



# **TalkCloud SDK for iOS**

**V2.2.9**

## 目录

TalkCloud SDK for iOS .....	1
TalkCloud SDK 简介 .....	5
1、集成TalkCloud SDK for iOS .....	5
1.1、环境准备 .....	5
1.2、导入 .....	5
1.3、工程设置 .....	7
1.4、使用示例程序 .....	9
2、重要概念 .....	9
2.1、房间 .....	9
2.2、用户 .....	9
2.3、音视频 .....	10
3、使用指南 .....	10
3.1、进出房间 .....	10
3.3、用户 .....	10
3.4、音视频 .....	10
3.5、自定义信令 .....	11
4、TalkCloud SDK for iOS –API参考说明 .....	11
4.1、TKRoomSDK.framework文件说明: .....	12
4.2、TKRoomSDK文件 .....	12
获取SDK版本号 .....	12
4.3、TKRoomManager接口类 .....	12
获取房间管理类单例（类方法） .....	12
注册appKey .....	13
注册房间管理类的相关代理对象 .....	13
设置日志打印等级 .....	13
设置房间类代理回调 .....	14
设置音视频数据代理回调 .....	14
设置服务器 .....	15
获取服务器列表 .....	15
加入房间 .....	15
离开房间 .....	16
获取房间属性 .....	16
获取房间配置项 .....	16
获取指定用户对象 .....	17
获取房间中人数 .....	17
获取房间用户信息 .....	17
设置视频分辨率 .....	18
修改指定用户的一个属性 .....	18
修改某一个用户的音视频状态 .....	19
设置发布音视频属性 .....	19
发布本地视频 .....	20

停止发布本地视频 .....	20
发布本地音频 .....	20
停止发布本地音频 .....	20
发送聊天消息 .....	21
发布自定义信令 .....	21
删去自定义信令 .....	23
播放视频 .....	23
停止播放视频 .....	24
播放音频 .....	24
停止播放音频 .....	25
发布（分享）流媒体 .....	25
取消发布（分享）媒体流 .....	25
播放媒体流 .....	26
停止播放媒体流 .....	26
暂停播放媒体流 .....	26
设置媒体流播放的进度 .....	27
设置远端用户的视频音量（包括分享的媒体流） .....	27
播放桌面共享 .....	27
停止播放桌面共享 .....	28
播放电影媒体流 .....	28
停止播放电影媒体流 .....	29
录制用户的音视频流 .....	29
停止录制指定用户的音视频流 .....	29
开始服务器录制 .....	30
停止服务器录制 .....	30
切换纯音频教室 .....	30
踢指定用户出房间 .....	31
切换本地前后摄像头 .....	31
当前本地摄像头是否启用 .....	31
设置启用/禁用本地摄像头 .....	31
本地麦克风是否启用 .....	32
设置本地音频开启/关闭 .....	32
设置其他用户的静音/非静音 .....	32
设置本地设备声音是否外放 .....	32
禁用本地音频 .....	33
禁用本地视频 .....	33
设置视频采集方向 .....	33
播放本地音频文件 .....	33
停止播放本地音频文件 .....	34
4.4、TKRoomDelegate回调函数 .....	34
TKRoomManagerDelegate 房间相关消息回调 .....	34
成功进入房间回调 .....	34
离开房间回调 .....	34
发生错误回调 .....	34

发生警告回调 .....	34
有用户进入房间回调 .....	34
有用户离开房间回调 .....	35
自己被踢出房间回调 .....	35
用户属性发生变化回调 .....	35
用户视频状态变化回调 .....	35
收到自定义信令消息-发布信令消息 .....	36
收到自定义信令消息-删去信令消息 .....	36
收到聊天消息 .....	37
视频数据统计回调 .....	37
音频数据统计回调 .....	37
用户音频音量大小变化回调 .....	38
纯音频与音视频教室切换回调 .....	38
播放用户视频时第一帧回调 .....	38
播放用户音频时第一帧回调 .....	38
网络发生问题回调 .....	39
网络环境发生变化回调 .....	39
用户发布（分享）的媒体流状态变化回调 .....	39
更新媒体流信息的回调 .....	39
媒体流第一帧回调 .....	40
桌面共享状态变化的回调 .....	40
电影共享状态变化的回调 .....	40
TKMediaFrameDelegate 音视频数据实时回调 .....	41
本地采集的音频数据回调 .....	41
本地采集的视频数据回调 .....	41
收到远端音频数据回调 .....	41
收到远端视频数据回调 .....	42
4.5、TKRoomUser 用户对象类 .....	42
4.6、TKRoomDefines.h文件定义说明 .....	42
TKRoomWarningCode 房间警告码 .....	42
TKRoomErrorCode 房间错误码 .....	43
5、FAQ .....	44

# TalkCloud SDK 简介

北京拓课网络科技有限公司由国内顶尖的音视频通讯专家创立，致力于用可靠的音视频技术服务在线教育产业。公司基于新一代通信协议标准打造的拓课云课堂，真正做到全平台互通，轻松实现国际间实时互动教学直播，丰富的API接口及SDK,可帮助机构更快速的接入原有业务系统，并且可以根据业务场景，定制开发相应的功能。

TalkCloud SDK for iOS 是专为iOS平台研发，基于WebRTC协议架构的。拓课云产品具有良好的普适性，支持全终端、全平台接入，从手机、PAD到PC、MAC，从iOS、Android到Windows，都可以运行拓课云产品，而强大的无客户端的网页式交互能力，更是最大程度上方便了数量众多、设备复杂、环境复杂、应用水平较低的学生群体。让他们可以无需安装、快速上手，让应用的学习成本、操作成本几乎为零，也让运营者的推广和教育变得更加便利。

