

# **ZJSDK 使用说明文档**

## 目录

<b>1. 编写目的 .....</b>	<b>4</b>
<b>2. SDK 集成说明 .....</b>	<b>4</b>
2.1 导入 SDK 文件 .....	4
2.2 重新导入库文件 .....	4
2.3 添加头文件路径 .....	5
2.4 添加 OTHER LINKER FLAGS .....	5
2.5 关闭 BITCODE .....	5
2.6 修改 INFO.PLIST .....	6
<b>3. 加入会议/点对点通讯 .....</b>	<b>6</b>
3.1 配置接口地址 .....	6
3.2 加入会议参数 .....	7
3.3 通话质量参数 .....	7
3.4 加入会议室（使用默认通话质量参数直接入会） .....	8
3.5 加入会议室（可设置通话质量参数） .....	8
3.6 加入会议室（DEPRECATED!） .....	9
3.7 获得远端视频和本端视频的视图 .....	11
<b>4. 点对点被呼叫 .....</b>	<b>11</b>
4.1 配置 MCU SERVER .....	12
4.2 配置 VOIP TOKEN .....	13
4.3 配置 CARTIFICATE NAME .....	13
4.4 注册 .....	13
4.5 接听呼叫/接听入会参数 .....	14
4.6 邀请参会者 .....	14
4.7 拒绝接听.....	14
<b>5. 会中管理 API .....</b>	<b>14</b>

- 5.1 切换静音 / 非静音状态 ..... 15
- 5.2 切换静画 / 非静画状态（只关闭回显视频） ..... 15
- 5.3 切换静画 / 非静画状态（关闭回显、发送视频） ..... 15
- 5.4 切换摄像头 ..... 16
- 5.5 退出当前会议 ..... 16
- 5.6 结束会议（断开当前所用与会者） ..... 17
- 6. 监听返回状态 ..... 17**
  - 6.1 获得丢包率 ..... 17
  - 6.2 获得挂断状态 ..... 18
  - 6.3 获得通话状态 ..... 18
- 7. 版本记录 ..... 18**

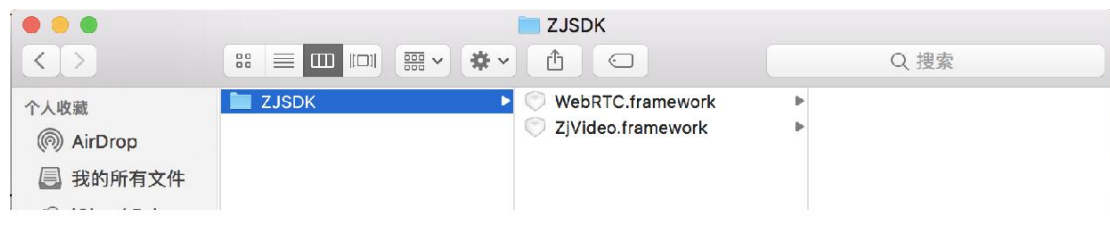
## 1. 编写目的

快速集成使用 ZJSDK 的说明

## 2. SDK 集成说明

### 2.1 导入 SDK 文件

导入 WebRTC.framework、ZjVideo.framework 到工程中。



Destination: ☒ Copy items if needed

Added folders: ☒ Create groups  
☐ Create folder references

### 2.2 重新导入库文件

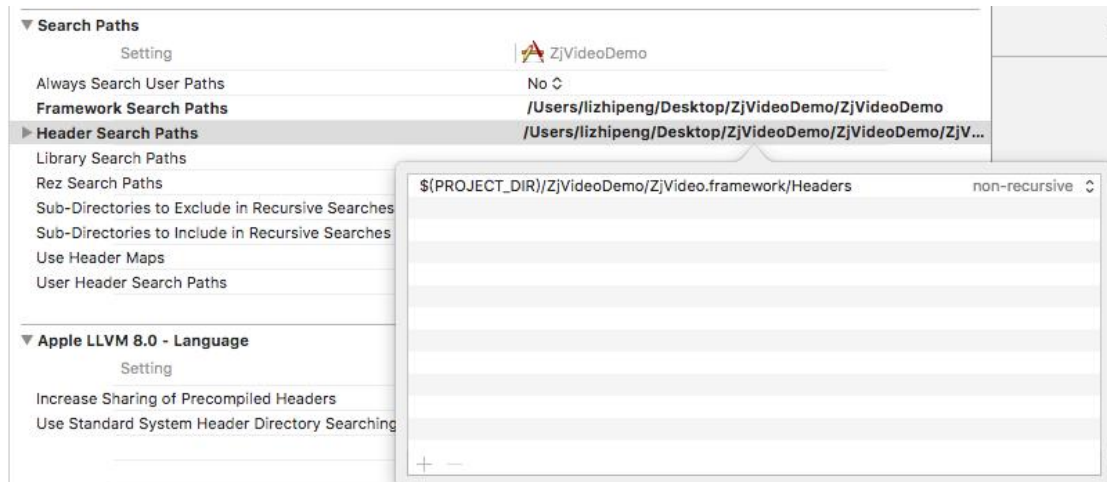
确保 Embedded Binaries 和 Linked Frameworks and Libraries 中添加 WebRTC.framework 和 ZjVideo.framework。



