

AnyChat SDK for Linux

开发流程指南(Qt)

(V8.2 版本)



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>

2020 年 10 月

目录

一、	简介	3
1.1	面向的读者.....	4
1.2	获取 ANYCHAT SDK FOR LINUX	4
1.3	技术支持	5
二、	编写说明	6
三、	工程准备	7
3.1	导入库文件.....	7
3.2	导入 SDK 文件	8
四、	基本流程	10
4.1	监听基本事件.....	10
4.2	初始化 SDK.....	12
4.3	应用签名	12
4.3.1	应用 Id.....	12
4.3.2	应用签名使用场景	12
4.4	连接、登录服务器	13
4.4.1	登录 SDK 平台	14
4.4.2	登录 AnyChat 视频云平台	15
4.5	进入房间	16
五、	音视频交互	17
5.1	打开本地音视频.....	17
5.2	关闭本地音视频.....	17
5.3	请求远程音视频.....	18
5.4	关闭远程音视频.....	18
六、	释放过程	19
七、	附录	20

一、简介

AnyChat SDK(AnyChat 音视频互动开发平台)是一套跨平台的音视频即时通讯解决方案，基于先进的 H.264 视频编码标准、AAC 音频编码标准与 P2P 技术，支持高清视频，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。

基于 Linux 的客户端 SDK 应用于 Linux 平台设备，您可以通过该套 SDK API 接口实现在 Linux 平台快速开发基于音视频通讯交互功能的应用程序，主要提供的功能如下：

- 音视频即时通讯：提供语音、视频一对一、一对多的实时通讯，支持高清视频和高品质音频效果；
- 录像：支持针对单个人的音视频录制、整个视频通话过程内容的合成音视频录制；
- 抓拍：可对本地视频和正在视频的对象进行抓拍；
- 文字聊天：支持多用户之间的文字交流；
- 透明通道：提供客户端之间、客户端跟服务器之间的数据通讯能力；
- 文件传输：支持客户端直接、客户端跟服务器之间的文件传输功能，支持断点续传；
- 动态设置音视频参数：提供音视频参数设置的接口，可以根据需要动态设置分辨率、码率、帧率等视频参数，满足各种应用场景的需求；
- 外部音视频输入：支持非标准采集设备以外的音视频源输入，满足更多的应用场景；
- 集成第三方外部音视频编解码器：可集成第三方音视频编解码器，满足特殊环境下面的硬件编解码要求。

1.1 面向的读者

《AnyChat SDK for Linux 开发流程指南(Qt)》文档是提供给具有一定的Linux下Qt编程经验和了解面向对象概念的读者使用，不要求具备音视频开发方面的经验。您在使用遇到任何问题，都可以通过访问 bbs.anychat.cn 反馈给我们。

1.2 获取 AnyChat SDK for Linux

目前 AnyChat 官方网站提供的“AnyChat for Linux SDK”分为 32-bit 和 64-bit 两个版本，请根据实际的系统版本进行下载，下载地址为: <http://sdk.anychat.cn/html/download.html>，如下图所示：



AnyChat for Linux SDK 包里面提供了可执行程序 and 动态库、开发接口文档、程序开发接口、开发示例代码，其解压之后的目录结构如下所示：

目录清单

----bin	AnyChat SDK动态库、演示程序
----doc	开发文档
----src	客户端与服务器Demo程序源代码
----sdk	客户端SDK、服务器SDK引用文件

1.3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联系：

- 1、在线论坛：<http://bbs.anychat.cn/>
- 2、知识中心：<http://www.anychat.cn/faq/>
- 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、38103410

二、编写说明

本指南的编写是为了帮助使用 AnyChat for Linux SDK 的用户快速地搭建 SDK 开发环境、熟悉 SDK 开发流程、掌握 SDK 开发功能接口而编写的，其中“工程准备”、“基本流程”、“音视频交互”三章的内容是基于 src\HelloAnyChat 目录中提供的 HelloAnyChat 工程来编写说明的，涉及到的开发环境配置、以及相关代码说明可以参考 HelloAnyChat 工程源码。

