## 视频会议

# 登录

## 1.1登录UC服务器

### 1.1.1 业务场景序图



### 1.1.2 业务场景流程描述

1、用户A启动客户端程序

2、客户端调用tup\_im\_init初始化IM组件

3、客户端调用tup\_im\_setserveraddress接口，设置登录服务器IP和端口

4、客户端调用tup\_im\_register\_callback接口，设置回调函数地址

5、客户端调用tup\_im\_setdispatchmessage接口，传入TUP\_TRUE参数，使回调可用

6、客户端调用tup\_im\_login接口，传入参数由账户和密码组成，发起登录请求

7、登录成功则启动一个大约30秒的定时器，调用tup\_im\_sendheartbeat，向UC服务器发送心跳消息

### 1.1.3 接口列表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块/组件 | 接口类型 | 接口名称 | 接口描述 |
| 客户端程序-> eSDK UC | DLL | tup\_im\_init | 初始化 |
| tup\_im\_setserveraddress | 设置登录服务器IP和端口 |
| tup\_im\_register\_callback | 设置回调函数地址 |
| tup\_im\_setdispatchmessage | 设置消息分发，使得sdk消息能够通知到应用程序回调函数 |
| tup\_im\_login | 登录 |
| tup\_im\_sendheartbeat | 发送心跳消息 |

## 1.2登录Call服务器

### 1.1.1 业务场景序图



### 1.1.2 业务场景流程描述

1、客户端登录UC服务器成功，调用tup\_im\_getserviceprofile接口获取业务参数接口，客户端在业务数据中获取登录的呼叫服务器地址、端口、SIP账号、SIP密码

2、客户端调用tup\_call\_register\_process\_notify接口，设置回调函数

3、调用tup\_call\_log\_start和tup\_call\_init，注意tup\_call\_log\_start接口先调用

4、客户端调用tup\_call\_set\_cfg，此接口需要被调用多次，分别设置传输模式、呼叫注册服务器、呼叫代理服务器（两个服务器一般相同）、用户代理（为固定字符串“Huawei SoftCoPCA”）、本地地址。（注意：如果本地有多个IP，需要使用和服务器互通的IP）

5、调用tup\_call\_register向Call服务器发送注册请求

### 1.1.3 接口列表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块/组件 | 接口类型 | 接口名称 | 接口描述 |
| 客户端程序-> eSDK UC | DLL | tup\_im\_getserviceprofile | 获取业务参数 |
| tup\_call\_register\_process\_notifiy | 设置注册回调函数 |
| tup\_call\_log\_start | 启动呼叫日志 |
| tup\_call\_init | 初始化呼叫模块 |
| tup\_call\_set\_cfg | 设置呼叫参数 |
| tup\_call\_register | 注册呼叫服务器 |
| eSDK UC-->客户端程序 | DLL(回调) | CALL\_E\_EVT\_SIPACCOUNT\_INFO | 注册结果回调消息 |
| eSDK UC-->客户端程序 | DLL(回调) | IM\_E\_EVENT\_IM\_KICKOUT-NOTIFY | 用户被踢下线消息通知 |

# 用户被踢下线

### 2.1 流程描述

账号被他人登录时，客户端会收到用户被踢通知IM\_E\_EVENT\_IM\_KICKOUT\_NOTIFY，调用相应处理函数

### 2.2 接口列表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块/组件 | 接口类型 | 接口名称 | 接口描述 |
| eSDK UC-->客户端程序 | DLL(回调) | IM\_E\_EVENT\_IM\_KICKOUT-NOTIFY | 用户被踢下线消息通知 |

# 会议开始/结束

## 3.1 会议开始

### 3.1.1 业务场景序图

### 

### 3.1.2 业务流程描述

### 1、用户A、B、C均登录客户端，用户A选择用户B和C启动语音会议

### 2、用户A调用tup\_call\_serverconf\_create接口向服务器发送创建会议接口，创建成功，服务器向客户端回调通知创建会议成功消息，回调消息ID为CALL\_E\_EVT\_SERVERCONF\_CREATE\_RESULT

### 3、Call服务器向客户端发送加入会议成功消息，ID为CALL\_E\_EVT\_SERVERCONF\_CONNECT\_RESULT，Call服务器通知主席加入会议成功

### 4、A客户端调用tup\_im\_queryinfo接口查询用户B和C的绑定好，调用tup\_call\_serverconf\_add\_attende发送增加会议成员请求，Call服务器向用户B和C客户端发送会议来电消息，消息ID为CALL\_E\_EVT\_SERVERCONF\_INCOMMING

### 5、用户B调用tup\_call\_serverconf\_accept接口接听会议来电，Call服务器向A发送B入会消息，消息ID为CALL\_E\_EVT\_SERVERCONF\_ATTENDEEJOINED\_SUCCESS，A收到B的入会成功消息，客户端维护会议的成员列表

