提示（3）
BRAC_USERSTATE_LOCALIP	PCHAR	用户内网 IP	
BRAC_USERSTATE_INTERNETIP	PCHAR	用户外网 IP	
BRAC_USERSTATE_VIDEOBITRATE	DWORD	用户当前的视频码率	单位：Bps
BRAC_USERSTATE_AUDIOBITRATE	DWORD	用户当前的音频码率	单位：Bps
BRAC_USERSTATE_P2PCONNECT	DWORD	P2P 连接状态 见提示（4）	0 P2P 不通 1 TCP 连接 2 UDP 连接 3 TCP、UDP 连接
BRAC_USERSTATE_NETWORKSTATUS	DWORD	查询指定用户的网络状态 (注：查询间隔需要>1s)，见提示（5）	0 优良 1 较好 2 一般 3 较差 4 非常差 5 未知状态
BRAC_USERSTATE_VIDEOSIZE	DWORD	查询指定用户的视频分辨率	低 16 位表示宽度 高 16 位表示高度
BRAC_USERSTATE_PACKLOSSRATE	DWORD	查询丢包率	见提示（6）
BRAC_USERSTATE_DEVICETYPE	DWORD	查询用户终端类型	0 Unknow, 1 Windows, 2 Android, 3 iOS, 4 Web, 5 Linux, 6 Mac, 7 Win Phone, 8 WinCE
BRAC_USERSTATE_SELFUSERSTATUS	DWORD	查询本地用户的当前状态	0 Unknow, 1 Connected, 2 Logined, 3 In Room, 4 Logouted, 5 Link Closed
BRAC_USERSTATE_SELFUSERID	DWORD	查询本地用户的 ID	若用户登录成功，返回用户实际的 userid，否则返回-1

(1)、如需要获取用户的说话音量，则必须在 SDK 初始化时，启用音量计算功能（参数：BRAC\_FUNC\_AUDIO\_VOLUME\_CALC），因为音量计算需要进行“快速傅立叶变换”，将采样的音频样本从时域变换到频域，以获取声音的能量值（音量大小），会占用一些 CPU 的运算资源，故 SDK 默认没有开启。

(2)、这里所查询到的用户级别，是用户在身份验证时，服务器端调用 SDK Filter 的“BRGS\_VerifyUser”方法时，由 SDK Filter 返回给服务器的 lpUserLevel 参数值。

(3)、这里所查询到的用户昵称，是用户在身份验证时，服务器端调用 SDK Filter 的“BRGS\_VerifyUser”方法时，由 SDK Filter 返回给服务器的 lpNickName 参数值，如果 lpNickName 为空，则默认采用登录用户名替代用户昵称。

(4)、这里所查询的 P2P 连接状态是指：查询本地用户与指定用户的当前 P2P 连接状态，即所传 dwUserId 参数为其它用户的 ID。

(5)、这里所查询指定用户的网络状态是指当自己请求其它用户的语音视频数据时，底层 SDK 根据网络丢包率、数据包的延迟等参数综合判定的结果，当没有请求某用户的数据时，查询该用户的网络状态会返回默认值：0，通常来说，“优良、较好”两种网络状态下语音、视频较流畅，“一般”网络状态下语音、视频会偶尔停顿，但不频繁，而“较差、非常差”两种网络状态下则语音、视频数据丢包较多，停顿会比较频繁，未知状态通常是指没有请求该用户的音视频数据流，内核没有相关的数据供参考。

(6)、可以查询自己，或是其它用户的网络丢包率，返回值为 0-100（如返回 5 则表示丢包率为 5%，返回 100 表示状态未知），只有当发生流媒体数据传输时内核才对丢包率进行统计。

## 5.4.6. 查询房间名称

**DWORD BRAC\_GetRoomName(DWORD dwRoomId, CHAR\* lpRoomName, DWORD dwLen);**

**功能：**查询根据房间 ID 获取房间名称

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwRoomId	房间编号；
lpRoomName	返回房间名称的缓冲区地址
dwLen	房间名称缓冲区大小

**备注：**

目前只能查询当前所在房间的房间名称，当用户离开房间后，查询将会失败。

## 5.5. 普通功能

### 5.5.1. 获取SDK版本信息

**DWORD BRAC\_GetSDKVersion(DWORD& dwMainVer, DWORD& dwSubVer, LPSTR lpCompileTime, DWORD dwBufLen);**

**功能：**获取 SDK 版本信息以及编译时间等信息。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwMainVer: 返回 SDK 的主版本号

dwSubVer: 返回 SDK 的从版本号

lpCompileTime: 返回 SDK 的编译时间

dwBufLen: 保存编译时间信息缓冲区的大小

**备注：**

通过调用该方法可以获得 SDK 的版本及编译时间等信息，在遇到版本不兼容问题时，有助于判断故障原因。

### 5.5.2. 获取当前房间在线用户列表

**DWORD BRAC\_GetOnlineUser(LPDWORD lpUserIDArray, DWORD& dwUserNum);**

**功能：**获取当前房间在线用户列表（不包含自己）

**返回值：**0 表示成功，否则为出错代码

**参数：**

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

dwUserNum 用户数量。

**备注：**

由于有时上层应用并不知道房间的在线用户有多少，所以不好确定该为 lpUserIDArray 分配多大的内存空间，典型的做法是，分两次调用，第一次调用，

将 `lpUserIDArray` 置为 `NULL`，只获取用户的具体数量，然后分配合适的缓冲区，再调用一次该方法，获得详细的用户列表。

获取在线用户列表，并不包含当前用户自己的 ID，自己的 ID 在登录消息（`WM_GV_LOGINSYSTEM`）中已通知给上层应用。

### 5.5.3. 传送文本消息

**DWORD BRAC\_SendTextMessage(DWORD dwUserid, BOOL bSecret, LPCTSTR lpMsgBuf, DWORD dwLen);**

**功能：**向指定的用户传送文本消息

**返回值：**0 表示成功，否则为出错代码

**参数：**

`dwUserid`: 目标用户编号，-1 表示大家（所有人）

`bSecret`: 是否为密语，只在 `dwUserid` 不为-1 时有效，选择密语时，其它用户看不到发送的消息

`lpMsgBuf`: 消息缓冲区

`dwLen`: 消息缓冲区的大小

**备注：**

可以利用该消息实现文字交流的功能，发送消息的对象可以是大家，也可以是指定的对象，如果是对指定的对象发送文字消息，可以选择密语。

对方收到该消息后，会触发 `BRAC_TextMessage_CallBack` 回调函数的调用。

文字消息发送功能需在进入房间成功之后才有效，即只能向当前房间中的用户发送文字消息。文字消息均经服务器转发，使用“AnyChat Server SDK”，可在服务器端保存文字消息，详细信息可参考《AnyChat Server SDK 开发指南》。

