**LibEasyPlayerPro API 说明**

***概述***

LibEasyPlayerPro 实现了多种输入源的音视频实时采集及解码显示,可实时调整显示的流畅度, 设置按比例显示视频; 可在视频上实现叠加任意字符;

支持 RTSP、RTMP、HLS、HTTP、外部编码后数据、外部解码后数据输入;

LibEasyPlayerPro 依赖于以下 4 个动态库:

FFSource.dll: 多种输入源 输出完帧的音视频帧

FFDecoder.dll: 音视频解码库 可解码H264,H264, MPEG4, MJPEG, G711(ualw),AAC FFEncoder.dll: 音频编码库 用于本地录像时将非 AAC 格式的音频转为 AAC; D3DRender.dll 视频显示库 支持格式: YV12 YUY2 UYVY ARGB8888 XRGB8888

RGB565 RGB555 RGB24;

**API 功能说明**

|  |  |
| --- | --- |
| 函数名称 | 说明 |
| libEasyPlayerPro\_Initialize | 初始化 SDK 内部参数 |
| libEasyPlayerPro\_Deinitialize | 反初始化 SDK 内部参数 |
| libEasyPlayerPro\_Create | 初始化 SDK 内部参数 |
| libEasyPlayerPro\_Release | 反初始化 SDK 内部参数 |
| libEasyPlayerPro\_OpenStream | 打开一路码流,返回一个可用的通道 ID |

|  |  |
| --- | --- |
| libEasyPlayerPro\_CloseStream | 关闭一路码流 |
| libEasyPlayerPro\_StartPlayStream | 开始播放一路码流 |
| libEasyPlayerPro\_PutFrameData | 外部输入帧数据 |
| libEasyPlayerPro\_ResetFrameQueue | 清空缓冲区帧队列 |
| libEasyPlayerPro\_StopPlayStream | 停止播放 |
| libEasyPlayerPro\_GetStreamInfo | 获取媒体流信息 |
| libEasyPlayerPro\_SetPlayFrameCache | 设置指定通道的缓存帧数(实时优先 or 流畅优先) |
| libEasyPlayerPro\_GetPlayFrameCache | 获取指定通道的缓存帧数 |
| libEasyPlayerPro\_ShowStatisticalInfo | 显示统计信息，如分辨率，帧类型，码率… |
| libEasyPlayerPro\_SetDecodeType | 设置解码帧类型,是否仅解码关键帧 |
| libEasyPlayerPro\_SetVideoFlip | 视频翻转(仅在显示格式为 RGB24 时有效) |
| libEasyPlayerPro\_SetRenderRect | 放大显示指定的视频区域 |
| libEasyPlayerPro\_SetScaleDisplay | 按比例显示视频 |
| libEasyPlayerPro\_SetOverlayText | 设置在视频上叠加文字 |
| libEasyPlayerPro\_ClearOverlayText | 清空叠加文字 |
| libEasyPlayerPro\_SetPlaySpeed | 设置播放速度(仅针对文件有效) |
| libEasyPlayerPro\_GetPlaySpeed | 获取播放速度 |
| libEasyPlayerPro\_StartRecording | 对指定通道进行本地录像(mpg 格式) |
| libEasyPlayerPro\_StopRecording | 停止对指定通道的本地录像 |
| libEasyPlayerPro\_SnapshotToFile | 抓图 |
| libEasyPlayerPro\_SetElectronicZoomStartPoint | 开始电子放大，设置电子放大起始点 |
| libEasyPlayerPro\_SetElectronicZoomEndPoint | 设置电子放大结束点 |
| libEasyPlayerPro\_SetElectronicZoom | 设置是否放大 |
| libEasyPlayerPro\_ResetElectronicZoom | 复位电子放大参数 |
| libEasyPlayerPro\_InstantReplay\_Start | 开始即时回放 |
| libEasyPlayerPro\_InstantReplay\_Pause | 暂停即时回放 |
| libEasyPlayerPro\_InstantReplay\_Resume | 恢复即时回放 |
| libEasyPlayerPro\_InstantReplay\_Stop | 停止即时回放 |
| libEasyPlayerPro\_InstantReplay\_Save | 保存即时回放片断为录像文件 |
| libEasyPlayerPro\_StartPlaySound | 开始播放声音 |
| libEasyPlayerPro\_StopPlaySound | 停止播放声音 |
| libEasyPlayerPro\_SoundPlaying | 获取指定通道是否在播放声音 |
| libEasyPlayerPro\_GetAudioOutputDeviceList | 获取音频输出设备列表 |

|  |  |
| --- | --- |
| libEasyPlayerPro\_SetAudioOutputDeviceId | 设置音频输出设备 ID |
| libEasyPlayerPro\_SetAudioOutputDeviceName | 设置音频输出设备名称 |
| libEasyPlayerPro\_SetAudioVolume | 设置音量 |
| libEasyPlayerPro\_GetAudioVolume | 获取音量 |

