



编号: CIN/CC-12

SIPPhone-SDK for Android 开发手册

册

V2.0.0.8

北京新方通信技术有限公司

2023年2月

目 录

第 1 章	介绍.....	3
1.1	范围.....	3
1.2	参考.....	3
1.3	术语及约定.....	3
第 2 章	整体结构.....	3
第 3 章	接口内容.....	3
3.1	属性.....	3
3.2	方法.....	6
3.2.1	<i>Instance</i>	6
3.2.2	<i>GetVersion</i>	6
3.2.3	<i>Initial</i>	7
3.2.4	<i>UnInitial</i>	7
3.2.5	<i>Register</i>	8
3.2.6	<i>UnRegister</i>	8
3.2.7	<i>DoCall</i>	9
3.2.8	<i>Answer</i>	9
3.2.9	<i>Disconnect</i>	10
3.2.10	<i>Reject</i>	10
3.2.11	<i>SendDtmf</i>	11
3.2.12	<i>SetSpeakMode</i>	11
3.2.13	<i>Mute</i>	12
3.2.14	<i>SendMessage</i>	12
3.2.15	<i>SendInfo</i>	13
3.2.16	<i>GetSpKey</i>	13
3.2.17	<i>GetSipCall</i>	13
3.2.18	<i>SetVedioPlay</i>	15
3.2.19	<i>SetVedioView</i>	16
3.2.20	<i>ChangeCamera</i>	16
3.2.21	<i>SetMediaConsultationResult</i>	17
3.2.22	<i>UploadLog</i>	17
3.2.23	<i>enableFaceDetection</i>	18
3.2.24	<i>RegisterEvent</i>	18
3.2.25	<i>StartScreenRecord</i>	19
3.2.26	<i>StopScreenRecord</i>	19
3.2.27	<i>Update</i>	19

3.2.28	<i>Share</i>	20
3.3	事件 (SIPPHONECTRL.IEVENT)	20
3.3.1	<i>onRegistrationStateChanged</i>	20
3.3.2	<i>onCallStateChanged</i>	21
3.3.3	<i>onMessage</i>	22
3.3.4	<i>onInfo</i>	22
3.3.5	<i>onMediaStatics</i>	23
3.3.6	<i>onMediaConsultation</i>	23
3.3.7	<i>onDebugEvent</i>	24
3.3.8	<i>onInitialStatusChange</i>	25
3.3.9	<i>onFacePositionChanged</i>	25
3.3.10	<i>onScreenRecord</i>	26
3.3.11	<i>onCallStatistics</i>	26
第 4 章	开发指南	27
4.1	开发步骤	27
4.1.1	<i>AndroidManifest.xml</i> 加入	27
4.1.2	把 <i>sdk</i> 和支持库复制到 <i>libs</i>	28
4.1.3	把资源文件复制到 <i>res/raw</i> 下.....	28
4.1.4	非 <i>SDK</i> 接口调用设置 (API 28)	28
4.1.5	混淆注意事项.....	29
4.2	DEMO.....	29
	修改历史	29

第1章 介绍

1.1 范围

本文是一个标准的 android 版本的 sip 电话 SDK。

1.2 参考

《rfc3621》

《rfc2327》

1.3 术语及约定

第2章 整体结构

SDK 结构：主要包括几个部分，sip 协议栈、rtp 协议栈等，对外提供属性、方法、事件。具体内容下面详细描述。

SDK 中有一个对外接口的全局静态对象 SipPhoneCtrl.Instance()，以前属性方法和事件都是来自这个对象。

第3章 接口内容

3.1 属性

【功能】

设置/得到属性

【函数】

```
String GetXXX() 或者 SetXXX(string strValue)
```

```
int GetXXX() 或者 SetXXX(int nValue)
```