针对移动设备CPU处理能力不高，影响音视频并发效率的状况，拓课云进行了针对性的GPU 技术优化，提高了移动设备的音视频并发能力，降低了CPU 占用和电力消耗。

## 1、集成TalkCloud SDK for iOS

在您阅读此文档时，我们假定您已经具备了基础的前端应用开发经验，并能够理解相关基础概念。

### 1.1、环境准备

您需要有iOS开发环境9.0以上版本Xcode，以及一部可以连接到互联网的手机iPhone或者iPad。

### 1.2、导入

用Xcode创建工程，如下所示：



图1

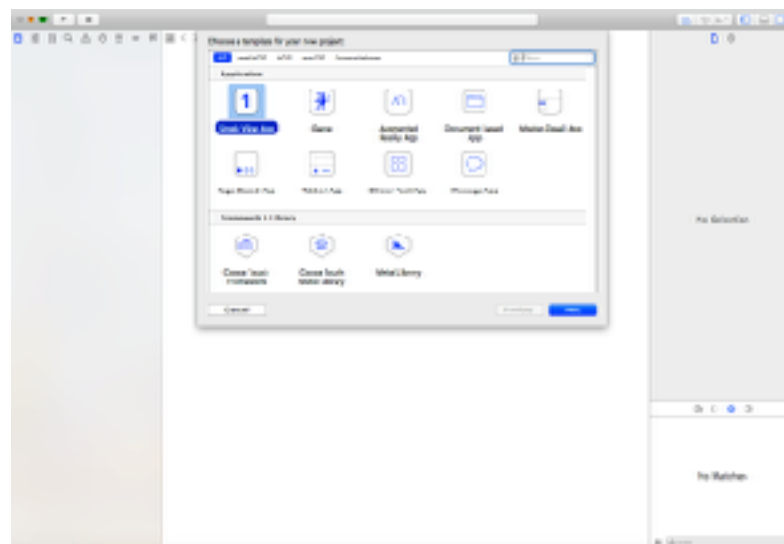


图2

向您的工程导入：TKRoomSDK.framework

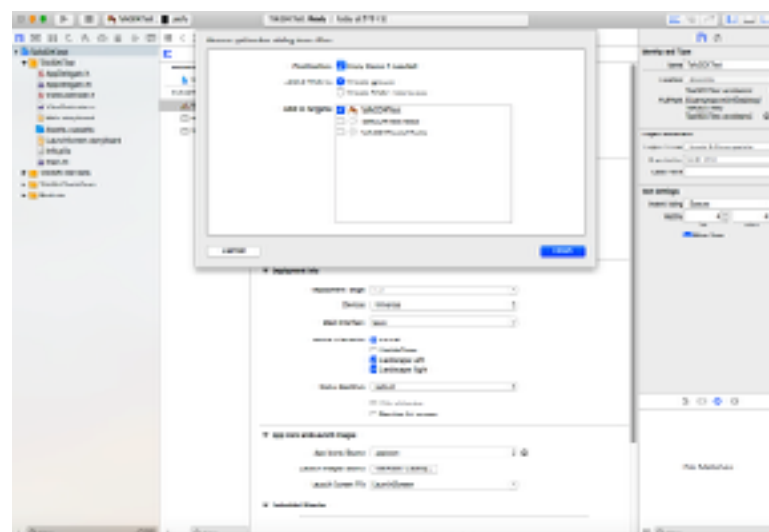


图3

### 1.3、工程设置

#### 配置工程文件

##### 1. Build Settings:

Build Settings -> Linking -> Other Linker Flags -> -all\_load

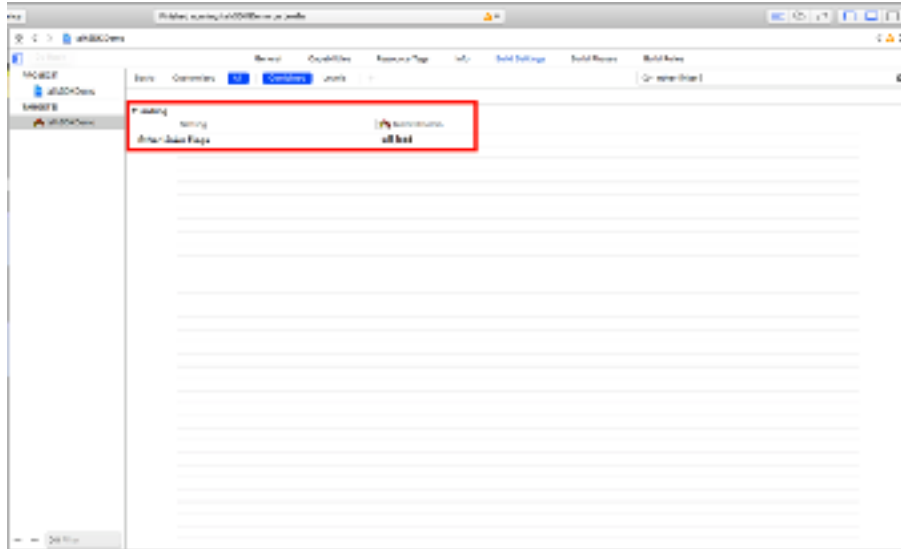


图4

Build Settings -> Build Options -> Enable Bitcode -> NO

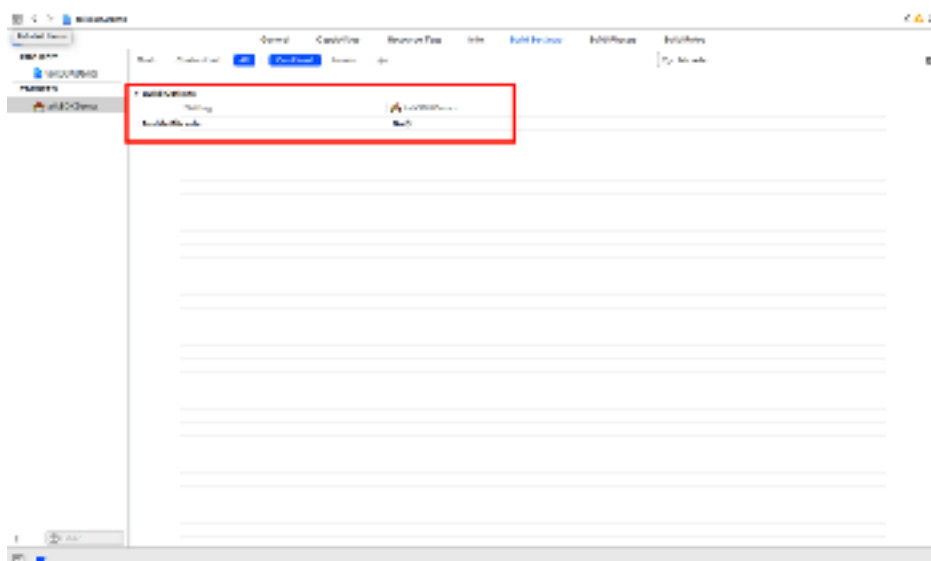


图5

Build Settings -> Build Options -> Always Embed Swift Standard Libraries -> YES

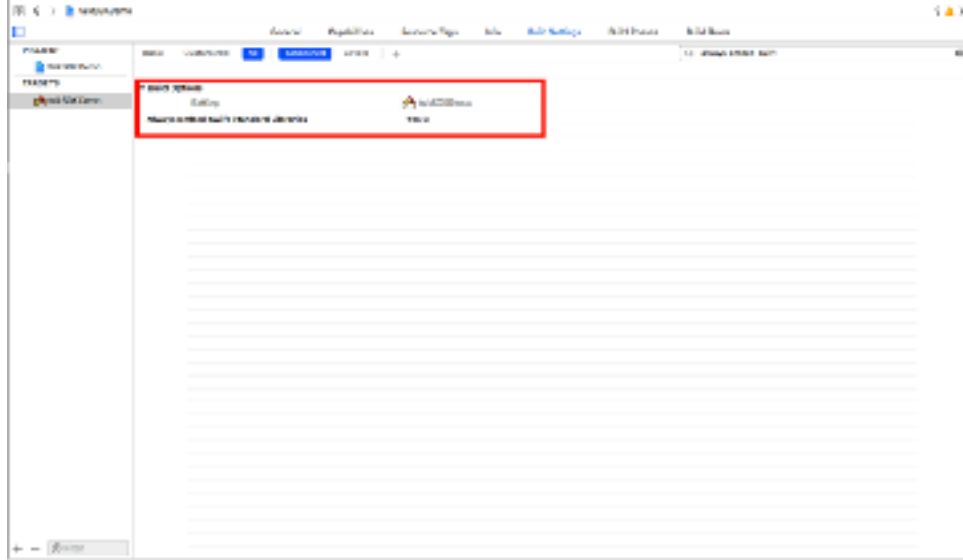


图6

## 2. Build Phases:

Build Phases->Copy Files->Destination->选择为Frameworks

Build Phases->CopyFiles->name->添加 TKRoomSDK.framework

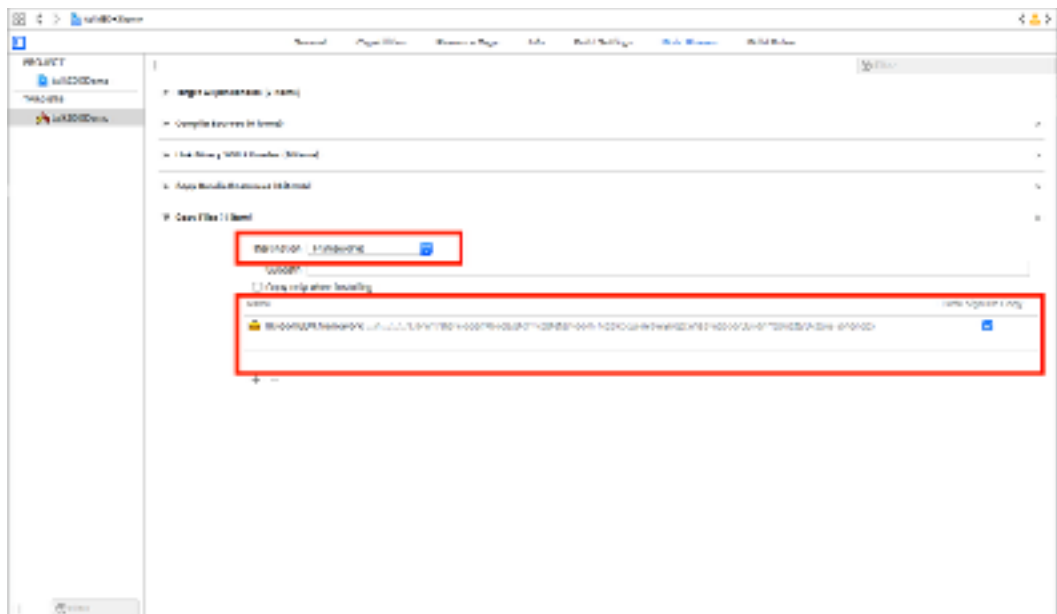


图7

## 3> Info:

info.plist -> Privacy - Camera Usage Description -> 主人需要您的同意，才能访问相机；

info.plist -> Privacy - Microphone Usage Description -> 主人需要您的同意，才能使用麦克风；

info.plist -> Privacy - Photo Library Usage Description -> 主人需要您的同意，才能访问相册；



info.plist -> Privacy - Bluetooth Peripheral Usage Description -> 主人需要您的同意，才能使用蓝牙；

