

# 良田多浏览器控件接口说明

## 目录

设备函数 .....	2
窗口函数 .....	7
视频函数 .....	9
录像函数 .....	14
缩略图函数 .....	17
图像函数 .....	18
图像列表函数 .....	24
文件/目录操作函数 .....	26
二代证函数 .....	26
读卡函数 .....	28
磁条卡操作 .....	28
指纹函数 .....	29
条码/二维码函数 .....	30
OCR 函数 .....	31
内存对象函数 .....	33
HTTP 函数 .....	34
FTP 函数 .....	35
区域 1 函数 .....	37
区域 2 函数 .....	38
区域列表函数 .....	41
字体函数 .....	42
模板函数 .....	43
人脸识别 .....	46
其它 .....	47
事件通知 .....	47
附录 .....	48

应用平台: winXP/win7/win8

位数: 32/64

支持编程语言: javascript

支持浏览器: IE7/8/9/10/11、chrome44 及以下版本 (chrome42/43/44 需要启用 NPAPI 插件功能)

## 设备函数

### 1) 初始化高拍仪 (必须在所有其它函数之前调用)

- 成功返回TRUE, 失败返回FALSE

```
var Global_InitDevs();
```

### 2) 反初始化高拍仪 (退出时释放资源)

- 成功返回TRUE, 失败返回FALSE

```
var Global_DeinitDevs();
```

### 3) 创建视频

- dev 设备句柄
- resolution 分辨率索引
- subtype 子类型, 1 表示YUY2, 2 表示MJPG, 4表示UYVY, 传0自动选择一个类型
- 返回视频句柄

```
var Device_CreateVideo(var dev, var resolution, var subtype);
```

### 4) 获取设备数量

- type 设备类型, 1 表示视频设备, 2 表示音频设备
- 返回设备数量

```
var Global_GetDevCount(var type);
```

### 5) 获取显示名称

- type 类型, 1 表示视频设备, 2 表示音频设备
- idx 设备索引
- 返回显示名称

```
var Global_GetDisplayName(var type, var idx);
```

### 6) 获取友好名称

- type 类型, 1 表示视频设备, 2 表示音频设备
- idx 设备索引
- 返回友好名称

```
var Global_GetFriendlyName(var type, var idx);
```

### 7) 获取设备摄像头序号

- **type** 类型, 1 表示视频设备, 2 表示音频设备
  - **idx** 设备索引
  - 返回设备摄像头序号, 1 为主摄像头, 2、3 为辅摄像头, -1 表示获取失败
- ```
var Global_GetEloamType(var type, var idx);
```

#### 8) 创建设备

- **type** 设备类型, 1 表示视频设备, 2 表示音频设备
  - **idx** 设备索引
  - 返回设备对象, 失败为0
  - **备注:** 若有多个设备, idx为依次从0开始编号的整数
- ```
var Global_CreateDevice(var type, var idx);
```

#### 9) 增加引用计数

- **dev** 设备句柄
  - 返回增加后的引用计数
- ```
var Device_AddRef(var dev);
```

#### 10) 减少引用计数 (减少为0 时会销毁)

- **dev** 设备句柄
  - 返回减少后的引用计数, 为0 表示已经销毁
- ```
var Device_Release(var dev);
```

#### 11) 获取设备类型

- **dev** 设备句柄
  - 返回设备的类型, 1 为视频设备, 2 为音频设备
- ```
var Device_GetType(var dev);
```

#### 12) 获取设备索引

- **dev** 设备句柄
  - 返回设备的索引, 设备丢失后索引会变为-1, 其它设备丢失可能影响自身的索引值
- ```
var Device_GetIndex(var dev);
```

#### 13) 获取设备状态

- **dev** 设备句柄
  - 返回设备的状态, 0 表示空闲, 1 表示打开中, 2 表示使用中, 3 表示暂停
- ```
var Device_GetState(var dev);
```

#### 14) 获取设备友好名称

- **dev** 设备句柄
  - 返回设备的友好名称
- ```
var Device_GetFriendlyName(var dev);
```

#### 15) 获取设备显示名称

- **dev** 设备句柄

- 返回设备的显示名称

```
var Device_GetDisplayName(var dev);
```

#### 16) 获取设备子类型

- dev 设备句柄
- 返回设备支持的子类型，对应下表：

1	YUY2
2	MJPG
3	YUY2, MJPG
4	UYVY
5	UYVY, YUY2
6	UYVY, MJPG
7	UYVY, MJPG, YUY2

```
var Device_GetSubtype(var dev);
```

#### 17) 获取扫描尺寸

- dev 设备句柄
- 返回支持的扫描尺寸

```
var Device_GetScanSize(var dev);
```

#### 18) 获取设备摄像头序号

- dev 设备句柄
- 返回设备的摄像头序号，1 为主摄像头，2、3 为辅摄像头，-1 表示获取失败

```
var Device_GetEloamType(var dev);
```

#### 19) 获取分辨率个数

- dev 设备句柄
- subtype 设备子类型  
返回该设备支持的分辨率个数

```
var Device_GetResolutionCountEx(var dev, var subType);
```

#### 20) 获取分辨率宽度

- dev 设备句柄
- subtype 设备子类型
- Idx 分辨率索引  
返回该分辨率的宽度

```
var Device_GetResolutionwidthEx (var dev, var subType, var idx);
```

#### 21) 获取分辨率高度

- dev 设备句柄
- subtype 设备子类型
- idx 分辨率索引  
返回该分辨率的高度

```
var Device_GetResolutionHeightEx (var dev, var subType, var
```

`idx);`

## 22) 获取分辨率数量 (不推荐使用)

- `dev` 设备句柄
- 返回该设备支持的分辨率个数
- 备注: 默认返回YUY2模式的分辨率个数, 不支持YUY2的设备返回MJPG模式分辨率个数, 不支持YUY2和MJPG的设备返回UYVY 分辨率个数

`var Device_GetResolutionCount(var dev);`

## 23) 获取分辨率宽度 (不推荐使用)

- `dev` 设备句柄
- `Idx` 分辨率索引
- 返回该分辨率的宽度
- 备注: 默认返回YUY2模式的分辨率宽度, 不支持YUY2的设备返回MJPG模式分辨率宽度, 不支持YUY2和MJPG的设备返回UYVY 分辨率宽度

`var Device_GetResolutionWidth(var dev, var idx);`

## 24) 获取分辨率高度 (不推荐使用)

- `dev` 设备句柄
- `idx` 分辨率索引
- 返回该分辨率的高度
- 备注: 默认返回YUY2模式的分辨率高度, 不支持YUY2的设备返回MJPG模式分辨率高度, 不支持YUY2和MJPG的设备返回UYVY 分辨率高度

`var Device_GetResolutionHeight(var dev, var idx);`

## 25) 暂停预览

- `dev` 设备句柄
- 成功返回TRUE, 失败返回FALSE

`var Device_PausePreview(var dev);`

## 26) 恢复预览

- `dev` 设备句柄
- 成功返回TRUE, 失败返回FALSE

`var Device_ResumePreview(var dev);`

## 27) 获取VidepProc 数据

- `dev` 设备句柄
- `prop` 视频的各个属性, 0x1表示亮度, 0x2表示对比度, 0x3表示饱和度, 0x4表示色调, 0x5表示清晰度, 0x6表示伽马, 0x7表示白平衡, 0x8表示逆光对比, 0x9表示启用颜色, 0xA表示增益
- `value` `prop`对应的值的类型, 0x1表示最小值, 0x2表示最大值, 0x3表示步长, 0x4表示默认值, 0x5表示实际值, 0x6表示是否自动设置
- 返回属性对应值

```
var Device_GetVideoProcAmp(var dev, var prop, var value);
```

### 28) 设置VideoProc 数据

- dev 设备句柄
- prop 视频的各个属性, 0x1表示亮度, 0x2表示对比度, 0x3表示饱和度, 0x4表示色调, 0x5表示清晰度, 0x6表示伽马, 0x7表示白平衡, 0x8表示逆光对比, 0x9表示启用颜色, 0xA表示增益
- value 要设置的prop对应属性的实际值
- isAuto 是否启用自动
- 成功返回TRUE, 失败返回FALSE

```
var Device_SetVideoProcAmp(var dev, var prop, var value, var isAuto);
```

### 29) 获取CameraControl

- dev 设备句柄
- prop 照相机的各个属性, 0x1表示曝光, 0x2表示全景, 0x3表示倾斜, 0x4表示滚动, 0x5表示缩放, 0x6表示光圈, 0x7表示焦点
- value prop对应的值的类型, 0x1表示最小值, 0x2表示最大值, 0x3表示步长, 0x4表示默认值, 0x5表示实际值, 0x6表示是否自动设置
- 返回属性对应值

```
var Device_GetCameraControl(var dev, var prop, var value);
```

### 30) 设置CameraControl

- dev 设备句柄
- prop 照相机的各个属性, 0x1表示曝光, 0x2表示全景, 0x3表示倾斜, 0x4表示滚动, 0x5表示缩放, 0x6表示光圈, 0x7表示焦点
- value 要设置的prop对应属性的实际值
- sAuto 是否启用自动
- 成功返回TRUE, 失败返回FALSE

