

# CKVisionCtrl.ocx 开发手册

## 一、 函数说明

### 1、 void Redraw()

功能：更新一次显示画面，执行检测后需要更新显示。

### 2、 void ZoomView(long IMode)

功能：放大或缩小图像画面。

参数：

IMode            0 为缩小图像, 1 为放大图像, 2 为自动适应, 3 为 1 倍显示图像。

### 3、 void LockView(BOOL bLock)

功能：锁定视图画面，锁定后不能对视图画面进行操作。

参数：

bLock            TRUE 为锁定, FALSE 为不锁定。

### 4、 BOOL LoadConfigure(LPCTSTR lpszPathName)

功能：装载项目配置文件。

参数：

lpszPathName    项目配置文件全路径名称。

### 5、 BOOL SaveConfigure(LPCTSTR lpszPathName)

功能：保存项目配置到文件中。

参数：

lpszPathName    项目配置文件全路径名称。

### 6、 long GetFirstTool()

功能：获取第一个工具指针地址。

返回： 成功返回工具指针地址，否则返回 NULL。

### 7、 long GetNextTool(long pTool)

功能：获取下一个工具指针地址。

参数：

pTool            输入当前工具指针地址。

返回： 成功返回工具指针地址，否则返回 NULL。

### 8、 long GetTool(LPCTSTR lpszName)

功能：根据工具名称获取工具指针地址。

参数：

lpszName        输入工具名称。

返回： 成功返回工具指针地址，否则返回 NULL。

**9、CString GetToolName(long pTool)**

功能：根据工具指针地址获取工具名称。

参数：

pTool            输入工具指针地址。

返回： 成功返回工具名称，否则返回空字符串。

**10、     BOOL EnableTool(long pTool, BOOL bEnable)**

功能：激活或屏蔽指定工具。

参数：

pTool            输入工具指针地址。

bEnable        TRUE 为激活，FALSE 为屏蔽。

返回： 成功返回 TRUE，否则返回 FALSE。

**11、     BOOL ShowEditor(LPCTSTR lpszTitle, long lStyle)**

功能：弹出流程编辑对话框界面。

参数：

lpszTitle       对话框标题名称。

lStyle           对话框属性风格，默认为 0。

返回： 成功返回 TRUE，否则返回 FALSE。

**12、     BOOL ShowProperty(long pTool)**

功能：弹出工具设置属性对话框。

参数：

pTool            输入工具指针地址。

返回： 点击“确认”按钮返回 TRUE，点击“取消”按钮返回 FALSE。

**13、     BOOL Execute(long lIndex)**

功能：执行一次检测流程。

参数：

lIndex           输入一个可以用于分支工具的索引值，默认为 0。

返回： 成功执行返回 TRUE，否则返回 FALSE。

**14、     BOOL SetValue(long pTool, long IID, long lIndex, VARIANT putVal)**

功能：设置“用户变量”工具中变量的值。

参数：

pTool            输入工具指针地址。

IID              输入变量对应的 ID 值。

lIndex           输入数组变量的索引值。

putVal           输入变得的值，根据变量类型来设置，

返回： 成功执行返回 TRUE，否则返回 FALSE。

**15、     BOOL GetValue(long pTool, long IID, long lIndex, VARIANT \* retVal)**

功能：获取工具中数据的值。

参数:

pTool	输入工具指针地址。
IID	输入数据对应的 ID 值。
lIndex	输入数组变量的索引值。
retVal	返回数据的值，

返回: 成功执行返回 TRUE，否则返回 FALSE。

16、 **long GetImageBuffer(long pTool, long\* Width, long\* Height, long\* chNum)**

功能: 获取工具中图像的数据首地址。

参数:

pTool	输入工具指针地址。
Width	返回图像的宽度。
Height	返回图像的高度。
chNum	返回图像的通道数，灰度图像为 1，彩色图像为 3，

返回: 成功返回指向图像的数据首地址，否则返回 NULL。

17、 **void UnloadConfigure(void)**

功能: 解除加载配置文件，释放当前所占用的资源。

18、 **long GetToolType(long pTool)**

功能: 获取工具的类型标识 ID。

参数:

pTool	输入工具指针地址。
-------	-----------

返回: 成功返回工具的类型标识 ID，否则返回 0。

19、 **BOOL SetImageBuffer(long pTool, long Width, long Height, long chNum, long buf)**

