

# AnyChat for Web SDK

## 开发手册

(版本: V1.7)



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>

2013年03月

# 目 录

<b>1. 系统概述 .....</b>	<b>5</b>
1.1    系统介绍 .....	5
1.2    系统特性 .....	5
1.2.1    视频技术 .....	6
1.2.2    音频技术 .....	6
1.2.3    P2P技术 .....	6
1.3    关于佰锐科技 .....	7
1.4    技术支持 .....	7
1.5    版权申明 .....	8
<b>2. 编程指南 .....</b>	<b>9</b>
2.1.    客户端SDK概述 .....	9
2.2.    函数调用顺序 .....	10
2.3.    编程要点 .....	12
2.4.    服务器SDK概述 .....	13
<b>3. 事件(EVENT)说明 .....</b>	<b>14</b>
3.1.    基本流程事件回调函数 .....	14
3.1.1.    事件定义 .....	14
3.1.2.    网络连接事件 .....	14
3.1.3.    用户登录事件 .....	15
3.1.4.    进入房间事件 .....	15
3.1.5.    房间在线用户事件 .....	15
3.1.6.    房间用户活动事件 .....	15
3.1.7.    网络连接关闭事件 .....	16
3.2.    状态变化事件回调函数 .....	16
3.2.1.    事件定义 .....	16
3.2.2.    音频设备状态改变事件 .....	16
3.2.3.    系统音量改变事件 .....	17
3.2.4.    视频设备状态改变事件 .....	17
3.2.5.    用户P2P连接改变事件 .....	17
3.2.6.    录像/拍照完成事件 .....	18
3.3.    私聊控制事件回调函数 .....	19
3.3.1.    事件定义 .....	19
3.3.2.    用户私聊请求事件 .....	19
3.3.3.    用户私聊回复事件 .....	19
3.3.4.    用户私聊退出事件 .....	19
3.4.    文字消息事件回调函数 .....	20
3.4.1.    事件定义 .....	20
3.4.2.    收到文字聊天消息事件 .....	20
3.5.    数据传输事件回调函数 .....	21

3.5.1. 事件定义 .....	21
3.5.2. 收到文件传输数据事件 .....	21
3.5.3. 收到透明通道数据事件 .....	22
3.5.4. 收到扩展透明通道数据事件 .....	22
3.5.5. 收到SDK Filter数据事件.....	22
<b>4. 函数说明 .....</b>	<b>24</b>
4.1. 初始化与资源释放 .....	24
4.1.1. 初始化SDK .....	24
4.1.2. 释放SDK资源 .....	24
4.2. 登录流程 .....	25
4.2.1. 设置服务器认证密码 .....	25
4.2.2. 连接服务器 .....	25
4.2.3. 登录系统 .....	25
4.2.4. 进入房间（根据房间编号） .....	26
4.2.5. 进入房间（根据房间名称） .....	27
4.2.6. 离开房间 .....	28
4.3. 音视频操作 .....	29
4.3.1. 用户音频控制 .....	29
4.3.2. 用户视频控制 .....	29
4.3.3. 设置视频显示位置 .....	30
4.4. 查询状态 .....	31
4.4.1. 查询摄像头的状态 .....	31
4.4.2. 查询用户音频设备采集状态 .....	31
4.4.3. 查询用户昵称 .....	32
4.4.4. 查询用户状态（字符串） .....	32
4.4.5. 查询用户状态（整型值） .....	32
4.5. 普通功能 .....	34
4.5.1. 获取SDK版本信息 .....	34
4.5.2. 获取当前房间在线用户列表 .....	34
4.5.3. 发送文字聊天消息 .....	34
4.5.4. 透明通道传送缓冲区 .....	35
4.5.5. 透明通道传送缓冲区扩展 .....	35
4.5.6. 传送文件 .....	36
4.5.7. 查询传输任务相关信息 .....	37
4.5.8. 发送SDK Filter通信数据 .....	37
4.5.9. 用户音视频录制 .....	38
4.5.10. 用户视频抓拍 .....	38
4.6. 私聊函数 .....	39
4.6.1. 更改当前的聊天模式 .....	39
4.6.2. 获取指定用户聊天模式 .....	40
4.6.3. 请求与对方私聊 .....	40
4.6.4. 回复对方的私聊请求 .....	40
4.6.5. 回复对方的私聊请求（扩展） .....	41
4.6.6. 4.6.5 回复对方的私聊请求（扩展） .....	41

---

4.6.7.    退出与某用户的私聊 .....	41
4.7.    系统设置 .....	42
4.7.1.    枚举本地采集设备 .....	42
4.7.2.    选择指定采集设备 .....	42
4.7.3.    获取当前采集设备 .....	42
4.7.4.    获取音频设备当前音量 .....	43
4.7.5.    设置指定音频设备音量 .....	43
4.7.6.    查询SDK内核参数（整形值） .....	43
4.7.7.    查询SDK内核参数（字符串值） .....	44
4.7.8.    SDK内核参数设置（整形值） .....	44
4.7.9.    SDK内核参数设置（字符串值） .....	44
<b>5.    附录一：错误代码参考 .....</b>	<b>46</b>