备注: 对于S520AF的设备,value为0表示手动对焦, 1表示单次对焦, 2表示自动对焦, isAuto参数忽略

```
var Device_SetCameraControl(var dev, var prop, var value, var isAuto);
```

### 31) 显示属性框

- dev 设备句柄
- view 所有者窗口
- 成功返回TRUE, 失败返回FALSE

```
var Device_ShowProperty(var dev, var view);
```

### 32) 获取设备唯一编号

- dev 设备句柄

- 返回设备唯一编号

```
var Device_GetSonixSerialNumber(var dev);
```

## 窗口函数

### 1) 设置窗口名称

- Name “view”表示预览窗口，“thumb”表示缩略图
- 无返回值

```
var Global_SetWindowName(var name);
```

### 2) 获取内部对象

- 返回内部对象

```
var View_GetObject();
```

### 3) 将图像选入窗口

- image 图像句柄
- 成功返回TRUE，失败返回FALSE

```
var View_SelectImage(var img);
```

### 4) 显示视频

- video 视频句柄
- 成功返回TRUE，失败返回FALSE

```
var View_SelectVideo(var video);
```

### 5) 清空窗口显示

- 成功返回TRUE，失败返回FALSE

```
var View_SelectNull();
```

### 6) 放大

- 成功返回TRUE，失败返回FALSE

```
var View_SetZoomIn();
```

### 7) 缩小

- 成功返回TRUE，失败返回FALSE

```
var View_SetZoomOut();
```

### 8) 按实际尺寸显示视频

- 成功返回TRUE，失败返回FALSE

```
var View_SetOriginal();
```

### 9) 视频铺满显示窗口

- 成功返回TRUE，失败返回FALSE

```
var View_SetCarpete();
```

## 10) 还原显示

- 成功返回TRUE，失败返回FALSE

```
var View_SetAdapt();
```

## 11) 设置是否全屏显示

- Full 是否全屏
- 成功返回TRUE，失败返回FALSE

```
var View_SetFullScreen(var bFull);
```

## 12) 获取窗口状态

- 返回窗口状态，-1为失败，1为普通状态，2为自定义状态

```
Var View_GetState();
```

## 13) 设置窗口状态

- stat 新状态，1为普通状态，2为自定义状态
- 成功返回TRUE，失败返回FALSE

```
var View_SetState(var stat);
```

## 14) 设置背景颜色

- clr 新的背景颜色值
- 成功返回TRUE，失败返回FALSE

```
var View_SetBkColor(var clr);
```

## 15) 设置文本

- text 新的文本值
- clr 文本颜色
- 成功返回TRUE，失败返回FALSE

```
var View_SetText(var text, var clr);
```

## 16) 获取选中区域

- 返回选中区域句柄。

```
var View_GetSelectedRect();
```

## 17) 设置选中区域

- rect 区域句柄
- 成功返回TRUE，失败返回FALSE

```
var View_SetSelectedRect(var rect);
```

## 18) 播放拍照效果

- 成功返回TRUE，失败返回FALSE

```
var View_PlayCaptureEffect();
```

## 19) 设置缩放比

- scale 缩放比，100表示1比1显示，最大值为2000，表示20倍



- 成功返回TRUE，失败返回FALSE

```
var View_SetScale(var scale);
```

20) 设置缩放比2

- ratio 缩放比，0为最小，100为最大

- 成功返回TRUE，失败返回FALSE

```
var View_SetRatio(var ratio);
```

## 视频函数

1) 增加引用计数

- video 视频句柄

- 返回增加后的引用计数

```
var Video_AddRef(var video);
```

2) 减少引用计数（减少为0时会销毁）

- video 视频句柄

- 返回减少后的引用计数，为0表示已经销毁

```
var Video_Release(var video);
```

3) 获取视频所属设备

- video 视频句柄

- 返回设备句柄

```
var Video_GetDevice(var video);
```

4) 获取分辨率索引

- video 视频句柄

- 返回对应的分辨率索引

```
var Video_GetResolution(var video);
```

5) 获取子类型

- video 视频句柄

- 返回子类型,对应下表:

1	YUY2
2	MJPG
3	YUY2, MJPG
4	UYVY
5	UYVY, YUY2
6	UYVY, MJPG
7	UYVY, MJPG, YUY2

```
var Video_GetSubtype(var video);
```

## 6) 获取宽度

- video 视频句柄
- 返回视频宽度

```
Var Video_Getwidth(var video);
```

## 7) 获取高度

- video 视频句柄
- 返回视频高度

```
Var Video_GetHeight(var video);
```

## 8) 拍照

- video 视频句柄
- scanSize 0 表示默认, 0x04 表示A3 , 0x08 表示A4 , 0x10 表示A5
- view 显示窗口句柄
- 返回创建的图像句柄

```
var Video_CreateImage(var video, var scanSize, var view);
```

## 9) 播放拍照声音

- 成功返回TRUE, 失败返回FALSE

```
var Global_PlayCaptureVoice();
```

## 10) 创建图像列表

- video 视频句柄
- scanSize 0 表示默认, 0x04 表示A3 , 0x08 表示A4 , 0x10 表示A5
- view 显示窗口句柄
- 返回创建的图像列表句柄

```
var Video_CreateImageList(var video, var scanSize, var view);
```

## 11) 向左旋转

- video 视频句柄
- 成功返回TRUE , 失败返回FALSE

```
var Video_RotateLeft(var video);
```

备注: 可以在显示视频 (View\_SelectVideo) 之前调用

## 12) 向右旋转

- video 视频句柄
- 成功返回TRUE , 失败返回FALSE

```
var Video_RotateRight(var video);
```

备注: 可以在显示视频 (View\_SelectVideo) 之前调用

## 13) 旋转180 度

- video 视频句柄
  - 成功返回TRUE，失败返回FALSE
- ```
var Video_Rotate180(var video);
```
- 备注：可以在显示视频（view\_SelectVideo）之前调用

#### 14) 上下镜像

- video 视频句柄
  - 成功返回TRUE，失败返回FALSE
- ```
var Video_Flip(var video);
```
- 备注：可以在显示视频（view\_SelectVideo）之前调用

#### 15) 左右镜像

- video 视频句柄
  - 成功返回TRUE，失败返回FALSE
- ```
var Video_Mirror(var video);
```
- 备注：可以在显示视频（view\_SelectVideo）之前调用

#### 16) 上下左右镜像

- video 视频句柄
  - 成功返回TRUE，失败返回FALSE
- ```
var Video_FlipAndMirror(var video);
```
- 备注：可以在显示视频（view\_SelectVideo）之前调用

#### 17) 启动灰度化

- video 视频句柄
  - 成功返回TRUE，失败返回FALSE
- ```
var Video_EnableGray(var video);
```

#### 18) 取消灰度化

- video 视频句柄
  - 成功返回TRUE，失败返回FALSE
- ```
var Video_DisableGray(var video);
```

#### 19) 启用二值化

- video 视频句柄
  - threshold 二值化阈值
  - 成功返回TRUE，失败返回FALSE
- ```
var Video_EnableThreshold(var video, var threshold);
```

#### 20) 取消二值化

- video 视频句柄

- 成功返回TRUE，失败返回FALSE

```
var Video_DisableThreshold(var video);
```

## 21) 启用自适应二值化

- video 视频句柄
- flag 保留值，必须为0
- 成功返回TRUE，失败返回FALSE

```
var Video_EnableAdaptiveThreshold(var video, var flag);
```

## 22) 取消自适应二值化

- video 视频句柄
- 成功返回TRUE，失败返回FALSE

```
var Video_DisableAdaptiveThreshold(var video);
```

## 23) 启用去底色

- video 视频句柄
- flag 保留，必须为0
- 成功返回TRUE，失败返回FALSE

```
var Video_EnableDelBkColor(var video, var flag);
```

## 24) 取消去底色

- video 视频句柄
- 成功返回TRUE，失败返回FALSE

```
var Video_DisableDelBkColor(var video);
```

## 25) 启用文字

- video 视频句柄
- font 字体句柄
- x 绘制的起始位置x坐标
- y 绘制的起始位置y坐标
- text 绘制的文本
- clr 文本的颜色
- weight 图片所占权值，取值范围为0-255，为0表示文字完全不透明，为255表示文字完全透明
- 成功返回TRUE，失败返回FALSE

```
var Video_EnableAddText(var video, var font, var x, var y, var text, var clr, var weight);
```

## 26) 取消文字

- video 视频句柄
- 成功返回TRUE，失败返回FALSE

```
var Video_DisableAddText(var video);
```

## 27) 启用纠偏

- video 视频句柄

- Flag 0 表示少裁, 1 表示多裁
- 成功返回TRUE , 失败返回FALSE

```
var Video_EnableDeskew(var video, var flag);
```

#### 28) 启用多张纠偏

- video 视频句柄
- Flag 0 表示少裁, 1 表示多裁
- 成功返回TRUE , 失败返回FALSE

```
var Video_EnableDeskewEx(var video, var flag);
```

