**VoIP C++ SDK 集成说明**



Copyright © 2013-2017 Easiio, Inc. All Rights Reserved.

Version 2.0.1更新

1、网络状态差下提高视频通话流畅度.

1.VoIP C++ SDK 介绍  
VoIP C++ SDK 针对语音及视频传输和编码做了相当多的技术上和性能上的优化。增加了多方视频会议功能，支持移动终端的高清语音和视频，优化网络通讯，降低丢包率，同时确保在高丢包率情况下也能保证语音和视频通话的正常进行。  
  
**2. VoIP C++ SDK接入步骤**2.1 VoIP C++ SDK文件说明：

\* EasiioLibrary.h 是SDK的函数头文件

\* EasiioLibrary.dll 是SDK 的动态库文件

EasiioLibrary.lib 是动态链接库的索引lib(参看dll隐式调用)

\* 其他dll文件都为必须动态库文件，需要一并放到运行文件夹下

dll文件放在：



2.2 VoIP C++ SDK 集成说明

2.2.1 初始化

（1）传入回调函数 EASIIO\_CALLBACK\_INTERFACE

（2）初始化Developer Key 和 Token

EASIIO\_CALLBACK\_INTERFACE \*cbInterface;

cbInterface = new EASIIO\_CALLBACK\_INTERFACE;

memset(cbInterface, 0, sizeof(EASIIO\_CALLBACK\_INTERFACE));

cbInterface->onLoginResult = &onLoginResult;

cbInterface->onPJSIPServiceStateChanged = &onPJSIPServiceStateChanged;

cbInterface->onCallStatusChanged = &onCallStatusChanged;

cbInterface->onTwoWayCallStatusChanged = &onTwoWayCallStatusChanged;

cbInterface->onIncomingCall = &onIncomingCall;

cbInterface->onMakeTwoWayCallResult = &onMakeTwoWayCallResult;

cbInterface->onCallRecordingReaponse = &onCallRecordingReaponse;

cbInterface->onGetAllRecordResponse = &onGetAllRecordResponse;

cbInterface->onGetRecordResponse = &onGetRecordResponse;

cbInterface->onMakeCallResult = &onMakeCallResult;

cbInterface->onMeetingResult = &onMeetingResult;

// 初始化回调函数

EasiioInit(cbInterface);

// 初始化Developer Key 和 Token

EasiioInitDeveloperKeyAndToken("[Developer Key]", "[Token]");

2.2.2 登录与退出

1. 登录

调用EasiioLogin(const char\* account, int paramCount, EasiioLoginParams\* params);

account: 第三方集成账号，直接使用集成应用原有的账号体系的登录用户名。

paramCount: 附加参数数量

params: 附加参数指针首地址,参数的具体使用可参考Demo。

登录结果回调void(\*onLoginResult)(int resultCode, const char\* resultMsg);

（2）退出

调用EasiioLogout(); SDK会注销PJSIP状态

2.2.3 PJSIP状态回调

void(\*onPJSIPServiceStateChanged)(int stateCode); // PJSIP状态变化通知

stateCode具体表示含义参看EasiioLibrary.h 的PJSIP State说明

2.2.4 VoIP呼叫

调用EasiioMakeCall(const char\* number, int callType, const char\* postValue);

Number: 呼叫的号码

callType: 呼叫类型

postValue: 传递参数，不超过256个字符，且使用无换行形式的BASE64加密，否 则不保证传递成功，且有可能导致呼叫失败。

状态回调：

void(\*onMakeCallResult)(int callId, const char\* toNumber，const char\* meetingkey);  
 呼叫结果，参考EasiioLibrary.h

void(\*onCallStatusChanged)(int callId, int callState, const char\* resultCode, const char\* callUUID，);

参看EasiioLibrary.h，其中CallUUID可用来获取录音文件。

2.2.5 双向呼叫

调用 EasiioMakeTwoWayCall(const char\* toNumber, const char\* fromNumber, int openRecord);

toNumber: 被叫号码

fromNumber: 主叫号码

openRecord: 是否开启录音

调用结果回调：

void(\*onMakeTwoWayCallResult)(EasiioResponseReason reason, const char\* caller, const char\* callee, const char\* callUUID, const char\* retryId);

双向呼叫通话状态回调：

void(\*onTwoWayCallStatusChanged)(int callState, const char\* callUUID, const char\* caller, const char\* callee, int retrying);

2.2.6 通话操作

1. 应答

调用 EasiioAnswerCall(int callId, int callType);

callId 是来电通知里的参数，如下：

void(\*onIncomingCall)(int callId, int callType, const char\* fromNumber, const char\* postValue); // 来电通知

callType目前支持Voice

1. 拒接

调用 EasiioRejectCall(int callId);

callId 是来电通知里的参数，同应答。

1. 挂断

调用EasiioHangupCall(int callId);

callId，如果是来电同应答，如果是外呼，则是调用make call的返回值。

1. 保持/解除保持

保持调用 EasiioHoldCall(int callId);

解除保持调用 EasiioUnHoldCall(int callId);

callId：同挂断说明

1. 静音/解除静音

静音调用 EasiioMuteCall(int callId);