4> 设置App后台运行模式，如需要可以设置如下：

Capabilities->Background Modes->设置成ON->勾选 Audio,AirPlay,and Picture in Picture

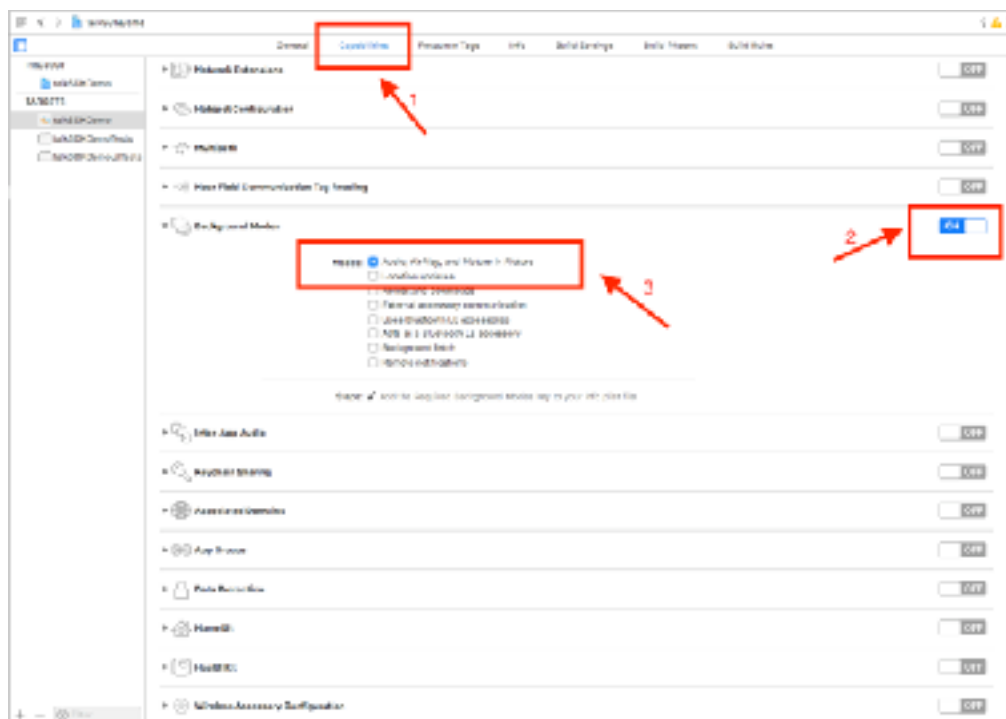


图8

5> SDK目前可以支持arm64、armv7、i386、x86\_64。

## 1.4、使用示例程序

在talkSDKDemo中包含了sdk基本调用方法如加入房间，发布音视频和离开房间。使用Xcode打开示例程序，直接在真机上编译运行。

## 2、重要概念

### 2.1、房间

使用本SDK之前，您需要使用webapi或访问管理系统页面来创建房间。使用获得的房间号即可与同房间用户进行音视频和其他数据的交换了。

### 2.2、用户

使用网页、客户端或手机app皆可进入房间，每一个接入房间的客户端实例，我们称为一个用户。

## 2.3、音视频

每个具有可用的音视频采集设备的用户都可以单独发布自己的音频数据，也可以同时发布音视频数据。

# 3、使用指南

## 3.1、进出房间

进入之前，您需要确保已经通过webapi创建了房间，并得到正确的房间号（详见web接口文档）。

调用joinroom接口加入房间。成功加入房间后，将收到事件room-connected（加入房间成功事件），同时房间内的其他用户将会收到事件room-participant\_join（有用户加入房间事件）。

当需要离开房间时，您需要调用leaveroom接口。离开房间成功后，您将收到事件room-leaveroom（离开房间事件），同时房间内其他用户将收到事件room-participant\_leave（有用户离开房间）。

当遇到网络闪断等异常情况时，您也会收到事件room-disconnected（房间连接中断事件）。在这种情况下，您需要做好项目的相关清除工作。

## 3.3、用户

每一个连接到房间内的客户端，被称为一个用户。用户的身份由userId来唯一标识。当您在调用joinroom接口时，如果传入一个userId，则本地用户的userId即为您设定的Id。如果没有传，则SDK会自动生成一个guid作为您的userId。但请注意，无论是哪种情况，系统不允许同一个房间内存在userId相等的两个用户。所以当您使用一个房间内已经存在的userId进入房间时，服务器会把之前使用这个id的用户请出房间。

除了userId外，每个用户还有一些其他属性（预定义的和自定义的），您可以使用changeUserProperty接口来修改任一用户的任一属性,并且可以指定把这项修改通知给哪个（些）用户。随后，您指定的用户就会收到一个事件room-userproperty-changed（用户属性改变），告知某个用户的某些属性被修改了。

## 3.4、音视频

您可以调用publishVideo和publishAudio接口来发布自己的音频或视频。一旦用户开始

发布音、视频数据，房间内的所有用户都会收到事件room-uservideostate-changed和room-useraudiostate-changed。要注意的是，您不需要发布本地用户的视频，都可以随时调用playVideo观看自己的视频图像。如果您不想观看某个用户的视频，可以通过调用unplayVideo结束观看。

您可以调用unpublishVideo和unpublishAudio接口来取消发布自己的音频或视频。一旦用户将自己的音视频都取消发布，房间内的所有用户都会收到事件room-uservideostate-changed和room-useraudiostate-changed，这时您需要调用unplayVideo和unplayAudio取消观看别人的视频和收听别人的音频。

### 3.5、自定义信令

利用自定义信令接口，您可以完成您设计的各种业务流程，创造各种激动人心的用户体验。我们把各种信令都抽象成了消息，和用户类似，消息有自己的id（msgId）和昵称（msgName），昵称可以和其他消息重名，但id必须在房间内唯一，id相同的消息，后来的会替换之前的。除了这两个属性，消息还有这些属性：

told：谁会接收到该消息的通知。可以填入目标用户的peerID，也可以填入如下4种关键字：

"\_\_allExceptSender"：通知给除了调用者以外的所有人

"\_\_allExceptAuditor"：通知给除了旁听用户以外的所有人

"\_\_none"：不通知给任何人

"\_\_all"：通知给所有人

data：数据（可以是NSNumber、NSString、NSDictionary或NSArray）

save：是否保存。当为真时，服务器会在房间内保存该条消息，新进入的用户会收到该条消息。如果连续发两条id相同且需要保存的消息，服务器只会保存后一条。；当为假时，服务器在转发完这条消息后，即将此消息抛弃

具体流程如下：

调用pubMsg接口发布一条消息，则所有指定用户将收到事件room-pubmsg（pubMsg消息事件）。

如果这条消息是指定保存的，则之后新进入房间的用户会收到事件room-connected(房间连接成功事件)的数据中带有当前服务器的所有保存的且未删除的消息数组。在发布消息之后，可以再次调用pubMsg接口，发布一条相同id的消息来覆盖它，则指定用户会再次收到事件room-pubmsg（pubMsg消息发布事件）。也可以调用delMsg接口来删除此消息，此时指定用户会收到事件room-delmsg（delMsg消息删除事件）。

## 4、TalkCloud SDK for iOS –API参考说明

### 4.1、TKRoomSDK.framework文件说明：

TalkCloud SDK for iOS是打包成framework的动态库，名称：  
TKRoomSDK.framework。

TKRoomSDK.framework中header文件夹中包含7个头文件。

1. TKRoomSDK.h：可以导入所有的头文件。
2. TKRoomManager.h：SDK的主接口类文件；
3. TKRoomDelegate.h：SDK的主代理回调类文件；
4. TKRoomDefines.h：SDK中使用的主定义类文件；
5. TKRoomUser.h：SDK用户类接口文件；
6. TKRoomWhiteBoardNotification.h：定义白板通知消息类文件；
7. TKUtils.h：SDK工具类头文件
8. TKRoomSDK-Swift.h: SDK内部OC与Swift混编，桥接文件。

