

AnyChat for Android SDK

开发手册

(版本: V1.8)



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 07 月

目 录

| | |
|---------------------------------|-----------|
| 1. 系统概述 | 5 |
| 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 |
| 2. 编程指南 | 9 |
| 2.1. 客户端SDK概述 | 9 |
| 2.2. 函数调用顺序 | 10 |
| 2.3. 编程要点 | 11 |
| 2.3.1. SDK的Java Package | 11 |
| 2.3.2. SDK与Activity | 11 |
| 2.3.3. SurfaceView视频显示 | 11 |
| 2.3.4. 字符编码 | 11 |
| 2.4. 服务器SDK概述 | 13 |
| 3. 接口(INTERFACE)说明 | 14 |
| 3.1. 基本流程事件接口 | 14 |
| 3.1.1. 接口定义 | 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. 状态变化事件接口 | 16 |
| 3.2.1. 接口定义 | 16 |
| 3.2.2. 音频设备状态改变事件 | 17 |
| 3.2.3. 视频设备状态改变事件 | 17 |
| 3.2.4. 用户聊天模式改变事件 | 17 |
| 3.2.5. 用户P2P连接改变事件 | 18 |
| 3.2.6. 用户视频大小改变事件 | 18 |
| 3.3. 数据传输事件接口 | 18 |
| 3.3.1. 接口定义 | 18 |
| 3.3.2. 收到文件传输数据事件 | 19 |
| 3.3.3. 收到透明通道数据事件 | 19 |
| 3.3.4. 收到扩展透明通道数据事件 | 20 |

| | | |
|-----------|-----------------------|-----------|
| 3.3.5. | 收到SDK Filter数据事件..... | 20 |
| 3.4. | 文字消息事件接口 | 21 |
| 3.4.1. | 接口定义 | 21 |
| 3.4.2. | 收到文字聊天消息事件 | 21 |
| 3.5. | 私聊控制事件接口 | 21 |
| 3.5.1. | 接口定义 | 21 |
| 3.5.2. | 用户私聊请求事件 | 22 |
| 3.5.3. | 用户私聊回复事件 | 22 |
| 3.5.4. | 用户私聊退出事件 | 22 |
| 4. | 函数说明 | 23 |
| 4.1. | 初始化与资源释放 | 23 |
| 4.1.1. | 初始化SDK | 23 |
| 4.1.2. | 设置基本事件通知接口 | 23 |
| 4.1.3. | 设置状态变化事件接口 | 24 |
| 4.1.4. | 设置私聊消息通知接口 | 24 |
| 4.1.5. | 设置文字聊天消息接口 | 25 |
| 4.1.6. | 设置数据传输消息接口 | 25 |
| 4.1.7. | 释放SDK资源 | 26 |
| 4.2. | 登录流程 | 26 |
| 4.2.1. | 设置服务器认证密码 | 26 |
| 4.2.2. | 连接服务器 | 27 |
| 4.2.3. | 登录系统 | 27 |
| 4.2.4. | 进入房间（根据房间编号） | 28 |
| 4.2.5. | 进入房间（根据房间名称） | 29 |
| 4.2.6. | 离开房间 | 29 |
| 4.2.7. | 注销系统 | 30 |
| 4.3. | 音视频操作 | 30 |
| 4.3.1. | 用户视频控制 | 30 |
| 4.3.2. | 用户语音控制 | 31 |
| 4.3.3. | 设置视频显示位置 | 31 |
| 4.4. | 查询状态 | 32 |
| 4.4.1. | 查询摄像头的状态 | 32 |
| 4.4.2. | 查询用户音频设备采集状态 | 32 |
| 4.4.3. | 查询用户昵称 | 33 |
| 4.4.4. | 查询用户IP地址 | 33 |
| 4.4.5. | 查询用户视频宽度 | 33 |
| 4.4.6. | 查询用户视频高度 | 34 |
| 4.4.7. | 查询用户状态（字符串） | 34 |
| 4.4.8. | 查询用户状态（整型） | 35 |
| 4.4.9. | 查询房间名称 | 35 |
| 4.5. | 普通功能 | 36 |
| 4.5.1. | 获取SDK版本信息 | 36 |
| 4.5.2. | 获取当前房间在线用户列表 | 36 |
| 4.5.3. | 传送文本消息 | 37 |

| | | |
|---------|-------------------------|----|
| 4.5.4. | 透明通道传送缓冲区 | 37 |
| 4.5.5. | 透明通道传送缓冲区扩展 | 38 |
| 4.5.6. | 传送文件 | 39 |
| 4.5.7. | 查询传输任务相关信息 | 39 |
| 4.5.8. | 激活（关闭）SDK调用日志 | 40 |
| 4.5.9. | 发送SDK Filter通信数据 | 41 |
| 4.5.10. | 音视频录制 | 41 |
| 4.6. | 私聊函数 | 42 |
| 4.6.1. | 更改当前的聊天模式 | 42 |
| 4.6.2. | 获取指定用户聊天模式 | 43 |
| 4.6.3. | 请求与对方私聊 | 43 |
| 4.6.4. | 回复对方的私聊请求 | 43 |
| 4.6.5. | 回复对方的私聊请求（扩展） | 44 |
| 4.6.6. | 退出与某用户的私聊 | 44 |
| 4.7. | 系统设置 | 45 |
| 4.7.1. | 枚举本地视频采集设备 | 45 |
| 4.7.2. | 选择指定的视频采集设备 | 45 |
| 4.7.3. | 获取当前视频采集设备 | 46 |
| 4.7.4. | 枚举本地音频采集设备 | 46 |
| 4.7.5. | 选择指定的音频采集设备 | 47 |
| 4.7.6. | 获取当前音频采集设备 | 47 |
| 4.7.7. | 获取音频设备的当前音量 | 47 |
| 4.7.8. | 设置指定音频设备的音量 | 48 |
| 4.7.9. | SDK内核参数设置（整形值） | 48 |
| 4.7.10. | SDK内核参数设置（字符串值） | 49 |
| 4.7.11. | SDK内核参数状态查询 | 49 |
| 5. | 版本变更记录 | 51 |
| 6. | 附录一：错误代码参考 | 53 |