#### 29) 取消纠偏

- video 视频句柄
- 成功返回TRUE , 失败返回FALSE

```
var Video_DisableDeskew(var video);
```

#### 30) 启用反色

- video 视频句柄
- 成功返回TRUE , 失败返回FALSE

```
var Video_EnableReverse(var video);
```

#### 31) 取消反色

- video 视频句柄
- 成功返回TRUE , 失败返回FALSE

```
var Video_DisableReverse(var video);
```

#### 32) 启用移动侦测

- video 视频句柄
- flag 保留, 必须为0
- 成功返回TRUE , 失败返回FALSE

```
var Video_EnableMoveDetec(var video, var flag);
```

#### 33) 取消移动侦测

- video 视频句柄
- 成功返回TRUE , 失败返回FALSE

```
var Video_DisableMoveDetec(var video);
```

#### 34) 启用日期

- video 视频句柄
- font 字体句柄
- x 绘制的起始位置x坐标
- y 绘制的起始位置y坐标
- clr 文本的颜色
- weight 图片所占权值, 取值范围为0-255, 为0表示文字完全不透明, 为255表示文字完全透明

- 成功返回TRUE，失败返回FALSE

```
var Video_EnableDate(var video, var font, var x, var y, var
clr, var weight);
```

### 35) 取消日期

- video 视频句柄
- 成功返回TRUE，失败返回FALSE

```
var Video_DisableDate(var video);
```

### 36) 启用平滑

- video 视频句柄
- Flag 保留值，必须为0
- 成功返回TRUE，失败返回FALSE

```
var Video_EnableSmooth(var video, var flag);
```

### 37) 取消平滑

- video 视频句柄
- 成功返回TRUE，失败返回FALSE

```
var Video_DisableSmooth(var video);
```

### 38) 开始录像（已淘汰）

- video 视频句柄
- filePath 录像文件保存路径
- flag 录像帧率，0表示5帧\秒；1表示10帧\秒；2表示15帧\秒；3表示20帧\秒；4表示25帧\秒
- 成功返回TRUE，失败返回FALSE

```
var Video_StartRecord(var video, var filePath, var flag );
```

### 39) 停止录像（已淘汰）

- video 视频句柄
- 成功返回TRUE，失败返回FALSE

```
var Video_StopRecord(var video);
```

## 录像函数

- 录像支持avi mp4 flv wmv 4种格式
- 建议选择合适分辨率录像，分辨率过高可能会导致录像失败

### 1) 初始化录像

- 成功返回TRUE，失败返回FALSE

```
Var eloamPluginAPI::Global_VideoCapInit()
```

## 2) 创建录像对象

- 成功返回录像对象

```
Var eloamPluginAPI::Global_CreatVideoCap()
```

## 3) 获取麦克风设备数量

- 成功返回麦克风设备数量

```
Var eloamPluginAPI::Global_VideoCapGetAudioDevNum()
```

## 4) 获取麦克风设备名称

- devIndex 麦克风设备名称索引
- 成功返回麦克风设备名称

```
Var eloamPluginAPI::Global_VideoCapGetAudioDevName(Var devIndex)
```

## 5) 释放录像对象

- videoCap 录像对象
- 成功返回TRUE，失败返回FALSE

```
Var eloamPluginAPI::VideoCap_Destroy(Var videoCap)
```

## 6) 录像设置

- videoCap 录像对象
- filename 录制文件的完整路径，包括文件名和后缀，例如 "D:\weike\videoCap\test.mp4"
- micIndex 麦克风设备索引
- framerate 录制视频帧率
- compressMode 视频压缩率，0:高；1:中；2:低，压缩率越高，文件越小但画面质量越差。
- width 输出视频宽
- height 输出视频高
- 成功返回TRUE，失败返回FALSE

```
Var eloamPluginAPI::VideoCap_PreCap(Var videoCap, Var fileName, Var micIndex, Var frameRate, Var compressMode, Var width, Var height)
```

## 7) 开始录像

- videoCap 录像对象
- 成功返回TRUE，失败返回FALSE

```
Var eloamPluginAPI::VideoCap_Start(Var videoCap)
```

## 8) 停止录像

- videoCap 录像对象
- 成功返回TRUE，失败返回FALSE

```
Var eloamPluginAPI::VideoCap_Stop(Var videoCap)
```

## 9) 暂停录像

- videoCap 录像对象
- 成功返回TRUE, 失败返回FALSE

`Var eloamPluginAPI::VideoCap_Pause(Var videoCap)`

#### 10) 设置水印

- videoCap 录像对象
- watermark 是否添加水印
- AddTime 是否增加时间水印, 例如: 2016.05.07 12.35.5.237
- mode 0: 文字水印, 1:图片水印
- pos 0: 左上角; 1:右上角; 2:左下角; 3:右下角; 4:中间
- alpha 0~255, 透明度控制
- imgPath 水印图片路径, 格式支持: jpg/bmp/png/jpeg/gif
- pStrText 水印文字
- color 字体颜色
- faceName 字体名称
- weight 字体大小
- Height 字体轻重
- Italic 是否斜体
- 成功返回TRUE, 失败返回FALSE

`Var eloamPluginAPI::VideoCap_SetWatermark(Var videoCap, Var watermark, Var AddTime, Var mode, Var pos, Var alpha, Var imgPath, Var pStrText, Var color, Var faceName, Var weight, Var Height, Var Italic)`

#### 11) 添加视频源

- videoCap 录像对象
- video 视频对象
- 成功返回TRUE, 失败返回FALSE

`Var eloamPluginAPI::VideoCap_AddVideoSrc(Var videoCap, Var video)`

#### 12) 添加视频源

- videoCap 录像对象
- video 视频对象
- x 视频源起点横坐标
- y 视频源起点纵坐标
- width 视频的宽
- height 视频的高
- 成功返回TRUE, 失败返回FALSE
- 备注: 如果要进行视频合成, 可添加多个视频源

`Var eloamPluginAPI::VideoCap_AddVideoSrcEx(Var videoCap, Var video, Var x, Var y, Var width, Var height)`

#### 13) 移除所有视频源



- videoCap 录像对象
- 成功返回TRUE，失败返回FALSE

```
var eloamPluginAPI::VideoCap_RemoveAllVideoSrc(Var videoCap)
```

#### 14) 获取录像状态

- videoCap 录像对象
- 0停止，1准备就绪，2正在录像，3暂停

```
eloamPluginAPI::VideoCap_GetState(Var videoCap)
```

## 缩略图函数

#### 1) 获取内部对象

- 返回内部对象

```
var Thumbnail_GetObject();
```

#### 2) 添加图片

- imagePath 图片路径
- 成功返回TRUE，失败返回FALSE

```
var Thumbnail_Add(var imagePath);
```

#### 3) 插入图片

- imagePath 图片路径
- pos 插入位置
- 成功返回TRUE，失败返回FALSE

```
var Thumbnail_Insert(var imagePath, var pos);
```

#### 4) 移除图片

- idx 图片索引
- bDel TRUE表示删除图片文件，FALSE表示不删除
- 成功返回TRUE，失败返回FALSE

```
var Thumbnail_Remove(var idx, var bDel);
```

#### 5) 清理图片

- bDel TRUE表示删除图片文件，FALSE表示不删除
- 成功返回TRUE，失败返回FALSE

```
var Thumbnail_Clear(var bDel);
```

#### 6) 获取图片数量

- 返回图片的数量

```
var Thumbnail_GetCount();
```

#### 7) 获取图片路径

- **Idx** 图片索引
  - 返回图片的全路径
- ```
var Thumbnail_GetFileName(var idx);
```

#### 8) 复选框是否勾选

- **Idx** 索引值
  - 选中返回TRUE，未选中返回FALSE
- ```
Var Thumbnail_GetCheck(var idx);
```

#### 9) 设置复选框勾选

- **Idx** 索引值
  - **bCheck** 是否选中
  - 成功返回TRUE，失败返回FALSE
- ```
Var Thumbnail_SetCheck(var idx, var bCheck);
```

#### 10) 设置语言

- **langId** 语言代号，表示识别的语言  
0x00000001表示英文，0x00000002表示简体中文，0x00000004表示繁体中文，0x00000008表示泰语，0x00000010表示德语，0x00000020表示俄语，0x00000040表示希腊语，0x00000080表示韩语，0x00000100表示西班牙语，0x00000200表示葡萄牙语，0x00000400表示意大利语；识别多种语言，则需要进行操作
  - 成功返回TRUE，失败返回FALSE
- ```
var Thumbnail_SetLanguage(var langId);
```

#### 11) 设置菜单选项

- **MenuId** 菜单Id
  - **Flag** 菜单项掩码
  - 成功返回TRUE，失败返回FALSE
- ```
var Thumbnail_SetMenuItem(var menuId, var flag);
```