### 4.2、TKRoomSDK文件

#import<KRoomSDK/TKRoomSDK.h>就可以导入TKRoomSDK.framework中所有头文件。

### 获取SDK版本号

TKRoomSDKVersionString[];  
调用该方法获取SDK版本号，返回值为指针地址。

### 4.3、TKRoomManager接口类

TKRoomManager类是使用TalkCloud SDK的主要接口类，对 SDK 功能的使用都通过这个类进行调用，TKRoomManager的接口最好在同一个线程进行，不建议在不同的线程同时调用。

### 获取房间管理类单例（类方法）

+ (instancetype)instance;

该方法实例化一个房间管理类单例对象。房间中所有操作，几乎都由该对象控制。

名称	描述
返回值 (instancetype)	房间管理类单例

TKRoomManager \*roomMgr = [TKRoomManager instance];

## 销毁房间管理类（类方法）

+ (void)destory;

注：离开房间时，必须要调用。

示例代码：

```
[TKRoomManager destory];
```

## 注册appKey

- (int)initWithAppKey:(NSString \*)appKey optional:(NSDictionary \* \_Nullable)optional;

该方法用于初始化Room对象

注：请在调用任何room方法前调用此方法

名称	描述
appKey	请填写正确的企业ID，形如： 82AUScgguvqXzhUh
optional	房间扩展信息

## 设置日志打印等级

- (int)setLogLevel:(TKLogLevel)level logPath:(NSString \*)logPath debugToConsole:(BOOL)debug;

名称	描述
level	日志等级 Verbose：表示所有的日志都会打印 info：打印info、warning、error warning：打印warning、error error：打印error
logPath(NSString)	日志写入本地沙盒路径，默认路径：沙盒 Documents/TKLog。 日志等级为TKLog_None时不会写入沙盒。

名称	描述
debug	是否是Debug模式， Debug模式：控制台打印 Release模式：控制台不打印
返回值 (int)	0:方法调用成功 非0:方法调用失败

### 设置房间类代理回调

```

-(int)registerRoomManagerDelegate:(id<TKRoomManagerDelegate>)
roomDelegate
-(int)registerRoomWhiteBoardDelegate:(id<TKRoomManagerDelegate>
_Nullable)roomDelegate andWB:(BOOL)notifyWB;
-(int)registerRoomManagerPlaybackDelegate:(id<TKRoomManagerDelegate>
_Nullable)roomDelegate andWB:(BOOL)notifyWB;

```

名称	描述
roomDelegate	实现了TKRoomManagerDelegate代理的对象
notifyWB	是否需要发送白板消息通知（如使用TalkCloud白板SDK功能，请传YES。）
返回值 (int)	0:方法调用成功 非0:方法调用失败

### 设置音视频数据代理回调

```

- (int)registerMediaDelegate:(id<TKMediaFrameDelegate>
_Nullable)mediaDelegate;

```

名称	描述
mediaDelegate	实现了TKMediaDelegate代理的对象
返回值 (int)	0:方法调用成功 非0:方法调用失败

## 设置服务器

- (int)changeCurrentServer:(NSString \*)serverName;

名称	描述
serverName (NSString)	服务器名字（可以通过调用getServerList获取服务器列表）
返回值 (int)	0:调用成功 非0:调用失败

## 获取服务器列表

-(NSArray \* \_Nullable)getServerList;

名称	描述
返回值NSArray	服务器信息列表

## 加入房间

- (int)joinRoomWithHost:(NSString \*)host  
port:(int)port  
nickName:(NSString \*)nickname  
roomParams:(NSDictionary \*)roomParams  
userParams:(NSDictionary \* \_Nullable)userParams

调用该方法加入已创建的房间。此方法应为Room类中首先被调用的方法。

名称	描述
host (NSString)	服务器地址
port (int)	服务器端口
nickname (NSString)	昵称
roomParams (NSDictionary)	房间属性参数（例如密码，用户角色等等）
userParams (NSDictionary)	用户属性（可以根据用户需求自定义）
lowConsume (BOOL)	是否是低端设备
返回值 (int)	0:调用成功 非0:调用失败

port:80  
nickName:@"test"

```
roomParams:@{
    @"serial": @"199593119",
    @"userrole": @(2)}
userParams:nil
lowConsume:NO];
```

## 离开房间

```
- (int)leaveRoom:(completion_block _Nullable)completion;
- (int)leaveRoom:(BOOL)force Completion:(completion_block
_Nullable)completion;
```

使用该方法离开房间。离开房间后，所有音视频通话都被终止。

名称	描述
completion	离开房间后的回调block
force	是否强行离开
返回值 (int)	0:调用成功 非0:调用失败

示例代码：

```
[_roomMgr leaveRoom:nil];
```

## 获取房间属性

```
- (TKRoomProperty *)getRoomProperty;
```

该方法获取房间信息。此信息中包括预约房间时的所有配置。

名称	描述
返回值 (TKRoomProperty)	返回TKRoomProperty对象（具体参考TKRoomDefines.h文件中定义）

示例代码：

```
TKRoomProperty *roomProperty = [_roomMgr getRoomProperty];
```

## 获取房间配置项

```
- (TKRoomConfiguration *)getRoomConfiguration
```

名称	描述
返回值 (TKRoomConfiguration)	返回TKRoomConfiguration对象



## 获取指定用户对象

- (TKRoomUser \*)getRoomUserWithUId:(NSString \*)peerId

该方法获取指定RoomUser对象

名称	描述
peerId (NSString)	指定用户的ID
返回 (TKRoomUser)	返回用户TKRoomUser对象（具体参考TKRoomUser.h文件中定义）

## 获取房间中人数

- (int)getRoomUserNumberWithRole:(NSArray \* \_Nullable)role callback:(void (^)(

(NSInteger num, NSError \*error))callback

该方法返回房间中所有用户人数。callback异步返回。

名称	描述
role (NSArray)	指定需要获取哪些角色的用户
callback	异步返回获取到的数据， num：人数，error：错误
返回 (int)	0:调用成功 非0:调用失败

## 获取房间用户信息

- (int)getRoomUsersWithRole:(NSArray \* \_Nullable)role startIndex:(NSInteger)start  
maxNumber:(NSInteger)max callback:(void (^)(NSArray <TKRoomUser \*> \_Nonnull users  
, NSError \*error))callback

该方法返回房间中指定角色的用户信息。callback异步返回。

名称	描述
role (NSArray)	指定需要获取哪些角色的用户
start (NSInteger)	起始位置
max (NSInteger)	需要获取的人数，默认最大200
callback	异步返回获取到的数据， users：NSArray类型，包含的是TKRoomUser类型对象的数组，error：错误
返回 (int)	0:调用成功 非0:调用失败

```

示例代码：
[roomMgr getRoomUsersWithRole:@[@"2"] startIndex:0 maxNumber:50
callback:^(NSArray <TKRoomUser *>* _Nonnull users, NSError *error) {
    //do something
    dispatch_async(dispatch_get_main_queue(), ^{
        //do something
    });
}];

```

## 设置视频分辨率

- (int)setVideoProfile:(TKVideoProfile \*)profile  
 此方法需要在发布视频成功后调用，方可设置成功。

名称	描述
profile (TKVideoProfile)	TKVideoProfile对象（具体参考TKRoomDefines.h文件中定义）
返回 (int)	0:调用成功 非0:调用失败

## 修改指定用户的一个属性

```

- (int)changeUserProperty:(NSString *)peerID
    tellWhom:(NSString *)tellWhom
    key:(NSString *)key
    value:(NSObject *)value
    completion:(completion_block _Nullable)completion

- (int)changeUserProperty:(NSString *)peerID
    tellWhom:(NSString *)tellWhom
    data:(NSDictionary *)data
    completion:(completion_block _Nullable)completion

```

名称	描述
peerID (NSString)	指定用户的ID
tellWhom (NSString)	要将此修改通知给谁。“__all”:所有人 “__allExceptSender”：除了自己之外的所有人 “__allExceptAuditor”：除了旁听用户以外的人 “__none”：不通知任何人，某用户的peerID：只发给发用户。 <b>注：被通知的用户会收到room-userproperty-changed事件</b>
key (NSString)	要修复的属性名称key值，可以自定义