### 5.5.4. 透明通道传送缓冲区

**DWORD BRAC\_TransBuffer(DWORD dwUserid, LPBYTE lpBuf, DWORD dwLen);**

**功能：**透明通道传送缓冲区

返回值:	0	表示成功，否则为出错代码
	20	函数功能不允许，可能是传输的缓冲区太大
	21	参数设置问题，lpBuf 值为空
	208	当前还未登录

**参数:**

**dwUserId:** 目标用户编号，-1 表示大家（用户当前房间所有人），0 表示向服务器传送数据

**lpBuf:** 缓冲区，≤1024Byte（1KB）

**dwLen:** 缓冲区的大小

**备注:**

可以利用该方法实现自定义功能，缓冲区采用透明传输，目标对象可以是大家，也可以是指定的对象。

该方法将会触发对方的 BRAC\_TransBuffer\_CallBack 回调函数。

当用户进入房间成功之后，目标用户（dwUserId）等于-1 时表示向房间内的所有用户广播数据，如果没有进入房间，则目标用户不能为-1。

自 2.8.7 版本开始，该 API 方法支持跨房间传输缓冲区数据，目标用户为指定用户时，目标用户可以与自己在不同的房间，或是目标用户没有进入任何房间，或是源用户（自己）没有进入任何房间，只要双方都登录服务器成功，则可利用该方法传输缓冲区，当目标用户编号为-1 时，则源用户（自己）必须已经在房间中，表示向该房间的其它用户广播数据（注：自己不能发送给自己）。

自 4.1 版本开始，该 API 方法支持向服务器传输数据（dwUserId 等于 0），将触发服务器（AnyChat Server SDK）的回调函数：BRAS\_OnTransBuffer\_CallBack。

该方法采用 TCP 通道传输，将保证数据传输的可靠性，且接收方接收顺序与发送顺序相同。

该方法不需要用户进入房间，即可以在服务器内部的任意用户之间传输数据，更多信息请参考：<http://www.anychat.cn/faq/index.php?action=artikel&cat=2&id=206>

### 5.5.5. 透明通道传送缓冲区扩展

**DWORD BRAC\_TransBufferEx (DWORD dwUserId, LPBYTE lpBuf, DWORD dwLen,**

**DWORD wParam, DWORD lParam, DWORD dwFlags , DWORD& dwTaskId);**

**功能：**透明通道传送缓冲区

**返回值：**

0	表示成功，否则为出错代码
20	函数功能不允许，可能是传输的缓冲区太大
21	参数设置问题，lpBuf 值为空
208	当前还未登录

**参数：**

**dwUserId:** 目标用户编号，只针对某一个用户，不能为-1（所有人），可以为 0，表示发给服务器

**lpBuf:** 缓冲区，≤1024KB（1MB），内部会自动分包处理

**dwLen:** 缓冲区的大小

**wParam:** 附带参数，由上层应用自定义

**lParam:** 附带参数 2，由上层应用自定义

**dwFlags:** 特殊功能标志，当对该缓冲区有特殊要求时，可通过使用相关的功能标志，通知 SDK 进行特殊的处理，默认为 0，SDK 将自动根据网络状态选择合适的传输途径（TCP、UDP or P2P）

**dwTaskId** 与该缓冲区对应的任务 ID（只有任务添加成功后，该 ID 才会自动生成，可利用该 ID 查询该任务的传输进度）

**备注：**

该方法与“BRAC\_TransBuffer”功能相同，都是传输上层应用自定义（透明通道）数据，区别在于该方法通过设置相应的功能标识，如可选择采用 UDP 通道传输，但是只针对指定的用户传输，而“BRAC\_TransBuffer”方法则固定采用 TCP 通道传输，且缓冲区大小不能超过 1024 个字节，但可以针对所有用户传输。

从应用来看：

（1）、BRAC\_TransBuffer 适合数据量小、要求实时传输的缓冲区传递，如控制指令等；

（2）、BRAC\_TransBufferEx 适合数据量大、对实时性要求不高的需求；

该方法将会触发对方的 BRAC\_TransBufferEx\_CallBack 回调函数。

自 4.1 版本开始，该 API 方法支持向服务器传输数据（dwUserId 等于 0），将触发服务器（AnyChat Server SDK）的回调函数：

**BRAS\_OnTransBufferEx\_CallBack。**

内核使用 UDP 通道传输，不保证接收方收到的顺序与发送顺序相同，但是保证数据可达，丢包自动重传；

该方法不需要用户进入房间，即可以在服务器内部的任意用户之间传输数据。

## 5.5.6. 传送文件

**DWORD BRAC\_TransFile(DWORD dwUserId, LPCTSTR lpLocalPathName, DWORD wParam, DWORD lParam, DWORD dwFlags, DWORD& dwTaskId);**

**功能：**传送文件给指定用户

**返回值：**0 表示任务建立成功（并非文件传输完成），否则为出错代码

**参数：**

**dwUserId:** 目标用户编号，只针对某一个用户，不能为-1（所有人），可以为 0，表示发给服务器

**lpLocalPathName:** 本地文件名，含路径

**wParam:** 附带参数 1，便于上层应用扩展

**lParam:** 附带参数 2

**dwFlags:** 特殊功能标志，参考：“BRAC\_TransBufferEx”方法

**dwTaskId** 与该文件传输对应的任务 ID（只有任务添加成功后，该 ID 才会自动生成，可利用该 ID 查询该任务的传输进度）

**备注：**

该方法传输效率与“BRAC\_TransBufferEx”方法相同，只是在 SDK 内部封装了文件的分组传输功能，实现对上层应用的透明，简化上层应用的开发难度。

对方收到完整的文件后，将会触发回调函数：**BRAC\_TransFile\_CallBack**；

自 4.1 版本开始，该 API 方法支持向服务器传输文件（dwUserId 等于 0），将触发服务器（AnyChat Server SDK）的回调函数：**BRAS\_OnTransFile\_CallBack**。

该方法采用 UDP 通道传输，将保证数据传输的可靠性，丢包重传，但不保证接收方接收顺序与发送顺序相同。

该方法不需要用户进入房间，即可以在服务器内部的任意用户之间传输数

据。

### 5.5.7. 查询传输任务相关信息

**DWORD BRAC\_QueryTransTaskInfo(DWORD dwUserId, DWORD dwTaskId, int infoname, char FAR\* infoval, int infolen)**

**功能：** 查询与传输任务相关的信息，如传输进度、传输状态、传输码率等

**返回值：** 0 表示查询成功，否则为出错代码

**参数：**

**dwUserId:** 任务发起者用户编号（并非传输目标用户编号）

**dwTaskId:** 需要查询的任务编号

**infoname** 需要查询的信息代码（见备注附表）

**infoval** 查询信息值的保存参数地址

**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，可以查询指定传输任务编号的缓冲区传输情况。用户编号与任务编号组合才具有唯一性，不同的用户可能存在相同的任务编号。

目前提供的查询的信息代码见下表：

信息代码定义	参数类型	用途		备注
BRAC_TRANSTASK_PROGRESS	DOUBLE	传输任务进度查询		0.0 ~ 100.0
BRAC_TRANSTASK_BITRATE	DWORD	传输任务当前码率		单位：bps
BRAC_TRANSTASK_STATUS	DWORD	传输任务当前状态：		
		1	准备状态	
		2	传输状态	
		3	完成状态	
		4	任务被发送者取消	
		5	任务被接收方取消	

