



# eSDK ICP V200R001C10 接口参考 01(CC, iOS)

文档版本 01  
发布日期 2017-03-17

华为技术有限公司



**版权所有 © 华为技术有限公司 2017。 保留一切权利。**

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI 和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <http://www.huawei.com>

客户服务邮箱： [support@huawei.com](mailto:support@huawei.com)

客户服务电话： 4008302118

---

# 目 录

---

<b>1 概述</b>	<b>1</b>
<b>2 数据类型</b>	<b>2</b>
2.1 通知	2
2.2 枚举	3
2.2.1 日志级别	3
2.2.2 视频角度	4
2.3 结构体	4
2.3.1 视频流信息	4
<b>3 初始化类接口</b>	<b>5</b>
3.1 获取版本信息	5
3.2 设置日志路径和日志级别	6
3.3 校验服务器证书	7
3.4 初始化 SDK	8
3.5 去初始化 SDK	8
3.6 设置接入网关地址	9
3.7 设置 SIP 服务器地址	10
3.8 设置数据加密模式	11
3.9 获取验证码	11
<b>4 登录类接口</b>	<b>13</b>
4.1 登录	13
4.2 登出	14
<b>5 呼叫类接口</b>	<b>15</b>
5.1 文字交谈	15
5.1.1 发起文字交谈	15
5.1.2 发送文字消息	16
5.1.3 释放文字呼叫	17
5.2 语音/视频呼叫	18
5.2.1 发起语音/视频呼叫	18
5.2.2 语音呼叫升级为视频呼叫	19

5.3 结束呼叫 .....	20
5.4 排队 .....	21
5.4.1 获取排队信息 .....	21
5.4.2 取消排队 .....	22
<b>6 设备类接口.....</b>	<b>23</b>
6.1 设置显示容器 .....	23
6.1.1 设置本地和远端视频显示容器.....	23
6.1.2 设置桌面共享显示容器 .....	24
6.2 媒体网络参数设置 .....	25
6.2.1 设置视频显示模式 .....	25
6.2.2 设置带宽 .....	26
6.3 设置视频旋转角度 .....	27
6.4 获取视频流信息 .....	27
6.5 前置/后置摄像头切换 .....	28
6.6 扬声器 .....	29
6.6.1 扬声器/听筒模式切换 .....	29
6.6.2 获取扬声器音量 .....	30
6.6.3 扬声器静音 .....	30
6.7 麦克风静音 .....	31
<b>7 错误码.....</b>	<b>33</b>
<b>8 修订记录.....</b>	<b>35</b>

# 1 概述

## 目的

本文主要介绍 eSDK CC iOS 接口参考。介绍内容包括相关数据类型、接口功能、方法定义、参数描述和使用示例等。

## 版本配套

与本文档相配套平台版本如下所示。

产品名称	版本
eSDK ICP CC	V200R001C10
TUP 库	V600R006C00B022

## 技术支持

- 开发过程中，您有任何问题可以在 [DevCenter](#) 系统中提单跟踪。
- 如果您在使用过程中有任何疑问，都可以通过以下方式联系我们。
  - 华为技术支持热线电话：400-8828-000。
  - 华为技术支持邮箱：esdk@huawei.com。

# 2 数据类型

- 2.1 通知
- 2.2 枚举
- 2.3 结构体

## 2.1 通知

事件通知名称	返回值	说明
AUTH_MSG_ON_LOGIN	通过 object 获取	登录结果通知，返回值为 @” 0” 表示登录成功，其它表示失败
AUTH_MSG_ON_LOGOUT	通过 object 获取	登出结果通知，返回值为 @” 0” 表示登出成功，其它表示失败
CALL_MSG_ON_QUEUEING	—	呼叫排队通知
CALL_MSG_ON_CANCEL_QUEUE	—	取消排队通知
CALL_MSG_ON_QUEUE_INFO	NSDictionary，通过 userInfo 获取	获取排队信息通知， position - 在队伍中的位置 totalWaitTime - 累计排队时长
CALL_MSG_ON_QUEUE_TIMEOUT	—	呼叫排队超时通知
CALL_MSG_ON_CONNECTED	通过 object 获取	呼叫成功通知，TP 环境返回值为空，MS 环境，返回值为空表示发起文字聊天成功，AUDIO_CALL 表示语音呼叫成功，