名称	描述
value (NSObject)	要修改的属性的value值，NSObject可以是NSNumber，NSString，NSDictionary，NSArray
data (NSDictionary)	要修改的属性（键值对）NSDictionary
completion	调用完的回调
返回 (int)	0:调用成功 非0:调用失败

示例代码：

```
[_roomMgr changeUserProperty:@"1234569"
    tellWhom:@"__all"
    key:@"key"
    value:@"value"
    completion:nil];
```

修改某一个用户的音视频状态

```
- (int)changeUserPublish:(NSString *)peerID
    publishState:(TKPublishState)publishState
    completion:(completion_block _Nullable)block
```

名称	描述
peerID(NSString)	指定的用户ID
publishState	TKPublishState类型的枚举值（具体参考TKRoomDefines.h文件中定义） <b>注：所有用户会收到room-userproperty-changed事件</b>
返回 (int)	0:调用成功 非0:调用失败

设置发布音视频属性

```
- (int)setAttributes:(NSDictionary *)attributes TK_Deprecated("Will deprecated!!!");
```

此方法是设置自定义关于发布音视频属性。不调用不影响音视频的发布。

名称	描述
attributes (NSDictionary)	自定义关于音视频的属性 NSDictionary
返回 (int)	0:调用成功 非0:调用失败

## 发布本地视频

- (int)publishVideo:(completion\_block \_Nullable)completion;

该方法将自己的视频发布出去，房间中的人会收到room-userpublishstate-changed事件，然后可通过调用playVideo方法观看发布者的视频。

名称	描述
completion	完成调用的回调block
返回 (int)	0:调用成功 非0:调用失败

## 停止发布本地视频

- (int)unPublishVideo:(completion\_block \_Nullable)completion

该方法停止发布自己的视频，房间中的人会收到room-userpublishstate-changed事件，然后可调用unplayVideo方法关闭播放器。

名称	描述
completion	完成调用的回调block
返回 (int)	0:调用成功 非0:调用失败

## 发布本地音频

- (int)publishAudio:(completion\_block \_Nullable)completion

该方法将自己的音频发布出去，房间中的人会收到room-userpublishstate-changed事件，然后可通过调用playAudio方法观收听布者的音频。

名称	描述
completion	完成调用的回调block
返回 (int)	0:调用成功 非0:调用失败

## 停止发布本地音频

- (int)unPublishAudio:(completion\_block \_Nullable)completion

该方法停止发布自己的音频，房间中的人会收到room-userpublishstate-changed事件，然后可调用unplayAudio方法关闭播放器。

名称	描述
completion	完成调用的回调block

名称	描述
返回 (int)	0:调用成功 非0:调用失败

## 发送聊天消息

```
- (int)sendMessage:(NSObject *)message
    toID:(NSString *)toID
    extensionJson:(NSObject * _Nullable)extension
```

该方法可以给指定的用户发送聊天消息（仅支持文本消息）

名称	描述
message(NSObject)	聊天文本消息，可以是NSString、NSDictionary
toID(NSString)	要将此修改通知给谁。“__all”:所有人 “__allExceptSender”：除了自己之外的所有人，“__allExceptAuditor”：除了旁听用户以外的所有人，“__none”：不通知任何人，某用户的peerID：只发给发用户 <b>注：被通知的用户会收到room-receivemessage事件</b>
extension(NSObject)	聊天消息的扩展信息，支持NSString(JSON字符串string)、NSDictionary
返回 (int)	0:调用成功 非0:调用失败

## 发布自定义信令

```
- (int)pubMsg:(NSString *)msgName
    msgID:(NSString *)msgID
    toID:(NSString *)toID
    data:(NSObject *)data
    save:(BOOL)save
    completion:(completion_block _Nullable)completion;
```

```
- (int)pubMsg:(NSString *)msgName
    msgID:(NSString *)msgID
    toID:(NSString *)toID
    data:(NSObject *)data
    save:(BOOL)save
    associatedMsgID:(NSString * _Nullable)associatedMsgID
    associatedUserID:(NSString * _Nullable)associatedUserID
    expires:(NSTimeInterval)expires
    completion:(completion_block _Nullable)completion;
```

```
- (int)pubMsg:(NSString *)msgName
```

```

msgID:(NSString *)msgID
toID:(NSString *)toID
data:(NSObject *)data
save:(BOOL)save
extensionData:(NSDictionary * _Nullable)extensionData
completion:(completion_block _Nullable)completion;

```

关于自定义信令的理解，可见详情3.5:自定义信令一节。

名称	描述
msgName(NSString)	信令名称
msgID(NSString)	信令ID
toID(NSString)	<p>要将此修改通知给谁。”__all”:所有人  “__allExceptSender”: 除了自己之外的所有人,  “__allExceptAuditor”: 除了旁听用户以外的所有人,  “__none”: 不通知任何人, 某用户的peerID: 只发给发用户。</p> <p><b>注: 接收者会收到room-pubmsg事件</b></p>
data(NSObject)	信令消息内容, NSObject可以是NSNumber, NSString, NSDictionary, NSArray
save	<p>[非必选项]消息是否保存到服务器, 缺省为false。若不保存到服务器, 晚于发布消息时刻进入房间的用户不会收到此消息, 只有发布消息时刻前进入房间的用户可收到此消息; 若保存到服务器, 在该消息未被删除的情况下, 任何时刻进入房间的用户都会收到此消息。</p> <p><b>注: 未调用删除消息接口 (delMsg) 情况下, 若房间中所有用户已退出, 此消息会被自动删除</b></p> <p>详见3.5:自定义信令</p>
associatedMsgID(NSString)	[非必选项]信令关联的消息ID, 假设此消息关联了一条msgID为'test-msg'的消息, 那么当'test-msg'消息被删除后, 此消息也被删除
associatedUserID(NSString)	[非必选项]信令关联的用户ID, 假设此消息关联了一个id为'user-id-123'的用户, 那么当'user-id-123'用户离开房间后, 此消息将被删除
expires	时间戳, 相对于classbegin, 表示信令在期望的时间后到期,
extensionData(NSDictionary)	扩展信息
completion	发布完成回调
返回 (int)	<p>0:调用成功  非0:调用失败</p>

## 删去自定义信令

```
- (int)delMsg:(NSString *)msgName
    msgID:(NSString *)msgID
    toID:(NSString *)toID
    data:(NSObject *)data
    completion:(completion_block _Nullable)completion;
```

关于自定义信令的理解，可见详情3.5:自定义信令一节。

名称	描述
msgName(NSString)	信令名称
msgID(NSString)	信令ID 注：删除消息的ID必须与pubMsg方法发布的消息ID对应，否则会失败
toID(NSString)	要将此修改通知给谁。“__all”:所有人 “__allExceptSender”：除了自己之外的所有人，“__allExceptAuditor”：除了旁听用户以外所有人，“__none”：不通知任何人，某用户的peerID：只发给发用户。 注：接收者会收到room-delmsg事件
data(NSObject)	信令消息内容，NSObject可以是NSNumber，NSString，NSDictionary，NSArray
completion	发布完成回调
返回 (int)	0:调用成功 非0:调用失败

## 播放视频

```
- (int)playVideo:(NSString *)peerID
    renderType:(TKRenderMode)renderType
    window:(UIView *)window
    completion:(completion_block _Nullable)completion;
```

该方法用于播放指定用户的视频。若播放其他用户视频必须在收到room-uservideostate-changed事件后且视频为发布状态方可调用；若播放本地视频，则可直接调用。

注：此方法必须在主线程调用。

名称	描述
peerID(NSString)	用户ID

名称	描述
renderType	视频渲染模式， 0:表示等比拉伸， 1:表示等比拉伸，并占满全屏。
window	播放视频窗口View指针
completion	完成回调
返回 (int)	0:调用成功 非0:调用失败

