

屏幕采集库 (Windows)

一、简介

本库完成 Windows 平台下屏幕或窗口采集功能，当系统连接多屏幕时，可以选择指定屏幕进行采集。当屏幕或窗口分辨率不等于用户指定输出分辨率时，自动进行保宽高比，左右或上下添加黑边，保证输出画面不变形。支持对外输出恒定帧率。因系统性能不足时自动补帧处理，系统性能过高时自动丢帧处理；被采集窗口关闭而无法继续采集时，持续输出最后一帧有效画面。支持对外输出用户指定的分辨率和色度空间，可直通编码器。内部采用高性能 libYUV 缩放和色度空间转换，支持设置缩放算法。

二、API 说明

1、环境初始化与反初始化

```
void SDDesktopCap_Environment_Init (const char* outputPath, int outputLevel);
```

```
void SDDesktopCap_Environment_Free ();
```

参数：

@param: outputPath 表示日志存放路径，支持相对路径和绝对路径，若目录不存在将自动创建

@param: outputLevel 表示日志输出的级别，只有等于或者高于该级别的日志输出到文件，取值范围参考 DESKTOP_CAP_LOG_OUTPUT_LEVEL。

上述 API 仅需在系统中调用一次。

2、创建 SDDesktopCap 对象

```
void* SDDesktopCap_New (const char* outputPath, int outputLevel);
```

参数：

@param: outputPath 表示日志存放路径，支持相对路径和绝对路径，若目录不存在将自动创建

@param: outputLevel 表示日志输出的级别，只有等于或者高于该级别的日志输出到文件，取值范围参考 DESKTOP_CAP_LOG_OUTPUT_LEVEL。

返回值：返回模块指针，为 NULL 则失败

3、销毁 SDDesktopCap 对象

```
void SDDesktopCap_Delete(void** ppScreenCap);
```

说明：使用者应该做好与其他 API 之间的互斥保护，调用本 API 后模块指针将被设置为 NULL。

参数：

@param ppScreenCap: 模块指针指针

4、开始启动 SDDesktopCap

```
BOOL SDDesktopCap_Start(void* pDesktopCap,  
DESKTOP_CAP_SOURCE eCapSource,  
unsigned int unSourceId, BOOL bCapCursor, unsigned int unCapFrameRate,  
unsigned int unCapWidth, unsigned int unCapHeight,  
DESKTOP_CAP_OUT_FORMAT eCapFormat,  
CaptureDeskTopVideoCallback pfCapCallback, void* pObject);
```

参数：

@param pDesktopCap: 模块指针

@param eCapSource: 采集源，屏幕或窗口

@param unSourceId: 采集源 ID，采集屏幕时通过 SDGetScreenDeviceList 获得屏幕 ID 列表，只有一个屏幕时，默认屏幕 ID 为 0。采集窗口时通过 SDGetWindowList 获得窗口 ID 列表。

@param bCapCursor: 窗口采集时，是否采集鼠标。

@param unCapFrameRate: 采集帧率

@param unCapWidth: 采集输出宽度，当源宽高与之不一致时，内部将缩放到输出宽高。

@param unCapHeight: 采集输出高度

@param eCapFormat: 采集输出格式，建议根据编码器输入格式来确定采集输出格式，尽量保持一致，以便直通编码，提高性能。

@param pfCapCallback: 采集输出回调接口，可以在回调中进行编码耗时操作。

@param pObject: 采集输出回调接口透传参数

返回值: TRUE 成功, FALSE 则失败

5、停止 SDDesktopCap

```
void SDDesktopCap_Stop (void* pDesktopCap);
```

参数:

@param pDesktopCap: 模块指针