#### 12) 获取选中索引

- 返回选中的索引
- ```
var Thumbnail_GetSelected();
```

## 图像函数

#### 1) 增加引用计数

- **img** 图像句柄
  - 返回增加后图像的引用计数
- ```
var Image_AddRef(var image);
```

## 2) 减少引用计数（如果引用计数为0 则销毁）

- `img` 图像句柄
- 返回减少后图像的引用计数，为0 表示图像已经销毁

```
var Image_Release(var image);
```

## 3) 创建图像

- `width` 图像宽度
- `height` 图像高度
- `channels` 3表示彩色图像，1表示灰度图像
- 返回图像对象，失败为0

```
var Global_CreateImage(var width, var height, var channels);
```

## 4) 从文件创建图像

- `fileName` 文件名
- `Flag` 保留，必须为0
- 返回图像对象，失败为0

```
var Global_CreateImageFromFile(var fileName, var flag);
```

## 5) 创建内存块对象

- `img` 图像句柄
- `fmt` 图像格式，-1 表示未知，0 表示bmp，1 表示ico，2 表示jpeg，13 表示png，18 表示tiff，25 表示gif
- `flag` 一般为0，对于jpg格式，0x0080表示高质量，0x0100表示较高质量，0x0200表示中等质量，0x0400表示较差质量，0x0800表示差质量
- 返回内存块句柄对象

```
var Image_CreateMemory(var image, var fmt, var flag);
```

## 6) 拷贝图像

- `imgDest` 目标图像句柄
- `imgSrc` 源图像句柄
- 成功返回TRUE，失败返回FALSE

```
var Image_Copy(var imageDest, var imageSrc);
```

## 7) 保存图像

- `img` 图像句柄
- `filename` 文件名
- `flag` 一般为0，对于jpg格式，0x0080表示高质量，0x0100表示较高质量，0x0200表示中等质量，0x0400表示较差质量，0x0800表示差质量
- 成功返回TRUE，失败返回FALSE

```
var Image_Save(var image, var fileName, var flag);
```

## 8) 保存图像到PDF

- `img` 图像句柄对象

- **fmt** 图像格式, 0 表示bmp, 1 表示ico, 2 表示jpeg , 13 表示png, 18 表示tiff , 25 表示gif
  - **fileName** 文件名
  - **flag** 一般为0 , 对于jpg 格式, 0x0080 表示高质量, 0x0100 表示较高质量, 0x0200 表示中等质量, 0x0400 表示较差质量, 0x0800 表示差质量
  - 成功返回TRUE , 失败返回FALSE
- ```
var Image_SaveToPDF(var image, var fmt, var fileName, var flag);
```

#### 9) 设置识别区域

- **img** 图像句柄
  - **rect** 区域对象, 传0 表示识别整个区域
  - 成功返回TRUE , 失败返回FALSE
- ```
var Image_SetDiscernRect(var image, var rect);
```

#### 10) 打印

- **image** 图像句柄
  - **x** 打印区域左上角x 坐标 (单位毫米)
  - **y** 打印区域左上角y 坐标 (单位毫米)
  - **width** 打印区域宽度 (单位毫米)
  - **height** 打印区域高度 (单位毫米)
  - **printer** 打印机名称, 为0 表示使用默认打印机
  - 成功返回TRUE , 失败返回FALSE
- ```
var Image_Print(var image, var x, var y, var width, var height, var printer);
```

#### 11) 打印

- **img** 图像句柄
  - **x** 打印区域左上角x 坐标 (单位毫米)
  - **y** 打印区域左上角y 坐标 (单位毫米)
  - **printer** 打印机名称, 为0 表示使用默认打印机
  - 成功返回TRUE , 失败返回FALSE
- ```
var Image_PrintByDPI(var image, var x, var y, var printer);
```

#### 12) 自适应打印

- **image** 图像句柄
  - **width** 打印区域宽度 (单位毫米)
  - **height** 打印区域高度 (单位毫米)
  - **printer** 打印机名称, 为0 表示使用默认打印机
  - 成功返回TRUE , 失败返回FALSE
- ```
var Image_AdaptivePrint(var image, var width, var height, var printer);
```

#### 13) 自适应打印

- **img** 图像句柄

- `printer` 打印机名称，为0 表示使用默认打印机
- 成功返回TRUE，失败返回FALSE

```
var Image_AdaptivePrintByDPI(var image, var printer);
```

#### 14) 获取图像宽度

- `img` 图像句柄
- 返回图像宽度

```
var Image_Getwidth(var image);
```

#### 15) 获取图像高度

- `img` 图像句柄
- 返回图像高度

```
var Image_GetHeight(var image);
```

#### 16) 获取通道数

- `image` 图像句柄
- 返回3 表示为彩色图像，返回1 表示灰度图像，返回-1 表示失败

```
var Image_GetChannels(var image);
```

#### 17) 获取每行字节数

- `img` 图像句柄
- 返回图像每行的字节数，一般为4 或8 的倍数

```
var Image_GetwidthStep(var image);
```

#### 18) 获取XDPI

- `img` 图像句柄
- 返回XDPI

```
var Image_GetXDPI(var image);
```

#### 19) 获取YDPI

- `img` 图像句柄
- 返回YDPI

```
var Image_GetYDPI(var image);
```

#### 20) 设置XDPI

- `img` 图像句柄
- `value` 新的XDPI
- 成功返回TRUE，失败返回FALSE

```
var Image_SetXDPI(var image, var value);
```

#### 21) 设置YDPI

- `img` 图像句柄
- `value` 新的YDPI
- 成功返回TRUE，失败返回FALSE

```
var Image_SetYDPI(var image, var value);
```

## 22) 获取Base64 数据

- `img` 图像句柄
- `fmt` 图像格式, -1 表示未知, 0 表示bmp, 1 表示ico, 2 表示jpeg, 13 表示png, 18 表示tiff, 25 表示gif
- `flag` 一般为0, 对于jpg格式, 0x0080表示高质量, 0x0100表示较高质量, 0x0200表示中等质量, 0x0400表示较差质量, 0x0800表示差质量
- 返回base64字符串(`fmt`为-1时返回不带文件头的图片数据的base64字符串)

```
var Image_GetBase64(var image, var fmt, var flag);
```

## 23) 在图像上书写文字

- `img` 图像句柄
- `font` 字体句柄
- `x` 绘制的起始位置x 坐标
- `y` 绘制的起始位置y 坐标
- `text` 绘制的文本
- `clr` 文本的颜色
- `weight` 图片所占权值, 取值范围为0-255, 为0 表示文字完全不透明, 为255 表示文字完全透明
- 成功返回TRUE, 失败返回FALSE

```
var Image_DrawText(var image, var font, var x, var y, var text, var clr, var weight);
```

## 24) 旋转图像

- `img` 图像句柄
- `angle` 旋转的角度(以角度为单位), 逆时针方向
- `clr` 旋转造成的空白区域被填充的颜色
- `flag` 0 表示旋转后尺寸不变, 1 表示旋转后内容不丢失
- 成功返回TRUE, 失败返回FALSE

```
var Image_Rotate(var image, var angle, var clr, var flag);
```

## 25) 裁剪图像

- `img` 图像句柄
- `rect` 区域句柄
- 成功返回TRUE, 失败返回FALSE

```
var Image_Crop(var image, var rect);
```

## 26) 重新设置图像大小

- `img` 图像句柄
- `width` 新大小宽度
- `height` 新大小高度
- `flag` 保留, 必须为0
- 成功返回TRUE, 失败返回FALSE

```
var Image_Resize(var image, var width, var height, var flag);
```

## 27) 合成图像

- `imgDest` 目标图像句柄
- `rectDest` 目标区域句柄
- `imgSrc` 源图像句柄
- `srcRect` 源区域句柄
- `weight` 本图像所占权值，取值范围为0-255，0表示完全不透明，255表示完全透明
- `flag` 保留，必须为0
- 成功返回TRUE，失败返回FALSE

```
var Image_Blend(var imageDest, var rectDest, var imageSrc, var rectSrc, var weight, var flag);
```

## 28) 将图像转化为彩色

- `img` 图像句柄
- 成功返回TRUE，失败返回FALSE

```
var Image_ToColor(var image);
```

## 29) 将图像转化为灰度

- `img` 图像句柄
- 成功返回TRUE，失败返回FALSE

```
var Image_ToGray(var image);
```

## 30) 将图像二值化

- `img` 图像句柄
- `Threshold` 二值化阈值
- 成功返回TRUE，失败返回FALSE

```
var Image_Threshold(var image, var threshold);
```

## 31) 图像自适应二值化

- `img` 图像句柄
- `Flag` 保留，必须为0
- 成功返回TRUE，失败返回FALSE

```
var Image_AdaptiveThreshold(var image, var flag);
```

## 32) 图像反色

- `img` 图像句柄
- 成功返回TRUE，失败返回FALSE

```
var Image_Reverse(var image);
```

## 33) 图像矫正

- `img` 图像句柄
- `Flag` 1 表示二代证
- 成功返回TRUE，失败返回FALSE

```
var Image_Rectify(var image, var flag);
```

## 34) 图像平滑

- `Img` 图像句柄
- `Flag` 保留值，必须为0
- 成功返回TRUE，失败返回FALSE

```
var Image_Smooth(var image, var flag);
```