三、工程准备

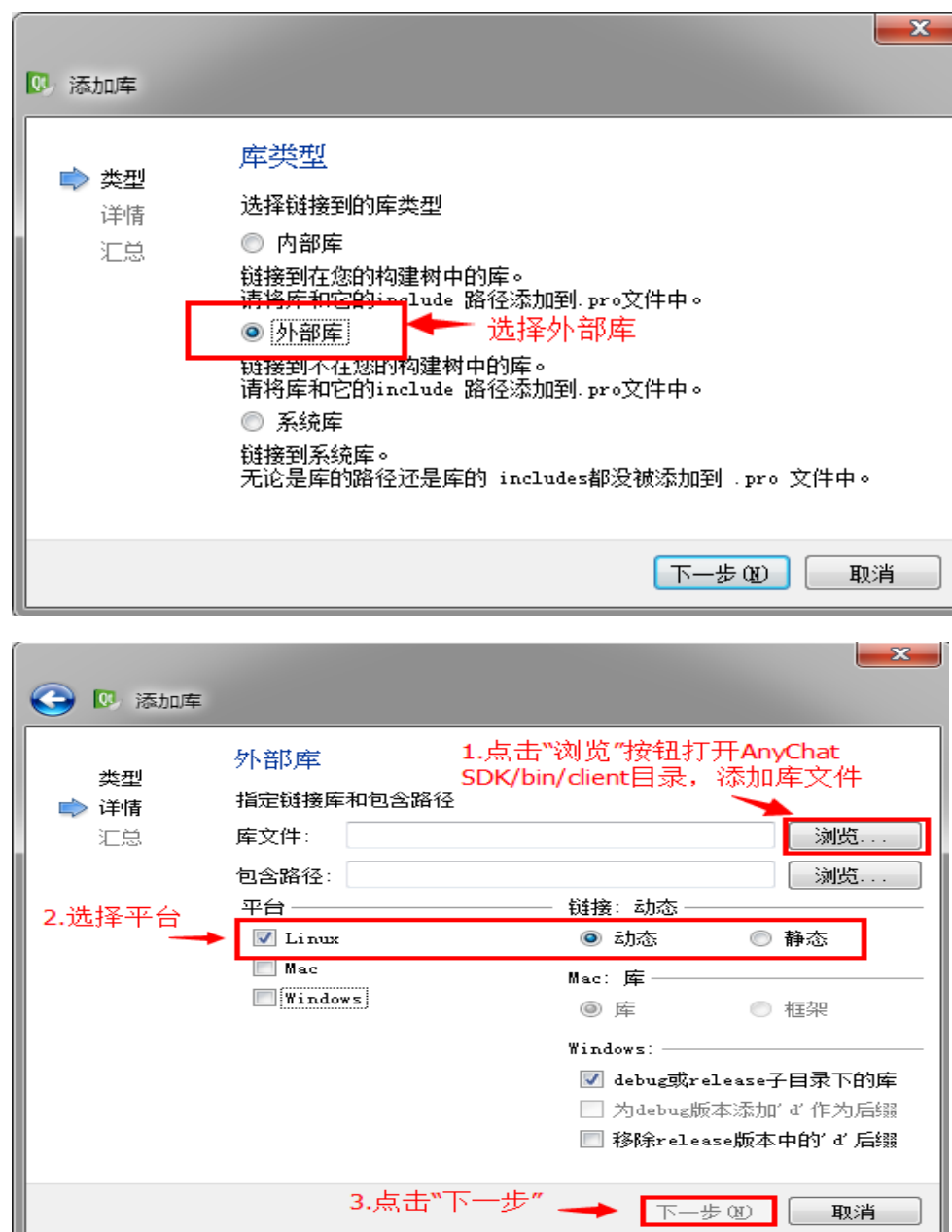
Linux 开发工具有很多，开发者可根据自己的喜好进行选择。在此，我们推荐开发者使用 Qt5 作为自己的开发工具，本套开发指南也是针对 Qt5 开发环境下进行编写的。在 Qt5 中新建一个 Qt 工程，对工程进行以下配置，搭建 AnyChat 的开发环境。

