

内置编解码客户端 SDK(C 版)说明(V1.0.0)

www.mediapro.cc

一、基本概念

房间：服务器上一个虚拟的概念，进入同一个房间的客户端才允许数据交互，不同房间之间的客户端并无交集。一个房间内客户端的最大数目理论不限，具体根据服务器端设置。服务器允许最大的房间数目同样依据服务端设置而定。最简单的模型是一个房间内一个发布者，一个接收者。

客户端 UID：用于标识每个客户端的唯一 ID，在系统中不允许有相同 UID 的客户端同时在线，否则新登录的客户端会将之前的客户端“顶下去”。

音视频收发者：具备上行音视频到服务器（房间）能力的客户端类型，同时也能接收来自服务器（房间）的音视频数据。

音视频发送者：只上行音视频到服务器（房间）的客户端类型，不具备下行能力。

音视频接收者：只从服务器（房间）接收音视频的客户端类型，不具备上行能力，比如直播业务中的普通观众。

音视频位置：一个房间内允许同时多个音视频发布者，它们发布音视频到不同的“位置”上，接收者可选择接收某个或某些位置的音视频。目前单个房间允许 32 个“位置”。发布者登录后可以选择由服务器自动分配空闲的“位置”供其发布，也可以指定“位置”发布，若当前该“位置”上已有其他发布者，后者将顶替前者以保证同一位置上同时只有一个发布者。

音视频接收掩码：客户端可以根据业务需要选择接收房间内某个或某几个位置的音频或视频数据，可以通过设置自己期望接收的掩码到服务器。音频和视频使用各自独立的掩码，每个掩码 32bit 对应房间内的 32 个位置。

传输参数：本文传输参数是指视频通道的 FEC 上行冗余度、上行 FEC Group 组大小、接收 Qos 丢包等待时间，音频通道在内部已经根据经验数据配置好了合适的值且不对外开放。对于纯接收端（纯下行）同样也可以设置 FEC 上行冗余度、上行 FEC Group 组大小，内部将忽略该参数。

二、API 接口

所有 API 接口定义均位于 `SDTerminalSdk.h` 文件中。本 API 主要实现音视频编码、网络收发、音视频解码三大功能，同时提供了上下行码流的 TS 录制、统计信息获取等辅助功能。

1、系统环境初始化，仅需调用一次

```
void SDDTerminal_Enviroment_Init(const char * outputPath, int outputLevel)
```

参数：

@param: outputPath: 日志文件输出的目录，若目录不存在，SDK 将自动创建，支持相对路径或绝对路径。日志对于问题定位非常重要，建议开启。

@param: outputLevel: 日志输出的级别，只有等于或者高于该级别的日志会输出到文件，日志级别有：DEBUG、INFO、WARNING、ERROR、ALARM、FATAL、NONE，当指定为 NONE 时，将不会生成日志文件。具体见 `TERMINAL_LOG_OUTPUT_LEVEL` 定义。

2、系统退出时调用一次反初始化

```
void SDDTerminal_Enviroment_Free ()
```

3、创建客户端 SDK 对象

```
void* SDDTerminal_New(TerminalEncodeParams* ptEncodeParams, RemoteVideoYuvDataCallback  
pfOutputVideo, RemoteAudioPcmDataCallback pfOutputAudio, SystemStatusCallback pfSysStatus,  
void* pObject);
```

参数：

@ ptEncodeParams, 音视频编码相关参数，包括码率、帧率、分辨率、采样率、声道数等等。

@ pfOutputVideo, 视频接收解码输出回调函数。

@ pfOutputAudio, 音频接收解码输出回调函数。

@ pfSysStatus, 系统状态输出回调函数。

@ pObject, 输出回调函数透传数据。

返回值：

对象指针，返回 NULL 表示失败。

4、销毁客户端 SDK 对象

```
void SDTerminal_Delete(void** ppTerminal);
```

参数：

@ ppTerminal, 模块指针的指针

说明：使用者应该做好与其他 API 之间的互斥保护

返回值：无

5、登录服务器

```
BOOL SDTerminal_Online(  
    void* pTerminal,  
    TerminalLogonParams* ptLogonParams,  
    TerminalTransParams* ptTransParams);
```

参数：

@ pTerminal, 模块指针

@ ptLogonParams, 登录服务器相关参数, 包括 UID、房间号、上行发送位置、下行接收掩码等。

@ ptTransParams, 音视频传输相关参数, 包括 FEC 相关、接收缓存相关等。

说明：同步登录模式下, 本 API 为阻塞式, 登录失败的超时时间较长, 请避免在 UI 线程中调用。

说明：通过设置音视频下行接收掩码, 可以选择从服务器接收哪几个位置的音视频数据。每一个 bit 对应一个位置, 最低位对应 0 号位置, 最高位对应 31 号位置。比如希望接收某个 index 位置的音视频时, 可以设置其对应 bit 设置为 1, 否则设置为 0。