## 2.3 添加头文件路径

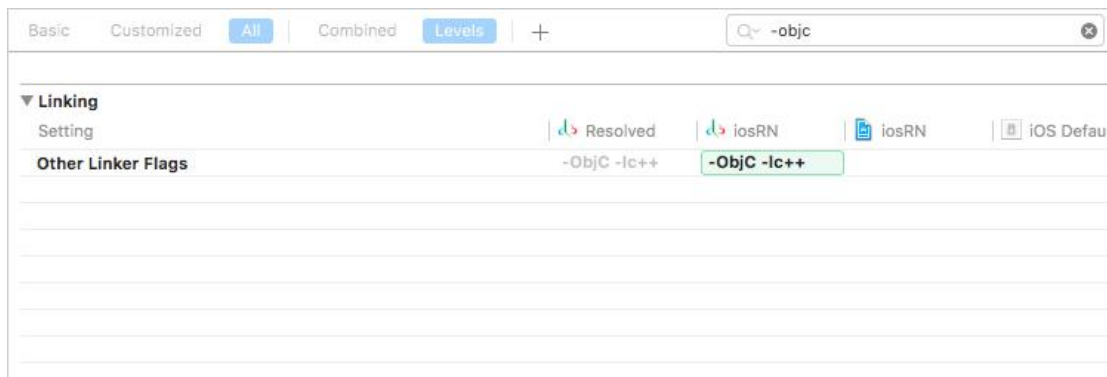
向工程中添加头文件

/path to ZjVideo.framework/Headers



## 2.4 添加 Other Linker Flags

-Objc -lc++



## 2.5 关闭 bitcode



## 2.6 修改 info.plist

添加麦克风、相机权限。

添加 ATS

Key	Type	Value
▼ Information Property List	Dictionary	(17 items)
Privacy - Microphone Usage Desc...	String	使用麦克风
Privacy - Camera Usage Description	String	使用相机
▼ App Transport Security Settings	Dictionary	(1 item)

## 3. 加入会议/点对点通讯

建立音视频通讯需要收集部分参数。

第一步：公有云项目使用默认 Api Server domain（接口地址），不需要设置 Api Server domain；私有云项目需要设置 Api Server domain。

第二步：收集加入会议/点对点通讯的需要参数，会议室/用户的短号或长地址、会议主持人/访客密码（在没有访客密码、呼叫用户的时候可以传空字符串）、接口地址、会中/主叫显示名称。

第三步：收集音视频通话质量相关参数，如果不设置会使用默认的值。关于通话质量的默认参数值为，上传呼叫速率（默认 576kbit/s）、下载呼叫速率（默认 576kbit/s）、视频上传帧率（默认 20fps）、视频下载帧率（默认 20fps）、视频上传分辨率（默认 20fps），视频下载分辨率（默认 20fps）。

第四步：在完成参数配置之后，就可以直接通过 SDK 中的加入会议接口，使用音视频通讯服务。

相关接口：

### 3.1 配置接口地址

Api 名称	参数
setApiServer:(NSString *)apiServer	apiServer：接口地址

实现方式：setter 方法

使用说明：API Server 默认使用公有云 API 地址，私有云请使用本方法设置 API 地址。主要用于调用请求接口。

### 3.2 加入会议参数

Api 名称	参数
ZJSDKVideoJoinConferenceWithTarget:(NSString *)target displayName:(NSString *)displayName password:(NSString *)pwd	target:入会长地址 / 短号 displayName:入会显示名称 pwd:入会密码，对应身份输入相应的密码