【参数 XXX】

属性	说明	备注
----	----	----

属性	说明	备注
Server	String: 服务器 IP 地址	必填
Domain	String: 服务器域名地址, 通常和 Server 一样	必填
ServerPort	Int : 服务器端口号	必填
Number	String: 电话号码	必填
PassWord	String: 密码	必填
AuthName	String: 鉴权名	必填
AuthType	integer : 鉴权类型 0: 不鉴权 1: DEGIST(默认)	
SipHeartBeatInterval	integer : 心跳消息间隔, 默认时间是 3 秒	
SipHeartBeatType	integer : 心跳消息类型, 0: 回车换行 1: option	
MeidaType	枚举类型: 媒体类型 SipCoreEvent.MediaType.RTPVoice (默认) SipCoreEvent.MediaType.AgoraVoice SipCoreEvent.MediaType.WebrtcVoice SipCoreEvent.MediaType.XYlinkVoice	
VideoMode	枚举类型: 视频展示方式, 对 webrtc 方式有效 Webrtc_Fit Webrtc_Full Webrct_Bananced	
TraceFlag	integer : 日志类型: 0: 没有日志 1: sipcore 日志 2: sipstack 日志 3: sipcore+sipstack 日志	
FileFlag	integer : 日志 0: 不写文件, 通过 OnDebugMessage 事件上报; TraceFlag 为 0, 日志不上报 1: 写文件, 不上报, 可以通过 UploadLogFile 上传;	
SipService	boolean : 是否使用服务, true: 使用, false: 不使用, 默认: 使用	
WakeUpApp	boolean : 是否后台唤醒, true: 唤醒 false: 不唤醒	
SurpportCodecs	String : 设置支持的编解码,	
DefaultCodec	String : 设置默认音频编解码	
DefaultVCodec	String : 设置默认视频编解码	

属性	说明	备注
AllCodecs	String[] : 得到 SIP 电话支持的所有编解码 get AllCodecs, 目前支持: "Ulaw;0", "Alaw;8", "ILbc;102", "Opus;120", "Agora;121", "XYlink;122", "H264;127", "VP8;100", "VP9;101"	
SurpportVedio	boolean : 是否支持视频, true: 支持 false: 不支持, 默认: 不支持	
ServiceKey	String: 注册是增加消息头, 用于特殊项目, 如增加一个消息头格式: "Token: 123213213213213"; 如果增加多个消息头中间使用 " " 分割 比如: "AAA: 123213213213213 BBB: 1232132132132"	
RouteKey	String : 用于区分业务使用	
SipEncryptType	Int: sip 消息加密 0: 不加密 (默认) 1: 加密	
AutoRegister	boolean : 是否自动注册 true: 自动注册 (默认) false: 不自动注册	
TransferMode	传输模式, 枚举值, 目前支持: TransferMode_Udp TransferMode_WebSocket	
XPath	当传输类型为 TransferMode_WebSocket 时, 设置的服务器路径, 例如: "acd"	
GmMode	boolean : 设置是否支持国密: (默认) false: 不支持国密。设置方式区别于其他属性, 示例代码: WebSocketCtrl.setGmMode(true);	
LoadSingleMediaLib	boolean : 加载单种媒体库 true: 是 false: 加载所有; 默认: false, 加载所有	
CallStatics	boolean : 媒体统计 true: 启动媒体统计 false: 不启动 默认: 不启动, 只对 RTP 有效	
EcEnable	boolean : 开启回声消除 true: 开启 false:不开启 默认:true	
NsEnable	boolean : 开启降噪 true: 开启 false:不开启 默认:true	
AgcEnable	boolean : 开启声音增益 true: 开启 false:不开启 默认:false	

属性	说明	备注
TimerType	枚举类型：定时器类型 TimerType_so, 内部 TimerType_tasktime, 内部 TimerType_outside, 外部 TimerType_handle 内部 TimerType_outside 时要求 app 端启动定时，每 100ms 调用一次 Iterate	
LogPath	设置日志上传的目录	

【示例代码】

```
SipPhoneCtrl.Instance().SetServer("192.168.2.136");
```

3.2 方法

3.2.1 Instance

【功能】

返回 SipPhoneCtrl 对象，静态对象

【预置条件】

1)、引用 cinsip-sdk 库成功;