**所有 API 的 int 类型返回值, 为 0 则表示成功,否则为失败;**

libEasyPlayerPro 分为两种调用方式:

1. 使用libEasyPlayerPro\_Initialize初始化, 在后面的所有调用中, PLAYER\_HANDLE为NULL,

结束调用libEasyPlayerPro\_Deinitialize

2. 使用libEasyPlayerPro\_Create 创建一个PLAYER\_HANDLE句柄, 在后面的所有调用中, 填上创建的句柄, 结束调用libEasyPlayerPro\_Release

# libEasyPlayerPro\_Initialize

输入: int supportMaxChannel: 按需指定最大通道数,最大不能超过宏定义MAX\_CHANNEL\_NUM 输 出 : int

说明: 初始化SDK内部参数, 以该种方式初始化SDK后, 在所有的API调用中, PLAYER\_HANDLE

都填为 NULL;

**libEasyPlayerPro\_Deinitialize**

输入: 无

输出: int 0 为成功,其它值为失败

说明: 反初始化 SDK 内部参数

**libEasyPlayerPro\_Create**

输 入 : PLAYER\_HANDLE \*handle: 初始化成功则为有效句柄

int supportMaxChannel: 按需指定最大通道数,最大不能超过宏定义MAX\_CHANNEL\_NUM

输 出 : int

说明: 初始化SDK内部参数, 以该种方式初始化SDK后, 在所有的API调用中, PLAYER\_HANDLE

都填为返回的 handle;

**libEasyPlayerPro\_Release**

输入: PLAYER\_HANDLE \*handle: 调用 libEasyPlayerPro\_Create 返回的 handle

输出: int 0 为成功,其它值为失败

说明: 反初始化 SDK 内部参数

**libEasyPlayerPro\_OpenStream**

|  |  |  |  |
| --- | --- | --- | --- |
| 输入: | Handle: 为NULL或有效的playerHandle channelType: 通道源类型 |  | |
|  | EASY\_CHANNEL\_SOURCE\_TYPE\_RTSP = 0, |  | //源为RTSP(内置) |
|  | EASY\_CHANNEL\_SOURCE\_TYPE\_RTMP, |  | //源为RTMP(内置) |
|  | EASY\_CHANNEL\_SOURCE\_TYPE\_HLS, |  | //源为HLS(内置) |
|  | EASY\_CHANNEL\_SOURCE\_TYPE\_FILE, |  | //源为本地文件(内置) |
|  | EASY\_CHANNEL\_SOURCE\_TYPE\_ENCODE\_DATA, |  | //源为外部的编码数据 |
|  | EASY\_CHANNEL\_SOURCE\_TYPE\_DECODE\_DATA, |  | //源为外部的解码数据 |

url: 有效的源地址， 如: rtsp://192.168.1.100 rtmp://192.168.1.100/1

<http://192.168.1.100/1> [file://D:\1.mp4](file://localhost/D:/1.mp4)

rtpOverTcp: 仅针对rtsp有效, 1为tcp,否则为udp mediaType: 媒体类型, 暂保留EasyPlayerProCallBack: 回调函数

userPtr: 用户自定义指针 reconnection: 是否重连 heartbeatType: 心跳类型: 填0

queueSize: 缓冲队列大小, 需大于1024\*512

multiplex: 0x01:复用源,即打开同一个url时，到前端的连接只有一个;

0x00:不复用, 打开多少个url,就有多少个连接\*

输出: int: 如果成功，则返回一个有效的通道号; 如果失败，则返回小于0的值说明: 以指定的连接方式, 打开一路码流, 并将编码数据进行回调;

# libEasyPlayerPro\_CloseStream

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id(libEasyPlayerPro\_OpenStream返回值) 输出: int 0为成功,其它值为失败

说明: 关闭指定的通道

# libEasyPlayerPro\_StartPlayStream

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id(libEasyPlayerPro\_OpenStream返回值) HWND hWnd: 窗口句柄