事件通知名称	返回值	说明
		VIDEO_CALL 表示会议成功
CALL_MSG_ON_DISCONNECTED	通过 object 获取	呼叫结束通知，TP 环境返回值为空，MS 环境，返回值为空表示文字聊天结束，AUDIO_CALL 表示语音通话结束
CALL_MSG_ON_FAIL	通过 object 获取	呼叫失败通知，返回值为失败错误码
CHAT_MSG_ON_SUCCESS	—	发送文字聊天消息成功通知
CHAT_MSG_ON_FAIL	通过 object 获取	发送文字聊天消息失败通知，返回值为失败错误码
CHAT_MSG_ON_RECEIVE	通过 object 获取	收到聊天消息通知，返回值为坐席发送的聊天内容。
CALL_MSG_ON_USER_LEAVE	—	用户离开会议通知
CALL_MSG_ON_SCREEN_DATA_RECEIVE	—	收到屏幕共享通知
CALL_MSG_ON_SCREEN_SHARE_STOP	—	屏幕共享停止通知
CALL_MSG_ON_NET_QUALITY_LEVEL	通过 object 获取	网络质量变化通知，返回值表示网络质量等级
CALL_GET_VERIFY_CODE	NSDictionary，通过 userInfo 获取。	获取验证码。关键字如下： verifyCode:base64 编码的验证码图片数据。

2.2 枚举

2.2.1 日志级别

类型定义

```
//日志级别
typedef NS_ENUM(NSInteger, CCLogLevel)
{
```

```
LOG_ERROR    = 0, //错误级别
LOG_WARNING  = 1, //警告级别
LOG_INFO     = 2, //一般级别
LOG_DEBUG    = 3, //调试级别
LOG_NONE     = 9      //无日志
};
```

## 2.2.2 视频角度

### 类型定义

```
typedef NS_ENUM(NSInteger, VIDEO_ROTATE)
{
    ROTATE_DEFAULT = 0, //不旋转
    ROTATE_90 = 90, //逆时针旋转 90°
    ROTATE_180 = 180, //逆时针旋转 180°
    ROTATE_270 = 270 //逆时针旋转 270°
};
```

## 2.3 结构体

### 2.3.1 视频流信息

#### 类型定义

```
typedef struct StreamParam
{
    float sendLossFraction;      //TP: 发送方丢包率(%)
    float sendDelay;            //TP: 发送方平均时延(ms)
    float receiveLossFraction;   //TP: 接收方丢包率(%)
    float receiveDelay;         //TP: 接收方平均时延(ms)
    char encodeSize[32];        //TP: 图像分辨率(发)
    char decodeSize[32];        //TP: 图像分辨率(收)
    int  videoWidth;            //TP: 视频分辨率-宽;MS:x 分辨率-宽
    int  videoHeight;          //TP: 视频分辨率-高;MS:y 分辨率-高
    int  frameRate;             //MS:帧率
    int  bitRate;               //MS:码流
}Stream_INFO;
```



# 3 初始化类接口

- 3.1 获取版本信息
- 3.2 设置日志路径和日志级别
- 3.3 校验服务器证书
- 3.4 初始化 SDK
- 3.5 去初始化 SDK
- 3.6 设置接入网关地址
- 3.7 设置 SIP 服务器地址
- 3.8 设置数据加密模式
- 3.9 获取验证码

## 3.1 获取版本信息

### 接口描述

获取当前 SDK 版本信息。

### 方法定义

```
//objc code
- (NSString *)getVersion;
```

### 使用说明

无

### 参数描述

无

## 返回值

类型	含义
NSString	版本信息字符串

## 使用示例

```
//objc code
[[CCSDK sharedInstance] getVersion];
```

## 3.2 设置日志路径和日志级别

### 接口描述

设置 SDK 的日志路径以及日志级别，便于定位出现的问题。

### 方法定义

```
//objc code
- (BOOL)setLogPath:(NSString *)path level:(CCLogLevel)level;
```

### 使用说明

在调用初始化接口之前调用。

### 参数描述

参数名	类型	描述
path	NSString	日志路径
level	枚举	级别，参考 2.2.1 日志级别

## 返回值

类型	含义
BOOL	YES 表示成功，NO 表示失败

## 使用示例

```
//objc code
NSString *path = [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomainMask, YES) objectAtIndex:0];
NSString *logPath = [path stringByAppendingPathComponent:@"TUP LOG"];
[[CCSDK sharedInstance] setLogPath:logPath level:CCLogLevelInfo];
```