3.1 导入库文件

在工程中直接添加库文件。步骤如下：

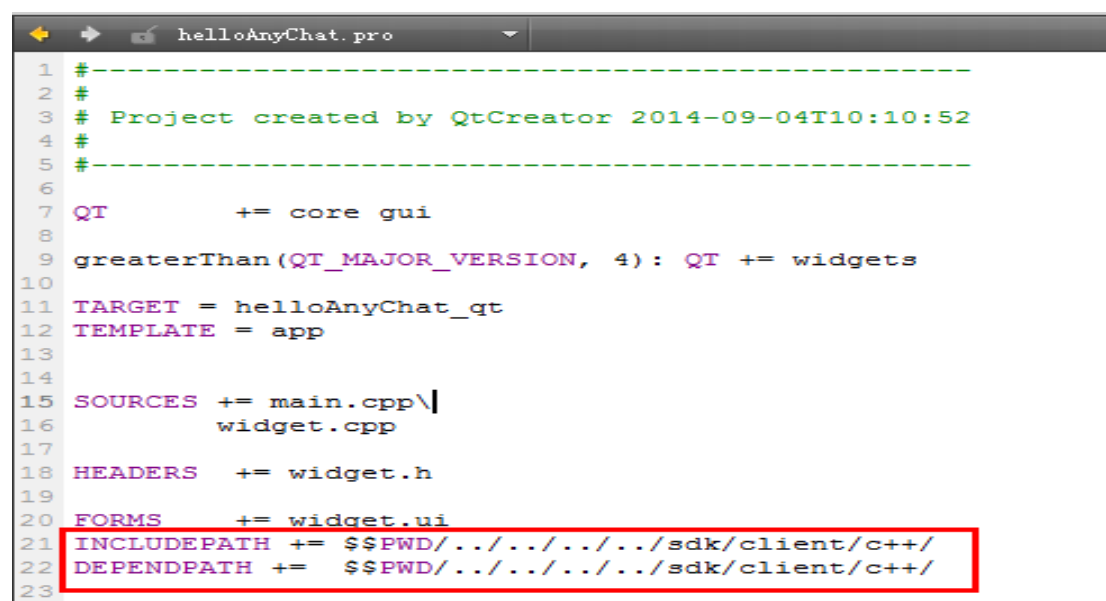
首先右键工程项目，选择“Add Library/添加库”菜单；然后在弹出的选项框中，选择“外部库”，点击“下一步”；最后点击“浏览”，打开 AnyChat SDK 包，并选择/bin/client 目录下的 libanychatcore.so 库文件，点击“下一步”和“完成”即可，如下图所示：





3.2 导入 SDK 文件

在工程的.pro 文件中添加 SDK 文件目录，方法如下图所示：



```
1 #-----
2 #
3 # Project created by QtCreator 2014-09-04T10:10:52
4 #
5 #-----
6
7 QT      += core gui
8
9 greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
10
11 TARGET = helloAnyChat_qt
12 TEMPLATE = app
13
14
15 SOURCES += main.cpp\
16          widget.cpp
17
18 HEADERS  += widget.h
19
20 FORMS    += widget.ui
21 INCLUDEPATH += $$PWD/../../../../sdk/client/c++/
22 DEPENDPATH +=  $$PWD/../../../../sdk/client/c++/
23
```

其中 INCLUDEPATH 为头文件包含路径，这里添加 AnyChat SDK 包的 SDK 目录；\$\$PWD 表示 .pro 文件当前目录；当然也可以使用绝对路径，直接填写绝对路径。