实现方式：实例方法

使用说明：收集将要加入会议室的信息，所以加入会议参数为必须赋值参数。

NSMutableDictionary+ZJSDKVideo.h 中定义的方法。

### 3.3 通话质量参数

Api 名称	参数
ZJSDKVideoParamWithMinBandWidth:(NSInteger )minBandWidth expectedBandwidth:(NSInteger )expectedBandWidth minFps:(NSInteger)minFps expectedFps:(NSInteger )expectedFps videoSize:(struct ZJVideoSize )videoSize expectedSize:(struct ZJVideoSize )expectedSize;	minBandWidth :上传呼叫速率 expectedBandWidth:下载呼叫速率 minFps:视频上传帧率 expectedFps：视频下载帧率 videoSize :视频上传分辨率 expectedSize :视频下载分辨率

实现方式：实例方法

使用说明：收集会中音视频通话的质量的参数，可以设置会中的分辨率、速率、帧率等参数。是非必须赋值参数，当没有赋值的时候，使用默认设置参数。

NSMutableDictionary+ZJSDKVideo.h 中定义的方法。

### 3.4 加入会议室（使用默认通话质量参数直接入会）

Api 名称	参数
connectWithModel:(NSDictionary *)conferenceModel showFunctionItem:(BOOL)isShow isAutoPrePresentVC:(BOOL)isAuto	conferenceModel :加入会议的参数设置 (必填参数, 密码为空时, 可传空字符串)  isShow :是否显示功能按钮（静音, 静画, 切换摄像头等功能）  isAuto : 是否自动跳转到控制器

实现方式：实例方法

使用说明：确定会议参数和展示视图的方式入会，可以使用默认的会中功能按钮，也可以直接通过本方法自动跳转到当前会中界面。isShow 能够设置是否使用我们默认的功能按钮的界面。isAuto 能够设置是否需要使用我们提供的 controller 自动跳转到会中界面。isAuto 为 NO 时可以自定义 controller，我们提供了 conferenceView（远端视频和本端视频的界面）来帮助你完成自定义 controller。

### 3.5 加入会议室（可设置通话质量参数）

Api 名称	参数
connectWithModel:(NSDictionary *)conferenceModel videoParams:(NSDictionary *)params showFunctionItem:(BOOL)isShow isAutoPrePresentVC:(BOOL)isAuto	conferenceModel :加入会议的参数设置（必填参数, 密码为空时, 可传空字符串）  params :通话质量的参数设置（非必填参数, 传空字典类型, 使用默认值）  isShow :是否显示功能按钮（静音, 静画, 切换摄像头等功能）  isAuto : 是否自动跳转到控制器

实现方式：实例方法

使用说明：确定会议参数和展示视图的方式入会，可以使用默认的会中功能按钮，也可以直接通过本方法自动跳转到当前会中界面。可以通过自定义的参数设



置通过音频，视频的质量。isShow 能够设置是否使用我们默认的功能按钮的界面。isAuto 能够设置是否需要使用我们提供的 controller 自动跳转到会中界面。isAuto 为 NO 时可以自定义 controller，我们提供了 conferenceView（远端视频和本端视频的界面）来帮助你完成自定义 controller。

使用示例：

### 3.6 加入会议室 (Deprecated!)

Api 名称	参数
connectTarget:(NSString*)target	target:会议室短号
name:(NSString*)name	name:会中显示名
password:(NSString*)pwd	pwd:入会密码
apiServer:(NSString*)server	server:会中 API 域名
bandwidthIn:(NSInteger)input	input:接受呼叫速率(默认 576kbit/s)
bandwidthOut:(NSInteger)output	output:上传呼叫速率(默认 576kbit/s)
minFps:(NSInteger)minFps	minFps:上传帧率(默认 20fps)
expectedFps:(NSInteger)expectedFps	expectedFps:接受帧率(默认 20fps)
showFunctionItem:(BOOL)isShow	isShow:默认展示功能 item
isAutoPrepresentVC:(BOOL)isAuto	isAuto:自动跳转界面
videoSize:(struct ZJVideoSize)videoSize	videoSize:上传分辨率(默认 640x480)
expectedSize:(struct ZJVideoSize)expectedSize	expectedSize:接受分辨率(默认 640x480)