### 5.5.8. 取消传输任务

**DWORD BRAC\_CancelTransTask(DWORD dwUserId, DWORD dwTaskId)**



**功能：**取消文件、扩展缓冲区等传输任务，中止任务。

**返回值：**0 表示任务取消成功，否则为出错代码

**参数：**

dwUserId: 任务发起者用户编号（并非传输目标用户编号）

dwTaskId: 需要取消的任务编号

**备注：**

通过调用该方法，可以中止传输任务，用户编号与任务编号组合才具有唯一性，不同的用户可能存在相同的任务编号。

### 5.5.9. 激活（关闭）SDK调用日志

**DWORD BRAC\_ActiveCallLog(BOOL bActive);**

**功能：**打开，或是关闭 SDK 调用期间所产生的日志

**返回值：**0 表示成功，否则为出错代码

**参数：**

bActive: 是否打开日志功能

**备注：**

通过调用该方法，可以记录应用程序调用该 SDK 的方法顺序及参数，对于分析利用该 SDK 进行开发所产生的异常现象有一定的帮助，建议在开发过程中打开该选项，在发布的版本中关闭该选项，所产生的日志文件名为：BRAnyChatCore.log，与调用者程序在同一级目录。

SDK 默认已打开日志记录功能，该方法必须在 SDK 初始化之前调用。

### 5.5.10. 发送SDK Filter通信数据

**DWORD BRAC\_SendSDKFilterData(LPBYTE lpBuf, DWORD dwLen);**

**功能：**向服务器发送 SDK Filter 通信数据

**返回值：** 0 表示成功，否则为出错代码

20 函数功能不允许，可能是传输的缓冲区太大

21 参数设置问题，lpBuf 值为空

208      当前还未登录

**参数:**

**lpBuf:**            缓冲区  
**dwLen:**            缓冲区的大小

**备注:**

服务器收到数据后, 会将该缓冲区数据全部提交给 SDK Filter, 由 SDK Filter 来解析, 该缓冲区的内容对于本 SDK 和服务端来说, 都是透明的。

客户端的 “BRAC\_SendSDKFilter” API 和服务端端的 “BRAC\_SendBufToUser”、“BRAC\_SendBufToRoom” API 都是使用 SDK Filter 通道。

建议用 “BRAC\_TransBuffer” (客户端)、 “BRAS\_TransBuffer” (服务端) 来进行通讯, SDK Filter 通道将在后续的版本中不被支持。

## 5.5.11. 音视频录制

**DWORD      BRAC\_StreamRecordCtrlEx(DWORD      dwUserId,      BOOL  
bStartRecord, DWORD dwFlags, DWORD dwParam,      LPCTSTR lpUserStr);**

**功能:** 对指定用户的音视频流进行录制, 保存为本地音视频文件

**返回值:** 0 表示录制指令被 SDK 成功接收, 否则为出错代码

**参数:**

**dwUserId:**          需要录制视频的用户编号, 可用-1 表示本地用户 (自己);  
**bStartRecord:**      指示当前指令是启动录像, 或是停止录像;  
**dwFlags:**            录制功能标志, 参考备注;  
**dwParam:**            录制指令附带参数 (整形), 录像任务结束时, 该参数将通过回调函数返回给上层应用;  
**lpUserStr:**          录制指令附带参数 (字符串), 录像任务结束时, 该参数将通过回调函数返回给上层应用;

**备注:**

该方法只是向 SDK 下达 (停止) 录像任务, 当指令 (bStartRecord) 为停止录像时, 而且已经录制到了数据时, SDK 将产生一次回调, 通知上层应用录像文件名。

视频录制可以在本地进行，也可以在服务器端进行，通过 `dwFlags` 标志进行控制，视频录制完成之后，将触发回调事件：`OnAnyChatRecordEvent`。

录像功能标志指示 SDK 在录制时，进行特殊的处理，0 表示默认（音视频同步录制），目前支持如下标志组合：

```

ANYCHAT_RECORD_FLAGS_VIDEO    ///< 录制视频
ANYCHAT_RECORD_FLAGS_AUDIO    ///< 录制音频
ANYCHAT_RECORD_FLAGS_SERVER   ///< 服务器端录制
ANYCHAT_RECORD_FLAGS_MIXAUDIO ///< 录制音频时，将其它人的声音混音后录制
ANYCHAT_RECORD_FLAGS_MIXVIDEO ///< 录制视频时，将其它人的视频迭加后录制
ANYCHAT_RECORD_FLAGS_ABREAST  ///< 录制视频时，将其它人的视频并列录制
ANYCHAT_RECORD_FLAGS_STEREO   ///< 录制音频时，将其它人的声音混合为立体声后录制
ANYCHAT_RECORD_FLAGS_SNAPSHOT ///< 拍照
ANYCHAT_RECORD_FLAGS_LOCALCB  ///< 触发本地回调

```

在服务器端录制音视频，需要单独部署中心录像服务器，参考：[Windows 平台中心录像服务器部署](#)、[Linux 平台中心录像服务器部署](#)

更多信息可参考：

[AnyChat 支持录像文件格式设置（MP4、WMV、FLV、MP3）](#)

[中心录像服务器返回录像文件路径可配置](#)

[中心服务器录像支持触发客户端回调事件](#)

## 5.5.12. 图像抓拍（拍照）

**DWORD BRAC\_SnapShot(DWORD dwUserId, DWORD dwFlags, DWORD dwParam);**

**功能：**对指定用户的视频进行抓拍，保存为本地图形文件

**返回值：**0 表示抓拍指令被 SDK 成功接收，否则为出错代码

**参数：**

**dwUserId:** 需要抓拍视频的用户编号，可用-1 表示本地用户（自己）；

**dwFlags:** 抓拍功能标志，备用；

**dwParam:** 抓拍指令附带参数，当抓拍到一张图片时，该参数将通过回调函数返回给上层应用。

**备注：**

该方法只是向 SDK 下达抓拍任务，当接收到用户的视频流时，SDK 将进行抓拍工作，同时产生一次回调，通知上层应用本地图形文件名（默认为 jpeg 格式）。

为了保证图片抓拍功能的正常使用，在 SDK 初始化时必须做好如下准备工作：

- 1、注册抓拍完成通知回调函数，参考：BRAC\_SetRecordSnapShotCallBack
- 2、设置照片文件保存路径，参考：BRAC\_SetSDKOption