允许接收某个 index 位置的音视频时, 可以设置为:

```
UINT unMask = 0x1 << (index);  
unAudioDownChannelsMask |= unMask;  
unVideoDownChannelsMask |= unMask;
```

当本客户端仅发送音视频, 不接收音视频时, 设置音视频接收掩码全为 0 即可。

返回值：

返回 TRUE 表示成功, 返回 FALSE 则为失败。

6、下线服务器

```
void SDTerminal_Offline(void* pTerminal);
```

参数：

@ pTerminal, 模块指针

说明： 下线服务器，若此时用户在音视频位置上，将同时从位置上下来。

返回值： 无

7、编码并发送一帧视频数据

```
void SDTerminal_SendVideoData(void* pTerminal, unsigned char* buf, unsigned int unWidth,  
unsigned int unHeight, TERMINAL_VIDEO_INPUT_FORMAT eFormat);
```

参数：

@ pTerminal, 模块指针

@ buf, 输入图像存放区。

@ unWidth, 输入图像宽度，当与编码宽高不一致时，内部自行缩放。建议与编码宽高一致。

@ unHeight, 输入图像高度，当与编码宽高不一致时，内部自行缩放。建议与编码宽高一致。

@ eFormat, 输入图像格式，建议与编码器当前使用的格式一致。

说明： 外层可先通过 SDTerminal_GetHardwareEncInfo 接口获得当前硬编码支持的输入格式，以此作为外层采集格式。这样避免内部再进行色度空间转换、缩放处理，提高性能。

返回值： 无

8、编码并发送一帧音频数据

```
void SDTerminal_SendAudioData(void* pTerminal, unsigned char* buf, unsigned int unLen);
```

参数：

@pTerminal, 模块指针

@ buf, 输入音频 PCM 存放区。

@ unLen, 输入音频 PCM 字节数。

返回值： 无

9、获取硬编码支持情况

```
BOOL SDTerminal_GetHardwareEncInfo(void* pTerminal, BOOL* pbWillUseHwEnc,  
TERMINAL_VIDEO_INPUT_FORMAT *peHwRecomInputFormat);
```

参数：

@ pTerminal, 模块指针

@ pbWillUseHwEnc, 是否将最终采用硬编码。用户在 Online 接口中启用硬编码, 当最终是否采用将取决于当前机器设备是否支持, 不支持时将自动使用软编码。

@ peHwRecomInputFormat, 若最终采用硬编码, 返回硬编码器支持的输入格式。

说明：需在 Online 调用成功后调用生效。

返回值：返回 TRUE 表示成功, 返回 FALSE 则为失败。

10、设置上行码流录制

```
BOOL SDTerminal_StartCapUpStream(void* pTerminal, const char* strTsFileDir);
```

```
void SDTerminal_StopCapUpStream(void* pTerminal);
```

参数：

@ pTerminal, 模块指针

@ strTsFileDir, 上行录制 TS 文件名, 需带完整路径。

说明：SDTerminal_StartCapUpStream 需在 Online 接口成功后调用生效。

返回值：返回 TRUE 表示成功, 返回 FALSE 则为失败。

11、设置下行码流录制

```
BOOL SDTerminal_StartCapDownStream(void* pTerminal, unsigned int unAvDownIndex, const char* strTsFileDir);
```

```
void SDTerminal_StopCapDownStream(void* pTerminal, unsigned int unAvDownIndex);
```

参数：

@ pTerminal, 模块指针

@ unAvDownIndex, 需要录制的下行位置。

@ strTsFileDir, 下行录制 TS 文件名, 需带完整路径。

说明：SDTerminal_StartCapDownStream 需在 Online 接口成功后调用生效。

返回值：返回 TRUE 表示成功, 返回 FALSE 则为失败。

12、获取当前 SDK 版本信息

```
unsigned int SDTerminal_GetVersion (void* pTerminal);
```

参数:

@ pTerminal, 模块指针

返回值: 获得当前 SDK 的版本信息

13、获取当前丢包率数据

```
void SDTerminal_GetVideoAudioUpDownLostRatio(void* pTerminal, float *pfVideoUpLostRatio,  
float *pfVideoDownLostRatio, float *pfAudioUpLostRatio, float *pfAudioDownLostRatio);
```

参数:

@pTerminal, 模块指针

@pfVideoUpLostRatio, 获取视频上行丢包率

@pfVideoDownLostRatio, 获取视频下行丢包率

@pfAudioUpLostRatio, 获取音频上行丢包率

@pfAudioDownLostRatio, 获取音频下行丢包率

返回值: 内部已经乘 100 得到百分比

14、获取当前视频通道的实时 RTT

```
unsigned int SDTerminal_GetCurrentRtt(void* pTerminal);
```

@pTerminal, 模块指针

返回值: 获得当前视频通道的实时 RTT 值