四、基本流程

在工程准备好了之后，需要实现以下基本流程，才能调用音视频交互等其他功能接口。

4.1 监听基本事件

监听“连接服务器、用户登录、进入房间、与服务器网络连接”等事件。在需要接收的类中实现以下两步。

(1) 在所需要监听的类里面实现 `NotifyMessage_CallBack` 接口，并实现其函数，参考代码如下：

```
void CALLBACK Widget::NotifyMessage_CallBack(DWORD dwNotifyMsg,
                                              DWORD wParam, DWORD lParam, LPVOID lpUserValue)
{
    Widget* pAnyChatSDKProc = (Widget*)lpUserValue;

    if(!pAnyChatSDKProc)
        return;
    switch(dwNotifyMsg)
    {
    case WM_GV_CONNECT:
        pAnyChatSDKProc->OnGVClientConnect(wParam,NULL); break;
    case WM_GV_LOGINSYSTEM:
        pAnyChatSDKProc->OnGVClientLogin(wParam,lParam); break;
    case WM_GV_ENTERROOM:
        pAnyChatSDKProc->OnGVClientEnterRoom(wParam,lParam); break;
    case WM_GV_USERATROOM:
        pAnyChatSDKProc->OnGVClientUserAtRoom(wParam,lParam); break;
    case WM_GV_LINKCLOSE:
        pAnyChatSDKProc->OnGVClientLinkClose(wParam, lParam); break;
    case WM_GV_ONLINEUSER:
        pAnyChatSDKProc->OnGVClientOnlineUser(wParam,lParam); break;
    default:
        break;
    }
    pAnyChatSDKProc->OnAnyChatNotifyMessageCallBack(dwNotifyMsg,
                                                    wParam,lParam);
}
```

其中 `Widget* pAnyChatSDKProc = (Widget*)lpUserValue` 是获取主线程对象进行操作。AnyChat 事件回调是在子线程中，而 Qt 程序的子线程不能直接操作主线程对象成员函数，需要注意在回调事件中操作主线程对象的处理。

(2) 设置基本回调事件接收。这个操作一般和第二步“初始化 SDK”放在一起，参考代码如下：

```
//设置异步消息回调
BRAC_SetNotifyMessageCallBack(NotifyMessage_CallBack,this);
```

4.2 初始化 SDK

加载资源，应用程序中只需要执行一次，其他的功能接口都必须在初始化之后才能正常使用，参考代码如下：

```
// 根据 BRAC_InitSDK 的第二个参数：dwFuncMode，来告诉 SDK 该如何处理相关 的任务（详情请参考开发文档）
DWORD dwFuncMode = BRAC_FUNC_VIDEO_AUTODISP |
                    BRAC_FUNC_AUDIO_AUTOPLAY |
                    BRAC_FUNC_CHKDEPENDMODULE |
                    BRAC_FUNC_AUDIO_VOLUME_CALC |
                    BRAC_FUNC_NET_SUPPORTUPNP |
                    BRAC_FUNC_FIREWALL_OPEN |
                    BRAC_FUNC_AUDIO_AUTOVOLUME |
                    BRAC_FUNC_AUDIO_VOLUME_CALC |
                    BRAC_FUNC_CONFIG_LOCALINI;

//初始化 SDK
BRAC_InitSDK((HWND*)this->winId(), dwFuncMode);
```

4.3 应用签名

4.3.1 应用 Id

在 AnyChat 视频云平台或 AnyChat 集群 Web 控制台中创建一个音视频应用，每个创建的应用都会有一个 AppGuid 属性值，该属性值即为应用 Id，如："78304AED-7DA7-4E14-92D9-E527AB1EA79A"。各平台的客户端通过该应用 Id 才能正常使用 AnyChat 视频云平台提供的服务或 AnyChat 服务集群。