### 5.5.13. 组播功能控制

**DWORD BRAC\_MultiCastControl(LPCTSTR lpMultiCastAddr, DWORD dwPort, LPCTSTR lpNicAddr, DWORD dwTTL, DWORD dwFlags);**

**功能：**加入、退出 IP 多播组，实现组播数据的传输与接收功能控制

**返回值：**0 表示调用成功，否则为出错代码

**参数：**

lpMultiCastAddr	组播 IP 地址（有效范围：224.0.0.0 到 239.255.255.255）
dwPort	组播端口号
lpNicAddr	本地网卡 IP 地址，可为空字符串，表示使用默认网卡
dwTTL	组播数据包生存周期，为 0 表示使用系统默认的 TTL 值，

最大值为：255

dwFlags 功能标志，定义如下：

```

///< 加入多播组
#define BRAC_MCFLAGS_JOININGROUP    0x00000001L
///< 离开多播组
#define BRAC_MCFLAGS_LEAVEGROUP    0x00000002L
///< 数据发送标志，指示该多播组用于发送数据
#define BRAC_MCFLAGS_SENDDATA    0x00000010L
///< 数据接收标志，指示该多播组用于接收数据
#define BRAC_MCFLAGS_RECVDATA    0x00000020L
  
```

**备注：**

使用该方法可以加入或是离开一个多播组，从多播组接收数据，或是向多播组发送数据。功能标志可组合使用，如：

(BRAC\_MCFLAGS\_JOINGROUP | BRAC\_MCFLAGS\_SENDDATA) 表示加入一个多播组，并向该组发送数据，(BRAC\_MCFLAGS\_JOINGROUP | BRAC\_MCFLAGS\_RECVDATA) 表示加入一个多播组，并从该组接收数据。

当系统有多块网卡，而需要向其中某一块网卡所连接的网络发送组播数据包时，可通过 lpNicAddr 参数指定该网卡的 IP 地址来实现网络的选择，如果 lpNicAddr 为空字符串，则表示使用系统默认的网卡。

TTL 原本用来控制数据包在网络中的存活时间，防止由于路由器配置错误导致出现数据包传播的死循环，系统默认的组播数据包生存周期 (TTL) 值为 1，表示只能在发送主机所在的一个子网内的传送，不会通过路由器转发，如果需要穿透多重路由器，则上层应用程序需要设置一个合适的 TTL 值。

只有多媒体数据（音频、视频）才通过 IP 组播通道传递。

**AnyChat 默认关闭了 IP 组播数据发送的功能，要让本地的音视频数据通过 IP 组播通道传递，需要进行如下的操作：**

1、调用 BRAC\_SetSDKOption，使用“BRAC\_SO\_NETWORK\_MULTICASTPOLITIC”参数设置 IP 组播传输策略：

- 0 执行服务器路由策略，禁止组播发送[默认]
- 1 忽略服务器路由策略，只向组播组广播媒体流
- 2 执行服务器路由策略，同时组播)

2、调用 BRAC\_MultiCastControl 方法，加入一个，或多个组播组，并指定数据发送标志 (BRAC\_MCFLAGS\_JOINGROUP | BRAC\_MCFLAGS\_SENDDATA)

更多信息可参考在线文档：

<http://www.anychat.cn/faq/index.php?action=artikel&cat=2&id=203&artlang=zh>

## 5.6. 私聊函数

### 5.6.1. 更改当前的聊天模式

**DWORD BRAC\_ChangeChatMode(DWORD dwChatMode);**

**功能：**更改自己的聊天模式（默认为公聊模式）

**返回值：**0 表示成功，否则为出错代码

**参数:**

dwChatMod: 聊天模式, 定义为:

GV\_CHATMODE\_PUBLIC      0    公聊模式

GV\_CHATMODE\_PRIVATE     1    私聊模式

**备注:**

不同的聊天模式下对语音视频数据的发送方式采取了不同的策略, 公聊模式下可以接收所有的视频和语音信息, 而私聊模式下只接收与自己私聊的用户的语音和视频数据, 在私聊模式下, 自己的语音视频数据也只向私聊对象传送。

该方法必须在进入房间成功之后调用方才有效, 在私聊模式下, 接收不到其它用户的语音和视频数据, 仅能接收到自己私聊对象的语音和视频数据, 文字消息不受模式影响, 关于私聊模式与公聊模式之间的关系, 请参考“关于私聊的实现思路” 部分。

## 5.6.2. 获取指定用户聊天模式

**DWORD      BRAC\_GetUserChatMode(DWORD      dwUserid,      DWORD& dwChatMode);**

**功能:** 获取指定用户当前的聊天模式。

**返回值:** 0 表示成功, 否则为出错代码

**参数:**

dwUserid:      用户 ID 号, -1 表示获取自己的聊天模式;

dwChatMod:    聊天模式

**备注:**

该方法必须在进入房间成功之后调用方才有效。

## 5.6.3. 请求与对方私聊

**DWORD BRAC\_PrivateChatRequest(DWORD dwUserid);**

**功能:** 请求与对方私聊, 向对方发起私聊请求。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwUserid: 对方的用户 ID 号

**备注：**

该方法必须在进入房间成功之后调用方才有效。如果对方接受请求，则系统将会自动切换到私聊模式，调用该方法后，对方将会收到消息：WM\_GV\_PRIVATEREQUEST。

### 5.6.4. 回复对方的私聊请求

**DWORD BRAC\_PrivateChatEcho(DWORD dwUserid, DWORD dwRequestid, BOOL bAccept);**

**功能：**回复对方的私聊请求。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwUserid: 对方的用户 ID 号

dwRequestid: 请求 ID 号，在请求消息中传过来，标示是哪一个请求

bAccept: 是否同意对方的私聊请求

**备注：**

该方法必须在进入房间成功之后调用方才有效。当收到其它用户的私聊请求时，可以调用该方法同意，或是拒绝对方的请求，调用该方法后，对方将会收到消息：WM\_GV\_PRIVATEECHO。

### 5.6.5. 回复对方的私聊请求（扩展）

**DWORD BRAC\_PrivateChatEchoEx(DWORD dwUserid, DWORD dwRequestid, DWORD dwErrorCode);**

**功能：**回复对方的私聊请求。

**返回值：**0 表示成功，否则为出错代码

**参数:**

**dwUserId:** 对方的用户 ID 号

**dwRequestid:** 请求 ID 号，在请求消息中传过来，标示是哪一个请求

**dwErrorCode:** 是否同意对方的私聊请求(**errorcode=GV\_ERR\_SUCCESS** 表示同意与对方私聊，否则表示拒绝与对方私聊)

**备注:**

该方法与“BRAC\_PrivateChatEcho”功能相同，但是该方法可以让上层传入相应的出错代码，指示拒绝的原因，便于上层扩展，传入的 **errorcode** 将会附带在对方收到的 **WM\_GV\_PRIVATEECHO** 消息中。