实现方式：实例方法

使用说明：收集入会需要的参数，并触发入会操作。

使用示例：

- (1) 自动跳转界面，无法自定义 Controller：

```

- (IBAction)JoinConference:(id)sender {
    struct ZJVideoSize minSize = {[self.minWidthTf.text integerValue],[self.minHeightTf.text integerValue]};
    struct ZJVideoSize expectedSize = {[self.exWidthTf.text integerValue],[self.exHeightTf.text integerValue]};
    self.manager = [ZJVideoManager sharedManager];
    [self.manager connectTarget:self.NumberTf.text
                     name:self.DisplayName.text
                     password:self.PwdTf.text
                     apiServer:self.API.text
                     bandwidthIn:[self.inputTf.text integerValue]
                     bandwidthOut:[self.outputTf.text integerValue]
                     minFps:20
                     expectedFps:20
                     showFunctionItem:YES
                     isAutoPrepresentVC:YES
                     videoSize:minSize
                     expectedSize:expectedSize];
}

```

(2) 通过 View 自定义 Controller :

```

- (IBAction)JoinWithCustom:(id)sender {
    struct ZJVideoSize minSize = {[self.minWidthTf.text integerValue],[self.minHeightTf.text integerValue]};
    struct ZJVideoSize expectedSize = {[self.exWidthTf.text integerValue],[self.exHeightTf.text integerValue]};
    self.manager = [ZJVideoManager sharedManager];
    [self.manager connectTarget:self.NumberTf.text
                     name:self.DisplayName.text
                     password:self.PwdTf.text
                     apiServer:self.API.text
                     bandwidthIn:[self.inputTf.text integerValue]
                     bandwidthOut:[self.outputTf.text integerValue]
                     minFps:20
                     expectedFps:20
                     showFunctionItem:YES
                     isAutoPrepresentVC:NO
                     videoSize:minSize
                     expectedSize:expectedSize];
    [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(EndedMeeting:)
                                             name:ZJCallDeclinedNotification object:nil];

    videoVC = [[VideoViewController alloc] init];
    videoVC.view = self.manager.conferenceView;

    [self presentViewController:videoVC animated:NO completion:nil];
}

- (void)EndedMeeting:(NSNotification *)sender{
    // 监听退出会议室
    if (videoVC != nil) {
        [[NSOperationQueue mainQueue] addOperationWithBlock:^(
            [videoVC dismissViewControllerAnimated:YES completion:nil];
        )];
    }
}

```

注意：使用自动跳转界面时,使用 View 将出现问题。

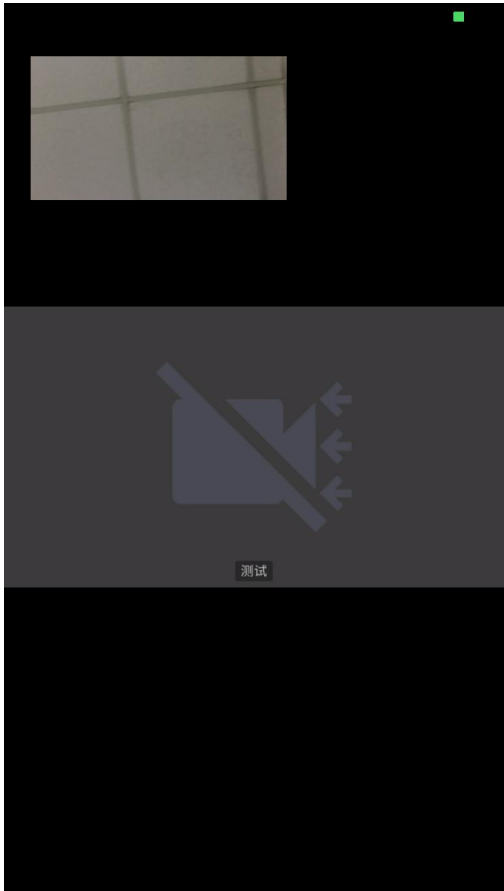
### 3.7 获得远端视频和本端视频的视图

Api 名称	参数
conferenceView;	无

实现方式：属性

使用说明：通过建立音视频通话，创建的会中界面。界面内容包括：远端视频、本端视频、默认功能按钮（静音、静画、切换摄像头、挂断通话）。

使用效果图：



## 4. 点对点被呼叫

完成点对点被呼叫，我们主要是利用 APNS 提供的 voip 推送服务（需要对苹果提供的推送服务 APNS 有一定的了解，会使用苹果提供的 pushkit 框架），来完成被呼叫、被邀请入会、后台拉起等功能。