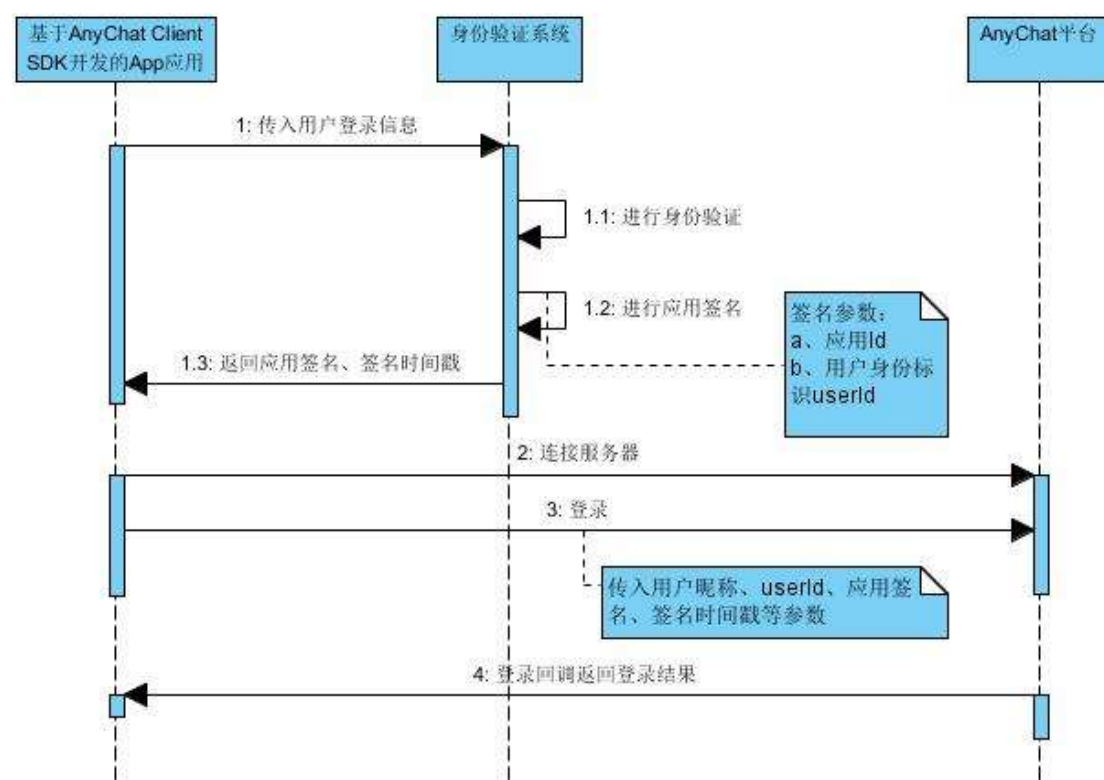
对于单独部署的非集群服务器（如独立部署的 SDK 核心服务器），则不需要关心应用 Id。

4.3.2 应用签名使用场景

在有更高安全级别的需求下,可以采用此非对称算法进行身份验证签名进行登录。在进行签名之前,开发人员可以通过 AnyChat 视频云平台或集群 Web 控制台生成应用的一对私钥、公钥信息,需要开发人员将私钥保存为一个文件;也可以由开发人员通过工具自己生成一对私钥、公钥文件。需要将私钥文件部署在身份验证系统中、公钥文件部署到 AnyChat 平台指定的位置。

当业务系统的使用用户登录系统时,需要在业务系统后端的身份验证系统中对登录的用户信息进行身份验证,验证成功之后传入需要签名的参数数据调用 AnyChat 提供的签名接口进行签名,签名接口会返回应用签名和签名时间戳数据,返回给客户端,在客户端调用 AnyChat 的 BRAC_LoginEx 接口进行登录,AnyChat 平台会对签名数据进行校验,通过登录回调返回登录结果。

应用签名过程如下图所示:



4.4连接、登录服务器

在 AnyChat 中,登录系统需要调用 BRAC_Connect、BRAC_Login、BRAC_LoginEx 等 API 接口来实现。其中 BRAC_Connect 为连接服务器的接口;BRAC_Login、BRAC_LoginEx 为登录服务器的接口,根据不同的应用场景使用不同

的登录接口，其中：**BRAC_Login** 接口实现输入用户名、密码来登录；**BRAC_LoginEx** 接口适用于应用签名登录，应用签名流程见 0 节内容介绍。

如果在系统或应用中设置了允许用户以游客的身份进行登录，则登录接口在不传入用户密码或用户身份签名数据的参数情况下，**AnyChat** 将不验证用户密码或用户身份签名，同样可以允许用户登录系统。

4.4.1 登录 SDK 平台

1) 调用 **BRAC_Login** 接口登录

独立部署的 **SDK** 核心服务器只支持一个应用连接到服务器中，通过业务服务器开发应用的业务逻辑。应用 **Id** 默认为空，核心服务器不对应用 **Id** 进行识别，因此客户端不需要设置应用 **Id**。

连接、登录代码如下所示：

```
// 连接服务器,第一个参数为你需要连接的 AnyChat 核心服务器地址,如果您部署 AnyChat 核心服务器的地址为 192.168.1.8, 则传入这个地址; 第二个参数为端口号
BRAC_Connect("192.168.1.8", 8906);
//登录服务器
BRAC_Login("AnyChat", "", 0);
```