## 5.6.6. 退出与某用户的私聊

**DWORD BRAC\_PrivateChatExit(DWORD dwUserId);**

**功能:** 退出与某用户的私聊，或者将某用户从自己的私聊列表中清除。

**返回值:** 0 表示成功，否则为出错代码

**参数:**

**dwUserId:** 对方的用户 ID 号

**备注:**

该方法必须在进入房间成功之后调用方才有效。因为同时可以和多个用户之间进行私聊，所以断开与该用户的私聊连接后，不影响与其它用户的私聊，调用该方法后，自己和对方都将会收到消息：**WM\_GV\_PRIVATEEXIT**。

## 5.7. 好友列表

自 AnyChat for Linux SDK V1.9 版本开始，AnyChat 提供了一个轻量级的用户好友解决方案，可以实现大厅好友列表、好友分组以及好友在线状态同步等功能。

好友列表需要业务服务器的支持，详情请参考《AnyChat Server SDK 开发指南》以及示例程序（AnyChatCallCenter）源代码。



### 5.7.1. 获取好友列表

**DWORD BRAC\_GetUserFriends(LPDWORD lpUserIDArray, DWORD& dwUserNum);**

**功能：**获取本地用户的好友列表，返回好友用户 ID 列表数组。

**返回值：**0 表示成功，否则为出错代码

**参数：**

lpUserIDArray 保存好友 ID 列表的缓冲区，可以为 NULL，表示只获取好友数量；

dwUserNum 好友 ID 列表缓冲区的大小，返回实际的好友数量

**备注：**

当客户端登录系统成功之后，会触发异步消息：WM\_GV\_LOGINSYSTEM，同时服务器会向客户端发送用户好友信息，当客户端接收完成之后，会触发客户端的异步消息：WM\_GV\_USERINFOUPDATE，且该消息的 lParam 为 0，表示好友列表有更新。所以该 API 通常在接收到 WM\_GV\_USERINFOUPDATE 异步消息之后调用。

### 5.7.2. 获取好友在线状态

**DWORD BRAC\_GetFriendStatus(DWORD dwFriendUserId, DWORD& dwStatus);**

**功能：**获取本地用户的好友在线状态，根据状态可以知道好友是否在线。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwFriendUserId 好友的用户 ID；

dwStatus 返回该好友的在线状态：0 离线， 1 在线

**备注：**

登录成功之后调用有效。当 dwFriendUserId 不正确时，函数返回值为出错代码，只有当函数返回值为 0 时，才能正确的获取好友在线状态值。

### 5.7.3. 获取好友分组列表

**DWORD BRAC\_GetUserGroups(LPDWORD lpGroupIDArray, DWORD& dwGroupNum);**

**功能：**获取本地用户的好友分组列表，返回好友分组 ID 列表数组。

**返回值：**0 表示成功，否则为出错代码

**参数：**

lpGroupIDArray 保存好友分组 ID 列表的缓冲区，可以为 NULL，表示只获取好友分组数量；

dwGroupNum 好友分组 ID 列表缓冲区的大小，返回实际的好友分组数量

**备注：**

登录成功之后调用有效。好友分组是指将好友归纳到某一个组别下，如“家人”、“大学同学”以及“老师”等。每一个分组对应一个分组 ID，通过分组 ID 可以获取分组的名称。

### 5.7.4. 获取分组名称

**DWORD BRAC\_GetGroupName(DWORD dwGroupId, TCHAR\* lpGroupName, DWORD dwLen);**

**功能：**根据分组 ID 获取分组名称。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwGroupId 分组 ID；

lpGroupName 保存分组名称的缓冲区

dwLen 保存分组名称缓冲区的大小

**备注：**

登录成功之后调用有效。分组名称由业务服务器设置。

### 5.7.5. 获取分组所对应的用户列表

**DWORD BRAC\_GetGroupFriends(DWORD dwGroupId, LPDWORD lpUserIDArray, DWORD& dwUserNum);**

**功能：**获取分组所对应的用户列表，即该分组下有多少用户。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwGroupId	分组 ID;
lpUserIDArray	保存用户 ID 列表的缓冲区，可以为 NULL，表示只获取用户数量;
dwUserNum	保存用户 ID 列表缓冲区的大小，返回实际的用户数量

**备注：**

登录成功之后调用有效。通过该 API 接口，可以获得每一个分组下面的用户列表，进而可以获得该分组下每一个用户的详细信息。

### 5.7.6. 获取好友用户信息

**DWORD BRAC\_GetUserInfo(DWORD dwUserId, DWORD dwInfoId, TCHAR\* lpInfoValue, DWORD dwLen);**

**功能：**获取好友用户的详细信息。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwUserId	好友用户 ID;
dwInfoId	用户信息类型 ID，业务层可自定义;
lpInfoValue	保存用户信息的缓冲区;
dwLen	保存用户信息缓冲区的大小;

**备注：**

登录成功之后调用有效。当业务服务器调用 API: BRAS\_SetUserInfo 设置了用户的信息之后，客户端便可以通过该 API 获取业务服务器所设置的信息，其中

dInfoId 由业务层（上层应用）自己定义。

关于好友用户信息这一部分，对于 AnyChat 来说是透明的，业务服务器设置了什么样的信息，客户端便可以获取到什么样的信息，AnyChat 只是提供了一个信息传输的中间通道，业务层可以自由扩展。

### 5.7.7. 用户信息控制

**DWORD BRAC\_UserInfoControl(DWORD dwUserId, DWORD dwCtrlCode, DWORD wParam, DWORD lParam, LPCTSTR lpStrValue);**

**功能：**对用户信息进行控制。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwUserId          用户 ID；

dwCtrlCode        控制代码，业务层自定义，其中<100 的值为系统保留，业务层使用时，其值必须>100

wParam            附带参数，业务层自定义；

lParam            附带参数，业务层自定义；

lpStrValue        附带参数（字符串类型），业务层自定义，可为 NULL；

**备注：**

登录成功之后调用有效。该 API 调用之后，会向业务服务器发送信息控制指令，将会触发业务服务器对应的回调函数。

## 5.8. 好友列表

自 AnyChat for Linux SDK V4.9 版本开始，AnyChat 提供了一个轻量级的用户好友解决方案，可以实现大厅好友列表、好友分组以及好友在线状态同步等功能。

好友列表需要业务服务器的支持，详情请参考《AnyChat Server SDK 开发指南》以及示例程序（AnyChatCallCenter）源代码。

### 5.8.1. 获取好友列表

**DWORD BRAC\_GetUserFriends(LPDWORD lpUserIDArray, DWORD& dwUserNum);**

**功能：**获取本地用户的好友列表，返回好友用户 ID 列表数组。

**返回值：**0 表示成功，否则为出错代码

**参数：**

lpUserIDArray 保存好友 ID 列表的缓冲区，可以为 NULL，表示只获取好友数量；

dwUserNum 好友 ID 列表缓冲区的大小，返回实际的好友数量

**备注：**

当客户端登录系统成功之后，会触发异步消息：WM\_GV\_LOGINSYSTEM，同时服务器会向客户端发送用户好友信息，当客户端接收完成之后，会触发客户端的异步消息：WM\_GV\_USERINFOUPDATE，且该消息的 lParam 为 0，表示好友列表有更新。所以该 API 通常在接收到 WM\_GV\_USERINFOUPDATE 异步消息之后调用。