## 35) 获取图像的MD5值

- `img` 图像句柄
- `fmt` 图像格式，-1 表示未知，0 表示bmp，1 表示ico，2 表示jpeg，13 表示png，18 表示tiff，25 表示gif
- `flag` 一般为0，对于jpg格式，0x0080表示高质量，0x0100表示较高质量，0x0200表示中等质量，0x0400表示较差质量，0x0800表示差质量
- 返回MD5 字符串

```
var Image_GetMD5(var image, var fmt, var flag)
```

## 图像列表函数

## 1) 增加引用计数

- `list` 图像列表句柄
- 返回增加后的引用计数

```
var ImageList_AddRef(var list);
```

## 2) 减少引用计数（如果减少为0 则销毁）

- `list` 图像列表句柄
- 返回减少后的引用计数，为0 表示已经销毁

```
var ImageList_Release(var list);
```

## 3) 创建图像列表

- 返回图像列表对象，失败为0

```
var Global_CreateImageList();
```

## 4) 从文件创建图像列表

- `fileName` 文件名
- `Flag` 保留，必须为0
- 返回图像列表对象，失败为0

```
var Global_CreateImageListFromFile(var fileName, var flag);
```

## 5) 添加图像

- `list` 图像列表句柄
- `img` 图像句柄



- 成功返回TRUE，失败返回FALSE

```
var ImageList_Add(var list, var img);
```

#### 6) 插入图像

- list 图像列表句柄
- img 图像句柄
- pos 图像插入位置
- 成功返回TRUE，失败返回FALSE

```
var ImageList_Insert(var list, var img, var pos);
```

#### 7) 移除图像

- list 图像列表句柄
- idx 要删除的索引
- 成功返回TRUE，失败返回FALSE

```
var ImageList_Remove(var list, var idx);
```

#### 8) 清理图像列表

- list 图像列表句柄
- 成功返回TRUE，失败返回FALSE

```
var ImageList_Clear(var list);
```

#### 9) list 图像列表句柄

- list 图像列表句柄
- 返回图像的数量

```
var ImageList_GetCount(var list);
```

#### 10) 获取图像

- list 图像列表句柄
- idx 图像索引
- 返回该索引下的图像句柄

```
var ImageList_GetImage(var list, var idx);
```

#### 11) 保存图像列表

- list 图像列表句柄
- filename 文件名
- flag 一般为0，对于jpg格式，0x0080表示高质量，0x0100表示较高质量，0x0200表示中等质量，0x0400表示较差质量，0x0800表示差质量
- 成功返回TRUE，失败返回FALSE

```
var ImageList_Save(var list, var fileName, var flag);
```

#### 12) 保存图像列表到PDF

- list 图像列表句柄
- fmt 图像格式，0表示bmp，1表示ico，2表示jpeg，13表示png，18表示tiff，25表示gif

- filename 文件名
- flag 一般为0，对于jpg格式，0x0080表示高质量，0x0100表示较高质量，0x0200表示中等质量，0x0400表示较差质量，0x0800表示差质量
- 成功返回TRUE，失败返回FALSE

```
var ImageList_SaveToPDF(var list, var fmt, var fileName, var flag);
```

### 13) 识别图像列表（异步，返回不代表识别结束）

- flag 1 表示文本识别，2 表示 条码/二维码 识别
- imageList 需要识别的图像列表，需要识别的图像列表，识别区域可以通过调用图像的Image\_SetDiscernRect 函数来设置
- 成功返回TRUE，失败返回FALSE

```
var Global_DiscernOcrList(var flag, var imageList);
```

## 文件/目录操作函数

### 1) 删除本地文件

- filename 文件名
- 成功返回TRUE，失败返回FALSE

```
var Global_DelFile(var fileName);
```

### 2) 获取临时文件名

- ext 扩展名
- 返回临时文件名，失败返回0

```
var Global_GetTempName(var ext);
```

### 3) 创建文件夹

- dirPath 文件夹路径
- 成功返回TRUE，失败返回FALSE

```
var Global_CreateDir(var dirPath);
```

### 4) 删除文件夹

- dirPath 文件夹路径
- 成功返回TRUE，失败返回FALSE

```
var Global_Removedir(var dirPath);
```

## 二代证函数

### 1) 初始化二代证，（必须在所有二代证函数前调用）

- 成功返回TRUE, 失败返回FALSE

```
var Global_InitIdCard();
```

## 2) 反初始化二代证

- 成功返回TRUE, 失败返回FALSE

```
var Global_DeinitIdCard();
```

## 3) 读取二代证

- 成功返回TRUE, 失败返回FALSE

```
var Global_ReadIdCard();
```

## 4) 识别二代证 （异步，返回不代表识别成功，需要在二代证初始化后调用）

- 成功返回TRUE, 失败返回FALSE

```
var Global_DiscernIdCard();
```

## 5) 获取二代证图像 （需要在二代证识别之后调用才有效）

- flag 1表示头像，2表示正面，3表示反面，4表示正反垂直合成，5表示反正垂直合成，6表示正反水平合成，7表示反正水平合成
- 返回图像对象，失败为0

```
var Global_GetIdCardImage(var flag);
```

## 6) 获取二代证数据

- Flag 代号，为以下数值：

|          |    |
|----------|----|
| 姓名       | 1  |
| 性别       | 2  |
| 民族       | 3  |
| 出生年      | 4  |
| 出生月      | 5  |
| 出生日      | 6  |
| 住址       | 7  |
| 身份证号码    | 8  |
| 签发机关     | 9  |
| 有效期限年(起) | 10 |
| 有效期限月(起) | 11 |
| 有效期限日(起) | 12 |
| 有效期限年(止) | 13 |
| 有效期限月(止) | 14 |
| 有效期限日(止) | 15 |
| 芯片序列号    | 16 |

- 返回数据

```
var Global_GetIdCardData(var flag);
```

## 7) 停止二代证识别

- 成功返回TRUE ， 失败返回FALSE

```
var Global_StopIdCardDiscern();
```

## 读卡函数

备注：用户刷卡会触发相应的事件通知，可以在接收到刷卡事件后读取数据

### 1) 初始化阅读器

- 成功返回TRUE，失败返回FALSE

```
var Global_InitReader();
```

### 2) 反初始化阅读器

- 成功返回TRUE，失败返回FALSE

```
var Global_DeinitReader();
```

### 3) 启动阅读 （检测到刷卡会触发Reader事件，请参照事件通知接口）

- 成功返回TRUE，失败返回FALSE

```
var Global_ReaderStart();
```

### 4) 关闭阅读

- 成功返回TRUE，失败返回FALSE

```
var Global_ReaderStop();
```

### 5) 获取CPU卡ID

- 返回ID

```
var Global_ReaderGetCpuId();
```

### 6) 获取CPU卡银行卡号

- 返回银行卡号

```
var Global_ReaderGetCpuCreditCardNumber();
```

### 7) 获取Mem卡ID

- 返回ID

```
var Global_ReaderGetMemoryId();
```

### 8) 获取M1卡ID

- 返回ID

```
var Global_ReaderGetM1Id();
```

## 磁条卡操作

### 1) 初始化磁卡

- 成功返回TRUE，失败返回FALSE

```
var Global_InitMagneticCard();
```

## 2) 反初始化磁卡

- 成功返回TRUE，失败返回FALSE

```
var Global_DeinitMagneticCard();
```

## 3) 启动磁卡阅读(检测到用户刷卡触发Mag事件，请参照事件通知接口)

- 成功返回TRUE，失败返回FALSE

```
var Global_MagneticCardReaderStart();
```

## 4) 停止磁卡阅读

- 成功返回TRUE，失败返回FALSE

```
var Global_MagneticCardReaderStop();
```

## 5) 读取磁道数据

- Type 0表示1磁道，1表示2磁道，2表示3磁道

- 返回磁道数据

```
var Global_MagneticCardGetData(var type);
```

## 6) 读取磁卡卡号

- 返回磁卡卡号

```
var Global_MagneticCardGetNumber();
```

# 指纹函数

备注：指纹数据采集完成会触发相应的事件通知，可以在接收到事件后读取数据

## 1) 初始化指纹仪

- 成功返回TRUE，失败返回FALSE

```
var Global_InitBiokey();
```

## 2) 反初始化指纹仪

- 成功返回TRUE，失败返回FALSE

```
var Global_DeinitBiokey();
```

## 3) 获取指纹模板(指纹模板采集完成触发Biokey事件，请参照事件通知接口)

- 成功返回TRUE，失败返回FALSE

备注：获取指纹模板要按压三次手指

```
var Global_GetBiokeyTemplate();
```

## 4) 停止获取指纹模板

- 成功返回TRUE，失败返回FALSE

```
var Global_StopGetBiokeyTemplate();
```

## 5) 获取模板数据

- 返回数据内存句柄，失败为0

```
var Global_GetBiokeyTemplateData();
```

## 6) 获取指纹特征（（指纹特征采集完成触发Biokey事件，请参照事件通知接口））

- 成功返回TRUE，失败返回FALSE

```
var Global_GetBiokeyFeature();
```

## 7) 停止获取指纹特征

- 成功返回TRUE，失败返回FALSE

```
var Global_StopGetBiokeyFeature();
```

## 8) 获取特征数据

- 返回数据内存对象，失败为0

```
var Global_GetBiokeyFeatureData();
```

## 9) 获取指纹图像

- 返回图像对象，失败为0

```
var Global_GetBiokeyImage();
```

## 10) 指纹数据对比

- mem1 指纹数据1
- mem2 指纹数据2
- 返回相似度，值越大表示越相似，一般来说大于50 表示匹配成功

```
var Global_BiokeyVerify(var mem1, var mem2);
```