## 停止播放视频

```
- (int)unPlayVideo:(NSString *)peerID completion:(completion_block
_Nullable)completion;
```

该方法停止播放指定用户的视频。

名称	描述
peerID(NSString)	用户ID
completion	完成回调
返回 (int)	0:调用成功 非0:调用失败

## 播放音频

```
- (int)playAudio:(NSString *)peerID completion:(completion_block
_Nullable)completion;
```

该方法用于播放指定用户的音频。请在收到room-useraudiostate-changed事件后且音频为发布状态时调用该方法。

**注：用户不能播放自己的音频**

名称	描述
peerID(NSString)	用户ID
completion	完成回调
返回 (int)	0:调用成功 非0:调用失败



## 停止播放音频

```
- (int)unPlayAudio:(NSString *)peerID completion:(completion_block _Nullable)completion;
```

该方法停止播放指定用户的音频。

名称	描述
peerID(NSString)	用户ID
completion	完成回调
返回 (int)	0:调用成功 非0:调用失败

## 发布（分享）流媒体

```
- (int)startShareMediaFile:(NSString *)mediaPath  
    isVideo:(BOOL)isVideo  
    toID:(NSString *)toID  
    attributes:(NSDictionary *)attributes  
    block:(completion_block _Nullable)completion;
```

该方法可以分享（发布）一个媒体视频给其他用户。如果正在分享视频中再次调用此方法，需要将之前分享的停止掉。例如：mp3、mp4文件。

名称	描述
mediaPath(NSString)	文件路径
isVideo	发布的文件是否是视频文件
toID(NSString)	发布媒体流给谁。 “__all”:所有人 “__allExceptSender”：除了自己之外的所有人, “__allExceptAuditor”：除了旁听用户以外的所有人, “__none”：不通知任何人, 某用户的peerID：只发给发用户。 <b>注：被通知用户会收到room-shareMediaState事件</b>
attributes	附带扩展信息
completion	完成回调
返回 (int)	0:调用成功 非0:调用失败

## 取消发布（分享）媒体流

```
- (int)stopShareMediaFile:(completion_block _Nullable)completion;
```

该方法可以取消已经发布（分享）的媒体流。

名称	描述
completion	完成回调 注：被通知用户会收到room-shareMediaState事件
返回 (int)	0:调用成功 非0:调用失败

## 播放媒体流

```
- (int)playMediaFile:(NSString *)peerId
    renderType:(TKRenderMode)renderType
    window:(UIView *)window
    completion:(completion_block _Nullable)completion;
```

此方法可以播放其他用户分享的媒体流。

注：此方法必须在主线程调用。

名称	描述
peerId(NSString)	播放分享媒体流的用户ID
renderType	媒体视频的渲染模式，0:表示等比拉伸，1:表示等比拉伸，并铺满全屏。只对视频起效果。
window(UIView)	播放视频的窗口视图
completion	完成回调
返回 (int)	0:调用成功 非0:调用失败

## 停止播放媒体流

```
- (int)unPlayMediaFile:(NSString *)peerId completion:(completion_block _Nullable)completion;
```

此方法可以停止播放正在分享（发布）的媒体流。

名称	描述
peerId(NSString)	播放分享媒体流的用户ID
completion	完成回调
返回 (int)	0:调用成功 非0:调用失败

## 暂停播放媒体流

```
- (int)pauseMediaFile:(BOOL)pause;
```

此方法可以暂停/继续播放媒体流。所有用户都会收到updatemediastream的消息

名称	描述
pause	YES：表示暂停，NO：表示继续
返回 (int)	0:调用成功 非0:调用失败

## 设置媒体流播放的进度

- (int)seekMediaFile:(NSTimeInterval)pos;

此方法可以设置当前正在播放的媒体流的进度，时间戳。所有用户都会收到updatemediastream的消息

名称	描述
pos	需要设置的播放时间位置
返回 (int)	0:调用成功 非0:调用失败

## 设置远端用户的视频音量（包括分享的媒体流）

- (int)setRemoteAudioVolume:(CGFloat)volume peerId:(NSString \*)peerId type:(TKMediaType)type;

此方法可以设置某一个用户的视频音量（包括其分享的媒体流）

名称	描述
Volume	音量大小，范围是： 0 - 1
peerId(NSString)	用户ID
type	视频类型（包括摄像头采集视频、共享的媒体视频）
返回 (int)	0:调用成功 非0:调用失败

## 播放桌面共享

- (int)playScreen:(NSString \*)peerID  
renderType:(TKRenderMode)renderType  
window:(UIView \*)window  
completion:(completion\_block \_Nullable)completion;

此方法可以播放其他用户发布的桌面共享。

注：此方法必须在主线程调用。

名称	描述
peerID(NSString)	发布桌面共享的用户ID
renderType	流的渲染模式，0:等比拉伸 1:表示等比拉伸，并平铺画面
window(UIView)	显示画面的窗口View
completion	调用完成的block回调
返回 (int)	0:调用成功 非0:调用失败

## 停止播放桌面共享

- (int)unPlayScreen:(NSString \*)peerID completion:(completion\_block \_Nullable)completion;

此方法可以停止播放正在发布的桌面共享流。

名称	描述
peerID(NSString)	发布桌面共享的用户ID
completion	调用完成的block回调
返回 (int)	0:调用成功 非0:调用失败

## 播放电影媒体流

- (int)playFile:(NSString \*)peerID  
renderType:(TKRenderMode)renderType  
window:(UIView \*)window  
completion:(completion\_block \_Nullable)completion;

此方法可以播放其他用户发布的电影媒体文件流。

**注：此方法必须在主线程调用。**

名称	描述
peerID(NSString)	发布电影媒体流的用户ID
renderType	流的渲染模式，0:等比拉伸 1:表示等比拉伸，并平铺画面
window(UIView)	显示画面的窗口View
completion	调用完成的block回调
返回 (int)	0:调用成功 非0:调用失败

## 停止播放电影媒体流

```
- (int)unPlayFile:(NSString *)peerId completion:(completion_block _Nullable)completion;
```

此方法可以停止播放正在发布的电影媒体流。

名称	描述
peerId(NSString)	发布电影媒体流的用户ID
completion	调用完成的block回调
返回 (int)	0:调用成功 非0:调用失败

## 录制用户的音视频流

```
- (int)startRecordUser:(NSString *)peerId  
    convert:(NSInteger)convert  
    completion:(void (^)(NSInteger ret, NSString *path))completion;
```

调用此方法可以开始录制指定用户的音视频流

名称	描述
peerId(NSString)	指定用户的ID
convert	录制的视频流转换格式： 0:表示不转换，1:表示webm，2:表示mp4
completion	调用完成的block回调， 参数 ret: 0表示成功，非0表示失败 参数path: 录制视频的路径
返回 (int)	0:调用成功 非0:调用失败

## 停止录制指定用户的音视频流

```
- (int)stopRecordUser:(NSString *)peerId completion:(void (^)(NSInteger, NSString *path))completion;
```

调用此方法可以停止录制指定用户的音视频流

名称	描述
peerId(NSString)	指定用户的ID
completion	调用完成的block回调， 参数 ret: 0表示成功，非0表示失败 参数path: 录制视频的路径

名称	描述
返回 (int)	0:调用成功 非0:调用失败

## 开始服务器录制

- (int)startServerRecord:(TKRecordType)recordType convert:(NSInteger)convert layout:(NSInteger)layout expiresabs:(NSInteger)expiresabs expires:(NSInteger)expires;

调用此方法可以开启服务器房间录制。

名称	描述
recordType	录制类型，表示生成的录制件的类型
convert	录制件的格式：只有在recordType = 0与1的情况下起效果。 0:表示不转换（mkv格式） 1:表示webm（recordType其他值时，默认状态） 2:表示mp4
layout	只有在recordType = 3的情况下起效果，
expiresabs	录制时长
expires	期望的结束录制的时间
completion	调用完成的block回调， 参数 ret：0表示成功，非0表示失败 参数path：录制视频的路径
返回 (int)	0:调用成功 非0:调用失败

## 停止服务器录制

- (int)stopServerRecord;  
此方法可以停止服务器录制。

## 切换纯音频教室

- (int)switchOnlyAudioRoom:(BOOL)isSwitch;

此方法可以在正常房间和纯音频房间之间切换。 所有用户都会收到room-pubmsg事件回调的消息

名称	描述
isSwitch	YES：表示纯音频，NO：表示正常房间（音视频）
返回（int）	0:调用成功 非0:调用失败

## 踢指定用户出房间

- (int)evictUser:(NSString \*)peerID evictReason:(NSNumber \*)reason  
completion:(completion\_block \_Nullable)completion;

此方法可以将指定用户踢出房间。

名称	描述
peerID	指定用户的ID
reason	踢出原因
completion	调用回调block
返回（int）	0:调用成功 非0:调用失败

## 切换本地前后摄像头

- (int)selectCameraPosition:(BOOL)isFront;

