

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：

名称	修改日期	类型
Bin	2017/7/10 13:03	文件夹
demo	2017/7/10 13:08	文件夹
dll	2017/7/10 10:35	文件夹
doc	2017/7/10 13:47	文件夹
include	2017/7/10 10:36	文件夹
lib	2017/7/10 10:39	文件夹

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->onCallRecordingResponse = &onCallRecordingResponse;
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  
EasiiInitDeveloperKeyAndToken("[Developer Key]", "[Token]");
```

2.2.2 登录与退出

(1) 登录

调用 `EasiiLogin(const char* account, int paramCount, EasiiLoginParams* 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 `EasiiLogout();` 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 `EasiiLibrary.h` 的 PJSIP State 说明

2.2.4 VoIP 呼叫

调用 `EasiiMakeCall(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 `EasiiLibrary.h`
`void(*onCallStatusChanged)(int callId, int callState, const char* resultCode, const char* callUUID,);`
参看 `EasiiLibrary.h`, 其中 CallUUID 可用来获取录音文件。

2.2.5 双向呼叫

调用 `EasiiMakeTwoWayCall(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
```

(2) 拒接

调用 EasiioRejectCall(int callId);

callId 是来电通知里的参数, 同应答。

(3) 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。

(4) 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

(5) 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

(6) 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

(7) 调节麦克风音量

调用 EasiioAdjustMicLevel(int callId, int level);

callId: 同挂断说明

Level: 0-40

(8) 开关通话中录音

调用 `EasiiSwitchRecordInCall(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(*onCallRecordingResponse)(EasiiResponseReason reason, int switchRecord, const char* callUUID);`

2.2.7 获取当前登录的 PBX 号码

调用: `EasiiGetCurrentPBXAccount(char* pbxAccount);`

2.2.8 获取录音

(1) 获取所有录音通话列表

调用: `EasiiGetAllRecord(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)(EasiiResponseReason reason, int count, EasiiRecord *record);`

(2) 获取单个通话录音列表

调用: `EasiiGetRecord(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)(EasiiResponseReason reason, const char* callUUID, int count, EasiiRecordItem *recordItem);`

2.2.9 监控摄像头

(1) 获取监控摄像头数量

调用: `EasiiGetDeviceCount(int *DevCount);`

(2) 获取单个摄像头信息

调用: `EasiiGetDeviceInfoById(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: `EasiiioGetMeetingStatus(const char* meeting_key);`

(8) 加入会议

调用: `EasiiioJoinMeeting(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: `EasiiioSetVideoResolution(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: `EasiiioUpdateJoinMeetingInfo(const char* meeting_key, const char* operate, const char* easiiio_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: `EasiiioSendSwitchUpstreamRequest(int callId, int type);`

2.3.1 通话结束后获取语音丢包率

调用: `EasiiioGetLastAvePktLostRate();`

返回值: 丢包率

2.3.2 销毁 CPP SDK

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

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 `EasiiioLibraryLog-2016-03-14.log` 或者调用 `void EasiiioSetLogDir(const char* logDir);` 可以设定 log 保存路径, 需要先创建好。