## 3.3 校验服务器证书

### 接口描述

对服务器进行证书校验。

### 使用说明

在网络请求中，开启校验服务器证书，如果证书校验失败，会导致访问服务器失败。如果需要证书验证功能，需要将证书放在工程中 assets 文件夹下，例如“assets/certs/server.cer”。

### 方法定义

```
//objc code
- (void)setNeedValidate:(BOOL)needValidate
needValidateDomain:(BOOL)needValidateDomain certificateData:(NSData
*)certificateData
```

### 参数描述

参数名	类型	描述
needValidate	BOOL	是否验证证书
needValidateDomain	BOOL	是否验证域名
certInputStream	NSData*	证书

### 返回值

无。

### 使用示例

```
//objc code
[[CCSDK sharedInstance] setHostAddress:self.ipText.text
port:self.portText.text
transSecurity: isHTTPS
sipServerType:serverType];
if ( isHTTPS)
{
```

```
NSString * cerPath = [[NSBundle mainBundle] pathForResource:@"server"
ofType:@"der"];
NSData *dataCA = [NSData dataWithContentsOfFile:cerPath];
[[CCSDK sharedInstance] setNeedValidate:YES needValidateDomain:NO
certificateData:dataCA];
}
```

## 3.4 初始化 SDK

### 接口描述

对用 TUP 以及会议组件进行初始化。

### 方法定义

```
//objc code
- (void)initSDK;
```

### 使用说明

当应用启动时调用此接口，该接口与 3.5 去初始化 SDK 接口配对使用。

### 参数描述

无

### 返回值

无

### 使用示例

```
//objc code
[[CCSDK sharedInstance] initSDK];
```

## 3.5 去初始化 SDK

### 接口描述

释放 SDK 资源。

### 方法定义

```
//objc code
- (void)unInitSDK;
```

## 使用说明

该接口与 3.4 初始化 SDK 接口配对使用。

## 参数描述

无

## 返回值

无

## 使用示例

```
//objc code
[[CCSDK sharedInstance] unInitSDK];
```

# 3.6 设置接入网关地址

## 接口描述

设置登录的网关地址。

## 方法定义

```
//objc code
- (NSInteger)setHostAddress:(NSString *)ip port:(NSString *)port
transSecurity:(BOOL)transSec sipServerType:(int)serverType;
```

## 使用说明

需要在调用登陆接口之前调用，不然无法获取服务器地址。

## 参数描述

参数名	类型	描述
ip	NSString	服务器地址
port	NSString	服务器端口，必须为数字
transSec	BOOL	YES 表示 HTTPS 请求， NO 表示 HTTP 请求。
serverType	int	SERVER_TYPE_TP 表示 TP 环境， SERVER_TYPE_MS 表示 MS 环境

## 返回值

类型	含义
NSInteger	0 表示接口调用成功，其它表示失败，详情见 7 错误码

## 使用示例

```
//objc code
NSInteger setRes = [[CCSDK sharedInstance] setHostAddress:self.ipText.text
port:self.portText.text transSecurity:NO];
if (setRes != RET_OK){
[self showMessage:[NSString stringWithFormat:@"设置错误:%ld", (long)setRes]];
}
```

## 3.7 设置 SIP 服务器地址

### 接口描述

设置呼叫 SIP 服务器地址。

### 方法定义

```
//objc code
- (NSInteger)setSIPServerAddress:(NSString *)ip port:(NSString *)port ;
```

### 使用说明

TP 的环境不调用默认使用从服务器获取的地址进行呼叫，MS 目前无法从服务器获取呼叫地址信息，需要根据环境配置好的 SIP 地址在登录后来手动设置。

### 参数描述

参数名	类型	描述
ip	NSString	服务器地址
port	NSString	服务器端口，必须为数字

## 返回值

类型	含义
NSInteger	0 表示接口调用成功，其它表示失败，详情见 7 错误码。

使用示例

```
//objc code
[[CCSDK shareInstance] setSIPServerAddress:@"10.174.5.54" port:@"5060"];
```

3.8 设置数据加密模式

接口描述

设置数据传输的加密模式，支持 TLS 以及 SRTP 加密。

方法定义

```
//objc code
- (void)setTransportSecurityUseTLS:(BOOL)enableTLS useSRTP:(BOOL)enableSRTP;
```

使用说明

无论是 TP 还 MS 都需要根据环境信息判断是否需要加密处理。

参数描述

参数名	类型	描述
enableTLS	BOOL	YES 表示进行 TLS 加密，NO 表示不加密
enableSRTP	BOOL	YES 表示进行 SRTP 加密，NO 表示不加密