# 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 for Web SDK 是一套即时通讯开发平台（SDK），包含了音视频处理模块与 P2P 网络模块两大部分，是 AnyChat Platform Core SDK 的重要组成部分，专为 Web 浏览器设计，即可以作为 PC 上的即时通讯，也可以作为视频会议、网络教育、文件传输、即时抓拍等网络互动平台。AnyChat for Web SDK 支持所有主流浏览器，如 IE、Google Chrome、Firefox 火狐浏览器等。整个平台由广州佰锐网络科技有限公司独立研发，具有自主知识产权。

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。AnyChat for Web SDK 属于客户端 SDK，需要配合 Windows（或 Linux）的服务器程序才能正常工作。AnyChat for Web SDK 提供 JavaScript 编程接口。

## 1.2 系统特性

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

AnyChat 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。可参考“AnyChat for Windows SDK”包中服务器开发相关文档。

### 1.2.1 视频技术

视频制式: PAL-B

分辨率: 176×144 — 1920×1080 (可调节)

帧 率: 5~30 FPS (可调节)

视频编码器: H.264

视频流码率: 10kbps ~ 1000kbps (VBR)

支持硬件编码、解码 (可定制, 需提供对应平台的接口)

### 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: AAC

音频流码率: 6kbps ~ 128kbps

音效处理: 回音抑制 (AES)、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 <——> Cone NAPT

Cone NAPT <——> 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研究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、知识中心：<http://www.anychat.cn/faq/>
- 2、在线论坛：<http://bbs.chinavideo.org>
- 3、官方网站：<http://www.anychat.cn>
- 4、电子邮件：[service@bairuitech.com](mailto:service@bairuitech.com)
- 5、24 小时客服电话：+86 (020) 85276986、38109065、87691886
- 6、即时通讯：QQ (75513809)、MSN (leon\_luck@hotmail.com)

## 1.5 版权申明

“AnyChat for Web 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。

## 2. 编程指南

### 2.1. 客户端SDK概述

“AnyChat for Web SDK” 属于客户端插件（简称“客户端”），对上层应用提供纯 Javascript 脚本的调用接口，通过对 AnyChat for Windows SDK 的二次封装形成浏览器支持的插件，浏览器的 html 页面通过 JavaScript 脚本与插件进行交互，包括调用插件的方法，插件主动触发 html 页面上的 JavaScript 方法（事件）等。

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 for Web SDK 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 JavaScript 事件通知上层应用。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。

JavaScript 脚本通过调用 BRAC\_InitSDK 时，将会自动创建插件，并检测 AnyChat for Web SDK 插件的版本信息，当用户本地没有安装插件，或是插件版本过低时，将返回出错代码，上层应用可根据返回值来显示提示界面，提示用户更新插件。

AnyChat for Web SDK 提供了一个视频显示插件，JavaScript 脚本通过调用 BRAC\_SetVideoPos 时将会自动在指定的 div 标签中插入该插件，用于视频的显示。

上层应用在编写应用时，不需要向 html 页面中手工插入任何“object”标签，只需要包含 AnyChat for Web SDK 的两个 JavaScript 函数库：anychatsdk.js 和 anychatevent.js 即可。

## 2.2. 函数调用顺序

调用顺序	函数	功能	备注
A	BRAC_InitSDK	初始化系统	初始化
	BRAC_SetSDKOption	设置 SDK 内核参数	
	BRAC_SetSDKOptionInt	设置 SDK 整形值相关参数	
	BRAC_SetSDKOptionString	设置 SDK 字符串相关参数	
B	BRAC_SetServerAuthPass	设置服务器连接密码	进入系统
	BRAC_Connect	连接服务器	
	BRAC_Login	用户登录系统	
	BRAC_EnterRoom	进入房间	
	BRAC_EnterRoomEx	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.GetUserLevel	查询用户级别	
	BRAC_GetUserName	查询用户名称	
	BRAC_QueryUserStateInt	查询指定用户相关状态（整形）	
	BRAC_QueryUserStateString	查询指定用户相关状态（字符串）	
	BRAC_ShowLVPProperty	显示本地视频画面调节对话框	
	BRAC_EnumDevices	枚举本地设备信息	
	BRAC_SelectVideoCapture	选择指定的视频采集设备	
	BRAC_MultiCastControl	组播功能控制	
	BRAC_GetCurrentDevice	获取当前使用的视频采集设备	
	BRAC_UserCameraControl	操作本地/远程用户视频	
	BRAC_UserSpeakControl	操作本地/远程用户语音	
	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_StreamRecordCtrl	用户音视频录制	
	BRAC_SnapShot	对远程用户视频抓拍	
	BRAC_TransBuffer	透明通道传送缓冲区	
	BRAC_TransBufferEx	透明通道传送缓冲区扩展	
	BRAC_TransFile	传送文件	
	BRAC_SendSDKFilterData	发送 SDK Filter 通信数据	
	BRAC_QueryTransTaskInfo	查询传输任务相关信息	
	BRAC_CancelTransTask	取消传输任务	
	BRAC_SendTextMessage	传输文本信息	
	BRAC_ChangeChatMode	更改当前聊天模式	

	BRAC_GetUserChatMode	获取指定用户当前聊天模式	
	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_PrivateChatEchoEx	回复对方的私聊请求（扩展）	
	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	