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 Android 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 Android 平台设计，并针对 ARM 系列 CPU 进行了汇编优化，可以作为 Android 平台上的即时通讯内核引擎，也可以做为视频会议、网络教育、即拍即传系统等互动平台的核心库。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 Android SDK 提供 Java 语言接口。

1.2 系统特性

“AnyChat for Android 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，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。

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。

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：1~25（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）

*高分辨率需硬件支持

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

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 NAT <——> Cone NAT

Cone NAT <——> Symmetric NAT

支持 UPNP 协议（wifi 网络中有效）

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、知识中心：<http://www.anychat.cn/faq/>
- 2、在线论坛：<http://bbs.anychat.cn/>
- 3、官方网站：<http://www.anychat.cn>
- 4、电子邮件：service@bairuitech.com
- 5、24 小时客服电话：+86 （020） 85276986、38109065、87691886
- 6、即时通讯：QQ：75513809

1.5 版权申明

“AnyChat for Android 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 Android”属于客户端组件（简称“客户端”），对上层应用提供纯 Java 语言的调用接口，内核是由一系列的.so 库（相当于 Win32 平台的 DLL）组合而成，采用 JNI 技术实现 Java 层与内核层的通信。

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 Android 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 Java 接口技术（Windows 平台是采用消息机制）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 for Android 客户端在房间中，收到其它用户的流媒体数据后，上层应用只需要提供一个 SurfaceView 控件的句柄，内核便可自动显示视频到该 SurfaceView 控件上，并自动播放声音。

AnyChat for Android 的工作流程与 Windows 平台的 SDK 一致，熟悉 Windows 平台的 SDK 工作机制将更有助于了解 AnyChat for Android 平台的工作机制。

2.2. 函数调用顺序

| 调用 顺序 | 函数 | 功能 | 备注 |
|----------|--------------------|----------------|------|
| A | InitSDK | 初始化系统 | 初始化 |
| | SetSDKOptionInt | 设置 SDK 整形值相关参数 | |
| | SetSDKOptionString | 设置 SDK 字符串相关参数 | |
| | | | |
| B | Connect | 连接服务器 | 进入系统 |
| | Login | 用户登录系统 | |
| | EnterRoom | 进入房间 | |
| | | | |
| C | GetOnlineUser | 获取当前房间在线用户数 | 其它功能 |
| | GetCameraState | 查询用户摄像头的状态 | |
| | GetSpeakState | 查询用户发言状态 | |
| | ShowLVProperty | 显示本地视频画面调节对话框 | |
| | EnumVideoCapture | 枚举本地视频采集设备 | |
| | SelectVideoCapture | 选择指定的视频采集设备 | |
| | GetCurVideoCapture | 获取当前使用的视频采集设备 | |
| | EnumAudioCapture | 枚举本地音频采集设备 | |
| | SelectAudioCapture | 选择指定的音频采集设备 | |
| | GetCurAudioCapture | 获取当前使用的音频采集设备 | |
| | AudioGetVolume | 获取指定音频设备的当前音量 | |
| | AudioSetVolume | 设置指定音频设备的音量 | |
| | SetVideoPos | 设置视频显示位置 | |
| | UserCameraControl | 操作用户视频 | |
| | UserSpeakControl | 操作用户语音 | |
| | SendTextMessage | 发送文本消息 | |
| | ChangeChatMode | 更改当前的聊天模式 | |
| | GetUserChatMode | 获取指定用户当前的聊天模式 | |
| | PrivateChatRequest | 请求与对方私聊，发起私聊请求 | |
| | PrivateChatEcho | 回复对方的私聊请求 | |
| | PrivateChatExit | 退出与某用户的私聊 | |
| | | | |
| | | | |
| D | LeaveRoom | 离开房间 | 退出系统 |
| | Logout | 用户注销 | |
| | Release | 释放资源 | |

2.3. 编程要点

2.3.1. SDK的Java Package

AnyChat for Android 的 package 路径是：com.bairuitech.anychat，由于内核采用了 JNI 技术，需要与该包进行交互，所以上层不能修改包的名称。