返回值

无

使用示例

```
//objc code
[[CCSDK shareInstance] setTransportSecurityUseTLS:YES useSRTP:YES];
```

3.9 获取验证码

接口描述

获取呼叫验证码。

## 方法定义

```
//objc code
- (void)getVerifyCode;
```

## 使用说明

在呼叫前获取，通过接收通知 2.1 通知获取返回的验证码，返回验证码信息为 base64 编码的图片数据，需要将数据显示成图片，才能看到验证码。在呼叫时将看到的验证码传入即可。服务器可以关闭验证码功能，在 IcsGateway 服务器中打开 home/prometheus/tomcat7/webapps/icsgateway/WEB-INF/config/verifycode.properties 这个文件，修改 VERIFYCODE\_ISUSERFORCALL = false ，然后重启 IcsGateway 服务器即可当。验证码功能关闭后，该接口不用调。

## 参数描述

无。

## 返回值

无。

## 使用示例

```
//objc code
//注册获取验证码通知
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(showVerifyCode:)
name:CALL_GET_VERIFY_CODE object:nil];

//调用获取验证码接口
[[CCSDK sharedInstance] getVerifyCode];
//处理获取验证码通知
- (void)showVerifyCode:(NSNotification *)notify
{
    if ([notify.object intValue] == 0) {
        NSString *encodedImageStr = [notify.userInfo objectForKey:@"verifyCode"];
        NSData *decodedImageData = [[NSData alloc]
initWithBase64EncodedString:encodedImageStr options:0];

        UIImage *decodedImage= [UIImage imageWithData:decodedImageData];
        dispatch_async(dispatch_get_main_queue(), ^{
            //抛回主线程进行显示
        });
    }
    else{
        dispatch_async(dispatch_get_main_queue(), ^{
            //抛回主线程进行提示，获取失败
        });
    }
}
```



# 4 登录类接口

- 4.1 登录
- 4.2 登出

## 4.1 登录

### 接口描述

用户登录软终端。

### 方法定义

```
//objc code
- (NSInteger)login:(NSString *)vndid userName:(NSString *)userName;
```

### 使用说明

返回值只能说明接口是否调用成功，登录之前注册通知，是否成功登录要根据通知 2.1 通知判断。

### 参数描述

参数名	类型	描述
vndid	NSString	虚拟呼叫中心 id， 1-999
userName	NSString	用户名 1-20 位数字、字母或其组合

### 返回值

类型	含义
----	----

类型	含义
NSInteger	0 表示接口调用成功，其它表示接口调用失败参考 7 错误码。

使用示例

```
//objc code
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(loginResult:)
name:AUTH MSG ON LOGIN object:nil];

NSInteger ret = [[CCSDK sharedInstance] login:@"1" userName:self.userNameText.text];
if (ret == RET_OK) {
}else{
[self showMessage:[NSString stringWithFormat:@"登录接口调用错误:%ld", (long)ret]];  };
```

4.2 登出

接口描述

用户登出软终端。

方法定义

```
//objc code
- (void)logout;
```

使用说明

登出之前注册通知，是否登出成功，根据通知 2.1 通知判断。

返回值

无

使用示例

```
//objc code
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(logoutResult:)
name:AUTH MSG ON LOGOUT object:nil];

[[CCSDK sharedInstance] logout];
```

# 5 呼叫类接口

- 5.1 文字交谈
- 5.2 语音/视频呼叫
- 5.3 结束呼叫
- 5.4 排队

## 5.1 文字交谈

### 5.1.1 发起文字交谈

#### 接口描述

用户与坐席可以进行文字交谈，该功能目前只支持 MS 环境。

#### 方法定义

```
//objc code
- (NSInteger)webChatCall:(NSString *)accessCode callData:(NSString *)callData
verifyCode:(NSString *)verifyCode;
```

#### 使用说明

返回值只能说明接口是否调用成功，发起文字交谈之前注册成功与失败通知，成功与坐席建立文字交谈收到 CALL\_MSG\_ON\_CONNECTED 通知，失败会收到 CALL\_MSG\_ON\_FAIL 通知，具体请见 2.1 通知。若服务器开启验证码功能，需先调用获取验证码接口，再调用此接口；若服务器验证码功能关闭，则直接调用本接口，且验证码参数可不传。