## 2.3. 编程要点

AnyChat for Web SDK是佰锐科技专为浏览器设计的音视频应用SDK，支持大多数主流浏览器，如IE、Google Chrome、Firefox等。AnyChat for Web SDK采用佰锐核心插件和视频插件进行视频显示，上层应用在收到远程用户的音视频数据时候只需提供一个HTML <object>标签加载插件，然后将该标签通过 BRAC\_SetVideoPos 函数传递给SDK，当视频数据到达时，内核将会自动将视频显示到视频插件上，不需要上层应用来处理视频的显示。

在编程时应该注意插件的加载时机：当页面加载的时候检查浏览器是否安装插件并提示相关信息，当初始化SDK失败时，可根据返回值判断是用户本地没有安装插件，或是插件版本过低造成的，然后提示用户安装插件，或升级插件。

初始化 SDK 的同时内核会自动关联相关事件回调函数，当内核产生相关的事件（如连接服务器、登录系统、进入房间等）时，将会主动调用（触发）HTML 页面中的 JavaScript 对应方法（参考“事件说明”章节）

## 2.4. 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。

“AnyChat for Web SDK”属于客户端 SDK，需要配合 Windows（或 Linux）的服务器程序才能正常工作，服务器的二次开发也将是基于“AnyChat for Windows SDK”或“AnyChat for Linux SDK”所提供的服务器 SDK 进行开发。

## 3. 事件(Event)说明

AnyChat for Web SDK 通过事件回调函数实现与 Web 应用的状态更新和数据交互。SDK 的很多调用都是异步的，如连接服务器、登录操作、进入房间等，调用函数完成之后，需要等待对应事件回调函数来触发是否登录成功，总体原则是：需要异步操作的地方，都采用事件回调函数来实现。

根据不同的请求响应类型，事件也分为几大类，在实际的开发过程中，可根据具体情况处理这些事件回调函数。

事件回调函数定义在 `anychatevent.js` 脚本文件里。

### 3.1. 基本流程事件回调函数

AnyChat 采用消息通知的方式来告知系统的各种当前状态变化。如网络连接或关闭、用户登录、进入房间等基本流程事件。“AnyChat for Windows SDK”是通过 Windows 消息来通知主窗口，而“AnyChat for Web SDK”则是内核主动调用 Web 上的 JavaScript 方法来实现（可以理解为触发回调函数，或是触发事件）。

#### 3.1.1. 事件定义

```
function OnAnyChatConnect(bSuccess, errorcode)
function OnAnyChatLoginSystem(dwUserId, errorcode)
function OnAnyChatEnterRoom(dwRoomId, errorcode)
function OnAnyChatRoomOnlineUser(dwUserCount, dwRoomId)
function OnAnyChatUserAtRoom(dwUserId, bEnterRoom)
function OnAnyChatLinkClose(reason, errorcode)
```

#### 3.1.2. 网络连接事件

**事件： OnAnyChatConnect(bSuccess, errorcode)**

**参数：** bSuccess 表示是否连接成功； errorcode 表示连接失败错误代码；

**说明：**当客户端连接服务器时被触发。

### 3.1.3. 用户登录事件

事件: **OnAnyChatLoginSystem(dwUserId, errorCode)**

参数:

dwUserId 表示自己的用户 ID 号,当 errorCode 为 0 时有效

errorCode 出错代码, 可判断进入房间是否成功

说明: 当客户端登录服务器时被触发。

### 3.1.4. 进入房间事件

事件: **OnAnyChatEnterRoom(dwRoomId, errorCode)**

参数:

dwRoomId 表示进入的房间 ID 号

errorCode 出错代码, 可判断进入房间是否成功

说明: 当客户端请求进入房间时被触发。

### 3.1.5. 房间在线用户事件

事件: **OnAnyChatRoomOnlineUser(dwUserCount, dwRoomId)**

参数:

dwUserCount 表示当前房间的在线用户数 (包含自己)

dwRoomId 房间编号

说明: 房间在线用户消息, 进入房间后触发一次。收到该消息后, 便可对房间中的用户进行音视频的相关操作, 如请求音频、请求视频等。

### 3.1.6. 房间用户活动事件

事件: **OnAnyChatUserAtRoom(dwUserId, bEnterRoom)**

参数:

dwUserId 表示当前房间活动用户的 ID 号

bEnterRoom 表示进入或离开房间

**说明:** 当成功进入房间之后，有新的用户进入房间，或是房间用户离开房间，均会触发该接口。

### 3.1.7. 网络连接关闭事件

**事件: OnAnyChatLinkClose(reason, errorcode)**

**参数:** reason 表示连接被关闭的原因

**说明:** 当连接服务器成功之后，网络连接关闭时触发该事件。

## 3.2. 状态变化事件回调函数

状态变化事件回调函数包含了 SDK 的用户视频设备状态、音频设备状态、P2P 连接状态变化等异步事件的通知。

### 3.2.1. 事件定义

```
function OnAnyChatMicStateChange(dwUserId, State)
function OnAnyChatVolumeChange(device, dwCurrentVolume)
function OnAnyChatCameraStateChange(dwUserId, State)
function OnAnyChatP2PConnectState(dwUserId, State)
function OnAnyChatRecordSnapShot(dwUserId, lpFileName, dwParam,
                                bRecordType)
```