第一步：配置主要信息 Voip Token(推送服务标识)、Certificate Name(推送证书名称)、向平台提交 VoIP Services Certificate 合成的.pem 文件。

第二步：搭建 Pushkit 环境，需要提供苹果账号下的证书名称、设备注册到 APNS 的 Voip Token。

第三步：向平台注册的账号信息，完成设备和账号的绑定过程。

第四步：然后当另一端呼叫当前注册的账号就能接受到 APNS 的推送信息。

第五步：推送参数为字典类型，根据推送信息可以知道主叫方的信息，选择接听和挂断。

推送参数说明：

格式	参数
<pre>{   .....,   data : {     conference_alias:xxx,     remote_display_name:xxx,     remote_alias:xxx,     time:0000,     token:xxxxxx,     type : xxxxxxxx   } }</pre>	<p>Conference_alias: 被叫账号</p> <p>Remote_display_name:主叫显示名称</p> <p>Remote_alias:主叫地址</p> <p>Time:呼叫时间戳</p> <p>Token:入会 token</p> <p>Type : 主叫用户类型</p>

相关 API：

#### 4.1 配置 mcu server

Api 名称	参数
setMcuServer:(NSString *)mcuServer	mcuServer : mcu 地址

实现方式：setter 方法

使用说明：MCU Server 默认使用公有云 MCU 地址，私有云请使用此方法设置

MCU 地址。设置使用的 MCU Server 的地址。

## 4.2 配置 voip token

Api 名称	参数
setVoipToken:(NSString *)voipToken	voipToken : 推送服务标识

实现方式：setter 方法

使用说明：iOS 提供的推送服务 APNS 当前注册设备的标识。

## 4.3 配置 certificate name

Api 名称	参数
setCertificateName:(NSString *)certificateName	certificateName : 接口地址

实现方式：setter 方法

使用说明：iOS 提供的推送服务的 voip 推送证书。由平台和用户双方约定名称，并向平台提供.p12 文件和.key 文件（用户秘钥）。

## 4.4 注册

Api 名称	参数
registerAccount:(NSString *)account withPassword:(NSString *)pwd completionHandler:(ZJVideoSDKCompletionHandler)completionHandler	account:注册的用户账号 pwd:注册的用户密码 completionHandler: 注册回调

实现方式：实例方法

使用说明：voip token 是向苹果提供的推送服务 APNS 注册的设备 token，来标识当前设备。通过标识（voip token）完成已注册账号（入网）和设备之间的绑

定。返回注册的状态。

#### 4.5 接听呼叫/接听入会参数

Api 名称	参数
ZJSDKVideoJoinConferenceWithTarget:(NSString *)target displayName:(NSString *)displayName token:(NSString *)token	target :入会长地址 / 短号 displayName :入会显示名称 token:入会 token

实现方式：实例方法

使用说明：当收到来电时，收集接通来电的参数，通过加入通话的方式接通(通过接口：[3.3 加入会议室（使用默认通话质量参数直接入会）](#)、[3.4 加入会议室（可设置通话质量参数）](#)）。

#### 4.6 邀请参会者

Api 名称	参数
inviteUserAccount:(NSString *)account withRole:(InviteParticipantRole )role withProtocol:(InviteParticipantProtocol )protocol;	account 用户账号 role 入会身份 protocol 入会协议

实现方式：实例方法

使用说明：主动邀请与会者入会可通过协议 sip h323 rtmp mssip 等方式邀请参会者入会。

#### 4.7 拒绝接听

Api 名称	参数
callRejectWithAccount:(NSString *)account withToken:(NSString *)token completionHandler:(ZJVideoSDKCompletionHandler)completionHandler	server : 接口地址 account :主叫用户账号 completionHandler : 拒绝回调