#### 参数描述

参数名	类型	描述
accessCode	NSString	接入码,平台配置好的数

参数名	类型	描述
		据,1-24 位数字组成
callData	NSString	呼叫随路数据,长度范围 [0,1024]
verifyCode	NSString*	传入接口 getVerifyCode 返回的验证码

返回值

类型	含义
NSInteger	0 表示接口调用成功，其它表示接口调用失败，参考 7 错误码。

使用示例

```
//objc code
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(callSuccess:)
name:CALL MSG ON CONNECTED
object:nil];
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(callFail:)
name:CALL MSG ON FAIL object:nil];

[[CCSDK sharedInstance] webChatCall:@"60011" callData:self.CallData
verifyCode:@"1234"];
```

5.1.2 发送文字消息

接口描述

收发文字聊天消息，该功能目前只支持 MS 环境。

方法定义

```
//objc code
- (NSInteger)sendMsg:(NSString *)message;
```

使用说明

返回值只能说明接口是否调用成功，发送文字消息之前注册成功、失败以及收到消息通知，发送成功收到 CHAT\_MSG\_ON\_SUCCESS 通知，发送失败会收到 CHAT\_MSG\_ON\_FAIL 通知，收到消息会有 CHAT\_MSG\_ON\_RECEIVE 通知，详情见 2.1 通知。

参数描述

参数名	类型	描述
message	NSString	消息内容，最小为 1 个字符，最大为 300

返回值

类型	含义
NSInteger	0 表示接口调用成功，其它表示接口调用失败，参考 7 错误码。

使用示例

```
//objc code
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(sendMsgSuccess)
name:CHAT_MSG_ON_SUCCESS object:nil];
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(sendMsgFail:)
name:CHAT_MSG_ON_FAIL object:nil];
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(receiveMsg:)
name:CHAT_MSG_ON_RECEIVE object:nil];
[[CCSDK sharedInstance] sendMsg:self.chatText.text]
```

5.1.3 释放文字呼叫

接口描述

结束当前的文字聊天，该功能目前只支持 MS 环境。

方法定义

```
//objc code
- (void)releaseWebChatCall;
```

使用说明

使用文字聊天功能前注册通知，文字聊天结束收到 2.1 通知通知。

参数描述

无

## 返回值

无

## 使用示例

```
//objc code
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(callEnd:)
name:CALL_MSG_ON_DISCONNECTED object:nil];

[[CCSDK sharedInstance] releaseWebChatCall];
```

## 5.2 语音/视频呼叫

如果出现语音连上就立即断开的情况，请到路由器的配置中，在“WAN 设置”中把 SIP ALG 开启。

### 5.2.1 发起语音/视频呼叫

#### 接口描述

与坐席建立语音/视频通话。

#### 方法定义

```
//objc code
- (NSInteger)makeCall:(NSString *)accessCode callType:(NSString *)callType
callData:(NSString *)callData verifyCode:(NSString *)verifyCode;
```

#### 使用说明

返回值只能说明接口是否调用成功，呼叫之前注册成功与失败通知，呼叫成功收到 CALL\_MSG\_ON\_CONNECTED 通知，呼叫失败会收到 CALL\_MSG\_ON\_FAIL 通知，详情见 2.1 通知。若服务器开启验证码功能，需先调用获取验证码接口，再调用此接口；若服务器验证码功能关闭，则直接调用本接口，且验证码参数可不传。在创建呼叫前必须先建立文字链接。

#### 参数描述

参数名	类型	描述
accessCode	NSString	接入码,平台配置好的数据,1-24 位数字组成
callType	NSString	呼叫类型,AUDIO_CALL 表示语音呼叫,VIDEO_CALL 表示视频呼叫

参数名	类型	描述
callData	NSString	呼叫随路数据,长度范围 [0,1024]
verifyCode	NSString*	传入接口 getVerifyCode 返回的验证码

返回值

类型	含义
NSInteger	0 表示接口调用成功，其它表示接口调用失败，具体参照 7 错误码。

使用示例

```
//objc code
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(callSuccess:)
name:CALL MSG ON CONNECTED
object:nil];
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(callFail:)
name:CALL MSG ON FAIL object:nil];
[[CCSDK sharedInstance] makeCall:self.aCode callType: AUDIO CALL
callData:self.CallData verifyCode:@"5678"];
[[CCSDK sharedInstance] makeCall:self.aCode callType: VIDEO CALL
callData:self.CallData verifyCode:@"5678"];
```