### 3.2.2. 音频设备状态改变事件

**事件: OnAnyChatMicStateChange(dwUserId, State)**

**参数:**

dwUserId 表示状态变化的用户 ID

State 表示该用户是否打开音频采集设备: (0 关闭, 1 打开)

0 音频设备关闭

1 音频设备打开

**说明:**

当进入房间成功之后，与其他用户会话建立的时候，当用户使用 API: BRAC\_UserSpeakControl 操作本地音频设备时将会触发该事件。

### 3.2.3. 系统音量改变事件

事件: **OnAnyChatVolumeChange(device, dwCurrentVolume)**

参数:

device	表示音频设备
dwCurrentVolume	表示音频设备音量

说明:

当进入房间成功后, 与其他用户正在进行会话的时候, 当用户使用 API:  
BRAC\_AudioSetVolume 操作音频音量的时候将会触发该事件。

### 3.2.4. 视频设备状态改变事件

事件: **OnAnyChatCameraStateChange(dwUserId, State)**

参数:

dwUserId	表示状态变化的用户 ID
State	表示该用户当前的视频设备状态;
0	没有摄像头设备
1	有摄像头设备, 但没有打开
2	已打开摄像头设备

说明: 当进入房间成功之后, 与其他用户会话建立的时候, 当用户使用 API:  
BRAC\_UserCameraControl 操作本地视频设备时将会触发该事件。

### 3.2.5. 用户P2P连接改变事件

事件: **OnAnyChatP2PConnectState(dwUserId, State)**

参数:

dwUserId	表示其它用户 ID 号
State	表示本地用户与其它用户的当前 P2P 网络连接状态:
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

说明: 当进入房间成功之后, 与其它用户建立 P2P 连接, 或是 P2P 连接被断开时触发该事件。

### 3.2.6. 录像/拍照完成事件

事件： **OnAnyChatRecordSnapShot(dwUserId, lpFileName, dwParam, bRecordType)**

参数：

dwUserId	表示其它用户 ID 号
lpFileName	表示保存文件名
dwParam	表示携带参数
bRecordType	表示保存文件类型（视频或抓拍图片）

说明：当与其他用户会话过程中，当用户使用 API: BRAC\_SnapShot 对远程用户录像或抓拍时将会触发该事件。

## 3.3. 私聊控制事件回调函数

### 3.3.1. 事件定义

```
function OnAnyChatPrivateRequest(dwUserId, dwRequestId)
function OnAnyChatPrivateEcho(dwUserId, errorcode)
function OnAnyChatPrivateExit(dwUserId, errorcode)
```

### 3.3.2. 用户私聊请求事件

事件: **OnAnyChatPrivateRequest(dwUserId, dwRequestId)**

参数:

dwUserId 表示发起者的用户 ID 号, 私聊请求者

dwRequestId 表示私聊请求编号, 标识该请求

说明: 当进入房间成功后, 用户发起私聊请求时, 在被请求者端将触发该事件。

### 3.3.3. 用户私聊回复事件

事件: **OnAnyChatPrivateEcho(dwUserId, errorcode)**

参数:

dwUserId 表示回复者的用户 ID 号

errorcode 返回代码, 表示对方是否同意自己的邀请

说明: 进入房间成功后, 当用户发起私聊请求时, 其它用户回复该请求时将触发该事件回调函数。

### 3.3.4. 用户私聊退出事件

事件: **OnAnyChatPrivateExit(dwUserId, errorcode)**

参数:

dwUserId 表示退出者的用户 ID 号

errorcode 返回代码, 表示退出私聊的原因

说明: 当进入房间成功之后, 在自己私聊列表中的用户退出, 或网络连接断开时将触发该事件回调函数。

## 3.4. 文字消息事件回调函数

### 3.4.1. 事件定义

```
function OnAnyChatTextMessage(dwFromUserId, dwToUserId, bSecret,  
                            lpMsgBuf, dwLen)
```

### 3.4.2. 收到文字聊天消息事件

事件: **OnAnyChatTextMessage(dwFromUserId, dwToUserId, bSecret, lpMsgBuf, dwLen)**

参数:

dwFromUserId	消息发送者用户 ID
dwToUserId	目标用户, -1 表示发送给大家, 即房间所有人
bSecret	是否为悄悄话, 当目标用户不为-1 时有效
lpMsgBuf	消息字符串
dwLen	消息字符串长度

说明: 当进入房间成功之后, 收到其他用户发送的文字聊天信息时将触发该事件回调函数, 本地用户向其它用户发送文字消息时, 将不会触发该事件。

## 3.5. 数据传输事件回调函数

### 3.5.1. 事件定义

```
function OnAnyChatTransFile(dwUserId, lpFileName, lpTempFilePath,
    dwFileLength, wParam, lParam, dwTaskId)
function OnAnyChatTransBuffer(dwUserId, lpBuf, dwLen)
function OnAnyChatTransBufferEx(dwUserId, lpBuf, dwLen, wParam, lParam,
    dwTaskId)
function OnAnyChatSDKFilterData(lpBuf, dwLen)
```