## 条码/二维码函数

支持条码类型： Code 128-AUTO、Code 128A、Code 128B、Code 128C、EAN-13、EAN128、UPC-A、UPC-E

## 1) 初始化 条码/二维码 识别

- 成功返回TRUE，失败返回FALSE

```
var Global_InitBarcode();
```

## 2) 反初始化 条码/二维码 识别

- 成功返回TRUE，失败返回FALSE

```
var Global_DeinitBarcode();
```

## 3) 条码/二维码 识别

- Img 要识别的图像，识别区域可以通过调用图像的Image\_SetDiscernRect函数来设置

- 成功返回TRUE，失败返回FALSE

```
var Global_DiscernBarcode(var image);
```

#### 4) 获取 条码/二维码 数量（需要在 条码/二维码 识别后才有效）

- 返回 条码/二维码 的数量

```
var Global_GetBarcodeCount();
```

#### 5) 获取 条码/二维码 类型（需要在 条码/二维码 识别后才有效）

- Idx 条码/二维码 索引
- 返回 条码/二维码 的类型，详细见附件

```
var Global_GetBarcodeType(var idx);
```

#### 6) 获取 条码/二维码 数据（需要在 条码/二维码 识别后才有效）

- Idx 条码/二维码 索引
- 返回的 条码/二维码 数据

```
var Global_GetBarcodeData(var idx);
```

#### 7) 获取 条码/二维码 类型

- page 页ID
- Blk 块ID
- 返回该块对应的 条码/二维码 的类型，若文字识别，则返回-1

```
var Global_GetOcrBarcodeType(var page, var blk);
```

## OCR 函数

#### 1) 初始化OCR库

- 成功返回TRUE，失败返回FALSE

```
var Global_InitOcr();
```

#### 2) 反初始化OCR 库

- 成功返回TRUE，失败返回FALSE

```
var Global_DeinitOcr();
```

#### 3) 设置语言

- Lang 语言
- 成功返回TRUE，失败返回FALSE

```
var Global_SetOcrLanguage(var lang);
```

#### 4) 启动OCR识别（异步，返回不代表识别结束）

- flag 1 表示识别文本，2 表示识别 条码/二维码
- img 需要识别的图像，识别区域可以通过调用图像Image\_SetDiscernRect函数来设置
- 成功返回TRUE，失败返回FALSE

```
var Global_DiscernOcr(var flag, var image);
```

5) 获取页的数量（结果为识别的图像的张数）

- 返回页的数量

```
var Global_GetOcrPageCount();
```

6) 获取块的数量（文字识别会自动分块，条码/二维码 识别表示 条码/二维码 个数）

- page 页ID
- 返回该页中块的数量

```
var Global_GetOcrBlockCount(var page);
```

7) 获取OCR 文本

- page 页ID
- 返回该页上的文字识别结果，忽略OCR时自动分块

```
var Global_GetOcrPlainText(var page);
```

8) 获取OCR 数据

- page 页ID
- blk 块ID
- 返回的该块上的信息（文字信息和 条码/二维码 信息）

```
var Global_GetOcrData(var page, var blk);
```

9) 保存OCR 识别结果

- filename 文件名，可以保存为pdf、doc、xls或txt文件
- flag 保留，必须为0
- 成功返回TRUE，失败返回FALSE

```
var Global_SaveOcr(var fileName, var flag);
```

10) 停止OCR 识别

- 成功返回TRUE，失败返回FALSE

```
var Global_StopOcrDiscern();
```

11) 等待OCR 识别（在识别后调用，即变为同步模式）

- 成功返回TRUE，失败返回FALSE

```
var Global_waitOcrDiscern();
```

12) 快速OCR识别

- fileName 图片路径
- resultPath 结果路径
- flag 0表示简体中文，1表示繁体中文，2表示英文
- 成功返回TRUE，失败返回FALSE

```
var Global_QuickOcr(var fileName, var resultPath, var flag);
```



## 内存对象函数

### 1) 增加引用计数

- mem 内存句柄
- 返回增加后的引用计数

```
var Memory_AddRef(var mem);
```

### 2) 减少引用计数（如果减少为0 则销毁）

- mem 内存句柄
- 返回减少后的引用计数，为0 表示已经销毁

```
var Memory_Release(var mem);
```

### 3) 创建内存对象

- 返回内存块句柄，失败返回0

```
var Global_CreateMemory();
```

### 4) 从文件创建内存块

- fileName 文件名
- 返回内存块句柄，失败返回0

```
var Global_CreateMemoryFromFile(var fileName);
```

### 5) 获取Base64 数据

- mem 内存句柄
- 返回base64 字符串

```
var Memory_GetBase64(var mem);
```

### 6) 创建图像

- mem 内存句柄
- flag 保留，必须为0
- 返回图像句柄，失败为0

```
var Memory_CreateImage(var mem, var flag);
```

### 7) 保存到文件

- mem 内存句柄
- filename 文件名
- 成功为TRUE，失败为FALSE

```
var Memory_Save(var mem, var fileName);
```

### 8) 获取内存对象的MD5

- mem 内存句柄
- 成功返回MD5值

```
var Memory_GetMD5(var mem)
```

## HTTP 函数

### 1) 增加引用计数

- http http 句柄
- 返回增加后的引用计数

```
var Http_AddRef(var http);
```

### 2) 减少引用计数（如果减少为0 则销毁）

- http http 句柄
- 返回减少后的引用计数，为0 表示已经销毁

```
var Http_Release(var http);
```

### 3) 创建HTTP对象

- httpPath 服务器路径
- 返回HTTP对象，失败为0

```
var Global_CreateHttp(var httpPath);
```

### 4) 上传文件

- http http 句柄
- flag 保留，必须传0
- localPath 本地文件名
- headers 要发送的数据头
- predata 前置数据
- taildata 后置数据
- 成功返回TRUE，失败返回FALSE

```
var Http_Upload(var http, var flag, var localPath, var headers, var predata, var taildata);
```

### 5) 上传内存数据

- http http 句柄
- flag 保留，必须传0
- mem 内存句柄
- headers 要发送的数据头
- predata 前置数据
- taildata 后置数据
- 成功返回TRUE，失败返回FALSE

```
var Http_UploadMemory(var http, var flag, var mem, var headers, var predata, var taildata);
```

### 6) 上传图像文件（上传数据格式参照附录）

- http http句柄
- filename 本地文件路径
- remoteName 服务器文件路径

- 成功返回TRUE，失败返回FALSE

```
var Http_UploadImageFile(var http, var fileName, var remoteName);
```

### 7) 上传图像内存块

- http http句柄
- mem 内存块对象
- remoteName 服务器文件路径
- 成功返回TRUE，失败返回FALSE

```
var Http_UploadImageMemory(var http, var mem, var remoteName);
```

### 8) 上传图像

- http http句柄
- img 图像对象
- fmt 图像格式，-1表示未知，0表示bmp，1表示ico，2表示jpeg，13表示png，18表示tiff，25表示gif
- flag 一般为0，对于jpg格式，0x0080表示高质量，0x0100表示较高质量，0x0200表示中等质量，0x0400表示较差质量，0x0800表示差质量
- remoteName 服务器文件路径
- 成功返回TRUE，失败返回FALSE

```
var Http_UploadImage(var http, var img, var fmt, var flag, var remoteName);
```

### 9) 停止上传

- http http 句柄
- 成功返回TRUE，失败返回FALSE

```
var Http_StopUpload(var http);
```

### 10) 等待上传

- http http 句柄
- 成功返回TRUE，失败返回FALSE

```
var Http_waitUpload(var http);
```

### 11) 获取服务器返回数据

- http http 句柄
- 成功返回服务器数据，失败返回空

```
var Http_GetServerInfo(var http)
```

## FTP 函数

### 1) 增加引用计数

- ftp ftp 句柄

- 返回增加后的引用计数

```
var Ftp_AddRef(var ftp);
```

## 2) 减少引用计数（如果减少为0 则销毁）

- ftp ftp 句柄
- 返回减少后的引用计数，为0 表示已经销毁

```
var Ftp_Release(var ftp);
```

## 3) 创建FTP对象

- ftpPath 服务器路径
- 返回FTP对象，失败为0

```
var Global_CreateFtp(var ftpPath);
```

## 4) 上传文件

- ftp ftp 句柄
- flag 保留，必须传0
- localPath 本地文件路径
- remotePath 服务器文件路径
- 成功返回TRUE，失败返回FALSE

```
var Ftp_Upload(var ftp, var flag, var localPath, var remotePath);
```