我们对外公开测试服务器地址为 **demo.anychat.cn**。

在客户端调用登录接口后，将会触发业务服务器的回调函数“**BRAS_VerifyUser_CallBack**”，由业务层服务器完成用户的身份验证工作；客户端根据 **OnAnyChatConnect**、**OnAnyChatLoginSystem** 回调函数的返回值来判定是否通过身份验证，在此函数中进行登录成功与否的逻辑处理。

在上述调用登录接口时，也可以放在 **OnAnyChatConnect** 回调函数中处理。

2) 调用 **BRAC_LoginEx** 接口登录

在调用 **BRAC_LoginEx** 接口登录系统时，需要传入 **AppId** 参数，该参数由用户自己根据实际情况设置应用 **Id**，应用签名流程见 0 节内容介绍。

```
// 连接服务器
BRAC_Connect("demo.anychat.cn", 8906);

//登录服务器
BRAC_LoginEx("AnyChat", 1001, "",
"C782383B-FC3C-4726-815B-9D6117BF681A",1452580453,"
TBK5M9FPcDfZ8/bawWDoYW82J9yAw1Sa8GURKf4fgRrOhYGk/ODOHODNih4ox2F
4gXPdq5UK1kRGbu114dNSK2ITLfhpQ5EMxIF9XyluR/z7mIWaViMGKMDLNd9b4
p6r9b3vDBmh1OXV9oknaEELLQLiw6Ka3JIj/qTmKKPD6E=", "");
```

如果部署了 AnyChat 服务集群，客户端接入服务集群的接口调用方法同上所述，只是服务器地址改为部署的寻址服务器的 IP 地址。如何部署 AnyChat 服务集群请参见《AnyChat 服务集群部署说明文档》。

4.4.2 登录 AnyChat 视频云平台

AnyChat 视频云平台支持多个云应用的接入。在 AnyChat 视频云平台创建的每个云应用都会有一个 AppId，见 4.3.1 节内容介绍。

1) 调用 BRAC_Login 接口登录

需要调用“BRAC_SetSDKOption”接口设置连接的应用 Id。代码如下：

```
// 设置应用Id
BRAC_SetSDKOption(BRAC_SO_CLOUD_APPGUID,
"C782383B-FC3C-4726-815B-9D6117BF681A");
```

我们云平台的接入服务器地址为 cloud.anychat.cn。

连接视频云平台服务器、登录服务器接口调用代码如下所示：

```
// 连接服务器
BRAC_Connect("cloud.anychat.cn", 8906);

//登录服务器
BRAC_Login("AnyChat", "", 0);
```

在客户端调用登录接口后，将会触发业务服务器的回调函数“BRAS_VerifyUser_CallBack”，由业务层服务器完成用户的身份验证工作，客户端根据 OnAnyChatLoginSystem 回调函数的返回值来判定是否通过身份验证，在此函数中进行登录成功与否的逻辑处理。

注意：该登录方式需要将业务服务器接入到 AnyChat 视频云平台中。

2) 调用 BRAC_LoginEx 接口登录

在调用 BRAC_LoginEx 接口之前需要先进行应用签名，具体应用签名流程见 0 节内容介绍。

```
// 连接服务器
BRAC_Connect("cloud.anychat.cn", 8906);

//登录服务器
BRAC_LoginEx("AnyChat", 1001, "", "C782383B-FC3C-4726-815B-9D6117BF681A",
1452580453,"TBK5M9FPcDfZ8/bawWDoYW82J9yAw1Sa8GURKf4fgRrOhYGk/ODOH
ODNih4ox2F4gXPdq5UK1kRGbu114dNSK2ITLfhupQ5EMxIF9XyluR/z7mIWaViM
GKMDLNd9b4p6r9b3vDBmh1OXV9oknaEELQLiw6Ka3JIj/qTmKKPD6E=", "");
```

相关 API 接口详细说明请参考《AnyChat for Linux SDK 开发手册》文档

4.5 进入房间

除了音视频交互功能需要本流程之外，没有特殊说明，其他功能都不需要本流程。应用层将 roomid 传入，进入指定的房间，只有在同一个房间内的用户才能进行音视频交互。参考代码如下：

```
//进入房间
BRAC_EnterRoom( 1 , " " , 0);
```