【函数】

```
SipPhoneCtrl Instance ()
```

【参数】

无

【返回值】

SipPhoneCtrl 对象

3.2.2 GetVersion

【功能】

得到版本号码

【预置条件】
【函数】

```
String GetVersion()
```

【参数】

无

【返回值】

3.2.3 Initial

【功能】

初始化电话条对象

【预置条件】

1)、已经设置了电话的各种参数(主要是 Server, ServerPort, Domain 等参数);

【函数】

```
boolean Initial(Context context, OneInitParamCallBack callback)
```

【参数】

参数	说明
Context	android 窗口对象
OneInitParamCallBack	初始化回调, 回调中 code: 200: 网络连接成功 其他值底层上报: 网络连接失败

【返回值】

true: 成功

false: 失败

【后处理事件】

触发事件 OnInitialStatusChange;

设置了 AutoRegister 参数为 true, 会自动注册触发事件

3.3.1 onRegistrationStateChanged;

【重要说明】

当 TransferMode 为 TransferMode_Udp, AutoRegister, OneInitParamCallBack 为非空, 当回调为 200 时表示网络可用。前提服务器支持 option。

3.2.4 UnInitial

【功能】

析构电话条对象

【预置条件】

1)、Initial 成功;

【函数】

```
boolean UnInitial()
```

【参数】

无

【返回值】

true:成功

false: 失败

3.2.5 Register

【功能】

注册

【预置条件】

- 1)、Initial 成功;
- 2)、已经设置了电话的各种参数(主要是 Number, PassWord, AuthName, AuthType 等参数);

【函数】

```
boolean Register()
```

【参数】

无

【返回值】

true:成功

false: 失败

【后处理事件】

触发事件 3.3.1 onRegistrationStateChanged;

3.2.6 UnRegister

【功能】

注销

【预置条件】

- 1)、Initial 成功;
- 2)、Register 成功;

【函数】

```
boolean UnRegister()
```

【参数】

无

【返回值】

true:成功

false: 失败

【后处理事件】

触发事件 3.3.1 onRegistrationStateChanged;

3.2.7 DoCall

【功能】

外呼一个目标号码

【预置条件】

1)、Initial 成功;

【函数】

```
boolean DoCall(String strNum, String extraHeader)
```

【参数】

参数	说明
strNum	String: 号码 特别说明: 如果是没有注册可以使用 1342622@211.103.220.86 呼到某个服务器
extraHeader	String, 外呼时增加了自定义的扩展协议头, 便于扩展 业务, 多个协议头之间使用“ ”作为分隔, 格式为: “AAAA:899889 BBBB:34534534”

【返回值】

true:成功

false: 失败

【特殊说明】

目前支持 rtp、webrtc 和 agora 呼叫, 呼叫类型可以通过 MediaType 来控制, 对于 agora 呼叫, 需要把 extraHeader 设置为 Agora-Info:agoravideo;

【后处理事件】

触发事件 3.3.2 onCallStateChanged;

Agora 类型呼叫会触发 onMediaConsultation, agora 呼叫可以参考 demo

3.2.8 Answer

【功能】

来电应答

【预置条件】

1)、电话初始化成功;

2)、有电话振铃;

【函数】

```
boolean Answer()
```

【参数】

无

【返回值】

true:成功

false: 失败

【后处理事件】

触发事件 3.3.2 onCallStateChanged;

3.2.9 Disconnect

【功能】

挂断电话

【预置条件】

- 1)、电话初始化成功;
- 2)、电话接通;

【函数】

```
boolean Disconnect()
```

【参数】

无

【返回值】

true:成功

false: 失败

【后处理事件】

触发事件 3.3.2 onCallStateChanged;

3.2.10 Reject

【功能】

唤醒拒接

【预置条件】

- 1)、Initial 成功;

【函数】

```
int Reject: (NSString*) destNum
```

【参数】

参数	说明
destNum	String: 拒接标示 格式: "APP 呼叫 callID",由唤醒消息传回

【返回值】

0:成功

其它: 失败

【后处理事件】

触发事件 3.3.2 onCallStateChanged;