## 5) 停止上传

- ftp ftp 句柄
- 成功返回TRUE，失败返回FALSE

```
var Ftp_StopUpload(var ftp);
```

## 6) 等待上传

- ftp ftp 句柄
- 成功返回TRUE，失败返回FALSE

```
var Ftp_WaitUpload(var ftp);
```

## 7) 创建文件夹

- ftp ftp 句柄
- dirPath 文件夹路径
- 成功返回TRUE，失败返回FALSE

```
var Ftp_CreateDir(var ftp, var dirPath);
```

## 8) 删除文件夹

- ftp ftp 句柄
- dirPath 文件夹路径
- 成功返回TRUE，失败返回FALSE

```
var Ftp_RemoveDir(var ftp, var dirPath);
```

## 区域 1 函数

备注：该区域为矩形，由起点坐标和长宽描述

### 1) 增加引用计数

- `rect` 区域句柄
- 返回增加后的引用计数

```
var Rect_AddRef(var rect);
```

### 2) 减少引用计数（减少为0将会销毁）

- `rect` 区域句柄
- 返回减少后的引用计数，为0表示已经销毁

```
var Rect_Release(var rect);
```

### 3) 创建区域对象

- `x` 区域的左上角x坐标
- `y` 区域的左上角y坐标
- `width` 区域宽度
- `Height` 区域高度
- 返回区域对象，失败为0

```
var Global_CreateRect(var x, var y, var width, var height);
```

### 4) 获取左上角x坐标

- `rect` 区域句柄
- 返回左上角x坐标

```
var Rect_GetLeft(var rect);
```

### 5) 设置左上角x坐标

- `rect` 区域句柄
- `left` 新的左上角x坐标
- 成功返回TRUE，失败返回FALSE

```
var Rect_SetLeft(var rect, var left);
```

### 6) 获取左上角y坐标

- `rect` 区域句柄
- 返回左上角y坐标

```
var Rect_GetTop(var rect);
```

### 7) 设置左上角y坐标

- `rect` 区域句柄
- `left` 新的左上角y坐标

- 成功返回TRUE，失败返回FALSE

```
var Rect_SetTop(var rect, var top);
```

#### 8) 获取宽度

- rect 区域句柄
- 返回区域宽度

```
var Rect_GetWidth(var rect);
```

#### 9) 设置宽度

- rect 区域句柄
- width 新的区域宽度
- 成功返回TRUE，失败返回FALSE

```
var Rect_SetWidth(var rect, var width);
```

#### 10) 获取高度

- rect 区域句柄
- 返回区域高度

```
var Rect_GetHeight(var rect);
```

#### 11) 设置高度

- rect 区域句柄
- height 新的区域高度
- 成功返回TRUE，失败返回FALSE

```
var Rect_SetHeight(var rect, var height);
```

#### 12) 拷贝

- rectDest 目标区域句柄
- rectSrc 源区域句柄
- 成功返回TRUE，失败返回FALSE

```
var Rect_Copy(var rectDest, var rectSrc);
```

## 区域 2 函数

备注：该区域为矩形，由四个顶点坐标描述

#### 1) 增加引用计数

- Region 区域句柄
- 返回增加后的引用计数

```
var Region_AddRef(var region);
```

#### 2) 减少引用计数（减少为0将会销毁）

- Region 区域句柄
- 返回减少后的引用计数，为0表示已经销毁

```
var Region_Release(var region);
```

### 3) 创建区域

- X1 x1 坐标
- Y1 y1 坐标
- X2 x2 坐标
- Y2 y2 坐标
- X3 x3 坐标
- Y3 y3 坐标
- X4 x4 坐标
- Y4 y4 坐标
- 返回区域对象，失败为0

```
var Global_CreateRegion(var x1, var y1, var x2, var y2, var x3,  
var y3, var x4, var y4);
```

### 4) 获取X1坐标

- Region 区域句柄
- 返回X1坐标

```
Var Region_GetX1(var region);
```

### 5) 获取Y1坐标

- Region 区域句柄
- 返回Y1坐标

```
Var Region_GetY1(var region);
```

### 6) 获取X2坐标

- Region 区域句柄
- 返回X2坐标

```
Var Region_GetX2(var region);
```

### 7) 获取Y2坐标

- Region 区域句柄
- 返回Y2坐标

```
Var Region_GetY2(var region);
```

### 8) 获取X3坐标

- Region 区域句柄
- 返回X3坐标

```
Var Region_GetX3(var region);
```

### 9) 获取Y3坐标

- Region 区域句柄
- 返回Y3坐标

```
Var Region_GetY3(var region);
```

## 10) 获取X4坐标

- Region 区域句柄
- 返回X4坐标

```
Var Region_GetX4(var region);
```

## 11) 获取Y4坐标

- Region 区域句柄
- 返回Y4坐标

```
Var Region_GetY4(var region);
```

## 12) 获取宽度

- Region 区域句柄
- 返回宽度

```
Var Region_Getwidth(var region);
```

## 13) 获取高度

- Region 区域句柄
- 返回高度

```
Var Region_GetHeight(var region);
```



## 区域列表函数

备注：列表中区域为区域 2 类型

### 1) 创建区域列表

- 返回区域对象，失败为0

```
var Global_CreateRegionList();
```

### 2) 增加引用计数

- List 区域列表句柄
- 返回增加后的引用计数

```
Var RegionList_AddRef(var list);
```

### 3) 减少引用计数（减少到0将会销毁）

- List 区域列表句柄
- 返回减少后的引用计数，为0表示已经销毁

```
Var RegionList_Release(var list);
```

### 4) 添加区域到区域列表

- List 区域列表句柄
- Region 区域句柄
- 成功返回TRUE, 失败返回FALSE

```
Var RegionList_Add(var list, var region);
```

### 5) 插入区域到区域列表

- List 区域列表句柄
- Region 区域句柄
- Pos 插入位置
- 成功返回TRUE, 失败返回FALSE

```
Var RegionList_Insert(var list, var region, var pos);
```

### 6) 从区域列表删除区域

- List 区域列表句柄
- Idx 索引
- 成功返回TRUE, 失败返回FALSE

```
Var RegionList_Remove(var list, var idx);
```

### 7) 清理区域列表

- List 区域列表句柄
- 成功返回TRUE, 失败返回FALSE

```
Var RegionList_Clear(var list);
```

## 8) 获取区域列表中的区域数量

- List 区域列表句柄
- 返回区域个数

```
Var RegionList_GetCount(var list);
```

## 9) 获取区域

- List 区域列表句柄
- Idx 索引
- 返回区域句柄，失败为0

```
Var RegionList_GetRegion(var list, var idx);
```

## 字体函数

## (1) 创建字体

- height 文字的高度
- width 文字的宽度
- escap 指定移位向量和设备X 轴之间的一个角度，以十分之一度为单位。移位向量平行于正文行的基线
- orien 指定每个字符的基线和设备X 轴之间的角度
- thickness 字体的粗细程度
- italic 1 为斜体，0 为非斜体
- underline 1 为画线字体，0 为非画线字体
- strick 1 表示增加删除线，0 表示不增加删除线
- font 字体名，例如：宋体
- 返回字体对象，失败则返回0

```
var Global_CreateTypeface(var height, var width, var escap, var orien, var thickness, var italic, var underline, var strike, var font);
```

## (2) 增加引用计数

- font 字体句柄
- 返回增加后的引用计数

```
var Font_AddRef(var font);
```

## (3) 减少引用计数（减少为0将会销毁）

- font 字体句柄
- 返回减少后的引用计数，为0表示已经销毁

```
var Font_Release(var font);
```

## 模板函数

### 1) 增加引用计数

- `templ` 模板句柄
- 返回增加后的引用计数

```
var Templ_AddRef(var templ);
```

### 2) 减少引用计数（减少为0将会销毁）

- `templ` 模板句柄
- 返回减少后的引用计数，为0表示已经销毁

```
var Templ_Release(var templ);
```

### 3) 创建模板

- `templName` 模板名称
- 返回模板对象

```
var Global_CreateTempl(var templName);
```

### 4) 通过文件创建模板

- `filename` 文件名称
- `flag`保留，必须传0
- 返回模板对象

```
var Global_CreateTemplFromFile(var fileName, var flag);
```

### 5) 通过数据创造模板

- `data` 数据名称
- `flag`保留，必须传0
- 返回模板对象

```
var Global_CreateTemplFromData(var data, var flag);
```

### 6) 初始化模板识别

- 成功返回TRUE，失败返回FALSE

```
var Global_InitTemplOcr();
```

### 7) 反初始化模板识别

- 成功返回TRUE，失败返回FALSE

```
var Global_DeinitTemplOcr();
```

### 8) 设置识别语言

- `Lang` 语言
- 成功返回TRUE，失败返回FALSE

```
var Global_SetTemplOcrLanguage(var lang);
```