名称	描述
isFront	YES：表示前置摄像头，NO：表示后置摄像头
返回（int）	0:调用成功 非0:调用失败

## 当前本地摄像头是否启用

- (BOOL)isVideoEnabled;

名称	描述
返回（BOOL）	YES：启用 NO：禁用

## 设置启用/禁用本地摄像头

- (int)enableVideo:(BOOL)enable;

名称	描述
enable	YES：表示启用， NO：表示禁用
返回（int）	0:调用成功 非0:调用失败

## 本地麦克风是否启用

- (BOOL)isAudioEnabled;

名称	描述
返回（BOOL）	YES： 启用 NO： 禁用

## 设置本地音频开启/关闭

- (int)enableAudio:(BOOL)enable;

名称	描述
enable	YES：表示启用， NO：表示禁用
返回（int）	0:调用成功 非0:调用失败

## 设置其他用户的静音/非静音

- (int)enableOtherAudio:(BOOL)enable;

名称	描述
enable	YES：表示静音， NO：表示非静音
返回（int）	0:调用成功 非0:调用失败

## 设置本地设备声音是否外放

- (int)useLoudSpeaker:(BOOL)use;



名称	描述
use	YES：表示外放（内置扬声器）， NO：表示非外放
返回（int）	0:调用成功 非0:调用失败

## 禁用本地音频

- (int)disableMyAudio:(BOOL)disable;

名称	描述
disable	YES：表示禁用， NO：表示启用
返回（int）	0:调用成功 非0:调用失败

## 禁用本地视频

- (int)disableMyVideo:(BOOL)disable;

名称	描述
disable	YES：表示禁用， NO：表示启用
返回（int）	0:调用成功 非0:调用失败

## 设置视频采集方向

- (int)setVideoOrientation:(UIDeviceOrientation)orientation;

名称	描述
orientation	设备取向
返回（int）	0:调用成功 非0:调用失败

## 播放本地音频文件

- (int)startPlayAudioFile:(NSString \*)filePath loop:(BOOL)loop;  
调用此方法可以播放一个本地音频文件。

名称	描述
filePath(NSString)	文件路径
loop	是否循环播放，YES：循环播放，NO：不循环播放
返回 (int)	返回一个即将播放音频的播放ID

## 停止播放本地音频文件

- (int)stopPlayAudioFile:(int)channel;

名称	描述
channel	指定停止播放音频的播放ID
返回 (int)	0:调用成功 非0:调用失败

## 4.4、TKRoomDelegate回调函数

### TKRoomManagerDelegate 房间相关消息回调

#### 成功进入房间回调

- (void)roomManagerRoomJoined;

#### 离开房间回调

- (void)roomManagerRoomLeft;

#### 发生错误回调

- (void)roomManagerDidOccuredError:(NSError \*)error;

#### 发生警告回调

- (void)roomManagerDidOccuredWaring:(TKRoomWarningCode)code;

#### 有用户进入房间回调

- (void)roomManagerUserJoined:(NSString \*)peerID inList:(BOOL)inList;

名称	描述
peerID(NSString)	进入房间的用户ID
inlist	YES：表示该用户是在自己进入房间之前进入的 NO：表示该用户是在自己进入房间之后进入的

## 有用户离开房间回调

- (void)roomManagerUserLeft:(NSString \*)peerID;

名称	描述
peerID(NSString)	离开房间的用户ID

## 自己被踢出房间回调

- (void)roomManagerKickedOut:(NSDictionary \*)reason;

名称	描述
reason	被踢出房间的原因

## 用户属性发生变化回调

- (void)roomManagerUserPropertyChanged:(NSString \*)peerID  
properties:(NSDictionary \*)properties  
fromId:(NSString \*)fromId;

名称	描述
peerID(NSString)	属性改变的用户ID
properties(NSDictionary)	改变的属性
fromId	修改用户属性的消息发送者ID

## 用户视频状态变化回调

- (void)roomManagerPublishStateWithUserID:(NSString \*)peerID publishState:  
(TKPublishState)state;

名称	描述
peerID(NSString)	视频状态改变的用户ID
state	视频状态，枚举值 TKUser_PublishState_NONE：音视频都没 TKUser_PublishState_AUDIOONLY：只有音频 TKUser_PublishState_VIDEOONLY：只有视频 TKUser_PublishState_BOTH：音视频都有 TKUser_PublishState_NONE_ONSTAGE：音视频都没，用户仍在台上的状态（注：特殊状态）

## 收到自定义信令消息-发布信令消息

```
- (void)roomManagerOnRemotePubMsgWithMsgID:(NSString *)msgID
    msgName:(NSString *)msgName
    data:(NSObject *)data
    fromID:(NSString *)fromID
    inList:(BOOL)inlist
    ts:(long)ts;
```

收到用户发布的自定义信令消息。

名称	描述
msgID(NSString)	信令消息ID
msgName(NSString)	信令消息名称
data(NSObject)	信令消息内容数据，可以是NSNumber、NSString、NSDictionary、NSArray数据结构类型
fromID(NSString)	发送此信令消息者的ID
inlist	表示此条信令是否是在自己进入房间之前发布的信令，YES：是， NO：否
ts	时间戳

## 收到自定义信令消息-删去信令消息

```
- (void)roomManagerOnRemoteDelMsgWithMsgID:(NSString *)msgID
    msgName:(NSString *)msgName
    data:(NSObject *)data
    fromID:(NSString *)fromID
    inList:(BOOL)inlist
    ts:(long)ts;
```

收到用户删去的自定义信令消息。

名称	描述
msgID(NSString)	信令消息ID
msgName(NSString)	信令消息名称
data(NSObject)	信令消息内容数据，可以是NSNumber、NSString、NSDictionary、NSArray数据结构类型
fromID(NSString)	发送此信令消息者的ID
inlist	表示此条信令是否是在自己进入房间之前发布的信令，YES：是， NO：否

名称	描述
ts	时间戳

## 收到聊天消息

```
- (void)roomManagerMessageReceived:(NSString *)message
    fromID:(NSString *)peerID
    extension:(NSDictionary *)extension;
```

收到某一个用户发送的聊天消息。

名称	描述
message(NSString)	聊天消息内容，NSString是JSON字符串，获取具体的消息内容需要解析
peerID(NSString)	发送聊天消息者ID
extension	有关于此条消息的扩展消息（比如：发送者昵称，发送者角色，时间戳等）

## 视频数据统计回调

```
- (void)roomManagerOnVideoStatsReport:(NSString *)peerId stats:
(TKVideoStats *)stats;
```

名称	描述
peerId(NSString)	用户id
state	TKVideoStats对象，表示视频统计数据。详见TKRoomDefines.h文件中对TKVideoStats定义。

## 音频数据统计回调

```
- (void)roomManagerOnAudioStatsReport:(NSString *)peerId stats:
(TKAudioStats *)stats;
```

名称	描述
peerId(NSString)	用户id
state	TKAudioStats对象，表示音频统计数据。详见TKRoomDefines.h文件中对TKAudioStats定义

## 用户音频音量大小变化回调

- (void)roomManagerOnAudioVolumeWithPeerID:(NSString \*)peerID volume:(int)volume;

凡在房间中发布了音频的用户音量大小的实时回调。

名称	描述
peerID(NSString)	用户ID
volume	音量大小

## 纯音频与音视频教室切换回调

- (void)roomManagerOnAudioRoomSwitch:(NSString \*)fromId onlyAudio:(BOOL)onlyAudio;

名称	描述
fromId(NSString)	切换房间模式的用户ID
onlyAudio	是否是纯音频房间 YES：表示纯音频房间 NO：表示音视频房间

## 播放用户视频时第一帧回调

- (void)roomManagerOnFirstVideoFrameWithPeerID:(NSString \*)peerID width:(NSInteger)width height:(NSInteger)height mediaType:(TKMediaType)type;

当在播放某用户视频时，渲染第一帧时，会收到此回调。如果在没有unplay某用户的视频，而在此play该用户的视频，不会收到此回调。

名称	描述
peerID(NSString)	用户ID
width	视频宽
height	视频高
type	视频类型

## 播放用户音频时第一帧回调

- (void)roomManagerOnFirstAudioFrameWithPeerID:(NSString \*)peerID

mediaType:(TKMediaType)type;