---

实现方式：实例方法、协议

使用说明：当接收到来电，不方便接听的时候或需要拒绝来电时，可以调用的 API。

使用示例：

## 5. 会中管理 API

以下的 API 可以帮助你完成会中完成静音、静画、切换摄像头、退出会议室等会

中对本地的管理。

### 5.1 切换静音 / 非静音状态

Api 名称	参数
toggleLocalAudio	无

实现方式：实例方法

使用说明：请保证使用 toggleLocalVideo 时正在使用会议通话功能,静音状态是远端收不到本端音频，但是仍然能收到远端的音频。

使用示例：

```
[[ZJVideoManager sharedManager]toggleLocalAudio];
```

### 5.2 切换静画 / 非静画状态（只关闭回显视频）

Api 名称	参数
toggleLocalVideo	无

实现方式：实例方法

使用说明：请保证使用 toggleLocalVideo 时正在使用会议通话功能,只关闭本地视频，远端仍能接受到本地视频。相当于本端不做回显展示。

使用示例：

```
[[ZJVideoManager sharedManager]toggleLocalVideo];
```

### 5.3 切换静画 / 非静画状态（关闭回显、发送视频）

Api 名称	参数
--------	----



toggleVideo	无
-------------	---

实现方式：实例方法

使用说明：请保证使用 toggleVideo 时正在使用会议通话功能, 同时关闭向远端发送视频。

使用示例：

## 5.4 切换摄像头

Api 名称	参数
toggleCamera	无

实现方式：实例方法

使用说明：请保证使用 toggleCamera 时正在使用会议通话功能, 前、后摄像头正常使用的情况下, 通过接口 toggleCamera 可以改变展示画面为前摄像头/后摄像头采集的图像。

使用示例：

```
[[ZJVideoManager sharedManager]toggleCamera];
```

## 5.5 退出当前会议

Api 名称	参数
outOfCurrentMeeting	无

实现方式：实例方法

使用说明：功能是挂断当前会议通话, 所以请保证使用 outOfCurrentMeeting 时正在使用会议通话功能。

使用示例：

```
[[ZJVideoManager sharedManager] outOfCurrentMeeting];
```

## 5.6 结束会议（断开当前所用与会者）

Api 名称	参数
endAllMeeting	无

实现方式：实例方法

使用说明：功能是断开当前所有与会者，所以请保证使用 endAllMeeting 时正在使用会议通话功能。

使用示例：

```
[[ZJVideoManager sharedManager] endAllMeeting];
```

## 6. 监听返回状态

### 6.1 获得丢包率

Api 名称	参数
zjLogPercentageLost:(NSDictionary *)packet	Packet：音视频的上下行丢包信息

实现方式：实例方法、协议

使用说明：键值代表实意 - incomingAudioPL:音频下行丢包率，incomingVideoPL：视频下行丢包率，outgoingAudioPL：音频上行丢包率，outgoingVideoPL：视频上行丢包率。默认为每两秒自动更新一次。

使用示例：

## 6.2 获得挂断状态

Api 名称	参数
zjOutofConference	无

实现方式：实例方法、协议

使用说明：以任何形式触发的挂断会议（包括异常、主动挂断、被动挂断）。

使用示例：

## 6.3 获得通话状态

Api 名称	参数
zjCallBackWithState:(ZJSDKCallState )state withReason:(NSString *)reason withUUID:(NSString *)uuid	state:返回通话状态 reason:发生当前状态的原因 uuid:当前与会者的 uuid

实现方式：实例方法、协议

使用说明：主要包括 5 个状态，Connected：正常连接；Error：建立错误；End：通话结束；StateMute：当前通话本地被静音，HappenError：通话过程中出现异常错误。会返回发生当前状态的原因。

使用示例：

## 7. 版本记录

版本	描述	日期
1.0	文档建立	2017/9/1
1.1	添加功能: 1. 入会打开扬声器和屏幕永亮状态,	2017/9/19

	2. 监听入会通话状态	
1.2	添加功能： <ol style="list-style-type: none"> <li>退出当前会议。</li> <li>结束会议</li> </ol>	2017/12/04
1.3	添加功能： <ol style="list-style-type: none"> <li>切换静音/非静音状态</li> <li>切换静画/非静画状态</li> <li>切换摄像头</li> <li>开放远端视频和本端视频视图</li> </ol>	2017/12/06
1.4	添加功能： <ol style="list-style-type: none"> <li>是否自动跳转默认 View，提供可自定义 Cotroller</li> <li>入会之后强制横屏展示</li> </ol>	2017/12/15
1.5	添加功能： <ol style="list-style-type: none"> <li>获得丢包率</li> </ol> 处理问题： <ol style="list-style-type: none"> <li>关闭本地视频功能出现黑窗。</li> <li>获取 uuid 异常</li> <li>弹窗中文提示</li> <li>崩溃异常不弹窗，在 reason 中提示</li> </ol>	2017 / 12/28
2.0	组织结构。	2018/1/22