如果为有效句柄, 则开启解码和显示;

如果为NULL, 则仅解码,不显示; 在回调时会回调解码后的帧数据; RENDER\_FORMAT renderFormat: 显示格式

unsigned char decodeType: 解码类型 -- 暂无效

输出: int 0为成功,其它值为失败说明: 开始播放指定的通道视频

# libEasyPlayerPro\_PutFrameData

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id

int mediaType: 媒体类型 MEDIA\_TYPE\_VIDEO 或 MEDIA\_TYPE\_AUDIO EASY\_FRAME\_INFO \*frameInfo: 帧信息

unsigned char \*pbuf: 编码或解码数据输出: int 0为成功,其它值为失败

说明: 送编码流到指定通道, 需配合libEasyPlayerPro\_OpenStream中的

channelType==EASY\_CHANNEL\_SOURCE\_TYPE\_ENCODE\_DATA或EASY\_CHANNEL\_SOURCE\_TYPE\_DECODE\_DATA

# libEasyPlayerPro\_ResetFrameQueue

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: int 0为成功,其它值为失败

说明: 清空帧队列, 将从下一个收到的关键帧开始播放

# libEasyPlayerPro\_StopPlayStream

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: int 0为成功,其它值为失败 说明: 停止播放指定通道的音视频

# libEasyPlayerPro\_GetStreamInfo

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id

EASY\_MEDIA\_INFO\_T \*pMediaInfo: 媒体信息

输出: int 0为成功,其它值为失败

说明: 获取指定通道的媒体信息. 该函数不能在libEasyPlayerPro\_OpenStream之后立即调用,如果需要尽快的获取媒体信息,则通过回调获取;

# libEasyPlayerPro\_SetPlayFrameCache

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id

int cache: 缓冲帧数 **范围 1-10** 输出: int 0为成功,其它值为失败

说明: 设置播放缓冲帧数, 越小实时性越好，越大越流畅, 可根据实际情况设置;

# libEasyPlayerPro\_GetPlayFrameCache

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: 缓冲帧数

说明: 获取指定通道的播放缓冲帧数

# libEasyPlayerPro\_ShowStatisticalInfo

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id unsigned char show: 显示统计信息

输出: int 0为成功,其它值为失败

说明: 设置是否显示统计信息,包括编码格式、分辨率、码率、当前缓存帧数等

# libEasyPlayerPro\_SetDecodeType

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id

unsigned char decodeKeyframeOnly: 显示仅解码显示关键帧

输出: int 0为成功,其它值为失败

说明: 设置是否仅解码显示关键帧

# libEasyPlayerPro\_SetVideoFlip

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id unsigned char flip: 视频翻转显示

输出: int 0为成功,其它值为失败

说明: 视频翻转显示, 仅限于显示格式为RGB24;

# libEasyPlayerPro\_SetRenderRect

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id LPRECT lpRect: 分辨率坐标点

输出: int 0为成功,其它值为失败

说明: 设置当前视频从指定的坐标开始显示; 用于放大局部细节;

# libEasyPlayerPro\_SetScaleDisplay

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id

unsigned char scaleDisplay: 0x00:铺满窗口 0x01:按比例显示\*

COLORREF bkColor: 背景色

输出: int 0为成功,其它值为失败

说明: 按比例显示视频或铺满窗口; 当按比例显示时, 参数bkColor有效;

# libEasyPlayerPro\_SetOverlayText

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id const char \*text: 设置叠加文字

输出: int 0为成功,其它值为失败

说明: 有窗口上显示自定义字符

# libEasyPlayerPro\_ClearOverlayText

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: int 0为成功,其它值为失败 说明: 清空自定义字符

# 

**libEasyPlayerPro\_SetPlaySpeed**

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id PLAY\_SPEED\_ENUM speed: 速度值

输出: int 0为成功,其它值为失败

说明: 设置速度值,仅针对文件有效, 正常，暂停，1x,2x,4x,8x,1/2,1/4,1/8

# libEasyPlayerPro\_GetPlaySpeed

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id

输出: PLAY\_SPEED\_ENUM speed: 速度值

说明: 获取当前播放速度

# libEasyPlayerPro\_StartRecording

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id

const char \*foldername: 文件夹名称 const char \*filename: 文件名称 unsigned int filesize: 录像文件大小 MB int duration: 录像时长(second)

unsigned char preRecording : 0x01:预录 0x00:不预录

输出: int 0为成功,其它值为失败

说明: 开始本地录像, file和 duration为互斥关系, 当filesize大于0时，手动录像至指定文件大小则自动停止, 如果filesize为0时, 则duration为有效值,录满指定时长后自动停止;