3.2.11 SendDtmf

【功能】

二次拨号

【预置条件】

- 1)、Initial 成功;
- 2)、有呼叫接通;

【函数】

```
boolean SendDtmf(int tapKey, int dtmfType)
```

【参数】

参数	说明
tapKey	int: 二次拨号键 键值: 1(1), 2(2), 3(3), 4(4), 5(5), 6(6), 7(7), 8(8), 9(9), 10(*), 0(0), 11(#), 12(A), 13(B), 14(C), 15(D)
dtmfType	Int 二次播放方式 0: Info 消息, application/dtmf-relay 1: Info 消息, application/dtmf 2: 带内

【返回值】

true: 成功

false: 失败

3.2.12 SetSpeakMode

【功能】

通话接通后, 在“免提”和“听筒”两种模式下切换

【预置条件】

- 1)、电话初始化成功;
- 2)、有电话接通;

【函数】

```
boolean SetSpeakMode()
```

【参数】

无

【返回值】

true: 设置为免提模式;

false: 设置为听筒模式;

注意: 每次新通话默认是听筒模式

3.2.13 Mute

【功能】

设置呼叫中本端静音

【预置条件】

1)、呼叫成功, 通话中

【函数】

```
boolean Mute(boolean bMute)
```

【参数】

参数	说明
bMute	Boolean, 是否本端静音。true: 静音, false: 不静音

【返回值】

true: 成功

false: 失败

【注意】: 每次新通话默认是双方呼通状态

3.2.14 SendMessage

【功能】

呼叫中给对方发送文本消息

【预置条件】

1)、呼叫成功, 通话中

【函数】

```
boolean SendMessage(String msg)
```

【参数】

参数	说明
msg	String, 文本信息

【返回值】

true: 成功

false: 不在呼叫中, 失败

3.2.15 SendInfo

【功能】

呼叫中使用 Info 方法给服务端发送 SIP 文本消息

【预置条件】

1)、呼叫成功, 通话中

【函数】

```
boolean SendInfo(String conType, String msg)
```

【参数】

参数	说明
conType	String, 发送消息的格式, 需要与服务端商定。例如: text/plain
msg	String, 文本信息

【返回值】

true: 成功

false: 不在呼叫中, 失败

3.2.16 GetSpKey

【功能】

得到呼叫中关键字

【预置条件】

1)、呼叫成功, 通话中

【函数】

```
String GetSpKey (String key)
```

【参数】

参数	说明
key	String, 关键字, 特殊业务商定

【返回值】

String, 得到关键字的值;

3.2.17 GetSipCall

【功能】

得到当前呼叫信息, 包括注册信息

【预置条件】

无

【函数】

```
SipCoreCall GetSipCall()
```

【参数】

无

【返回值】

SipCoreCall 对象

说明：

属性	说明	备注
callState	类型：CallState： 呼叫状态类型 CallState.IncomingReceived 来电振铃 CallState.OutgoingInit 去电振铃 CallState.CallEnd 电话结束 CallState.Error 电话异常 CallState.Connected 电话条接通 CallState.StreamsRunning 语音通话	呼叫状态
sNum	类型：String	对方号码
sCurID	类型：String	会话 ID
nCode	类型：int： sip 的标准错误码	错误码
sDescription	类型：String	错误描述
regState	类型：RegistrationState： 注册消息类型： RegistrationState.RegistrationNone 没有注册 RegistrationState.RegistrationFailed 注册失败 RegistrationState.RegistrationCleared 注销成功 RegistrationState.RegistrationProgress 正在注册 RegistrationState.RegistrationOk 注册成功	注册状态
regCode	类型：int 注册码： 200： OK 401： Unauthorized; 403： Forbidden; 404： Not Found; 404： Request Timeout;	注册码