### 3.5.2. 收到文件传输数据事件

**事件： OnAnyChatTransFile(dwUserId, lpFileName, lpTempFilePath,  
dwFileLength, wParam, lParam, dwTaskId)**

**参数：**

dwUserId: 用户 ID，指示发送用户

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: 该次文件传输对应的任务编号

**说明：**

当收到其它用户使用“BRAC\_TransFile”方法发送的文件时，将会触发该事件回调函数。

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。

### 3.5.3. 收到透明通道数据事件

事件: **OnAnyChatTransBuffer(dwUserId, lpBuf, dwLen)**

参数:

- dwUserId: 用户 ID, 指示发送用户
- lpBuf: 缓冲区地址
- dwLen: 缓冲区大小

说明:

当收到其它用户使用“BRAC\_TransFile”方法发送的缓冲区数据时, 将会触发该事件回调函数。

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。

### 3.5.4. 收到扩展透明通道数据事件

事件: **OnAnyChatTransBufferEx(dwUserId, lpBuf, dwLen, wParam, lParam, dwTaskId)**

参数:

- dwUserId: 用户 ID, 指示发送用户
- 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

说明:

当收到其它用户使用“BRAC\_TransBufferEx”方法发送的缓冲区数据时, 将会触发该事件回调函数。

### 3.5.5. 收到SDK Filter数据事件

事件: **OnAnyChatSDKFilterData(lpBuf, dwLen)**

**参数:**

lpBuf: 缓冲区地址

dwLen: 缓冲区大小

**说明:**

当收到服务器“BRAC\_SDKFilterData”或是“Server SDK”相关接口发送的缓冲区数据时，将会触发该事件回调函数

## 4. 函数说明

### 4.1. 初始化与资源释放

#### 4.1.1. 初始化SDK

**函数:** **BRAC\_InitSDK(apilevel)**

**功能:** 初始化SDK。

**参数:**

apilevel 字符串值，最低 API 版本号，默认为“0”

**返回值:** 0 表示成功，否则为出错代码

**说明:**

当给定的版本高于SDK实际的版本时，函数会返回出错代码，指示版本过低，上层应用需要提示用户安装新的版本；

当用户本地没有安装插件时，函数会返回出错代码，初始化SDK失败；

初始化SDK时，该方法会自动创建插件的对象，并关联相关的异步（回调）事件。

apilevel 表示当前 API 版本，在 BRAC\_InitSDK 函数里，需要检查插件是否安装、判断插件版本是否太旧、创建 Anychat SDK 插件、参数 apilevel 判断当前 API 是否满足业务层需要和关联相关事件回调函数。

#### 4.1.2. 释放SDK资源

**函数:** **BRAC\_Logout()**

**说明:** 该函数最后被调用，当用户退出 AnyChat 系统的时候被调用并释放 SDK 内部所占用的系统资源。

## 4.2. 登录流程

### 4.2.1. 设置服务器认证密码

函数: **BRAC\_SetServerAuthPass(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。

参数: lpPassword 字符串值, 表示认证密码 (大小写敏感)

返回值: 0 表示成功, 否则为出错代码

说明: 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。

### 4.2.2. 连接服务器

函数: **BRAC\_Connect(lpServerAddr, dwPort)**

功能: 用于与服务器建立连接

参数:

lpServerAddr 字符串值, 服务器 IP 地址或网站域名 (URL) 地址;

dwPort 整形值, 服务器端口号 (默认端口号为: 8906);

返回值: 0 表示成功, 否则为出错代码;

说明: 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, 都将会通过相应的接口通知上层应用, 这里是一个异步的过程。

### 4.2.3. 登录系统

函数: **BRAC\_Login(lpUserName, lpPassword, dwParam)**

功能: 登录服务器, 请求身份验证;

**参数:**

lpUserName 字符串值，登录用户名  
lpPassword 字符串值，登录密码  
dwParam 整形值，备用参数（登录时传 0）

返回值: 0 表示成功，否则为出错代码；

**说明:**

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，都将会通过相应的接口通知上层应用，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》。

#### 4.2.4. 进入房间（根据房间编号）

函数: BRAC\_EnterRoom(dwRoomid, lpRoomPass, dwParam)

功能: 根据房间编号进入房间

**参数:**

dwRoomid 整形值，房间编号，系统唯一；

lpRoomPass 字符串值，房间密码（当房间需要密码时有效，如果没有可为空）；

dwParam 整形值，备用参数（进入房间时传 0）

**返回值：**0 表示成功，否则为出错代码；

**说明：**

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，都将会通过相应的接口通知上层应用，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\_PreparesEnterRoom，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\_PreparesEnterRoom\_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》。

#### 4.2.5. 进入房间（根据房间名称）

**函数：**BRAC\_EnterRoomEx(lpRoomName, lpRoomPass) {}

**功能：**根据房间名称进入房间

**参数:**

lpRoomName 字符串值，房间名称；

lpRoomPass 字符串值，房间密码（当房间需要密码时有效，如果没有可为空）；

返回值：0 表示成功，否则为出错代码；

**说明:**

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，通过基本事件回调函数返回给上层应用。

## 4.2.6. 离开房间

**函数:** BRAC\_LeaveRoom(dwRoomid)

**功能:** 离开房间

**参数:** dwRoomid 整形值，表示当前房间编号，-1 表示退出当前房间；

**返回值:** 0 表示成功，否则为出错代码

**说明:**

在用户退出系统前需要调用该函数告知业务服务器用户离开房间，或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。