解除静音调用 EasiioUnMuteCall(int callId);

callId：同挂断说明

1. 发送DTMF

调用EasiioSendDTMF(int callId, const char\* dtmf);

callId：同挂断说明

Dtmf： 数字0-9，\*，# 单个字符的字符串

1. 调节麦克风音量

调用EasiioAdjustMicLevel(int callId, int level);

callId：同挂断说明

Level: 0-40

1. 开关通话中录音

调用 EasiioSwitchRecordInCall(int recordSwitch, const char\* callUUID);

recordSwitch: 录音开关

CallUUID: 通过通话状态回调获取

void(\*onCallStatusChanged)(int callId, int callState, const char\* resultCode, const char\* callUUID);

调用结果回调：

void(\*onCallRecordingReaponse)(EasiioResponseReason reason, int switchRecord, const char\* callUUID);

2.2.7 获取当前登录的PBX号码

调用：EasiioGetCurrentPBXAccount(char\* pbxAccount);

2.2.8 获取录音

（1）获取所有录音通话列表

调用： EasiioGetAllRecord(int recordType);

recordType: 分为通话中录音和整个通话录音两种type，请区分使用

结果回调：

void(\*onGetAllRecordResponse)(EasiioResponseReason reason, int count, EasiioRecord \*record);

1. 获取单个通话录音列表

调用： EasiioGetRecord(int recordType, const char\* callUUID);

recordType: 同上

CallUUID: 可通过通话状态变化获取或者通过获取所有通话录音返回的值中的

CallUUID.

结果回调：

void(\*onGetRecordResponse)(EasiioResponseReason reason, const char\* callUUID, int count, EasiioRecordItem \*recordItem);

2.2.9 监控摄像头

(1) 获取监控摄像头数量

调用： EasiioGetDeviceCount(int \*DevCount);

(2) 获取单个摄像头信息

调用：EasiioGetDeviceInfoById(int index, char \*deviceInfo);

index：设备索引，如果摄像头数量为n，索引的大小为0到(n-1);

(3) 启动监控预览

调用：EasiioStartPreview(int size, char \*\*urlArray);

size: url数组的数量大小

urlArray： 指具有rtsp流的url数组，url主要从摄像头信息中获取

(4) 关闭监控预览

调用：EasiioStopPreview();

(5) 获取监控预览状态

调用：EasiioPreviewStatus(int \*status);

2.3.0 视频会议

开会前需要确定是否支持视频会议(若不支持请联系本公司管理员),视频会议分为单层和多层，单层：只有主持人能收到加入会议的来电。 多层：允许除了主持人外的会议成员收到加入会议的来电，层层递增，让会议人数达到更多。 (默认的是单层,如要修改请联系本公司管理员)

(1)创建会议

调用：EasiioCreateMeeting(const char\* desc, int priority);

回调结果：

void(\*onMeetingResult)(EasiioResponseReason reason,int count, MeetingInfo \*info);

(2)获取会议列表

调用：EasiioGetMeetingList();

回调结果：

void(\*onMeetingResult)(EasiioResponseReason reason,int count, MeetingInfo \*info);

(3)删除会议

调用：EasiioDeleteMeeting(const char\* meeting\_key);

(4)开启会议

调用：EasiioStartMeeting(const char\* meeting\_key,const char\* resolution,const char\* platform,char\* start\_time);

resolution: 视频会议时的分辨率，与结束会议的resolution不一定相同

(5)结束会议

调用：EasiioStopMeeting(const char\* meeting\_key, const char\* resolution, const char\* platform);

resolution: 正常视频分辨率，与会议时的resolution不一定相同

(6)获取参会人员

主持人显示参会人员和加入会议时调用

调用：EasiioGetParticipantInfo(const char\* meeting\_key, ParticipantInfo \*participantinfo);

(7)会议状态

开启会议和加入会议前可以先检测会议状态

调用：EasiioGetMeetingStatus(const char\* meeting\_key);

(8)加入会议

调用：EasiioJoinMeeting(const char\* meeting\_key);

状态回调：void(\*onMakeCallResult)(int callId, const char\* toNumber，const char\* meetingkey);

Meetingkey: 有返回meetingkey就是呼叫成功

(9)设置视频分辨率

参会人员退出会议后需要重新设置下视频分辨率,因为会议的分辨率和正常视频分辨率不一定相同。

调用：EasiioSetVideoResolution(const char\* resolution);

(10)更行参会时的信息

参会人员退出会议后需要调用该函数

调用：EasiioUpdateJoinMeetingInfo(const char\* meeting\_key,const char\* operate,const char\* easiio\_id);

(11)参会人员退出会议时切换请求(多层会议时需要)

通知下一层参会人员重新参会

调用：EasiioSendSwitchUpstreamRequest(int callId, int type);

2.3.1 通话结束后获取语音丢包率

调用：EasiioGetLastAvePktLostRate();

返回值：丢包率

2.3.2 销毁CPP SDK

关闭应用时候请调用EasiioDestroy(); 区别于账号退出，账号退出请调用EasiioLogout.

2.3.3 CPP SDK日志

SDK会保存运行的日志在安装文件夹下，例如EasiioLibraryLog-2016-03-14.log

或者调用void EasiioSetLogDir(const char\* logDir);可以设定log保存路径，需要先创建好。