### 5.8.2. 获取好友在线状态

**DWORD BRAC\_GetFriendStatus(DWORD dwFriendUserId, DWORD& dwStatus);**

**功能：**获取本地用户的好友在线状态，根据状态可以知道好友是否在线。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwFriendUserId 好友的用户 ID；

dwStatus 返回该好友的在线状态：0 离线， 1 在线

**备注：**

登录成功之后调用有效。当 dwFriendUserId 不正确时，函数返回值为出错代码，只有当函数返回值为 0 时，才能正确的获取好友在线状态值。

### 5.8.3. 获取好友分组列表

**DWORD BRAC\_GetUserGroups(LPDWORD lpGroupIDArray, DWORD& dwGroupNum);**

**功能：**获取本地用户的好友分组列表，返回好友分组 ID 列表数组。

**返回值：**0 表示成功，否则为出错代码

**参数：**

**lpGroupIDArray**      保存好友分组 ID 列表的缓冲区，可以为 NULL，表示只获取好友分组数量；

**dwGroupNum**          好友分组 ID 列表缓冲区的大小，返回实际的好友分组数量

**备注：**

登录成功之后调用有效。好友分组是指将好友归纳到某一个组别下，如“家人”、“大学同学”以及“老师”等。每一个分组对应一个分组 ID，通过分组 ID 可以获取分组的名称。

### 5.8.4. 获取分组名称

**DWORD BRAC\_GetGroupName(DWORD dwGroupId, TCHAR\* lpGroupName, DWORD dwLen);**

**功能：**根据分组 ID 获取分组名称。

**返回值：**0 表示成功，否则为出错代码

**参数：**

**dwGroupId**            分组 ID；

**lpGroupName**        保存分组名称的缓冲区

**dwLen**                保存分组名称缓冲区的大小

**备注：**

登录成功之后调用有效。分组名称由业务服务器设置。

### 5.8.5. 获取分组所对应的用户列表

**DWORD BRAC\_GetGroupFriends(DWORD dwGroupId, LPDWORD lpUserIDArray, DWORD& dwUserNum);**

**功能：**获取分组所对应的用户列表，即该分组下有多少用户。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwGroupId	分组 ID；
lpUserIDArray	保存用户 ID 列表的缓冲区，可以为 NULL，表示只获取用户数量；
dwUserNum	保存用户 ID 列表缓冲区的大小，返回实际的用户数量

**备注：**

登录成功之后调用有效。通过该 API 接口，可以获得每一个分组下面的用户列表，进而可以获得该分组下每一个用户的详细信息。

### 5.8.6. 获取好友用户信息

**DWORD BRAC\_GetUserInfo(DWORD dwUserId, DWORD dwInfoId, TCHAR\* lpInfoValue, DWORD dwLen);**

**功能：**获取好友用户的详细信息。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwUserId	好友用户 ID；
dwInfoId	用户信息类型 ID，业务层可自定义；
lpInfoValue	保存用户信息的缓冲区；
dwLen	保存用户信息缓冲区的大小；

**备注：**

登录成功之后调用有效。当业务服务器调用 API：BRAS\_SetUserInfo 设置了用户的信息之后，客户端便可以通过该 API 获取业务服务器所设置的信息，其中

dInfoId 由业务层（上层应用）自己定义。

关于好友用户信息这一部分，对于 AnyChat 来说是透明的，业务服务器设置了什么样的信息，客户端便可以获取到什么样的信息，AnyChat 只是提供了一个信息传输的中间通道，业务层可以自由扩展。

### 5.8.7. 用户信息控制

**DWORD BRAC\_UserInfoControl(DWORD dwUserId, DWORD dwCtrlCode, DWORD wParam, DWORD lParam, LPCTSTR lpStrValue);**

**功能：**对用户信息进行控制。

**返回值：**0 表示成功，否则为出错代码

**参数：**

dwUserId          用户 ID;

dwCtrlCode        控制代码，业务层自定义，其中<100 的值为系统保留，业务层使用时，其值必须>100

wParam            附带参数，业务层自定义;

lParam            附带参数，业务层自定义;

lpStrValue        附带参数（字符串类型），业务层自定义，可为 NULL;

**备注：**

登录成功之后调用有效。该 API 调用之后，会向业务服务器发送信息控制指令，将会触发业务服务器对应的回调函数。

## 5.9. 系统设置

### 5.9.1. 显示本地视频调节对话框

**DWORD BRAC\_ShowLVProperty(HWND hParent, LPCTSTR szCaption, DWORD dwX, DWORD dwY);**



**功能：**显示本地视频画面调节对话框，可以调节亮度，对比度，灰度等。

**返回值：**0 表示成功，否则为出错代码

**参数：**

hParent      父窗口句柄，视频调节对话框将在该窗口上弹出；  
szCaption    属性框的标题，默认为 NULL；  
dwX          显示的横坐标，相对于父窗体，默认为 0；  
dwY          显示的纵坐标，相对于父窗体，默认为 0；

**备注：**

不同摄像头的调节对话框是不同的，该对话框由摄像头的驱动控制，本 SDK 无法单独控制里面的各项参数调节

## 5.9.2. 枚举本地视频采集设备

**DWORD BRAC\_EnumVideoCapture(TCHAR\*\* lpDeviceName, DWORD& 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，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，具体使用方法请参考 Demo 程序中的源代码。

## 5.9.3. 选择指定的视频采集设备

**DWORD BRAC\_SelectVideoCapture(LPCTSTR 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。

## 5.9.4. 获取当前视频采集设备

**DWORD BRAC\_GetCurVideoCapture(TCHAR\* lpDeviceName, DWORD dwLen);**

**功能：**获取当前使用的视频采集设备名称

**返回值：**0 表示成功，否则为出错代码

**参数：**

lpDeviceName    保存视频采集设备名称的缓冲区；

dwLen            保存名称的缓冲区长度；

**备注**

缓冲区由外部分配，并通过 dwLen 参数传递缓冲区的大小。

## 5.9.5. 枚举本地音频采集设备

**DWORD BRAC\_EnumAudioCapture(TCHAR\*\* lpDeviceName, DWORD& 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，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，具体使用方法请参考 Demo 程序中的源代码。