属性	说明	备注
mediaState	类型: MediaState 媒体类型: MediaState.mediaNone 没有媒体 MediaState.mediaAudio 音频 MediaState.mediaVedio 视频 MediaState.mediaAudioVedio 音频和视频	媒体类型 (通话中)
mediaType	类型: 呼叫的媒体传输方式 MediaType.NoneVoice 没有通话 MediaType.AgoraVoice, agora 方式 MediaType.RTPVoice, rtp 方式 MediaType.WebrtcVoice webrtc 方式	
mediaVedioInfo	类型: VedioInfo (包含): inFps int: coming framerate outFps int: going framerate inKbps int: coming bitrate outKbps int: going bitrate inWidth int: coming Width inHeight int: coming Height packetLoss int: coming packetLoss percent 100	视频信息 视频通话
endCause	CalEndCause 上一通呼叫结束类型: CalEndCause.NoCall -1 没有呼叫 CalEndCause.NoAnswerCall 0 无应答呼叫 CalEndCause.RejectCall 1 拒接呼叫 CalEndCause.ConnectCall 2 接通呼叫 CalEndCause.OtherFailureCall 3 其他原因呼叫	只有 callState 为 CallState.CallEnd 或者 CallState.Error 时有效。

3.2.18 SetVedioPlay

【功能】

呼叫时设置视频是否发送接收视频流

【预置条件】

1)、呼叫成功, 视频通话中

【函数】

```
boolean SetVedioPlay (boolean bSend, boolean bRecv)
```

【参数】

参数	说明
bSend	Boolean, 是否发送视频流。true: 发送, false: 不发送
bRecv	Boolean, 是否接收视频流。true: 接收, false: 不接收

【返回值】

true: 成功

false: 失败

3.2.19 SetVedioView

【功能】

设置视频本地和远程显示窗口

【预置条件】

1)、初始化成功

【函数】

```
void SetVedioPlay (ViewGroup localView, ViewGroup remoteView)
```

【参数】

参数	说明
localView	ViewGroup, 本地视频窗口
remoteView	ViewGroup, 远程视频窗口

【返回值】

无

3.2.20 ChangeCamera

【功能】

视频通话接通后, 在“前面摄像头”和“后面摄像头”两种模式下切换

【预置条件】

1)、电话初始化成功;

2)、有 webrtc 视频电话接通;

【函数】

```
boolean ChangeCamera ()
```

【参数】

无

【返回值】

true: 设置为前面摄像头模式;

false: 设置为后面摄像头模式;

注意: 每次新通话默认是前面摄像头模式

3.2.21 SetMediaConsultationResult

【功能】

设置 Agora 呼叫加入呼叫的结果

【预置条件】

- 1)、初始化成功
- 2)、有 Agora 呼叫呼出或者呼入
- 3)、有 onMediaConsultation 触发

【函数】

```
boolean SetMediaConsultationResult(String strKey, String strParam)
```

【参数】

参数	说明
strKey	String, 类型 “OnAgoraVoiceJoinSuccess”：加入 agora 会议成功； “OnAgoraVoiceError”：加入 agora 会议失败；
strParam	String, 错误原因, 来自 agora 的错误码

【返回值】

0

3.2.22 UploadLog

【功能】

上传底层日志到指定服务器上

【预置条件】

- 1)、初始化成功
- 2)、设置了本机号码, 属性 FileFlag 为 1

【函数】

```
void int UploadLog(String strServer, int port, Context context)
```

【参数】

参数	说明
strServer	String, 日志文件服务器
port	port, 端口号
Context	android 窗口对象

【返回值】

-1: 属性 FileFlag 为 0, 无法上传文件;

0: 上传文件命令执行成功;

【后处理事件】

触发事件 3.3.3 onMessage(code=100);

3.2.23 enableFaceDetection

【功能】

开启人脸检测

【预置条件】

- 1)、媒体通道选择 WebRTC
- 2)、开启视频通话
- 3)、呼叫成功, 通话中

【函数】

```
void enableFaceDetection(boolean enableFaceDetect, Context context)
```