5.2.2 语音呼叫升级为视频呼叫

接口描述

将语音通话升级为视频通话，目前只支持 MS 环境。

方法定义

```
//objc code
- (NSInteger)updateToVideo;
```

使用说明

在语音通话存在的情况下调用，返回值只能说明接口是否调用成功，升级之前注册成功与失败通知，升级成功收到 CALL\_MSG\_ON\_CONNECTED 通知，升级失败会收到 CALL\_MSG\_ON\_FAIL 通知，详情见 2.1 通知。

参数描述

无

返回值

类型	含义
NSInteger	0 表示接口调用成功，其它表示接口调用失败，参考 7 错误码。

使用示例

```
//objc code
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(callSuccess:)
name:CALL_MSG_ON_CONNECTED
object:nil];
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(callFail:)
name:CALL_MSG_ON_FAIL object:nil];
[[CCSDK sharedInstance] updateToVideo];
```

5.3 结束呼叫

接口描述

结束当前的通话。

方法定义

```
//objc code
- (void)releaseCall;
```

使用说明

注册通知，TP 环境下通话结束收到 CALL\_MSG\_ON\_DISCONNECTED 通知；MS 环境下通话结束收到 CALL\_MSG\_ON\_DISCONNECTED 通知，通话离开会议时调用接口即可离开会议，详情见 2.1 通知。

参数描述

无

返回值

无



## 使用示例

```
//objc code
//TP 环境
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(callEnd:)
name:CALL_MSG_ON_DISCONNECTED object:nil];

[[CCSDK sharedInstance] releaseCall];
//MS 环境语音呼叫
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(callEnd:)
name:CALL_MSG_ON_DISCONNECTED object:nil];

[[CCSDK sharedInstance] releaseCall];

//MS 环境视频呼叫
[[CCSDK sharedInstance] releaseCall];
```

## 5.4 排队

### 5.4.1 获取排队信息

#### 接口描述

呼叫排队状态下，获取排队信息。

#### 方法定义

```
//objc code
- (void)getCallQueueInfo;
```

#### 使用说明

注册通知，获取排队信息收到 CALL\_MSG\_ON\_QUEUE\_INFO2.1 通知。

#### 参数描述

无

#### 返回值

无

#### 使用示例

```
//objc code
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(receiveQueueInfo:)
```

```
name:CALL_MSG_ON_QUEUE_INFO object:nil];  
[[CCSDK sharedInstance] getCallQueueInfo];
```

## 5.4.2 取消排队

### 接口描述

呼叫排队状态下，取消排队，结束呼叫。

### 方法定义

```
//objc code  
- (void)cancelQueue;
```

### 使用说明

注册通知，取消排队收到 CALL\_MSG\_ON\_CANCEL\_QUEUE 通知，排队超时收到 CALL\_MSG\_ON\_QUEUE\_TIMEOUT 通知，详情见 2.1 通知。

### 参数描述

无

### 返回值

无

### 使用示例

```
//objc code  
[[NSNotificationCenter defaultCenter] addObserver:self  
selector:@selector(queueIsCancel:)  
name:CALL_MSG_ON_CANCEL_QUEUE object:nil];  
[[NSNotificationCenter defaultCenter] addObserver:self  
selector:@selector(callQueueTimeOut:)  
name:CALL_MSG_ON_QUEUE_TIMEOUT object:nil];  
[[CCSDK sharedInstance] cancelQueue
```

# 6 设备类接口

- 6.1 设置显示容器
- 6.2 媒体网络参数设置
- 6.3 设置视频旋转角度
- 6.4 获取视频流信息
- 6.5 前置/后置摄像头切换
- 6.6 扬声器
- 6.7 麦克风静音

## 6.1 设置显示容器

### 6.1.1 设置本地和远端视频显示容器

#### 接口描述

设置本地、远端视频窗口句柄。

#### 方法定义

```
//objc code
- (void)setVideoContainer:(id)localView remoteView:(id)remoteView;
```

#### 使用说明

必须在视频通话建立之前调用，以保证视频窗口正常显示。视频显示会根据重力感应进行适配，若想仅用横屏，在 app 工程里设置 device orientation 里为 landscape 即可，若是竖屏，设置为 portrait 即可。

#### 参数描述

参数名	类型	描述
-----	----	----

参数名	类型	描述
localView	id	本地视频容器，参考EAGLView以及子类
remoteView	id	远端视频容器，参考EAGLView以及子类

返回值

无

使用示例

```
//objc code
self.remoteView = [EAGLView getRemoteVideoView];
self.localView = [EAGLView getLocalVideoView];
[[CCSDK sharedInstance] setVideoContainer:self.localView remoteView:self.remoteView];
```