2.3.2. SDK与Activity

Android 应用是由一个或多个 Activity 组合而成，每个 Activity 都有其生命周期（可参考 Android 相关开发文档），而 AnyChat for Android 内核采用 JNI 技术，并进行了特殊的设计，使得 AnyChat for Android 可以在多个 Activity 中共享，即 AnyChat for Android 的生命周期是从第一个 Activity 调用 InitSDK 方法开始，到最后一个 Activity 调用 Release 方法结束，中间的 Activity 不需要再初始化 SDK。这样便可实现在第一个 Activity 中登录之后，便于立即切换第二个 Activity 继续操作，而不需要在第二个 Activity 中再进行登录的操作。

2.3.3. SurfaceView视频显示

AnyChat for Android 采用 SurfaceView 进行视频显示，上层应用只需要在 Activity 中创建一个 SurfaceView 控件，然后将控件句柄通过 SetVideoPos 方法传递给 SDK，则当有视频数据到达时，内核将会自动将视频显示到该 SurfaceView 控件上，不需要上层应用来处理视频的显示。

需要注意的是，部分 Android 设备显示本地视频时，需要启动 Overlay 模式，否则不能进行本地视频的预览和采集，当启动 Overlay 模式时，需要对 SurfaceView 控件进行相关属性的设置，详细信息可参考随 AnyChat for Android 一同发布的 Demo 程序源代码。

2.3.4. 字符编码

Android 平台上层是 Java 虚拟机，采用 Unicode 编码，AnyChat for Android

内核在处理 Android 与 Windows 平台的通信时，会自动将字符串（如文字聊天数据）转换为上层平台所对应的编码，不需要应用层进行转换，但是当上层应用使用透明通道在客户端与服务器，或是客户端之间传输数据，而需要显示时，就需要上层应用手动来处理编码转换的任务，否则显示将会出现乱码。

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。

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. 接口(Interface)说明

AnyChat for Android SDK 通过接口（类似于 C++ 的回调函数）实现与上层应用的状态更新和数据交互。SDK 的很多调用都是异步的，如登录操作，调用函数完成之后，需要等待对应接口的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，都采用接口来实现。

根据不同的类型，接口也分为几大类，在实际的开发过程中，可根据具体情况实现这些接口。

3.1. 基本流程事件接口

“AnyChat 在线音视频互动平台”采用消息通知的方式来告知系统的各种状态变化。

3.1.1. 接口定义

```
package com.bairuitech.anychat;
//AnyChat基本事件接口
public interface AnyChatBaseEvent {
    public void OnAnyChatConnectMessage(boolean bSuccess);
    public void OnAnyChatLoginMessage(INT UserId, INT ErrorCode);
    public void OnAnyChatEnterRoomMessage(INT RoomId, INT ErrorCode);
    public void OnAnyChatOnlineUserMessage(INT UserNum, INT RoomId);
    public void OnAnyChatUserAtRoomMessage(INT UserId, boolean bEnter);
    public void OnAnyChatLinkCloseMessage(INT ErrorCode);
}
```

3.1.2. 网络连接事件

方法： public void OnAnyChatConnectMessage(boolean bSuccess)

参数： bSuccess 表示是否连接成功，BOOLEAN 类型；

说明：当客户端连接服务器时被触发，等同于 WIN32 平台的 WM_GV_CONNECT 消息。

3.1.3. 用户登录事件

方法: `public void OnAnyChatLoginMessage(int dwUserId, int dwErrorCode)`

参数:

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

`dwErrorCode` 出错代码, 可判断登录是否成功

说明: 当客户端登录服务器时被触发, 等同于 WIN32 平台的 WM_GV_LOGINSYSTEM 消息。

3.1.4. 进入房间事件

方法: `public void OnAnyChatEnterRoomMessage(int dwRoomId, int dwErrorCode)`

参数:

`dwRoomId` 表示进入的房间 ID 号

`dwErrorCode` 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, 等同于 WIN32 平台的 WM_GV_ENTERROOM 消息。

3.1.5. 房间在线用户事件

方法: `public void OnAnyChatOnlineUserMessage(int dwUserNum, int dwRoomId)`

参数:

`dwUserNum` 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, 等同于 WIN32 平台的 WM_GV_ONLINEUSER 消息。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

方法: `public void OnAnyChatUserAtRoomMessage(int dwUserId, boolean bEnter)`

参数:

`dwUserId` 表示当前房间活动用户的 ID 号
`bEnter` `true` 表示进入房间, `false` 表示离开房间

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, 等同于 WIN32 平台的 `WM_GV_USERATROOM` 消息。

3.1.7. 网络连接关闭事件

方法: `public void OnAnyChatLinkCloseMessage(int dwReason)`

参数:

`dwReason` 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接口, 等同于 WIN32 平台的 `WM_GV_LINKCLOSE` 消息。如果已打开本地摄像头, 则上层应用必须在网络连接关闭事件中关闭本地摄像头, 否则可能造成异常。

3.2. 状态变化事件接口

状态变化事件接口定义在“AnyChatStateChgEvent.java”接口类中包含了 SDK 的用户视频设备状态、音频设备状态、用户聊天模式切换、P2P 连接状态变化等异步事件的通知。

3.2.1. 接口定义

```
package com.bairuitech.anychat;
// AnyChat状态变化事件通知接口
public interface AnyChatStateChgEvent {
    public void OnAnyChatMicStateChgMessage(INT UserId, boolean bOpenMic);
    public void OnAnyChatCameraStateChgMessage(INT UserId, INT State);
    public void OnAnyChatChatModeChgMessage(INT UserId, int bPublicChat);
    public void OnAnyChatActiveStateChgMessage(INT UserId, INT State);
    public void OnAnyChatP2PConnectStateMessage(INT UserId, INT State);
}
```