# libEasyPlayerPro\_StopRecording

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: int 0为成功,其它值为失败 说明: 停止本地录像

# libEasyPlayerPro\_SnapshotToFile

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id

unsigned char imageFormat: 图 片 格 式 0:bmp 1:jpg, char \*filename : 文件名

unsigned char sync: 0:异步: 1:同步, unsigned char useQueue: 1:使用队列 0:不使用队列

输出: int 0为成功,其它值为失败

说明: 抓拍图片到指定文件

# libEasyPlayerPro\_SetElectronicZoomStartPoint

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id

float fXPercent: point.x百分比float fYPercent: point.y百分比

unsigned char showBox: 0x01: 显示所画框 0x00:不显示

输出: int 0为成功,其它值为失败

说明: 设置电子放大起始点, fXPercent和fYPercent均为鼠标点相对于窗口的百分比;

# libEasyPlayerPro\_SetElectronicZoomEndPoint

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id

float fXPercent: point.x百分比float fYPercent: point.y百分比

输出: int 0为成功,其它值为失败

说明: 设置电子放大结束点, fXPercent和fYPercent均为鼠标点相对于窗口的百分比;

# libEasyPlayerPro\_SetElectronicZoom

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id

int zoomIn: 1:放大显示 0:不放大

输出: int 0为成功,其它值为失败

说明: 设置电子放大标识

# libEasyPlayerPro\_ResetElectronicZoom

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: int 0为成功,其它值为失败 说明: 还原电子放大

# libEasyPlayerPro\_InstantReplay\_Start

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: int 0为成功,其它值为失败

说明: 开始即时回放; 缓冲区的大小直接影响到即时回放, 缓冲区越大则即时回放时长越长;

还和相机的GOP相关;

# libEasyPlayerPro\_InstantReplay\_Pause

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: int 0为成功,其它值为失败 说明: 暂停即时回放

# libEasyPlayerPro\_InstantReplay\_Resume

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: int 0为成功,其它值为失败 说明: 恢复即时回放

# 

**libEasyPlayerPro\_InstantReplay\_Stop**

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: int 0为成功,其它值为失败 说明: 停止即时回放

# libEasyPlayerPro\_InstantReplay\_Save

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: int 0为成功,其它值为失败

说明: 保存即时回放视频片段为文件;

# 

**libEasyPlayerPro\_StartPlaySound**

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: int 0为成功,其它值为失败

说明: 开始播放指定通道声音;

# libEasyPlayerPro\_StopPlaySound

输入: PLAYER\_HANDLE handle: NULL或有效handle

输出: int 0为成功,其它值为失败

说明: 停止播放声音;

# libEasyPlayerPro\_SoundPlaying

输入: PLAYER\_HANDLE handle: NULL或有效handle

int channelId: 通道id 输出: int 0为播放中,<0非播放中; 说明: 获取当前播放状态;

# 

**libEasyPlayerPro\_GetAudioOutputDeviceList**

输入: PLAYER\_HANDLE handle: NULL或有效handle

MIXER\_DEVICE\_INFO\_T \*\*devicelist: 设备列表 int \*deviceNum: 设备数量

输出: int 0为成功,其它值为失败

说明: 获取音频输出设备列表

# libEasyPlayerPro\_SetAudioOutputDeviceId

输入: PLAYER\_HANDLE handle: NULL或有效handle

int deviceId: 设备id 输出: int 0为成功,其它值为失败

说明: 设置音频播放设备ID, 即使用该设备进行声音播放;

# libEasyPlayerPro\_SetAudioOutputDeviceName

输入: PLAYER\_HANDLE handle: NULL或有效handle

char \*deviceName: 设备名称 输出: int 0为成功,其它值为失败

说明: 设置音频播放设备名称, 即使用该设备进行声音播放

# libEasyPlayerPro\_SetAudioVolume

输入: PLAYER\_HANDLE handle: NULL或有效handle

int volume: 音 量 值 0 - 100 输出: int 0为成功,其它值为失败

说明: 设置音量

# libEasyPlayerPro\_GetAudioVolume

输入: PLAYER\_HANDLE handle: NULL或有效handle

输出: 0 – 100的音量值

说明: 获取音量