## 5.9.6. 选择指定的音频采集设备

**DWORD BRAC\_SelectAudioCapture(LPCTSTR szDeviceName);**

**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：**

szDeviceName      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。

## 5.9.7. 获取当前音频采集设备

**DWORD BRAC\_GetCurAudioCapture(TCHAR\* lpDeviceName, DWORD dwLen);**

**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：**

lpDeviceName    保存音频采集设备名称的缓冲区；

dwLen            保存名称的缓冲区长度；

## 备注：

缓冲区由外部分配，并通过 dwLen 参数传递缓冲区的大小。

### 5.9.8. 枚举本地音频播放设备

**DWORD BRAC\_EnumAudioPlayback(TCHAR\*\* lpDeviceName, DWORD& 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，具体使用方法请参考 Demo 程序中的源代码。

### 5.9.9. 选择指定的音频播放设备

**DWORD BRAC\_SelectAudioPlayback(LPCTSTR 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。

### 5.9.10. 获取当前音频播放设备

**DWORD BRAC\_GetCurAudioPlayback(TCHAR\* lpDeviceName, DWORD**

**dwLen);**

**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：**

lpDeviceName 保存音频播放设备名称的缓冲区；

dwLen 保存名称的缓冲区长度；

**备注：**

缓冲区由外部分配，并通过 dwLen 参数传递缓冲区的大小。

### 5.9.11. 获取音频设备的当前音量

**DWORD BRAC\_AudioGetVolume(BRAC\_AudioDevice device, DWORD& 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。

### 5.9.12. 设置指定音频设备的音量

**DWORD BRAC\_AudioSetVolume(BRAC\_AudioDevice device, DWORD 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。

### 5.9.13. SDK内核参数设置

**DWORD BRAC\_SetSDKOption(int optname, const char FAR\* optval, int optlen);**

**功能：**SDK 内核参数设置

**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

**备注**

可以通过该方法对 AnyChat Core SDK 内部的参数进行设置，实现特殊的功能要求。

目前提供的可设置内核参数名称代码见下表：

信息代码定义	参数类型	用途	备注
BRAC_SO_AUDIO_VADCTRL	DWORD	音频静音检测控制	0: 关闭 1: 打开（默认）
BRAC_SO_AUDIO_NSCTRL	DWORD	音频噪音抑制控制	0: 关闭 1: 打开（默认）
BRAC_SO_AUDIO_ECHOCTRL	DWORD	音频回音消除控制	0: 关闭 1: 打开（默认）

BRAC_SO_AUDIO_AGCCTRL	DWORD	音频自动增益控制 (*)	0: 关闭 1: 打开 (默认)
BRAC_SO_AUDIO_CAPTUREMODE	DWORD	音频采集模式设置	0 发言模式 1 放歌模式 2 卡拉 OK 模式 3 线路输入模式
BRAC_SO_AUDIO_MICBOOST	DWORD	音频采集 Mic 增强控制	0 取消 Mic 增强 1 选中 Mic 增强
BRAC_SO_AUDIO_AUTOPARAM	DWORD	根据音频采集模式，自动选择合适的相关参数，包括编码器、采样参数、码率参数等	1 启用 0 关闭[默认]
BRAC_SO_AUDIO_MONOBITRATE	DWORD	设置单声道模式下音频编码目标码率	单位: bps
BRAC_SO_AUDIO_STEREOBITRATE	DWORD	设置双声道模式下音频编码目标码率	单位: bps
BRAC_SO_AUDIO_SOFTVOLMODE	DWORD	设置软件音量模式控制	0 关闭[默认], 1 打开 启用软件音量模式将不会改变系统的音量设置
BRAC_SO_RECORD_VIDEOBR	DWORD	录像视频码率设置	单位: bps
BRAC_SO_RECORD_AUDIOBR	DWORD	录像音频码率设置	单位: bps
BRAC_SO_RECORD_TMPDIR	PCHAR	录像文件临时目录设置	有效的目录完整路径
BRAC_SO_SNAPSHOT_TMPDIR	PCHAR	快照文件临时目录设置	有效的目录完整路径
BRAC_SO_CORESDK_TMPDIR	PCHAR	SDK 临时文件目录	便于保存临时文件
BRAC_SO_CORESDK_PATH	PCHAR	SDK 相关组件路径	便于 SDK 自动注册组件

BRAC_SO_CORESDK_DUMP_COREINFO	DWORD	输出内核信息到日志文件中，	便于分析故障原因
BRAC_SO_LOCALVIDEO_BITRATECTRL	DWORD	视频编码码率	单位 bps, VideoBitrate
BRAC_SO_LOCALVIDEO_QUALITYCTRL	DWORD	视频编码质量因子	VideoQuality
BRAC_SO_LOCALVIDEO_GOPCTRL	DWORD	视频编码关键帧间隔	VideoGOPSize
BRAC_SO_LOCALVIDEO_FPSCTRL	DWORD	视频编码帧率	VideoFps
BRAC_SO_LOCALVIDEO_PRESETCTRL	DWORD	视频编码预设参数	取值范围：1-5
BRAC_SO_LOCALVIDEO_APPLYPARAM	DWORD	应用视频编码参数，使得设置的参数生效	1 使用新参数， 0 使用默认参数
BRAC_SO_LOCALVIDEO_VIDEOSIZEPOLITIC	DWORD	本地视频采集分辨率控制策略	0 自动向下逐级匹配[默认]； 1 使用采集设备默认分辨率
BRAC_SO_LOCALVIDEO_DEINTERLACE	DWORD	本地视频反交织参数控制	0 不进行反交织处理[默认]；1 反交织处理
BRAC_SO_LOCALVIDEO_WIDTHCTRL	DWORD	视频采集宽度设置	VideoWidth
BRAC_SO_LOCALVIDEO_HEIGHTCTRL	DWORD	视频采集高度设置	VideoHeight
BRAC_SO_LOCALVIDEO_ROTATECTRL	DWORD	本地视频旋转控制	0 表示不进行旋转[默认] 1 表示垂直翻转
BRAC_SO_NETWORK_P2PPOLITIC	DWORD	本地网络 P2P 策略控制	0 禁止本地 P2P 1 服务器控制 P2P[默认]
BRAC_SO_NETWORK_P2PCONNECT	DWORD	尝试与指定用户建立 P2P 连接	目标用户 ID
BRAC_SO_NETWORK_P2PBREAK	DWORD	断开与指定用户的 P2P 连接	目标用户 ID[暂不支持，保留]
BRAC_SO_NETWORK_TCPSERVICEPORT	DWORD	设置本地 TCP 服务端端口	连接服务器之前设置有效
BRAC_SO_NETWORK_UDPSERVICEPORT	DWORD	设置本地 UDP 服务端	连接服务器之前设置有效