3.2.2. 音频设备状态改变事件

```
public void OnAnyChatMicStateChgMessage(int dwUserId, boolean bOpenMic)
```

参数:

| | |
|-----------------------|------------------|
| <code>dwUserId</code> | 表示状态变化的用户 ID |
| <code>bOpenMic</code> | 表示该用户是否已打开音频采集设备 |

说明: 当进入房间成功之后, 当用户使用 API: `UserSpeakControl` 操作本地音频设备时将会触发该接口, 等同于 WIN32 平台的 `WM_GV_MICSTATECHANGE` 消息。

3.2.3. 视频设备状态改变事件

```
public void OnAnyChatCameraStateChgMessage(int dwUserId, int dwState)
```

参数:

| | |
|-----------------------|-----------------|
| <code>dwUserId</code> | 表示状态变化的用户 ID |
| <code>dwState</code> | 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, 当用户使用 API: `UserCameraControl` 操作本地视频设备时将会触发该接口, 等同于 WIN32 平台的 `WM_GV_CAMERASTATE` 消息。

3.2.4. 用户聊天模式改变事件

```
public void OnAnyChatChatModeChgMessage(int dwUserId, boolean bPublicChat)
```

参数:

| | |
|--------------------------|--------------------------|
| <code>dwUserId</code> | 表示状态变化的用户 ID |
| <code>bPublicChat</code> | 表示该用户当前是否为公聊状态, 否则为私聊状态: |

说明: 当进入房间成功之后, 当用户改变聊天模式时将会触发该接口, 等同于 WIN32 平台的 `WM_GV_CHATMODECHG` 消息。

3.2.5. 用户P2P连接改变事件

```
public void OnAnyChatP2PConnectStateMessage(int dwUserId, int dwState)
```

参数:

| | |
|----------|----------------------------|
| dwUserId | 表示其它用户 ID 号 |
| dwState | 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接口, 等同于 WIN32 平台的 WM_GV_P2PCONNECTSTATE 消息。

3.2.6. 用户视频大小改变事件

```
public void OnAnyChatVideoSizeChgMessage(int dwUserId, int dwWidth, int dwHeight)
```

参数:

| | |
|----------|--------------|
| dwUserId | 表示状态变化的用户 ID |
| dwWidth | 表示该用户当前的视频宽度 |
| dwHeight | 表示该用户当前的视频高度 |

说明: 当进入房间成功之后, 成功打开本地视频设备, 或是修改视频设备采集分辨率之后将触发该接口, 等同于 WIN32 平台的 WM_GV_VIDEOSIZECHG 消息。

3.3. 数据传输事件接口

3.3.1. 接口定义

```
package com.bairuitech.anychat;
// 数据传输通知接口
public interface AnyChatTransDataEvent {
    public void OnAnyChatTransFile(int dwUserId, String FileName, String TempFilePath,
int dwFileLength, int wParam, int lParam, int dwTaskId);
```

```
public void OnAnyChatTransBuffer(int dwUserId, byte[] lpBuf, int dwLen);
public void OnAnyChatTransBufferEx(int dwUserId, byte[] lpBuf, int dwLen, int
wParam, int lParam, int taskId);
public void OnAnyChatSDKFilterData(byte[] lpBuf, int dwLen);
}
```

3.3.2. 收到文件传输数据事件

```
public void OnAnyChatTransFile(int dwUserId, String FileName, String TempFilePath, int
dwFileLength, int wParam, int lParam, int dwTaskId);
```

参数:

| | |
|---------------|--------------------------------|
| dwUserId: | 用户 ID, 指示发送用户 |
| FileName: | 文件名 (含扩展名, 不含路径) |
| TempFilePath: | 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 |

说明:

当收到其它用户使用“TransFile”方法发送的文件时, 将会触发该接口, 等同于回调函数: BRAC_TransFile_CallBack。

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。

3.3.3. 收到透明通道数据事件

```
public void OnAnyChatTransBuffer(int dwUserId, byte[] lpBuf, int 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“TransBuffer”方法发送的缓冲区数据时，将会触发该接口，等同于回调函数：BRAC_TransBuffer_CallBack。

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。

3.3.4. 收到扩展透明通道数据事件

```
public void OnAnyChatTransBufferEx(int dwUserId, byte[] lpBuf, int dwLen, int wparam,
int lparam, int taskid);
```

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“TransBufferEx”方法发送的缓冲区数据时，将会触发该接口，等同于回调函数：BRAC_TransBufferEx_CallBack。

3.3.5. 收到SDK Filter数据事件

```
public void OnAnyChatSDKFilterData(byte[] lpBuf, int dwLen);
```

参数：

| | |
|--------|-------|
| lpBuf: | 缓冲区地址 |
| dwLen: | 缓冲区大小 |

说明：

当收到服务器“SDK Filter”或是“Server SDK”相关接口发送的缓冲区数据时，将会触发该接口，等同于回调函数：BRAC_SDKFilterData_CallBack。

3.4. 文字消息事件接口

3.4.1. 接口定义