进入房间的操作是一个异步的操作，会触发 OnGVClientEnterRoom 和 OnGVClientOnlineUser 回调。

五、音视频交互

AnyChat for Linux SDK 为开发者提供了便捷的建立音视频通讯的接口，通过以下几步操作，即可在您的应用中集成音视频交互功能。需要注意的是只有在同一个房间内的用户才能进行音视频通讯。

5.1 打开本地音视频

打开本地音视频数据需要在进入房间成功之后（参考 4.4）才有效，只有打开本地音视频，其他客户端才能请求到你的音视频数据。首先调用 `BRAC_SetVideoPos` 设置显示区域，然后调用 `BRAC_UserCameraControl` 打开视频，调用 `BRAC_UserSpeakControl` 打开音频。参考代码如下：

```
//设置本地显示区域
QRect g_LocalUserRect;
g_LocalUserRect = ui->horizontalWidgetLocalUser->geometry();
BRAC_SetVideoPos(-1, (HWND*)this->winId(), g_LocalUserRect.left(), g_LocalUserRect.top(), g_LocalUserRect.right(), g_LocalUserRect.bottom());
//打开本地音视频
BRAC_UserCameraControl(-1, TRUE);
BRAC_UserSpeakControl(-1, TRUE);
```

5.2 关闭本地音视频

打开本地音视频后，可以在音视频交互的过程中选择关闭本地音视频。同时，还可以在关闭之后重新打开本地音视频（参考 5.1）；在音视频交互结束之后需要调用该操作，释放本地摄像头和音频采集设备，参考代码如下：

```
//关闭本地音频, -1 表示自己
BRAC_UserCameraControl(-1, FALSE);
//关闭本地视频, -1 表示自己
BRAC_UserSpeakControl(-1, FALSE);
```

5.3 请求远程音视频

在触发 OnGVClientOnlineUser 或者 OnGVClientUserAtRoom 并判断通话目标对象已经进入当前房间之后，该操作才有效。首先调用 BRAC_SetVideoPos 设置显示区域，然后调用 BRAC_UserCameraControl 打开视频，调用 BRAC_UserSpeakControl 打开音频。参考代码如下：

```
QRect g_RemoteUserRect;  
g_RemoteUserRect = ui->horizontalWidgetRemoteUser->geometry();  
BRAC_SetVideoPos(userid, (HWND*)this->winId(), g_RemoteUserRect.left()  
( ), g_RemoteUserRect.top(), g_RemoteUserRect.right(), g_RemoteUserRect.bottom());  
g_RemoteUserRect.bottom());  
//打开新请求用户视频  
BRAC_UserCameraControl(userid, TRUE);  
BRAC_UserSpeakControl (userid, TRUE);
```

5.4 关闭远程音视频

请求远程音视频后，可以在音视频交互的过程中选择关闭远程音视频。同时，还可以在关闭之后重新请求远程音视频（参考 5.3）；在音视频交互结束之后需要调用该操作，释放远程音视频资源，参考代码如下：

```
//关闭远程音频，userid 为远程目标用户 userid  
BRAC_UserCameraControl(userid, FALSE);  
//关闭远程视频  
BRAC_UserSpeakControl(userid, FALSE);
```

六、释放过程

(1) 离开房间

释放当前房间内的音视频资源。参考代码如下：

```
//离开指定房间，-1 表示离开当前房间  
BRAC_LeaveRoom(-1);
```

在音视频交互结束后，可调用该操作。离开当前房间之后，可再次选择进入指定房间。

(2) 退出

断开与 AnyChat 通讯服务器连接。参考代码如下：

```
//断开与服务器的连接  
BRAC_Logout();
```

在需要断开跟 AnyChat 服务器通讯连接的时候，可调用该操作。退出之后，可以再次调用连接、登录服务器。

(3) 释放 SDK

释放整个 SDK 资源。参考代码如下：

```
//释放资源  
BRAC_Release();
```

建议在退出整个应用程序的时候调用该操作。释放 SDK 之后，需要重新初始化 SDK 之后才能进行连接、登录、进入房间等操作。

七、附录

AnyChat for Linux SDK 包提供的 HelloAnyChat 程序（源码在“src\client\Qt5\helloAnyChat”目录下）最终运行效果如下图所示：