【参数】

参数	说明
enableFaceDetect	是否开启人脸检测功能, 默认关闭
context	android 上下文对象

【后处理事件】

触发事件 3.3.8 onFacePositionChanged

3.2.24 RegisterEvent

【功能】

设置 SipCtrl 的回调事件

【预置条件】

- 1)、初始化之前

【函数】

```
void RegisterEvent (SipPhoneCtrl.IEvent event)
```

【参数】

参数	说明
event	SipPhoneCtrl.IEvent, 呼叫事件对象, 详细 3.3

【返回值】

无

3.2.25 StartScreenRecord

【功能】

开启屏幕共享

【预置条件】

呼叫成功，通话中

【函数】

```
Int StartScreenRecord
```

【返回值】

0:成功

其它：失败

3.2.26 StopScreenRecord

【功能】

关闭屏幕共享

【预置条件】

呼叫成功，通话中

【函数】

```
Int StopScreenRecord
```

【返回值】

0:成功

其它：失败

3.2.27 Update

【功能】

呼叫接通后，进行音视频切换

【预置条件】

呼叫成功，通话中

【函数】

```
Boolean Update(boolean bVideo)
```

【参数】

参数	说明
bVideo	Boolean true 表示目标是视频；false 表示目标是音频

【返回值】

true:成功

false: 失败,

原因: 1) 不在呼叫中;

2)、目标媒体类型和秒类型一样;

SipCall 中 mediaState 为 mediaAudioVedio (视频) 不能切换到视频;

SipCall 中 mediaState 为 mediaAudio (音频) 不能切换到音频

【后处理事件】

触发事件 3.3.2 onCallStateChanged;

CallState.Connected 下 mediaState 会发生变化

3.2.28 Share

【功能】

呼叫接通后, 本地视频源切换 (摄像头和手机屏幕)

【预置条件】

呼叫成功, 通话中

【函数】

```
Int SwitchVideoSource (String srcType, Intent data)
```

【参数】

参数	说明
srcType	String 表示视频源类型 “screen” 和 “camera”
data	Intent 屏幕共享的 MediaProjectionManager

【返回值】

true:成功

false: 失败,

【后处理事件】

触发事件 3.3.2 onCallStateChanged;

【说明】

本方法只对 MediaType 为 SipCoreEvent.MediaType.WebrtcVoice 的媒体有效;

3.3 事件 (SipPhoneCtrl.IEvent)

3.3.1 onRegistrationStateChanged

【功能】

电话注册事件

【预置条件】

- 1)、Initial 成功;
- 2)、调用 Register 消息;

【事件】

```
void onRegistrationStateChanged(final RegistrationState state, String message)
```

【参数】

参数	说明
state	RegistrationState 注册消息类型:
	RegistrationState.RegistrationNone 没有注册
	RegistrationState.RegistrationFailed 注册失败
	RegistrationState.RegistrationCleared 注销成功
	RegistrationState.RegistrationProgress 正在注册
RegistrationState.RegistrationOk 注册成功	
message	String, 状态对应的文本描述

【触发条件】

电话的注册状态发生变化时候触发;

3.3.2 onCallStateChanged

【功能】

电话呼叫事件

【预置条件】

- 1)、Initial 成功;
- 2)、调用 Register 消息;

【事件】

```
void onCallStateChanged(final SipCoreCall sipcall, final CallState state, String message)
```

【参数】

参数	说明
sipcall	SipCoreCall 呼叫信息, 参考 3.2.9 参数说明
state	CallState 呼叫状态:
	CallState.IncomingReceived 来电振铃
	CallState.OutgoingInit 去电振铃
	CallState.CallEnd 电话结束
	CallState.Error 电话异常
	CallState.Connected 电话条接通
CallState.StreamsRunning 语音通话	
message	String, 状态对应的文本描述

【触发条件】

电话呼叫状态发生变化时候触发;

3.3.3 onMessage

【功能】

文本消息事件

【预置条件】

1)、Initial 成功;