```
package com.bairuitech.anychat;  
// 文字聊天通知接口  
public interface AnyChatTextMsgEvent {  
    public void OnAnyChatTextMessage(int dwFromUserId, int dwToUserId, int bSecret,  
String message);  
}
```

3.4.2. 收到文字聊天消息事件

```
public void OnAnyChatTextMessage(int dwFromUserId, int dwToUserId, boolean bSecret,  
String message);
```

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 |
| message | 消息字符串 |

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接口, 等同于 WIN32 平台的回调函数: BRAC_TextMessage_CallBack。本地用户向其它用户发送文字消息时, 将不会触发该接口。

3.5. 私聊控制事件接口

3.5.1. 接口定义

```
package com.bairuitech.anychat;  
// AnyChat私聊消息通知接口  
public interface AnyChatPrivateChatEvent {  
    public void OnAnyChatPrivateRequestMessage(int dwUserId, int dwRequestId);  
}
```

```
public void OnAnyChatPrivateEchoMessage(int dwUserId, int dwErrorCode);  
public void OnAnyChatPrivateExitMessage(int dwUserId, int dwErrorCode);  
}
```

3.5.2. 用户私聊请求事件

```
public void OnAnyChatPrivateRequestMessage(int dwUserId, int 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: 当进入房间成功之后，当用户发起私聊请求时，在被请求者端将触发该接口，等同于 WIN32 平台的 WM_GV_PRIVATEEREQUEST 消息。

3.5.3. 用户私聊回复事件

```
public void OnAnyChatPrivateEchoMessage(int dwUserId, int dwErrorCode)
```

参数:

dwUserId 表示回复者的用户 ID 号

dwErrorCode 返回代码，表示对方是否同意自己的邀请

说明: 当进入房间成功之后，当用户发起私聊请求时，其它用户回复该请求时将触发该接口，等同于 WIN32 平台的 WM_GV_PRIVATEEECHO 消息。

3.5.4. 用户私聊退出事件

```
public void OnAnyChatPrivateExitMessage(int dwUserId, int dwErrorCode)
```

参数:

dwUserId 表示退出者的用户 ID 号

dwErrorCode 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接口，等同于 WIN32 平台的 WM_GV_PRIVATEEXIT 消息。

4. 函数说明

4.1. 初始化与资源释放

4.1.1. 初始化SDK

INT InitSDK(INT osver, INT flags)

功能：初始化 SDK

返回值：0 表示成功，否则为出错代码

参数：

osver Android 平台 API 版本，系统常量：android.os.Build.VERSION.SDK_INT

flags 功能模式组合，默认为 0，参考 WIN32 平台 SDK 的 InitSDK 的参数

dwFuncMode 定义

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，默认为 0，SDK 内部会自动设置常用的标志。

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

4.1.2. 设置基本事件通知接口

public void SetBaseEvent(AnyChatBaseEvent e);

功能：只有设置基本事件通知接口后，AnyChat 内核产生的异步事件才能通知到 Activity，并触发 Activity 中对应的接口方法。

返回值：无

参数：

E 实现 AnyChatBaseEvent 接口的对象

备注：

如有多个 Activity 需要响应事件，则每个 Activity 都必须实现对应的接口，

并调用该方法向 AnyChat 内核注册该接口，只有当前活动的 Activity 才能接收到异步消息，触发接口方法。

4.1.3. 设置状态变化事件接口

public void SetStateChgEvent(AnyChatStateChgEvent e);

功能：只有设置状态变化事件通知接口后，AnyChat 内核产生的状态变化事件才能通知到 Activity，并触发 Activity 中对应的接口方法。

返回值：无

参数：

E 实现 AnyChatStateChgEvent 接口的对象

备注：

如有多个 Activity 需要响应事件，则每个 Activity 都必须实现对应的接口，并调用该方法向 AnyChat 内核注册该接口，只有当前活动的 Activity 才能接收到异步消息，触发接口方法。

4.1.4. 设置私聊消息通知接口

public void SetPrivateChatEvent(AnyChatPrivateChatEvent e);

功能：只有设置私聊事件通知接口后，AnyChat 内核产生的私聊事件才能通知到 Activity，并触发 Activity 中对应的接口方法。

返回值：无

参数：

E 实现 AnyChatPrivateChatEvent 接口的对象

备注：

如有多个 Activity 需要响应事件，则每个 Activity 都必须实现对应的接口，并调用该方法向 AnyChat 内核注册该接口，只有当前活动的 Activity 才能接收到异步消息，触发接口方法。

4.1.5. 设置文字聊天消息接口

public void SetTextMessageEvent(AnyChatTextMsgEvent e);

功能：只有设置文字聊天消息接口后，AnyChat 内核产生的文字聊天事件才能通知到 Activity，并触发 Activity 中对应的接口方法。

返回值：无

参数：

E 实现 AnyChatTextMsgEvent 接口的对象

备注：

如有多个 Activity 需要响应事件，则每个 Activity 都必须实现对应的接口，并调用该方法向 AnyChat 内核注册该接口，只有当前活动的 Activity 才能接收到异步消息，触发接口方法。

4.1.6. 设置数据传输消息接口

public void SetTransDataEvent(AnyChatTransDataEvent e);