		口	效
BRAC_SO_PROXY_FUNCTIONCTRL	DWORD	本地用户代理功能控制	1 启动代理 0 关闭代理[默认]
BRAC_SO_PROXY_VIDEOCTRL	DWORD	本地用户代理视频控制	参数为被代理用户 ID
BRAC_SO_PROXY_AUDIOCTRL	DWORD	本地用户代理音频控制	参数为被代理用户 ID
BRAC_SO_VIDEOSHOW_DRIVERCTRL	DWORD	视频显示驱动控制	0 默认驱动 1 DirectShow 驱动 2 Windows GDI
BRAC_SO_NETWORK_MULTICASTPOLITIC	DWORD	组播策略控制	0 仅服务器路由策略，禁止组播发送[默认]； 1 禁止服务器路由策略，只向组播组广播 2 执行服务器路由策略，同时组播
BRAC_SO_CORESDK_EXTVIDEOINPUT	DWORD	外部视频输入控制	0 关闭外部视频输入[默认] 1 启用外部视频输入
BRAC_SO_CORESDK_EXTAUDIOINPUT	DWORD	外部音频输入控制	0 关闭外部音频输入[默认] 1 启用外部音频输入
BRAC_SO_STREAM_MAXBUFFERTIME	DWORD	最大流媒体缓冲时间	500ms – 5000ms，默认 800ms
BRAC_SO_STREAM_SMOOTHPLAYMODE	DWORD	平滑播放模式控制	0 关闭[默认]，丢包时视频短暂停顿，不会出现马赛克； 1 打开，流畅播放，丢包时可能出现马赛克

**注 1:** 自动增益能提高 Mic 输入信号的能量（音量），使得说话声音较小时，也能自动放大，提升用户体验，启用自动增益后，也会带来一定的副作用，如当说话声音较小，或是用户环境噪音较大时，少部分背景噪音也将会被放大，影响用户感受，故上层软件应为用户提供界面，打开（关闭）自动增益功能。

**注 2:** “静音检测”、“回音消除”、“噪音抑制”以及“自动增益”等语音特效处理主要是针对说话的声音进

行改进，如果需要通过“立体声混音”播放歌曲，则需要关闭这些语音特效处理（SDK 初始化之后关闭才能生效，可动态更改），否则将会导致所播放的歌曲音质变差。

注 3：路径（目录）相关内核参数设置，可在 SDK 初始化之前调用，其它的内核参数都必须在 SDK 初始化成功之后再调用才能有效。

注 4：有关视频质量相关参数设置，请参考在线文档：

<http://www.anychat.cn/faq/index.php?action=artikel&cat=2&id=175&artlang=zh>

注 5：有关用户流媒体数据代理功能，请参考在线文档：

<http://www.anychat.cn/faq/index.php?action=artikel&cat=2&id=178&artlang=zh>

注 6：有关本地网络 P2P 策略控制，请参考在线文档：

<http://www.anychat.cn/faq/index.php?action=artikel&cat=2&id=179&artlang=zh>

注 7：有关音频通信质量设置，请参考在线文档：

<http://www.anychat.cn/faq/index.php?action=artikel&cat=2&id=187&artlang=zh>

注 8：有关音频工作模式功能，请参考在线文档：

<http://www.anychat.cn/faq/index.php?action=artikel&cat=2&id=185&artlang=zh>

注 9：有关视频显示驱动设置，请参考在线文档：

<http://www.anychat.cn/faq/index.php?action=artikel&cat=2&id=196&artlang=zh>

注 10：有关 IP 组播的应用，请参考在线文档：

<http://www.anychat.cn/faq/index.php?action=artikel&cat=2&id=203&artlang=zh>

注 11：有关外部音视频输入功能，请参考在线文档：

<http://www.anychat.cn/faq/index.php?action=artikel&cat=2&id=254&artlang=zh>

注 12：有关最大流媒体缓冲时间以及平滑播放模式，请参考在线文档：

<http://www.anychat.cn/faq/index.php?action=artikel&cat=2&id=263&artlang=zh>

## 5.9.14. SDK 内核参数状态查询

**DWORD BRAC\_GetSDKOption(int optname, const char FAR\* 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

**备注**

可以通过该方法对 AnyChat Core SDK 内部的参数进行状态查询，获取当前的设置。

目前提供的可查询内核参数名称代码同“BRAC\_SetSDKOption”所附表格。



## 6.附录一：错误代码参考

#define GV_ERR_SUCCESS	0	///< 成功
#define GV_ERR_DB_ERROR	1	///< 数据库错误
#define GV_ERR_NOTINIT	2	///< 系统没有初始化
#define GV_ERR_NOTINROOM	3	///< 还未进入房间
#define GV_ERR_FUNCNOTALLOW	20	///< 函数功能不允许
#define GV_ERR_FUNCOPTERROR	21	///< 函数参数设置不正确
//连接部分		
#define GV_ERR_CONNECT_TIMEOUT	100	///< 连接服务器超时
#define GV_ERR_CONNECT_ABORT	101	///< 与服务器的连接中断
#define GV_ERR_CONNECT_AUTHFAIL	102	///< 未能通过服务器的认证,属于非法连接
//登录部分		
#define GV_ERR_CERTIFY_FAIL	200	///< 认证失败,用户名或密码有误
#define GV_ERR_ALREADY_LOGIN	201	///< 该用户已登录
#define GV_ERR_ACCOUNT_LOCK	202	///< 帐户已被暂时锁定
#define GV_ERR_IPADDR_LOCK	203	///< IP 地址已被暂时锁定
#define GV_ERR_VISITOR_DENY	204	///< 游客登录被禁止
#define GV_ERR_INVALID_USERID	205	///< 无效的用户 ID (用户不存在)
#define GV_ERR_SERVERSDK_FAIL	206	///< 与业务服务器连接失败,禁止登录
#define GV_ERR_SERVERSDK_TIMEOUT	207	///< 业务服务器执行任务超时
#define GV_ERR_NOTLOGIN	208	///< 没有登录
//进入房间		
#define GV_ERR_ROOM_LOCK	300	///< 房间已被锁住,禁止进入
#define GV_ERR_ROOM_PASSERR	301	///< 房间密码错误,禁止进入
#define GV_ERR_ROOM_FULLUSER	302	///< 房间已满员,不能进入
#define GV_ERR_ROOM_INVALID	303	///< 房间不存在
#define GV_ERR_ROOM_EXPIRE	304	///< 房间服务时间已到期
#define GV_ERR_ROOM_REJECT	305	///< 房主拒绝进入
#define GV_ERR_ROOM_OWNERBEOUT	306	///< 房主不在,不能进入房间
#define GV_ERR_ROOM_ENTERFAIL	307	///< 不能进入房间
#define GV_ERR_ROOM_ALREADYIN	308	///< 已经在房间里面了
//私聊		
#define GV_ERR_ROOM_PRINULL	401	///< 用户已经离开房间
#define GV_ERR_ROOM_REJECTPRI	402	///< 用户拒绝了私聊邀请
#define GV_ERR_ROOM_PRIDENY	403	///< 不允许与该用户私聊,或是用户禁止私聊
#define GV_ERR_ROOM_PRIREQIDERR	420	///< 私聊请求 ID 号错误,或请求不存在
#define GV_ERR_ROOM_PRIALRCHAT	421	///< 已经在私聊列表中

更多错误代码可参考：**SDK\Client\C++\GVErrorCodeDefine.h** 文件。