6.1.2 设置桌面共享显示容器

接口描述

设置桌面共享窗口句柄，目前只支持MS环境。

方法定义

```
//objc code
- (void)setDesktopShareContainer:(UIImageView *)shareView;
```

使用说明

目前只支持MS平台，必须在对方开启桌面共享之前调用，以保证共享窗口正常显示。

参数描述

参数名	类型	描述
shareView	UIImageView	共享显示容器

返回值

无

## 使用示例

```
//objc code
CGRect sFrame = self.remoteView.frame;
self.ScreenshareView = [[UIImageView alloc] initWithFrame:sFrame];
[[CCSDK sharedInstance] setDesktopShareContainer:self.ScreenshareView];
```

## 6.2 媒体网络参数设置

### 6.2.1 设置视频显示模式

#### 接口描述

设置视频画质优先还是流畅优先。

#### 方法定义

```
//objc code
- (BOOL)setVideoMode:(int)videoMode;
```

#### 使用说明

TP 环境默认为画质优先，MS 环境默认为流畅优先。如果需要改变默认值，TP 环境下在视频通话之前调用，MS 环境下在会议过程中调用。

#### 参数描述

参数名	类型	描述
videoMode	int	0 表示图像质量优先，1 表示流畅度优先

#### 返回值

类型	含义
BOOL	YES 表示设置成功，NO 表示设置失败。

## 使用示例

```
//objc code
[[CCSDK sharedInstance] setVideoMode:0];
```

## 6.2.2 设置带宽

### 接口描述

设置媒体网络的带宽，保证画面正常显示。

### 方法定义

```
//objc code
- (BOOL)setDataRate:(int)dataRateValue;
```

### 使用说明

默认值 512K。如果需要改变默认值，TP 环境下在视频通话之前调用，MS 环境下在会议过程中调用。

### 参数描述

参数名	类型	描述
dataRateValue	int	TP 环境：带宽值范围[1-768]。 1<=带宽值<=128 取 128K 128<带宽值<=256 取 256K 256<带宽值<=384 取 384K 384<带宽值<=512 取 512K 512<带宽值<=768 取 768K 数值越高，画面越清晰。 MS 环境：带宽值范围[1-768]。 带宽值小于 512K 表示画质流畅，大于 512K，表示画质越清晰

### 返回值

类型	含义
BOOL	YES 表示设置成功，NO 表示设置失败。

使用示例

```
//objc code
[[CCSDK sharedInstance] setDataRate:768];
```

6.3 设置视频旋转角度

接口描述

用于视频通话过程中手动调整视频角度。

方法定义

```
//objc code
- (BOOL)setVideoRotate:(VIDEO_ROTATE) rotate;
```

使用说明

视频通话过程中可能因横竖屏不同或者切换前后置摄像头的时候出现视频角度倾斜问题，调用该接口进行调整。

参数描述

参数名	类型	描述
rotate	2.2.2 视频角度	可选值：0、90、180、270。

返回值

类型	含义
BOOL	YES 表示成功，NO 表示失败

使用示例

```
//objc code
[[CCSDK sharedInstance] setVideoRotate:180];
```

6.4 获取视频流信息

接口描述

视频通话过程中查看视频分辨率、网络延迟、丢包率等。

方法定义

```
//objc code
- (Stream_INFO) getChannelInfo;
```

使用说明

在视频成功后，能看到双方画面的情况下，才能查看视频信息。MS 环境与 TP 环境下获得的数据有差异，具体详见 2.3.1 视频流信息

参数描述

无

返回值

类型	含义
2.3.1 视频流信息	视频流信息

使用示例

```
//objc code
- (void) getVideoStreamInfo{
    Stream INFO info = [[CCSDK sharedInstance] getChannelInfo];
    self.videoInfoView.text = [NSString stringWithFormat:@"发送方丢包率:%f\n 发送方平均延时:%fms\n 接收方丢包率:%f\n 接收方平均延时:%fms\n 发送分辨率:%s\n 接收分辨率:%s",info.sendLossFraction,info.sendDelay,info.receiveLossFraction,info.receiveDelay,info.encodeSize,info.decodeSize];
}
```