### 9) 通过模板进行OCR 识别

- `image` 需要识别的图像

- `templ` 需要识别的模板
- 成功返回TRUE，失败返回FALSE

```
var Global_DiscernTempl(var image, var templ);
```

#### 10) 获取模板识别结果

- 返回模板对象

```
var Global_GetTemplResult();
```

#### 11) 停止模板识别

- 成功返回TRUE，失败返回FALSE

```
var Global_StopTemplDiscern();
```

#### 12) 等待模板识别结果

- 成功返回TRUE，失败返回FALSE

```
var Global_waitTemplDiscern();
```

#### 13) 获取模板名称

- `templ` 模板句柄

- 返回模板名称

```
var Templ_GetName(var templ);
```

#### 14) 获取模板ID

- `templ` 模板句柄

- 返回模板ID

```
var Templ_GetId(var templ);
```

#### 15) 添加字段

- `templ` 模板句柄

- `fieldname` 字段名称

- `type` 类型，1表示文本，2表示 条码/二维码

- `left` 左上角X坐标

- `top` 左上角Y坐标

- `right` 右上角X坐标

- `bottom` 右上角Y坐标

- 成功返回TRUE，失败返回FALSE

```
var Templ_AppendField(var templ, var fieldName, var type, var left, var top, var right, var bottom);
```

#### 16) 获取字段数量

- `templ` 模板句柄

- 返回字段数量

```
var Templ_GetFieldCount(var templ);
```

#### 17) 获取字段名称

- `templ` 模板句柄

- idx 字段索引
- 返回字段名称

```
var Temp1_GetFieldName(var temp1, var idx);
```

#### 18) 获取字段类型

- temp1 模板句柄
- idx 字段索引
- 返回字段类型，1表示文本，2表示 条码/二维码，3表示图片（base64编码）

```
var Temp1_GetFieldType(var temp1, var idx);
```

#### 19) 清除所有字段

- temp1 模板句柄
- 成功返回TRUE，失败返回FALSE

```
var Temp1_ClearField(var temp1);
```

#### 20) 获取字段区域左上角X坐标

- temp1 模板句柄
- idx 字段索引
- 返回左上角X坐标

```
var Temp1_GetFieldRectLeft(var temp1, var idx);
```

#### 21) 获取字段区域左上角Y坐标

- temp1 模板句柄
- idx 字段索引
- 返回左上角Y坐标

```
var Temp1_GetFieldRectTop(var temp1, var idx);
```

#### 22) 获取字段区域右下角X坐标

- temp1 模板句柄
- idx 字段索引
- 返回右下角X坐标

```
var Temp1_GetFieldRectRight(var temp1, var idx);
```

#### 23) 获取字段区域右下角Y坐标

- temp1 模板句柄
- idx 字段索引
- 返回右下角Y坐标

```
var Temp1_GetFieldRectBottom(var temp1, var idx);
```

#### 24) 获取字段结果

- temp1 模板句柄
- idx 字段索引
- 返回字段结果

```
var Temp1_GetFieldResult(var temp1, var idx);
```

## 25) 设置字段结果

- templ 模板句柄
- idx 字段结果
- result 结果
- 成功返回TRUE，失败返回FALSE

```
var Templ_SetFieldResult(var templ, var idx, var result);
```

## 26) 获取模板数据

- templ 模板句柄
- flag 保留，必须传0
- 返回模板数据

```
var Templ_GetData(var templ, var flag);
```

## 27) 保存到XML

- templ 模板句柄
- filename 文件名
- flag 保留，必须传0
- 成功返回TRUE，失败返回FALSE

```
var Templ_Save(var templ, var fileName, var flag);
```

## 人脸识别

## 1) 初始化人脸识别（应在视频关闭状态时调用）

- 成功返回TRUE，失败返回FALSE
- 成功初始化需要较长时间，根据计算机性能不同在20s~40s之间

```
var InitFaceDetect (void);
```

## 2) 反初始化人脸识别（应在视频关闭状态时调用）

- 成功返回TRUE，失败返回FALSE

```
var DeinitFaceDetect (void);
```

## 3) 人脸识别

- pImage1 图像对象， 图像1
- pImage2 图像对象， 图像2
- 返回相似度， 范围在0至100， 一般认为大于50， 则匹配成功
- 返回-1， 则表示识别失败

```
var DiscernFaceDetect (var pImage1, var pImage2);
```

## 其它

### 1) 是否以管理员权限运行

- 是以管理员权限运行返回TRUE，不是返回FALSE

`var Global_IsUserAdmin()`

## 事件通知

### 1) 设备接入和丢失

- `type` 设备类型，1 表示视频设备，2 表示音频设备
- `idx` 设备索引
- `dbt` 1 表示设备到达，2 表示设备丢失
- **备注：**若有多个设备，`idx`为依次从0开始编号的整数

`VOID DevChange(int type, int idx, int dbt);`

### 2) 检测到二代证

- `ret` 1表示检测到二代证

`VOID IdCard(int ret);`

### 3) OCR识别完成

- `flag` 1表示OCR识别完成，2 识别进度
- `ret` 当`flag`为1时，`ret` 0表示识别成功，-1表示识别失败；当`flag`为2时，`ret` 表示识别百分比

`VOID Ocr(int flag, int ret);`

### 4) 视频帧

- `video` 视频对象
- `id` 视频帧序号

`VOID Arrival(variant video, int id);`

### 5) 触摸捕捉

- `video` 视频对象

`VOID Touch(variant video);`

### 6) 移动检测

`VOID MoveDetec(variant video, int id);`

- `video` 视频对象
- `id` 移动检测计数

## 7) 纠偏

- video 视频对象
- view 窗口对象
- list 纠偏区域列表

```
VOID Deskew(variant video,variant view,variant list);
```

## 8) 获取指纹模板和指纹特征完成

- 9) ret 4 表示获取指纹模板完成, 8 表示获取特征完成,9-14表示指纹刷取状态: 9刷的不错, 10图像质量太差, 11图像点数太少, 12太快, 13太慢, 14其它质量问题

```
VOID Biokey(int ret);
```

## 10) 模板识别完成

- ret 此值为0

```
VOID Templocr(int ret);
```

## 11) 检测到IC卡

- type IC卡类型。 2 表示M1卡, 3 表示Memory卡, 4 表示CPU卡
- subtype 此值为0

```
VOID Reader(int type, int subtype);
```

## 12) 检测到磁条卡

- type 此值为0

```
VOID Mag(int type);
```

## 13) 开始播放视频

- video 视频对象
- videoID 视频索引
- view 窗口对象
- viewID 窗口索引

```
VOID VideoAttach(variant video, int videoID,variant view,  
int viewID);
```

## 附录

### 1) 上传图像文件接口Http\_UploadImageFile 上传数据格式

上传流程:

- (1) 向 HTTP 服务器发送请求 (附带文件头)
- (2) 发送前置数据



- (3) 发送图片文件数据
- (4) 发送后置数据
- (5) 结束 HTTP 请求

文件头:

```
Content-Type: multipart/form-data;
boundary=-----7db29f2140360
```

前置数据:

```
--Content-Type: multipart/form-data;
boundary=-----7db29f2140360
Content-Disposition: form-data; name="file"; filename="1.jpg"
Content-Type:image/jpeg
Content-Transfer-Encoding: binary
```

图片文件数据: 图片文件的二进制数据

后置数据:

```
--Content-Type: multipart/form-data;
boundary=-----7db29f2140360
Content-Disposition: form-data; name="submitted"
submit
--Content-Type: multipart/form-data;
boundary=-----7db29f2140360--
```

## 2) chrome42/43/44启用NPAPI插件功能

多浏览器控仅支持 Chrome 45 以下版本的浏览器, 另外 chrome43、chrome44、要手动启用 NPAPI 插件。

启用方法:

- (1) 地址栏输入 chrome://flags/#enable-npapi 回车, 点“启用”, 重启浏览器

**启用 NPAPI** Mac, Windows

允许使用 NPAPI 插件。自 Chrome 45 起, 我们将不再提供此项以及 NPAPI 支持。

**停用**

- (2) 地址栏输入 chrome:plugins, 启用 eloamplugin, 并勾选“始终运行”

## 3) 多个摄像头不能同时打开

一个USB2.0的最高传输速率为480Mbps, 即60MB/s, 有些电脑配置较低, 可能出现带宽不够而无法同时显示多个头的情况。

解决办法:

- a. 如果不是必须同时显示多个头, 建议只显示一个摄像头, 可以切换显示其它

头。

b. 必须同时显示多个头时，建议降低分辨率打开视频以节省带宽，或者升级电脑硬件配置。

#### 4) 使用MJPG模式无法显示视频或显示为绿色

电脑上缺少高帧率解码器，导致使用MJPG模式无法正常显示。

**解决办法：**

去官网下载“良田高拍仪高速驱动”并安装。

#### 5) 能打开视频，但是无法拍照

某些系统，如win8，系统权限控制的很严格，例如用户无法在在根目录下面新建文件或文件，导致拍照无法保存。

**解决办法：**

a. 以管理员权限运行IE

b. 拍照保存于有读写权限的盘，如D盘