【事件】

```
void onMessage (int code, String message)
```

【参数】

参数	说明
code	Integer, code 100: 上传文件的结果 101: 对方推送过来特殊格式图片(application/xml) 102: 对方推送过来特殊格式视频(application/xml) 110: 对方推送的 Alert-info; 111: 对方推送过来特殊的 message(text/plain)
message	String, code 对应的表述

【触发条件】

有 image/msg 信息触发;

3.3.4 onInfo

【功能】

接收 SIP 的 Info 文本消息事件

【预置条件】

1)、Initial 成功;

【事件】

```
void onInfo(final String conType, final String message)
```

【参数】

参数	说明
conType	String, 接收文本格式, 例如: text/plain
message	String, 接收的文本消息

【触发条件】

有 info 信息触发;

3.3.5 onMediaStatics

【功能】

视频信息报告事件

【预置条件】

- 1)、Initial 成功;
- 2)、有视频呼叫事件;

【事件】

```
void onMediaStatics (VedioInfo vedio)
```

【参数】

参数	说明
vedio	VedioInfo, inFps int: coming framerate outFps int: going framerate inKbps int: coming bitrate outKbps int: going bitrate inWidth int: coming Width inHeight int: coming Height packetLoss int: coming packetLoss percent 100

【触发条件】

视频呼叫状态变化时候触发;

3.3.6 onMediaConsultation

【功能】

Agora 或者 xylink 呼叫发起协商事件

【预置条件】

- 1)、Initial 成功;
- 2)、有呼出操作;

【事件】

```
void onMediaConsultation (final String key, final String param)
```

【参数】

参数	说明
----	----

key	String 消息类型, “JoinAgoraAudio”: 加入 agora 或者 xylink 会议 “LeafAgoraAudio”: 退出 agora 或者 xylink 会议 “JoinAgoraScreen”: 加入屏幕共享 “LeafAgoraScreen”: 退出屏幕共享
param	String: Key 为: “JoinAgoraAudio”: param 为下面 Agora: agora-AgoraAppId-AgoraChannelId-AgoraUid-AgoraCertification-AgoraTokenExpires-AgoraChannelKey-AgoraDataEncrypt-AgoraToken AgoraChannelKey: 1 aes-128-xts 2 aes-128-ecb 3 aes-256-xts XYlink: xylink-appID-MeetingID Key 为: “LeafAgoraAudio”: param 为空 屏幕共享: Key 为: “JoinAgoraScreen”: param 同“JoinAgoraAudio” Key 为: “LeafAgoraScreen”: param 为空

【触发条件】

视频呼叫状态变化时候触发;

3.3.7 onDebugEvent

【功能】

底层消息报告, 用于调试

【预置条件】

- 1)、Initial 成功;
- 2)、设置属性 FileFlag 为 0

【事件】

```
void onMessage (String tagName, String message)
```

【参数】

参数	说明
tagName	String 消息类型
message	String 消息体

【触发条件】

当属性 TraceFlag 不为 0 同时 FileFlag 不为 1 时底层消息上报时触发

3.3.8 onInitialStatusChange

【功能】

底层通信连接的创建状态

【预置条件】

- 1、已经设置了电话的各种参数(主要是 Server, ServerPort, Domain 等参数);
- 2、调用了 Initial 方法

【事件】

```
void onInitialStatusChange(int code, String des)
```

【参数】

参数	说明
code	Int 状态码。200: 连接成功, 0: 表示关闭, 其他: 连接错误
des	String 连接信息描述

【触发条件】

使用 WebSocket 方式连接, 调用 Initial 方法

3.3.9 onFacePositionChanged

【功能】

WebRTC 人脸检测结果回调

【预置条件】

- 1、WebRTC 媒体通话;
- 2、enableFaceDetection 开启了人脸检测功能

【事件】

```
void onFacePositionChanged(int width, int height, int faceNum)
```

【参数】

参数	说明
width	人脸检测图像宽度
height	人脸检测图像高度
faceNum	检测的人脸个数

【触发条件】

enableFaceDetection 方法中设置第一个参数为 true

3.3.10 onScreenRecord

【功能】

获取屏幕共享协商结果获取

【预置条件】

呼叫成功，通话中

【事件】

```
void onScreenRecord(String key)
```

【参数】

参数	说明
key	屏幕共享的协商结果，枚举值： “begin”：屏幕共享开始； “end”：屏幕共享结束

【触发条件】

调用 StartScreenRecord 方法后接收回调

3.3.11 onCallStatistics

【功能】

视频信息报告事件

【预置条件】

- 1)、Initial 成功;
- 2)、有呼叫事件;