当在播放某用户音频时，会收到此回调。如果在没有unplay某用户的音频，而在此play该用户的音频，不会收到此回调。

名称	描述
peerID(NSString)	用户ID
type	视频类型

### 网络发生问题回调

- (void)roomManagerReportNetworkProblem;

### 网络环境发生变化回调

- (void)roomManagerReportNetworkChanged;

### 用户发布（分享）的媒体流状态变化回调

- (void)roomManagerOnShareMediaState:(NSString \*)peerId  
state:(TKMediaState)state  
extensionMessage:(NSDictionary \*)message;

收到某一用户发布或者分享的媒体流的回调消息。

名称	描述
perrID(NSString)	发布（分享）媒体流的用户ID
state	发布状态，0:表示停止发布，1:表示发布
message	扩展信息

### 更新媒体流信息的回调

- (void)roomManagerUpdateMediaStream:(NSTimeInterval)duration  
pos:(NSTimeInterval)pos  
isPlay:(BOOL)isPlay;

会实时收到媒体流的播放状态、播放进度、总时长等信息。

名称	描述
duration	媒体流总时长
pos	媒体流当前播放的进度

名称	描述
isPlay	播放状态； YES：播放 NO：未播放

## 媒体流第一帧回调

- (void)roomManagerMediaLoaded;  
如果是视频流，回收到第一帧画面的回调。

## 桌面共享状态变化的回调

- (void)roomManagerOnShareScreenState:(NSString \*)peerId  
state:(TKMediaState)state  
extensionMessage:(NSDictionary \*)message;  
收到某一用户发布或者分享的桌面共享的回调消息。

名称	描述
perrID(NSString)	发布（分享）桌面共享的用户ID
state	发布状态，0:表示停止发布，1:表示发布
message	扩展信息

## 电影共享状态变化的回调

- (void)roomManagerOnShareFileState:(NSString \*)peerId  
state:(TKMediaState)state  
extensionMessage:(NSDictionary \*)message;  
收到某一用户发布或者分享的电影共享的回调消息。

名称	描述
perrID(NSString)	发布（分享）电影共享的用户ID
state	发布状态，0:表示停止发布，1:表示发布
message	扩展信息



## TKMediaFrameDelegate 音视频数据实时回调

注：此部分回调函数均是线程不安全的。

### 本地采集的音频数据回调

```
- (void)onCaptureAudioFrame:(TKAudioFrame *)frame sourceType:(TKMediaType)type;
```

名称	描述
frame	音频数据结构，TKAudioFrame，详见：TKRoomDefines.h文件中TKAudioFrame。
type	采集源类型

### 本地采集的视频数据回调

```
- (void)onCaptureVideoFrame:(TKVideoFrame *)frame sourceType:(TKMediaType)type;
```

名称	描述
frame	视频数据结构，TKVideoFrame，详见：TKRoomDefines.h文件中TKVideoFrame。
type	采集源类型

### 收到远端音频数据回调

```
- (void)onRenderAudioFrame:(TKAudioFrame *)frame uid:(NSString *)peerId sourceType:(TKMediaType)type;
```

名称	描述
frame	音频数据结构，TKAudioFrame，详见：TKRoomDefines.h文件中TKAudioFrame。
peerId(NSString)	远端用户的ID
type	采集源类型

## 收到远端视频数据回调

- (void)onRenderVideoFrame:(TKVideoFrame \*)frame uid:(NSString \*)peerId  
sourceType:(TKMediaType)type;

名称	描述
frame	视频数据结构, TKVideoFrame, 详见: TKRoomDefines.h文件中TKVideoFrame。
peerId(NSString)	远端用户的ID
type	采集源类型

## 4.5、TKRoomUser 用户对象类

TKRoomUser类定义了关于房间用户对象的相关属性, 完全支持自定义用户属性。可以通过该类的properties属性中添加需要自定义的用户属性, 就可以实现自定义用户属性。

## 4.6、TKRoomDefines.h文件定义说明

### TKRoomWarningCode 房间警告码

名称	描述
TKRoomWarning_UnKnow	0
TKRoomWarning_Microphone_NotWorking	麦克风不可用, 111
TKRoomWarning_Micphone_InterruptionBegan	麦克风开始被打断, 112
TKRoomWarning_Micphone_InterruptionEnded	麦克风打断结束, 113
TKRoomWarning_AudioRouteChange_Headphones	声音输出耳机模式, 121
TKRoomWarning_AudioRouteChange_BuiltInReceiver	声音输出听筒模式 (只有手机端有内置听筒), 122
TKRoomWarning_AudioRouteChange_BuiltInSpeaker	声音输出内置扬声器模式, 123
TKRoomWarning_AudioRouteChange_Bluetooth	声音输出蓝牙模式, 124
TKRoomWarning_RequestAccessForVideo_Failed	请求获取摄像头权限失败, 131
TKRoomWarning_RequestAccessForAudio_Failed	请求获取麦克风权限失败, 132
TKRoomWarning_CheckRoom_Success	checkRoom成功, 5001
TKRoomWarning_ReConnectSocket_ServerChanged	切换服务器, 5002

## TKRoomErrorCode 房间错误码

方法调用失败是返回的错误码如下：

名称	描述
TKErrorCode_UnKnow	-2
TKErrorCode_Internal_Exception	初始化异常， -1
TKErrorCode_OK	OK， 0
TKErrorCode_Not_Initialized	没有初始化， 101
TKErrorCode_Bad_Parameters	参数错误， 102
TKErrorCode_Room_StateError	房间状态错误， 103
TKErrorCode_Publish_StateError	发布状态错误， 104
TKErrorCode_Stream_StateError	流状态错误， 105
TKErrorCode_Stream_NotFound	未知流， 106

房间出现推流相关的错误码如下：

名称	描述
TKErrorCode_Publish_NoAck	推流没有ack， 401
TKErrorCode_Publish_RoomNotExist	房间不存在， 402
TKErrorCode_Publish_RoomMaxVideoLimited	房间最大视频路数限制， 403
TKErrorCode_Publish_ErizoJs_Timeout	socket超时， 404
TKErrorCode_Publish_Agent_Timeout	代理超时， 405
TKErrorCode_Publish_UndefinedRPC_Timeout	未定义的RPC超时， 406
TKErrorCode_Publish_AddingInput_Error	添加输入错误， 407
TKErrorCode_Publish_DuplicatedExtensionId	重复的用户ID， 408
TKErrorCode_Publish_Unauthorized	未授权， 409

房间出现订阅相关的错误码如下：

名称	描述
TKErrorCode_Subscribe_RoomNotExist	房间不存在， 501
TKErrorCode_Subscribe_StreamNotDefine	流不存在， 502
TKErrorCode_Subscribe_MediaRPC_Timeout	请求超时， 503

名称	描述
TKErrorCode_SubscribeStreamFail	订阅失败，504

CheckRoom失败相关的错误码如下：

名称	描述
TKErrorCode_ConnectSocketError	连接socket错误，601
TKErrorCode_JoinRoom_WrongParam	Joinroom参数错误，701
TKErrorCode_CheckRoom_ServerOverdue	checkRoom服务器过期，3001
TKErrorCode_CheckRoom_RoomFreeze	公司被冻结，3002
TKErrorCode_CheckRoom_RoomDeleteOrOverdue	房间被删除或过期，3003
TKErrorCode_CheckRoom_RequestFailed	checkRoom请求失败，4001
TKErrorCode_GetConfig_RequestFailed	getConfig请求失败，4002
TKErrorCode_CheckRoom_RoomNonExistent	房间不存在，4007
TKErrorCode_CheckRoom_PasswordError	房间密码错误，4008
TKErrorCode_CheckRoom_WrongPasswordForRole	密码与身份不符，4012
TKErrorCode_CheckRoom_NeedPassword	该房间需要密码，4110
TKErrorCode_CheckRoom_RoomAuthenError	房间认证错误，4109
TKErrorCode_CheckRoom_RoomPointOverrun	企业点数超限，4112
TKErrorCode_CheckRoom_RoomNumberOverRun	房间人数超限，4103

## 5、FAQ

1、Q：SDK支持哪些CPU指令集？

A：TalkCould iOS SDK目前可以支持arm64、armv7、i386、x86\_64；

2、Q：SDK是静态库还是动态库？

A：TalkCould iOS SDK是动态库；

3、Q：SDK最低支持iOS版本？

A：SDK最低支持到iOS 8.0。