功能：只有设置数据传输消息接口后，AnyChat 内核产生的数据传输事件才能通知到 Activity，并触发 Activity 中对应的接口方法。

返回值：无

参数：

E 实现 AnyChatTransDataEvent 接口的对象

备注：

如有多个 Activity 需要响应事件，则每个 Activity 都必须实现对应的接口，并调用该方法向 AnyChat 内核注册该接口，只有当前活动的 Activity 才能接收到异步消息，触发接口方法。

数据传输接口内部实现了缓冲区传输、扩展缓冲区传输、文件传输等传输数据的异步事件通知接口。

4.1.7. 释放SDK资源

INT 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，即最后一个 Activity 销毁时调用，而中间 Activity 销毁时不需要，也不能调用该方法。

4.2. 登录流程

4.2.1. 设置服务器认证密码

INT SetServerAuthPass(STRING Password);

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。

4.2.2. 连接服务器

INT Connect(String ServerAddr, INT Port);

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

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 8906）

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

INT Login(String UserName, String Password);

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

参数：

lpUserName 注册用户名；
lpPassword 登录密码（为空表示游客）；

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）

INT EnterRoom(INT Roomid, STRING RoomPass);

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

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）；

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_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》。

4.2.5. 进入房间（根据房间名称）

INT EnterRoomEx(STRING RoomName, STRING RoomPass)

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）；

备注：

该方法与“EnterRoom”功能相同，区别在于房间的标识不同，其中“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，通过基本事件接口（AnyChatBaseEvent）返回给上层应用，上层应用可以通过方法“GetRoomName”来获取房间名称。

4.2.6. 离开房间

INT LeaveRoom(INT Roomid);

功能：离开房间。

返回值：0 表示成功，否则为出错代码

参数:

Roomid 房间编号，为-1 表示退出当前房间

备注:

在用户变换房间之前，需要调用该方法离开房间，然后才能进入新的房间。

4.2.7. 注销系统

INT 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

4.3. 音视频操作

4.3.1. 用户视频控制

INT UserCameraControl(INT Userid, BOOLEAN 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。

4.3.2. 用户语音控制

INT UserSpeakControl(INT Userid, BOOLEAN 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。

4.3.3. 设置视频显示位置

INT SetVideoPos(INT userid, Surface s, INT Left, INT Top, INT Right, INT Bottom);

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：

userid 用户编号，为-1 表示操作自己的视频显示位置

s 视频显示表面

dwLeft、dwTop、dwRight、dwBottom 位置信息，默认为 0

备注：

该方法在打开用户视频之前调用，调用之前需在 java 层设置该表面的视频大小（SetFixedSize(w,h)），用户视频大小可通过方法 GetUserVideoWidth、GetUserVideoHeight 获取。

4.4. 查询状态

4.4.1. 查询摄像头的状态

INT GetCameraState(INT Userid);

功能：查询用户摄像头的状态

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

参数：

userid 用户编号，为-1时表示获取自己的摄像头状态；

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。

4.4.2. 查询用户音频设备采集状态

INT GetSpeakState(INT userid);

功能：查询用户音频设备采集状态

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

参数：

userid 用户编号，为-1时表示获取自己的音频设备状态；

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。

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

String GetUserName(INT userid);

功能：查询用户昵称

返回值：用户昵称字符串

参数：

userid 用户编号，为-1时表示获取自己的昵称；

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。

4.4.4. 查询用户IP地址

String GetUserIPAddr (INT userid);

功能：查询用户互联网 IP 地址

返回值：用户 IP 字符串

参数：

userid 用户编号，为-1时表示获取自己的 IP 地址；

备注：

这里所查询到的用户 IP 为互联网 IP 地址，可能是用户本机的真实 IP，也可能是用户接入互联网的网关 IP 地址。

4.4.5. 查询用户视频宽度

INT GetUserVideoWidth (INT userid);

功能：查询用户视频分辨率的宽度值

返回值：返回指定用户的视频宽度，如果用户视频没有打开，或是打开视频设备失败，则获取的值为 0

参数：

`userid` 用户编号，为-1 时表示获取自己的视频宽度；

备注：

如查询本地的视频宽度值，则必须在打开本地视频设备成功之后方能查询成功；如查询其它用户的视频宽度值，则必须在对方打开视频设备成功，且状态同步到本地后，才能查询成功，故上层应用可用一个定时器间隔查询。

4.4.6. 查询用户视频高度

INT GetUserVideoHeight (INT userid);

功能：查询用户视频分辨率的高度值

返回值：返回指定用户的视频高度，如果用户视频没有打开，或是打开视频设备失败，则获取的值为 0

参数：

`userid` 用户编号，为-1 时表示获取自己的视频高度；

备注：

如查询本地的视频高度值，则必须在打开本地视频设备成功之后方能查询成功；如查询其它用户的视频高度值，则必须在对方打开视频设备成功，且状态同步到本地后，才能查询成功，故上层应用可用一个定时器间隔查询。

4.4.7. 查询用户状态（字符串）

String QueryUserStateString(INT userid, int infoname);

功能：查询指定用户状态（字符串类型）

返回值：相关状态的字符串

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 WIN32 平台 SDK 相关定义）

备注：

通过调用该方法，可以查询指定用户的相关状态值。

4.4.8. 查询用户状态（整型）

int QueryUserStateInt(INT userid, int infoname);