### 6、用户C收到来电消息，加入会议流程与B相同

### 7、Call服务器向已经入会的用户B和A同时推送消息，消息ID为CALL\_E\_EVT\_SERVERCONF\_REFRESHLIST,客户端C解析出CALL\_S\_CONF\_MEMBER结构内容，展示所有入会成员

### 3.2.3 接口列表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块/组件 | 接口类型 | 接口名称 | 接口描述 |
| 客户端程序-> eSDK UC | DLL | tup\_call\_serverconf\_create | 创建会议 |
| DLL | tup\_call\_serverconf\_add\_attendee | 增加成员 |
| DLL | tup\_call\_serverconf\_accept | 接听会议来电 |
| eSDK UC-->客户端程序 | DLL(回调) | CALL\_E\_EVT\_SERVERCONF\_CREATE\_RESULT | 会议创建成功 |
| CALL\_E\_EVT\_SERVERCONF\_ATTENDEEJOINED\_SUCCESS | 增加成员成功 |
| CALL\_E\_EVT\_SERVERCONF\_CONNECT\_RESULT | 主席入会成功 |
| CALL\_E\_EVT\_SERVERCONF\_INCOMING | 会议来电 |
| CALL\_E\_EVT\_SERVERCONF\_REFRESHLIST | 成员刷新 |

## 3.2 启动语音会议转多媒体会议

### 3.2.1 业务流程时序图



### 3.2.2 业务流程描述

1、主席、与会者1和与会者2均已登录客户端，主席选择与会者1、与会者2召开语音会议，主席调用tup\_call\_serverconf\_transferto\_dataconf接口将语音会议转成多媒体会议

2、服务器内部创建好多媒体会议，向主席、与会者1、与会者2发送多媒体会议的入会信息，消息ID为CALL\_E\_EVT\_SERVERCONF\_DATACONF\_PARAM,对应的消息结构体为CALL\_S\_DATACONF\_PARAM

3、主席、与会者1、与会者2分别调用接口tup\_conf\_new在本地创建出会议，得到会议句柄handle，然后调用接口tup\_conf\_join，传入handle加入会议。服务器收到主席、与会者1、与会者2的入会请求，处理后，向三方分别发送会议通知ID，消息ID为CONF\_MSG\_ON\_CONFERENCE\_JOIN

4、主席、与会者1、与会者2根据CONF\_MSG\_ON\_REFERENCE\_JOIN消息中的福袋消息，判断自己是否成功入会，成功入会则调用接口tup\_conf\_load\_component加载组件，调用接口tup\_conf\_reg\_component\_callback设置共享业务的回调函数，注意：组件加载成功时，会上报CONF\_MSG\_ON\_COMPONENT\_LOAD消息，告知客户端组件是否加载成功

5、主席、与会者1、与会者2分别会收到服务器下发的其他用户加入通知消息ID，消息ID为CONF\_MSG\_USER\_ON\_ENTERN\_IND，对应的消息结构体为TC\_Conf\_User\_Recourd，此时主席、与会者1、与会者2均已成功加入多媒体会议

### 3.2.3 接口列表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块/组件 | 接口类型 | 接口名称 | 接口描述 |
| 客户端程序-> eSDK UC | DLL | tup\_call\_serverconf\_transferto\_dataconf | 语音会议转多媒体会议 |
| DLL | tup\_conf\_new | 创建会议句柄 |
| DLL | tup\_conf\_join | 加入会议 |
| DLL | tup\_conf\_load\_component | 加载业务组件 |
| DLL | tup\_conf\_reg\_component\_callback | 注册业务组件回调函数 |
| eSDK UC-->客户端程序 | DLL(回调) | CALL\_E\_EVT\_SERVERCONF\_DATACONF\_PARAM | 多媒体会议参数通知消息 |
| CONF\_MSG\_ON\_CONFERENCE\_JOIN | 加入会议结果通知 |
| CONF\_MSG\_ON\_COMPONENT\_LOAD | 组件加载结果通知 |
| CONF\_MSG\_USER\_ON\_ENTER\_IND | 与会者刷新通知 |

## 3.3 结束会议

### 3.3.1 结束多媒体会议

#### 3.3.1.1 业务流程时序图



#### 3.3.1.2 业务场景流程

1、主席调用接口tup\_conf\_terminate结束会议（普通用户离开会议调用tup\_call\_conf\_leave接口），服务器通知与会者1、与会者2多媒体会议结束消息，消息ID为CONF\_MSG\_ON\_CONFERENCE\_TERMINATE

2、所有与会者均调用释放会议接口tup\_conf\_release接口

#### 3.3.1.3 接口列表

|  |  |  |  |
| --- | --- | --- | --- |
| 客户端程序-> eSDK UC | DLL | tup\_call\_serverconf\_end | 结束会议，主席调用 |
| DLL | tup\_call\_serverconf\_leave | 离开会议，非主席成员调用 |
| eSDK UC-->客户端程序 | DLL(回调) | CALL\_E\_EVT\_SERVERCONF\_ENDED | 会议结束通知消息 |

### 3.3.2 结束语音会议

#### 3.3.2.1 业务流程时序图

#### 3.3.2.2 业务流程描述

1. 主席A调用tup\_call\_serverconf\_end结束会议
2. 服务器向其他与会成员发送语音会议结束通知消息，消息ID为CALL\_E\_EVT\_SERVERCONF\_ENDED