6.5 前置/后置摄像头切换

接口描述

视频通话中前置/后置摄像头切换。

方法定义

```
//objc code
- (BOOL) switchCamera:(int) index;
```

使用说明

默认为前置摄像头



### 参数描述

参数名	类型	描述
index	int	1-前置摄像头,0-后置摄像头

### 返回值

类型	含义
BOOL	YES 表示切换成功，NO 表示切换失败

### 使用示例

```
//objc code
[[CCSDK sharedInstance] switchCamera:0];
```

## 6.6 扬声器

### 6.6.1 扬声器/听筒模式切换

#### 接口描述

语音通话过程中，扬声器和听筒模式切换。

#### 方法定义

```
//objc code
- (BOOL)changeAudioRoute:(int) route;
```

#### 使用说明

默认为听筒模式。

#### 参数描述

参数名	类型	描述
route	int	0-听筒模式，1-扬声器。

## 返回值

类型	含义
BOOL	YES 表示切换成功，NO 表示切换失败。

## 使用示例

```
//objc code
[[CCSDK sharedInstance] changeAudioRoute:1];
```

## 6.6.2 获取扬声器音量

### 接口描述

用于获取扬声器音量的大小。

### 方法定义

```
//objc code
- (NSInteger)getSpeakerVolume;
```

### 使用说明

无

### 参数描述

无

## 返回值

类型	含义
NSInteger	扬声器音量的大小值

## 使用示例

```
//objc code
NSInteger volume = [[CCSDK sharedInstance] getSpeakerVolume];
NSLog(@"speak volume:%ld", (long) volume);
```

## 6.6.3 扬声器静音

### 接口描述

用于设置扬声器静音或者取消静音。

## 方法定义

```
- (BOOL) setSpeakerMute: (BOOL) isMute;
```

## 使用说明

在有语音通话的过程中使用。

## 参数描述

参数名	类型	描述
isMute	BOOL	YES-扬声器静音，NO-取消静音。

## 返回值

类型	含义
BOOL	YES-设置成功，NO 表示失败。

## 使用示例

```
//objc code  
[[CCSDK sharedInstance] setSpeakerMute:NO];
```

## 6.7 麦克风静音

### 接口描述

语音通话过程中，关闭/开启麦克风。

### 方法定义

```
//objc code  
- (BOOL) setMicMute: (BOOL) isMute;
```

### 使用说明

无。

### 参数描述

参数名	类型	描述
isMute	BOOL	YES-麦克风静音，NO-取消静音。

参数名	类型	描述
		消麦克风静音。

返回值

类型	含义
BOOL	YES 表示成功，NO 表示失败。

使用示例

```
//objc code
[[CCSDK sharedInstance] setMicMute:YES];
```

# 7 错误码

错误码	说明
0	成功。
-1	参数错误。
-2	文字交谈未建立。
-3	语音未连接。
-4	无人应答。
-5	网络发生错误。
-6	创建会议失败。
10-100-002	服务端 platform.properties 中的 EVENT_METHOD 配置错误。
10-100-003	没有连接到 CCS，Was 服务不可用。
10-100-004	WebmAnyService 服务不可用。
10-100-005	无权限调用接口。
10-100-006	用户未登陆，网络长时间断连后，后续的请求都会失败，返回该错误码。
10-100-007	用户请求参数为空或不合法。
10-100-008	用户已登录。
10-100-009	资源不可用。
10-100-010	方法不支持。
10-100-011	状态错误。
10-100-012	用户的 WebmServer 为空。
10-100-013	vdnId 不存在。

错误码	说明
10-100-014	接入码不存在。
10-100-015	已达到最大用户登录数。
10-100-016	配置代理为空。
10-100-017	配置代理发送异常。
10-200-001	接入码对应的呼叫已存在。
10-200-002	超过用户最大呼叫数。
10-200-003	呼叫不存在。
10-200-004	编码转换失败。
10-200-006	呼叫不是排队态。
10-200-007	呼叫未建立。
10-300-001	正在会议中或申请会议中。
10-300-002	会议不存在。

# 8 修订记录

发布日期	文档版本	修订说明
2017-3-15	V2.1.10	文档 V200R001C10 版本发布。 全量适配 MS&TP 功能。
2017-3-8	1.1.T1	文档 1.1.T1 版本发布。 新增： <ul style="list-style-type: none"><li>5.1 文字交谈</li><li>5.2.2 语音呼叫升级为视频呼叫</li><li>6.1.2 设置桌面共享显示容器</li></ul> 修改： 2.1 通知（添加 MS NAT 功能）
2016-12-31	V2.1.00	第一次正式发布。