【事件】

```
void onCallStatistics (CallStatistics callstat)
```

【参数】

参数	说明
----	----

callstat	CallStatistics, fractionLost int: 丢失百分比(两次统计间隔间的丢包率) cumulativeLost int: 总的累积丢包 extendedMax int: 最大 extended jitterSamples int: 延时抖动 rttMs int: 发送者计算的发送来回时间, 大约 300ms 质量不好 bytesSent int: 发送字节数 packetsSent int: 发送媒体包数 bytesReceived int: 接收字节数 packetsReceived int: 接收媒体包数
----------	--

【触发条件】

媒体类型为 RTP, 设置了 CallStatics 为 true; 2s 一次

第4章 开发指南

4.1 开发步骤

4.1.1 AndroidManifest.xml 加入

```

<!-- SIPPhone-SDK needed-->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.CAMERA" />
    
```

```
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />

<service android:name="com.cincc.siphone.core.SipCoreService"
        android:label="CinSipCoreService"
        android:stopWithTask="false"/>
```

4.1.2 把 sdk 和支持库复制到 libs

```
cinsip-sdk.jar
armeabi\libcinsipcore.so
armeabi\libwebrtcdemo-jni.so
armeabi\libjingle_peerconnection_so.so
```

4.1.3 把资源文件复制到 res/raw 下

```
busy.ogg
end.ogg
oldphone_mono.ogg
overtimer.ogg
reqfailure.ogg
ringback.ogg
tap.ogg
toy_mono.ogg
```

4.1.4 非 SDK 接口调用设置 (API 28)

如果加密方式采用国密，需要在 application 的 attachBaseContext 方法中，开启非 sdk 接口调用权限

```
@Override
protected void attachBaseContext(Context base) {
    super.attachBaseContext(base);
    ReflectHelper.unseal(this);
}
```

4.1.5 混淆注意事项

```
-keep class com.cincc.** {*;}  
-keep class org.** {*;}
```

4.2 Demo

参考 demo

修改历史

版本 2.0.0.8

修订人 吴胜军 创建时间 2022-10-14

主要修订项目 1)、SendDtmf 增加 dtmfType 参数；

版本 2.0.0.7

修订人 吴胜军 创建时间 2022-10-11

主要修订项目 1)、增加 SetLogPath, 按照目录来保存 SDK 的日志；
2)、保存两天的日志, 上传压缩为 zip 再上传；

版本 2.0.0.6

修订人 吴胜军 创建时间 2022-07-22

主要修订项目 1)、BackServer、BackDomain、BackServerPort 属性；
2)、Initial 中如果不为 null, 会检测 udp 的网络链接
3)、增加主备用服务器的链接切换
2.X.X.X 支持 websocket 消息；

版本 1.0.0.5

修订人 吴胜军 创建时间 2022-07-06

主要修订项目 1)、增加 SetLogPath, 按照目录来保存 SDK 的日志；

版本 1.0.0.5

修订人	吴胜军	创建时间	2022-06-15
主要修订项目	1)、增加方法 Share; 2)、增加属性: LoadSingleMediaLib、CallStatics、EcEnable、NsEnable、AgcEnable、TimerType 3)、增加了事件: onCallStatistics		
版本	1.0.0.4		
修订人		创建时间	2022-06-08
主要修订项目	1)、增加方法 Update, 支持音视频切换		
版本	1.0.0.1		
创建人		创建时间	2016-01-19
主要修订项目	新建		