

# 视频渲染库 (Windows)

## 一、简介

本库完成 Windows 平台下视频渲染，支持传入窗口句柄，支持单个窗口上多路画面渲染。模块输入为 YUV420 数据，当传入 YUV420 宽高比与窗口宽高比不一致时，可以选择通过填黑边保持宽高比或全屏铺满两种方式。

## 二、API 说明

### 1、环境初始化与反初始化

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

```
void SDVideoRender_Enviroment_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 VIDEO\_RENDER\_LOG\_OUTPUT\_LEVEL

上述 API 仅需在系统中调用一次。

### 2、创建 SDVideoRender 对象

```
void* SDVideoRender_New (void* pWindowHandle, BOOL bKeepRatio);
```

**参数：**

@param: pWindowHandle 待渲染画面的窗口句柄。

@param: bKeepRatio 是否通过填黑边方式保持宽高比渲染。

**返回值：**返回模块指针，为 NULL 则失败

### 3、销毁 SDVideoRender 对象

```
void SDVideoRender_Delete (void** ppVideoRender);
```

**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 pVideoRender: 模块指针指针

#### 4、添加待渲染的视频流

```
BOOL SDVideoRender_AddStream(void* pVideoRender, unsigned int unStreamId,  
unsigned int unZOrder, float fLeft, float fTop, float fRight, float fBottom);
```

参数:

@param pVideoRender: 模块指针

@param unStreamId: 视频流 ID, 用户自行维护该 ID 并保证唯一性。

@param unZOrder: 视频流的相对显示层级, 0 表示最上层显示。

@param fLeft: 视频流在画面中的相对位置, 左, 范围[0~1]。

@param fTop: 视频流在画面中的相对位置, 上, 范围[0~1]。

@param fRight: 视频流在画面中的相对位置, 右, 范围[0~1]。

@param fBottom: 视频流在画面中的相对位置, 下, 范围[0~1]。

返回值: TRUE 成功, FALSE 则失败

#### 5、启动 SDVideoRender

```
BOOL SDVideoRender_StartRender(void* pVideoRender);
```

参数:

@param pVideoRender: 模块指针

#### 6、停止 SDVideoRender

```
void SDVideoRender_StopRender(void* pVideoRender);
```

参数:

@param pVideoRender: 模块指针

#### 7、渲染一帧 YUV420 到指定视频流

```
BOOL SDVideoRender_RenderYuv420Frame(void* pVideoRender, unsigned int  
unStreamId, unsigned char* data, unsigned int unWidth, unsigned int unHeight);
```

**参数:**

@param pVideoRender: 模块指针

@param unStreamId: 需要渲染本帧到指定的视频流 ID

@param data: 指向视频 YUV420 数据

@param unWidth: 视频 YUV420 数据宽度

@param unHeight: 视频 YUV420 数据高度