## 4.3. 音视频操作

### 4.3.1. 用户音频控制

函数： BRAC\_UserSpeakControl(dwUserId, bOpen)

功能： 用户发言控制

参数：

dwUserId 整形值，用户编号，为-1 表示对本地发言进行控制

bOpen 整形值，是否打开（请求）音频设备（数据流）

返回值： 0 表示成功，否则为出错代码

说明：

当 dwUserId= -1（或自己的用户 ID）时，表示对本地用户的操作，否则表示对远程用户的操作。

对于本地用户，该方法是直接操作用户的音频设备，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音频设备。当对方已打开本地音频设备之后，调用该方法，可以获取到对方的音频数据流，内核会自动播放，即可以听到对方的声音。

### 4.3.2. 用户视频控制

函数： BRAC\_UserCameraControl(dwUserId, 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

参数：

dwUserId: 整形值，用户编号，为-1 表示对本地视频进行控制

bOpen 整形值，是否打开视频

返回值： 0 表示成功，否则为出错代码

说明：

当 dwUserId= -1（或自己的用户 ID）时，表示对本地用户的操作，否则表示对远程用户的操作。

对于本地用户，该方法是直接操作用户的视频设备（USB 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视频设备。当对方已打开本地视频设备之后，调用该方法，可以获取到对方的视频数据流，当上层调用 BRAC\_SetVideoPos 方法设置视频的位置后，内核会自动播放，显示对方的视频到指定的位置。

### 4.3.3. 设置视频显示位置

**函数:** BRAC\_SetVideoPos(userid, parentid, videoid)

**功能:** 设置视频显示位置，或是刷新视频显示

**参数:**

userid 整形值，用户编号，不能为-1

parentobj 整形值，表示显示视频位置的 div 层的 ID

videoed 整形值，表示视频显示插件的 ID

**返回值:** 0 表示成功，否则为出错代码

**说明:**

SDK 会自动在 parentid 所指示的 DIV 层内插入视频显示插件，并将该插件与用户 ID 进行关联，内核收到该用户 ID 的视频数据的时候，将会自动在新插入的插件中显示用户的视频。

## 4.4. 查询状态

### 4.4.1. 查询摄像头的状态

**函数:** BRAC\_GetCameraState(dwUserId)

**功能:** 查询用户摄像头的状态

**参数:** dwUserId 为用户编号, 为-1 时表示获取自己的摄像头状态;

**返回值:** 返回指定用户的摄像头状态, 定义为:

0 没有摄像头

1 有摄像头但没有打开

2 摄像头已打开

**说明:** 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, 可以判断用户当前摄像头的状态, 以及判断用户是否有摄像头。

### 4.4.2. 查询用户音频设备采集状态

**函数:** BRAC\_GetSpeakState(dwUserId)

**功能:** 查询用户音频设备采集状态

**参数:** dwUserId 为用户编号, 为-1 时表示获取自己的音频设备状态;

**返回值:** 返回指定用户的音频设备状态, 定义为:

0 音频采集关闭

1 音频采集开启

**说明:** 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。

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。

#### 4.4.3. 查询用户昵称

函数: BRAC\_GetUserName(dwUserId)

功能: 查询用户昵称

参数: dwUserId 为用户编号, 为-1 时表示获取自己的昵称;

返回值: 用户昵称字符串

说明:

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。

#### 4.4.4. 查询用户状态 (字符串)

函数: BRAC\_QueryUserStateInt(dwUserId, infoname)

功能: 查询指定用户状态 (字符串类型)

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(查看 AnyCht SDK 通用常量定义)

返回值: 相关状态的字符串

备注:

通过调用该方法, infoname 表示查询指定用户的状态类型, infoname 常量定义在 anychatsdk.js 文件里。通过参数 infoname 可以查询指定用户的相关状态值, 包括用户摄像头状态、音频设备状态、用户名、用户级别、用户远程 IP 地址等详细信息。

#### 4.4.5. 查询用户状态 (整型值)

函数: BRAC\_QueryUserStateInt(dwUserId, infoname)

功能: 查询指定用户状态 (字符串类型)

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（查看 AnyCht SDK 通用常量定义）

**返回值：**相关状态的整型值

**说明：**

通过调用该函数，**infoname** 表示查询指定用户的状态类型，**infoname** 常量定义在 **anychatsdk.js** 文件里。通过参数 **infoname** 可以查询指定用户的相关状态值（整型值），包括用户摄像头状态、音频设备状态、用户名、用户级别、用户远程 IP 地址等详细信息。

## 4.5. 普通功能

### 4.5.1. 获取SDK版本信息

函数: BRAC\_GetVersion(type)

功能: 获取 SDK 版本信息

参数: type 表示查询 SDK 信息类型 (包括主版本、从版本、编译时间)

BRAC\_SO\_CORESDK\_MAINVERSION =22 查询主版本

BRAC\_SO\_CORESDK\_SUBVERSION =23 查询从版本

BRAC\_SO\_CORESDK\_BUILDTIME =24 查询编译时间

返回值: SDK 版本信息

说明: 通过调用该函数、通过参数 type 查询 SDK 信息

### 4.5.2. 获取当前房间在线用户列表

函数: BRAC\_GetOnlineUser()

功能: 获取当前房间在线用户列表 (不包含自己)

参数: 无

返回值: 返回在线用户 ID 数组

说明:

获取在线用户列表，并不包含当前用户自己的 ID，自己的 ID 在登录事件中已通知给上层应用。

### 4.5.3. 发送文字聊天消息

函数: BRAC\_SendTextMessage(dwUserId, bSecret, lpMsgBuf)

功能: 向指定的用户传送文本消息

参数:

dwUserId 目标用户编号, -1 表示大家 (所有人)

bSecret 是否私聊, 如果是私聊, 则房间内其他用户看不到私聊信息

lpMsgBuf: 文字聊天字符串

**返回值:** 0 表示成功，否则为出错代码

**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，可以选择私聊。

收到对方消息后，会触发 OnAnyChatPrivateEcho 事件回调函数。

#### 4.5.4. 透明通道传送缓冲区

**函数:** BRAC\_TransBuffer(dwUserId, lpBuf)

**功能:** 透明通道传送缓冲区

**参数:**

dwUserId: 目标用户编号，-1 表示大家（用户当前房间所有人）

lpBuf: 缓冲区

**返回值:** 0 表示成功，否则为出错代码

**说明:**

可以利用该方法实现自定义功能，缓冲区采用透明传输，目标对象可以向房间用户发送文件，也可以是指定的对象发送文件。

该方法将会触发对方的 OnAnyChatTransBuffer 事件回调函数。

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）。

#### 4.5.5. 透明通道传送缓冲区扩展

**函数:** BRAC\_TransBufferEx(dwUserId, lpBuf, wParam, lParam, dwFlags)

**功能:** 扩展透明通道传送缓冲区

**参数:**

userid: 目标用户编号，只针对某一个用户，不能为-1（所有人）

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）

**返回值:** >0 表示任务 ID 号（可利用该 ID 查询文件传输进度），否则表示出错。

**说明:**

该函数与“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；

#### 4.5.6. 传送文件

**函数:** BRAC\_TransFile(dwUserId, lpLocalPathName, wParam, lParam, dwFlags)

**功能:** 传送文件给指定用户

**返回值:** >0 表示任务 ID 号（可利用该 ID 查询文件传输进度），否则表示出错。

**参数:**

**dwUserId:** 目标用户编号，只针对某一个用户，不能为 -1（所有人）

**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

**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。

### 4.5.7. 查询传输任务相关信息

**函数:** BRAC\_QueryTransTaskInfo(dwUserId, dwTaskId, infoname)

**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

**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（SDK 内核产生任务 ID）

infoname 需要查询的信息代码（见 anychatsdk.js 传输信息参数）

**返回值:** 0 表示查询成功，否则为出错代码

**说明:**

通过调用该函数，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
TRANSTASK_PROGRESS	DOUBLE	传输任务进度查询	0.0 ~ 100.0
TRANSTASK_BITRATE	INT	传输任务当前码率	单位：bps
TRANSTASK_STATUS	INT	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

### 4.5.8. 发送SDK Filter通信数据

**函数:** BRAC\_SendSDKFilterData(lpBuf)

**功能:** 向服务器发送 SDK Filter 通信数据

**参数:** lpBuf: 缓冲区

**返回值:** 0 表示成功，否则为出错代码

**说明:** 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 和服务器来说，都是透明的。

#### 4.5.9. 用户音视频录制

**函数:** BRAC\_StreamRecordCtrl(dwUserId, bStartRecord, dwFlags, dwParam)

**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

**参数:**

dwUserId 目标用户编号，-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，录像任务结束时，该参数将通过回调函数返回给上层应用。

**返回值:** 0 表示录制指令被 SDK 成功接收，否则为出错代码

**说明:**

该函数只是向 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。

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：

BRAC\_SO\_RECORD\_VIDEOBR 录制视频

BRAC\_SO\_RECORD\_AUDIOBR 录制音频

为了保证音视频录制功能的正常使用，在 SDK 初始化时必须做好如下准备工作：

1、注册录像完成通知回调函数，参考：OnAnyChatRecordSnapShot

2、设置录像文件保存路径，参考：BRAC\_SetSDKOption

如果需要调节录制文件的音视频质量，可通过调节录制码率来实现，参考方法：

BRAC\_SetSDKOption（注：必须在开始录制之前设置）。

#### 4.5.10. 用户视频抓拍

**函数:** BRAC\_SnapShot(dwUserId, dwFlags, dwParam)

**功能:** 对用户视频进行抓拍，保存为本地图片文件

**参数:**

dwUserId 目标用户编号, -1 表示本地用户（自己）；  
dwFlags: 抓拍功能标志, 参考备注；  
dwParam: 抓拍指令附带参数, 拍照任务结束时, 该参数将通过回调函数返回给上层应用。

**返回值:** 0 表示录制指令被 SDK 成功接收, 否则为出错代码

**说明:**

该函数只是向 SDK 下达抓拍任务, 当得到了图片数据时, SDK 将产生一次回调, 通知上层应用图片文件名。

为了保证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、注册图片完成通知回调函数, 参考: [OnAnyChatRecordSnapShot](#)
- 2、设置图片文件保存路径, 参考: [BRAC\\_SetSDKOption](#)

