

libEasyPlayer RTSP 播放器库 API 接口说明

概述

libEasyPlayer 实现对 RTSP 直播流进行实时采集和解码显示，稳定，高效，低延时；解码可采用 intel 硬件解码和软件解码两种方式，能实时进行录像和快照抓图，OSD 叠加等功能。

API 接口函数定义

➤ int EasyPlayer_Init();

函数说明：

播放器初始化，播放器使用之前调用；

参数说明：

➤ void EasyPlayer_Release();

函数说明：

播放器资源释放，播放器不再使用以后调用；

参数说明：

➤ int EasyPlayer_OpenStream(const char *url, HWND hWnd, RENDER_FORMAT renderFormat, int rtpovertcp, const char *username, const char *password, MediaSourceCallBack callback, void *userPtr, bool bHardDecode);

函数说明：

播放器开始进行流播放；返回值为当前播放的通道 ID，该 ID 在停止推流时需要用到；

参数说明：

Url： [IN] 字符串类型，表示当前要播放的流地址， Eg: rtsp://127.0.0.1:554/stream.sdp

HWnd: [IN] 窗口句柄类型，表示为当前播放器将显示的窗口的句柄；

renderFormat: [IN] 播放渲染类型，详见 RENDER_FORMAT 结构；

Rtpovertcp: [IN] 整数型，拉取流的传输模式，0=udp, 1=tcp

Username: [IN] 字符串，访问流的用户名（如果存在）

Password: [IN] 字符串，访问流的用户名（如果存在）

Callback: [IN] 播放器回调音视频数据回调函数

userPtr: [IN] 用户自定义传入数据

bHardDecode: [IN] 是否采用硬件解码 1=是，0=否

➤ void EasyPlayer_CloseStream(int channelId);

函数说明：

播放器停止流播放；

参数说明：

channelId [IN] 当前播放的流通道 ID，该 ID 是 EasyPlayer_OpenStream 函数打开流的返回值；

➤ `int EasyPlayer_SetFrameCache(int channelId, int cache);`

函数说明:

播放器设置当前流播放缓存帧数;

参数说明:

`channelId [IN]` 当前播放的流通道 ID , 该 ID 是 `EasyPlayer_OpenStream ()` 函数打开 流的返回值;

`cache [IN]` 当前通道的流播放设置的缓存的视频帧数, Eg: 缓存 10 帧, 则 `cache = 10;`

➤ `int EasyPlayer_SetShownToScale(int channelId, int shownToScale);`

函数说明:

播放器按比例进行显示;

参数说明:

`channelId [IN]` 当前播放的流通道 ID , 该 ID 是 `EasyPlayer_OpenStream ()` 函数打开 流的返回值;

`shownToScale [IN]` 0=整个窗口区域显示, 1=按比例显示;

➤ `int EasyPlayer_SetDecodeType(int channelId, int decodeKeyframeOnly);`<不常用>

函数说明:

播放器设置解码类型; 分为所有帧解码和只解码关键帧;

参数说明:

channelId [IN] 当前播放的流通道 ID，该 ID 是 EasyPlayer_OpenStream（）函数打开流的返回值；

decodeKeyframeOnly [IN] 0=所有帧解码，1=只解码关键帧；

➤ int EasyPlayer_SetRenderRect(int channelId, LPRECT lpSrcRect);<不常用>

函数说明：设置视频显示时渲染区域

参数说明：

channelId [IN] 当前播放的流通道 ID，该 ID 是 EasyPlayer_OpenStream 函数打开流的返回值；

lpSrcRect [IN] 矩形框结构体指针，指向设置渲染区域的矩形结构体；

➤ int EasyPlayer_ShowStatisticalInfo(int channelId, int show);

函数说明：

播放器设置是否显示码流信息；

参数说明：

channelId [IN] 当前播放的流通道 ID，该 ID 是 EasyPlayer_OpenStream（）函数打开流的返回值；

Show [IN] 0=不显示，1=显示；

int EasyPlayer_ShowOSD(int channelId, int show, EASY_PALYER_OSD

osd);

➤ (int channelId, int show);

函数说明:

播放器设置自定义显示 OSD 信息;

参数说明:

channelId [IN] 当前播放的流通道 ID , 该 ID 是 EasyPlayer_OpenStream () 函数打开 流的返回值;

Show [IN] 0=不显示, 1=显示;

Osd [IN] osd 显示信息填充结构, 定义如下:

typedef struct tagEASY_PALYER_OSD

{

char stOSD[128]; //OSD 字幕信息

DWORD alpha; //透明通到 0-255

DWORD color; //RGB(0xf9,0xf9,0xf9)

DWORD shadowcolor; //OSD 背景颜色 RGB(0x4d,0x4d,0x4d)

全为 0 背景透明

RECT rect; //OSD 基于图像右上角显示区域

}EASY_PALYER_OSD;

➤ int EasyPlayer_SetDragStartPoint(int channelId, POINT pt);<不可用>

函数说明:

参数说明：

- `int EasyPlayer_SetDragEndPoint(int channelId, POINT pt);`<不可用>

函数说明：

参数说明：

- `int EasyPlayer_ResetDragPoint(int channelId);`<不可用>

函数说明：

参数说明：

- `int EasyPlayer_StartManuRecording(int channelId);`

函数说明：

播放器开始将流音视频数据进行录制，录制格式为 MP4；

参数说明：

`channelId` [IN] 当前播放的流通道 ID，该 ID 是 `EasyPlayer_OpenStream` 函数打开流的返回值；

- `int EasyPlayer_StopManuRecording(int channelId);`

函数说明：

播放器停止录制 MP4；

参数说明：

`channelId` [IN] 当前播放的流通道 ID，该 ID 是

EasyPlayer_OpenStream 函数打开 流的返回值；

➤ int EasyPlayer_PlaySound(int channelId);

函数说明：

播放器开始播放音频；

参数说明：

channelId [IN] 当前播放的流通道 ID ， 该 ID 是

EasyPlayer_OpenStream 函数打开 流的返回值；

➤ int EasyPlayer_StopSound();

函数说明：

播放器停止播放音频；