3.3.2.3 接口列表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块/组件 | 接口类型 | 接口名称 | 接口描述 |
| 客户端程序-> eSDK UC | DLL | tup\_call\_serverconf\_end | 结束会议，主席调用 |
| DLL | tup\_call\_serverconf\_leave | 离开会议，非主席成员调用 |
| eSDK UC-->客户端程序 | DLL(回调) | CALL\_E\_EVT\_SERVERCONF\_ENDED | 会议结束通知消息 |

# 4、增加/删除会议成员

## 4.1 添加会议成员

调用tup\_call\_serverconf\_add\_attendee接口发送增加会议成员请求

## 4.2 删除会议成员

调用tup\_conf\_user\_kickout和tup\_call\_serverconf\_kick\_attendee接口发送删除会议成员请求

# 5、接听/拒绝会议来电

## 5.1 接听会议来电

调用tup\_call\_serverconf\_accept接口接听会议来电

## 5.2 拒绝会议来电

直接单击取消即可

# 6、获取视频设备

用户已加入多媒体会议，调用获取视频设备信息接口tup\_conf\_video\_get\_deviceinfo获取视频设备的序列号

# 7、开启/关闭视频

## 7.1 开启视频

### 7.1.1 业务流程时序图



### 7.1.2 业务场景流程描述

1、主席调用开始视频接口tup\_conf\_video\_open，服务器向主席、与会者1、与会者2发送视频开始消息，ID为COMPT\_MSG\_VIDEO\_ON\_SWITCH，主席、与会者1、与会者2调用视频窗口设置接口，将主席视频数据显示在某个窗口上

2、主席调用接口tup\_conf\_video\_notify要求与会者1

与会者2打开视频

3、服务器向与会者1、其他与会者发送发送打开视频消息，ID为COMPT\_MSG\_VIDEO\_ON\_NOTIFY，假设与会者2无视频，与会者不打开视频

4、与会者1调用接口开启视频tup\_conf\_video\_open接口，服务器向主席、与会者1、与会者2发送与会者1视频开启信息，ID为COMPT\_MSG\_VIDEO\_ON\_SWITCH，主席、与会者1、与会者2调用视频窗口设置接口，将与会者1的视频数据显示在某个窗口上

### 7.1.3 接口列表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块/组件 | 接口类型 | 接口名称 | 接口描述 |
| 客户端程序-> eSDK UC | DLL | tup\_conf\_video\_get\_deviceinfo | 获取本地视频信息 |
| DLL | tup\_conf\_video\_open | 打开本地视频 |
| DLL | tup\_conf\_video\_notify | 通知其他与会者开启视频或者关闭视频 |
| eSDK UC-->客户端程序 | DLL(回调) | COMPT\_MSG\_VIDEO\_ON\_SWITCH | 视频开启或者关闭通知 |
| COMPT\_MSG\_VIDEO\_ON\_NOTIFY | 要求开启火或者关闭视频 |

## 7.2 关闭视频

### 7.2.1 业务流程时序图



### 7.2.2 业务流程描述

1、主席调用接口tup\_conf\_video\_close关闭主席视频，服务器通知与会者1、与会者2主席视频关闭，消息ID为COMPT\_MSG\_VIDEO\_ON\_SWITCH,主席、与会者1、与会者2的界面刷新，停止显示主席视频

2、主席调用接口tup\_conf\_video\_notify通知与会者1关闭视频，与会者1收到视频关闭请求消息，ID为COMPT\_MSG\_VIDEO\_ON\_NOTIFY

3、与会者1调用tup\_conf\_video\_close接口关闭与会者1视频，三方收到回调消息后刷新界面，停止显示与会者1的视频

### 7.2.3 接口列表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块/组件 | 接口类型 | 接口名称 | 接口描述 |
| 客户端程序-> eSDK UC | DLL | tup\_conf\_video\_notify | 通知其他与会者开启视频或者关闭视频 |
| DLL | tup\_conf\_video\_open | 打开本地视频 |
| eSDK UC-->客户端程序 | DLL(回调) | COMPT\_MSG\_VIDEO\_ON\_SWITCH | 视频开启或者关闭通知 |
| COMPT\_MSG\_VIDEO\_ON\_NOTIFY | 要求开启火或者关闭视频 |

# 8、调整视频显示模式

全景会场转换成展示模式时，先调用tup\_conf\_video\_detach接口将视频与全景会场下的窗口进行解绑，再调用tup\_conf\_video\_attach接口将视频与展示模式下的窗口进行绑定。

# 9、控制会场视频布局（仅主席有权限）

主席切换模式时调用接口tup\_conf\_update\_layout接口向所有与会者发送更新会议布局请求，消息ID为CONF\_MSG\_ON\_LAYOUT\_UPDATE，与会者再根据相应模式进行视频的解绑和绑定即可

# 10、登出

判断会议是否结束，若没结束，则先结束会议，参考3.3