功能： 查询指定用户状态（整型）

返回值： 相关状态值

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 WIN32 平台 SDK 相关定义） |

备注：

通过调用该方法，可以查询指定用户的相关状态值。

4.4.9. 查询房间名称

String GetRoomName(INT roomid);

功能： 查询根据房间 ID 获取房间名称

返回值： 返回房间名称，如查询失败，则返回空字符串

参数：

| | |
|--------|-------|
| roomid | 房间编号； |
|--------|-------|

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。

4.5. 普通功能

4.5.1. 获取SDK版本信息

public int GetSDKMainVersion();

功能：获取 SDK 主版本号。

返回值：主版本号

参数：无

public int GetSDKSubVersion ();

功能：获取 SDK 从版本号。

返回值：从版本号

参数：无

public String GetSDKBuildTime ();

功能：获取 SDK 编译时间字符串。

返回值：编译时间字符串

参数：无

4.5.2. 获取当前房间在线用户列表

public native int[] GetOnlineUser();

功能：获取当前房间在线用户列表（不包含自己）

返回值：在线用户 ID 数组

参数：

无。

备注：

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（AnyChatBaseEvent）中已通知给上层应用。

4.5.3. 传送文本消息

int SendMessage(int userid, boolean secret, String message);

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）

secret: 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

message: 消息字符串

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 AnyChatTextMsgEvent 接口的接口函数的调用。

4.5.4. 透明通道传送缓冲区

int TransBuffer(int userid, byte[] buf, int len);

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

参数：

dwUserId: 目标用户编号，-1 表示大家（用户当前房间所有人）

buf: 缓冲区

len: 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 AnyChatTransDataEvent 接口的对应接口函数。

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，表示向该房间的其他用户广播数据（注：自己不能发送给自己）。

4.5.5. 透明通道传送缓冲区扩展

int TransBufferEx(int userid, byte[] buf, int len, int wparam, int lparam, int flags, AnyChatTransTaskOutParam outParam);

功能：透明通道传送缓冲区

返回值：0 表示函数调用成功，否则为出错代码

参数：

userid: 目标用户编号，只针对某一个用户，不能为-1（所有人）
buf: 缓冲区，≤1024KB（1MB），内部会自动分包处理
len: 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）

outParam: 若函数调用成功，则内核通过该参数输出任务 ID（TaskId）

备注：

该方法与“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“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）、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）、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

```
int TransFile(int userid, String filepath, int wparam, int lparam, int flags,
AnyChatTransTaskOutParam outParam);
```

功能：传送文件给指定用户

返回值：0 表示函数调用成功，否则为出错代码

参数：

| | |
|-----------|---------------------------------|
| userid: | 目标用户编号，只针对某一个用户，不能为-1（所有人） |
| filepath: | 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：“TransBufferEx”方法 |
| outParam: | 若函数调用成功，则内核通过该参数输出任务 ID（TaskId） |

备注：

该方法传输效率与“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

```
INT QueryTransTaskInfo(INT Userid, INT TaskId, int infoname, char FAR* infoval, int
infoflen)
```

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 |
| infoflen | 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 |
|--------------------|--------|-----------|-------------|
| 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. 激活（关闭）SDK调用日志

INT ActiveCallLog(BOOLEAN 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。

4.5.9. 发送SDK Filter通信数据

int SendSDKFilterData(byte[] buf, int len);

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

参数：

buf: 缓冲区

len: 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 和服务器来说，都是透明的。

4.5.10. 音视频录制

INT StreamRecordCtrl(INT dwUserId, INT bStartRecord, INT dwFlags, INT dwParam);

功能：对指定用户的音视频流进行录制，在中心录像服务器上保存音视频文件

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，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。

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_RECORD_FLAGS_VIDEO 录制视频

BRAC_RECORD_FLAGS_AUDIO 录制音频

BRAC_RECORD_FLAGS_SERVER 服务器端录制

目前 AnyChat for Android SDK 暂时不支持客户端录像。

在服务器端录制音视频，需要单独部署中心录像服务器，参考论坛相关介绍：

<http://bbs.anychat.cn/forum.php?mod=viewthread&tid=20&extra=page%3D1>。

4.6. 私聊函数

4.6.1. 更改当前的聊天模式

INT ChangeChatMode(INT ChatMode);

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。

4.6.2. 获取指定用户聊天模式

INT GetUserChatMode(INT Userid, INT& 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。

4.6.3. 请求与对方私聊

INT PrivateChatRequest(INT Userid);

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。

4.6.4. 回复对方的私聊请求

INT PrivateChatEcho(INT Userid, INT Requestid, BOOLEAN 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。

4.6.5. 回复对方的私聊请求（扩展）

INT PrivateChatEchoEx(INT UserId, INT Requestid, INT ErrorCode);

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“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。

4.6.6. 退出与某用户的私聊

INT PrivateChatExit(INT UserId);

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。

4.7. 系统设置

4.7.1. 枚举本地视频采集设备

INT EnumVideoCapture(CHAR lpDeviceName, INT& dwDeviceNum);**

功能: 枚举本地视频采集设备

返回值: 0 表示成功，否则为出错代码

参数:

lpDeviceName 视频设备名称，可为空，表示只获取设备数量；

dwDeviceNum 视频设备数量；

备注:

该方法将会在内部分配缓冲区，外部使用完成之后，必需手工释放这些缓冲区，否则会造成内存泄露，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. 选择指定的视频采集设备

INT SelectVideoCapture(STRING 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。

4.7.3. 获取当前视频采集设备

INT GetCurVideoCapture(CHAR* lpDeviceName, INT Len);

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。

4.7.4. 枚举本地音频采集设备

INT EnumAudioCapture(CHAR lpDeviceName, INT& dwDeviceNum);**

功能：枚举本地音频采集设备

返回值：0 表示成功，否则为出错代码

参数：

lpDeviceName 音频设备名称；

dwDeviceNum 音频设备数量；

备注

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

4.7.5. 选择指定的音频采集设备

INT SelectAudioCapture(String szCaptureName);

功能：选择指定的音频采集设备

返回值：0 表示成功，否则为出错代码

参数：

szCaptureName 所获取设备的名称；

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。

4.7.6. 获取当前音频采集设备

INT GetCurAudioCapture(Char* lpDeviceName, INT Len);

功能：获取当前使用的音频采集设备

返回值：0 表示成功，否则为出错代码

参数：

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。

4.7.7. 获取音频设备的当前音量

INT AudioGetVolume(AudioDevice device, INT& dwVolume);

功能：获取指定音频设备的当前音量

返回值：0 表示成功，否则为出错代码

参数：

device 设备类型，定义为：

```
AD_WAVEIN = 0,          ///< 输入设备: Mic
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。

4.7.8. 设置指定音频设备的音量

INT AudioSetVolume(AudioDevice device, INT Volume);

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：

```
AD_WAVEIN = 0,          ///< 输入设备: Mic
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。

4.7.9. SDK内核参数设置（整形值）

int SetSDKOptionInt(int optname, int optvalue);

功能：SDK 内核参数设置（整形值参数）

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 设置的参数值

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 WIN32 平台 SDK 相关定义。

4.7.10. SDK内核参数设置（字符串值）

int SetSDKOptionString(int optname, String optvalue);

功能：SDK 内核参数设置（字符串值参数）

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 设置的参数值

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 WIN32 平台 SDK 相关定义。

4.7.11. SDK内核参数状态查询

int GetSDKOptionInt(int optname);

功能：SDK 内核参数状态查询（整形值）

返回值：返回查询结果

参数：

optname 内核参数名称；

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。

String GetSDKOptionString(int optname);

功能：SDK 内核参数状态查询（字符串）

返回值：返回查询结果

参数：

optname 内核参数名称；

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。

5. 版本变更记录

2011-05-15 V1.0

初始版本，采用 AnyChat Platform Core V4.0 内核

2011-11-15 V1.2

基于 AnyChat Platform Core SDK V4.2 版本编译；
修正内核多线程操作界面元素导致程序异常的 Bug；
增加回音抑制模块，提高语音通信体验；

2012-02-20 V1.3

基于 AnyChat Platform Core SDK V4.3 版本编译；
优化语音通信效果
支持 Android 4.x 平台

2012-05-11 V1.4

基于 AnyChat Platform Core SDK V4.4 版本编译；
优化 Android 4.0 内核显示模块，效率更高；
优化 Android 4.x 硬件编码模块（需内核及硬件的支持）

2012-09-10 V1.5

基于 AnyChat Platform Core SDK V4.5 版本编译；
修正 android 平台下不能传输中文文件名的 Bug；
修正在部分 android 设备上长时间运行会导致程序崩溃的 Bug；
增加平滑播放模式以及最大缓冲时间设置接口，增强系统对各种网络环境的适应性；
修正某些情况下视频卡、顿的问题以及播放时可能音视频不同步的 Bug；

2012-11-22 V1.6

基于 AnyChat Platform Core SDK V4.6 版本编译；
优化音视频播放模块，音视频同步更精准，缩短通讯延迟；
增强系统容错性，在网络较差的环境下用户体验有明显的改善；
流媒体传输针对 3G 网络进行优化，播放更平滑；
增加流媒体传输过程中丢包率统计功能并提供查询 API 接口；
新增 Android 4.1 平台支持，并兼容更多 Android 设备；

2013-03-20 V1.7

基于 AnyChat Platform Core SDK V4.7 版本编译；
优化回声消除算法，极大提高音频通话体验；
视频会话时，默认打开前置摄像头；
增加 Java 音视频采集、显示驱动，兼容更多 Android 设备；
优化网络传输模块，提高音视频的流畅性；

2013-07-28 V1.8

基于 AnyChat Platform Core SDK V4.8 版本编译；
支持中心服务器录像，音视频录像文件服务器集中保存；
支持瑞芯微（RockChip）全系列硬件编解码，实现高清视频通话；
优化 P2P 算法，提升复杂网络环境下音视频体验；
修改 TransFile、TransBufferEx 两个 API 调用参数，增加 TaskId 输出接口；
新增 Android 4.2 平台支持，并兼容更多 Android 设备；

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（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 | ///< IP 地址已被暂时锁定 |
| #define GV_ERR_VISITOR_DENY | 204 | ///< 游客登录被禁止（登录时没有输入密码） |
| #define GV_ERR_INVALID_USERID | 205 | ///< 无效的用户 ID（用户不存在） |
| //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，本次进入房间请求忽略 |
| //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。