## 4.6. 私聊函数

### 4.6.1. 更改当前的聊天模式

**函数:** [BRAC\\_ChangeChatMode\(dwChatMode\)](#)

**功能:** 更改自己的聊天模式

**参数:** dwChatMode 参数常量定义为 [WM\\_GV\\_CHATMODECHG](#)

**返回值:** 0 表示成功, 否则为出错代码

**说明:**

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。

## 4.6.2. 获取指定用户聊天模式

**函数:** BRAC\_GetUserChatMode(dwUserId)

**功能:** 获取指定用户当前的聊天模式。

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

**返回值:** 获得指定用户当前聊天模式;

**说明:** 该方法必须在进入房间成功之后调用方才有效。

## 4.6.3. 请求与对方私聊

**函数:** BRAC\_PrivateChatRequest(dwUserId)

**功能:** 请求与对方私聊, 向对方发起私聊请求。

**参数:** dwUserId: 对方的用户 ID 号

**返回值:** 0 表示成功, 否则为出错代码

**说明:**

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。

## 4.6.4. 回复对方的私聊请求

**函数:** BRAC\_PrivateChatEcho(dwUserId, dwRequestId, bAccept)

**功能:** 回复对方的私聊请求。

**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

**返回值:** 0 表示成功, 否则为出错代码

**说明:**

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。

## 4.6.5. 回复对方的私聊请求（扩展）

### 4.6.6. 4.6.5 回复对方的私聊请求（扩展）

**函数:** BRAC\_PrivateChatEchoEx(dwUserId, dwRequestId, dwErrorCode)

**功能:** 回复对方的私聊请求。

**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)

**返回值:** 0 表示成功, 否则为出错代码

**说明:**

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。

## 4.6.7. 退出与某用户的私聊

**函数:** BRAC\_PrivateChatExit(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。

**参数:** dwUserId 对方的用户 ID 号

**返回值:** 0 表示成功, 否则为出错代码

**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。

## 4.7. 系统设置

### 4.7.1. 枚举本地采集设备

**函数:** BRAC\_EnumDevices(dwDeviceType)

**功能:** 枚举本地音视频采集设备

**参数:** dwDeviceType 采集设备类型、常量定义为:

```
BRAC_DEVICE_VIDEOCAPTURE = 1; //视频采集设备  
BRAC_DEVICE_AUDIOCAPTURE = 2; // 音频采集设备  
BRAC_DEVICE_AUDIOPLAYBACK = 3; // 音频播放设备
```

**返回值:** 返回设备名称数组

**说明:**

该函数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。

### 4.7.2. 选择指定采集设备

**函数:** BRAC\_SelectVideoCapture(dwDeviceType, szCaptureName)

**功能:** 选择指定的视频采集设备

**参数:**

dwDeviceType 采集设备类型（同上）

szCaptureName 所获取设备的名称；

**返回值:** 0 表示成功，否则为出错代码

**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。

### 4.7.3. 获取当前采集设备

**函数:** BRAC\_GetCurrentDevice(dwDeviceType)

**功能:** 获取当前使用的音视频采集设备名称

**参数:** dwDeviceType 采集设备类型

**返回值:** 返回当前采集设备名。

#### 4.7.4. 获取音频设备当前音量

**函数:** BRAC\_AudioGetVolume(device)

**功能:** 获取指定音频设备的当前音量

**参数:** device 当前音频设备

**返回值:** 当前设备音量: 0~100

**说明:** 获取音频设备当前音量。

#### 4.7.5. 设置指定音频设备音量

**函数:** BRAC\_AudioSetVolume(device, dwVolume)

**功能:** 设置指定音频设备的音量

**参数:**

device 设备类型, 定义为:

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

**返回值:** 0 表示成功, 否则为出错代码

**说明:**

根据设备类型 (device) 参数的不同, 可以调节设备音量大小。

#### 4.7.6. 查询SDK内核参数 (整形值)

**函数:** BRAC\_GetSDKOptionInt(optname) {

**功能:** SDK 内核参数查询 (整形值参数)

**参数:** optname 内核参数名称 (见 anychatsdk.js 内核参数类型定义)

**返回值:** 返回查询内核参数值

**说明:**

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。

#### 4.7.7. 查询SDK内核参数（字符串值）

**函数:** BRAC\_GetSDKOptionString(optname)

**功能:** SDK 内核参数查询（字符串值参数）

**参数:** optname 内核参数名称（见 anychatsdk.js 内核参数类型定义）

**返回值:** 返回查询内核参数值

**说明:**

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。

#### 4.7.8. SDK内核参数设置（整形值）

**函数:** BRAC\_SetSDKOptionInt(optname, value)

**功能:** SDK 内核参数设置

**参数:**

optname 内核参数名称（见 anychatsdk.js 内核参数类型定义）

value 设置的参数值，整型值

**返回值:** 0 表示成功，否则为出错代码

**说明:**

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

#### 4.7.9. SDK内核参数设置（字符串值）

**函数:** BRAC\_SetSDKOptionString(optname, value)

**功能:**     SDK 内核参数设置

**参数:**

optname     内核参数名称（见 anychatsdk.js 内核参数类型定义）

value        设置的参数值，字符串值

**返回值:** 0 表示成功，否则为出错代码

**说明:**

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

## 5.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(初始化时没有指定该功能)
//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    ///
```

更多错误代码可参考：[SDK\Client\C++\GVErrorCodeDefine.h](#) 文件。