功能: 设置工具中图像的数据缓冲。

参数:

pTool	输入工具指针地址。
Width	输入图像的宽度。
Height	输入图像的高度。
chNum	输入图像的通道数，灰度图像为 1，彩色图像为 3。
buf	输入图像数据缓冲首地址。

返回: 成功返回 TRUE，否则返回 FALSE。

20、 **void SetLanguageKey(LPCTSTR lpszKey)**

功能: 设置当前使用的语言

参数:

lpszKey	输入语言字符串，sch-简体中文，eng-英文。
---------	--------------------------

返回: 无。

21、 **void ViewToImage(double\* x, double\* y)**

功能: 将坐标从控件视图转到图像坐标系

参数:

x            输入/输出 x 轴坐标。  
y            输入/输出 y 轴坐标。  
返回： 无。

22、        void ImageToView (double\* x, double\* y)

功能： 将坐标从图像转到控件视图坐标系

参数：

x            输入/输出 x 轴坐标。  
y            输入/输出 y 轴坐标。

返回： 无。

23、        long GetValID(long pTool, LPCTSTR svName)

功能： 根据数据名称获取数据对应的 ID 值，数据名称会根据软件设置的语言而改变。

参数：

pTool       输入工具指针地址。  
svName      输入数据名称。

返回： 成功返回对应的数据 ID，否则返回 0。

24、        BOOL SaveImage( long pTool, LPCTSTR strPathName, long IFlag)

功能： 以 BMP 格式保存图像到文件。

参数：

pTool       输入带有图像数据工具的指针地址，如果设为 NULL 则保存当前显示的图像。

strPathName 保存图像文件的全路径名。

IFlag       选项标记，低 8 位的值为图像缩小级别，0 为 1 倍大小，1 为 1/2 缩小，2 为 1/4 缩小，3 为 1/8 缩小，带有 0x100 标记为保存带图形图像。

返回： 成功返回对应的数据 ID，否则返回 0。

25、        long GetHaspKeyID(void)

功能： 获取当前加密锁的唯一标识 ID。

返回： 成功返回加密锁 ID，否则返回 0。

26、        VARIANT GetData(long pTool, long nID, long nIndex)

功能： 通过返回值的方式获取工具中数据的值，有些语言无法通过参数返回数据时可用使用，功能和 GetValue 函数一样。

参数：

pTool       输入工具指针地址。  
nID          输入数据 ID，可通过查看工具 ID 对话框获得。  
nIndex      如果数据是数组，则为数组的索引号，否则无效。

返回： 成功返回对应数据的值，否则返回无效值。

## 二、        使用范例

1、获取工具的数据方法

获取数据要使用 `GetValue` 函数，该函数需要输入工具指针和数据 ID，数据 ID 可以在流程编辑对话框中点击工具栏中的“ID”图标查看（参见下图），返回数据使用 `VARIANT` 类型，根据工具内部不同的数据类型，在 `VARIANT` 数据结构中使用对应的数据类型（参见以下表格）。



内部数据类型	VARIANT 对应类型	VARIANT 对应数据
bool	VT_BOOL	boolVal
int	VT_I4	lVal
dou	VT_R8	dblVal
str	VT_BSTR	bstrVal

2、获取“形状匹配”工具的数据

```
// 执行检测
m_ckvisionctrl1.Execute(0);

// 获取工具指针地址
```

```
long pTool = m_ckvisionctrl1.GetTool("形状匹配");
```

```
VARIANT var;
```

```
long count;
```

```
double dx, dy, da;
```

```
// 获取匹配目标数量
```

```
m_ckvisionctrl1.GetValue(pTool,100,0,& var);
```

```
count = var.lVal;
```

```
// 获取匹配位置 x
```

```
m_ckvisionctrl1.GetValue(pTool,102,0,& var);
```

```
dx = var.dblVal;
```

```
// 获取匹配位置 y
```

```
m_ckvisionctrl1.GetValue(pTool,103,0,& var);
```

```
dy = var.dblVal;
```

```
// 获取匹配角度
```

```
m_ckvisionctrl1.GetValue(pTool,104,0,& var);
```

```
da = var.dblVal;
```