

# **CKVisionAppDoNet.dll 接口函数说明**

**Ver3.3.2.3**

**修订时间 2021.8.3**

## **1 类型定义** 5

---

APPROI	5
APPVIEW_WM	5
APPVIEW_MK	6
APPVIEW_PARAM_ID	7
APPPROC_PARAM_ID	7
HVIEW_MSG_CALLBACK	7
HVIEW_DRAW_CALLBACK	8

## **2 CKAPP 主要操作函数** 9

---

INITVISIONAPP	11
EXITVISIONAPP	11
GETHASPKEYID	12
SETLANGUAGE	12
CREATEPROC	13
FREEPROC	13
LOADPROC	13
SAVEPROC	14
GETPROCNAME	14
SETPROCNAME	15
GETPROCPARAM	15
SETPROCPARAM	16
SHOWEDITOR	17
STICKEDITOR	17
EXECUTEPROC	18
ADDTOOL	18
DELETETOOL	19
MOVETOOL	19
GETFIRSTTOOL	20
GETNEXTTOOL	20
GETTOOL	21
GETTOOLTYPE	21
GETTOOLSTYLE	22
ENABLETOOL	22
GETTOOLNAME	23
SETTOOLNAME	23
GETTOOLNOTE	24
SETTOOLNOTE	24
SHOWPROPERTY	24
GETDATAID	25
SETBOOLVALUE	25
GETBOOLVALUE	26

SETINTVALUE	27
GETINTVALUE	27
SETDBLVALUE	28
GETDBLVALUE	29
SETSTRVALUE	29
GETSTRVALUE	30
GETIMAGEBITS	30
SETIMAGEBITS	31
SAVEIMAGEFILE	32
CREATEVIEW	32
MOVEVIEW	33
SHOWVIEW	34
SETVIEWPROC	34
SETVIEWTOOL	35
ZOOMVIEW	35
SETVIEWPARAM	36
GETVIEWPARAM	36
SETMSGCALLBACK	37
SETDRAWCALLBACK	37
VIEWTOIMAGE	38
IMAGETOVIEW	39
REDRAWVIEW	39
FREEVIEW	39

---

### 3 附加视图 ROI 绘制函数（与图像处理无关单独添加显示） 42

GETROI MODE	42
SETROI MODE	43
ADDROI	43
GETROI COUNT	44
GETACTIVEROI	44
SETACTIVEROI	45
GETROI COLOR	45
SETROI COLOR	46
GETROI TYPE	46
DELETEROI	47
DELETEALLROI	47
GETROIDATA	47
SETROIDATA	48
SAVEROIS	49
LOADROIS	49

---

### 附录 1. 50



CKVisionAppDoNet.dll 是 CKVisionBuilder3 的集成开发版本，也是 CKVisionApp.dll 的 DoNet 引用版本，且兼容打开 CKVisionCtrl.ocx 保存的项目文件。

## 1 类型定义

名称	描述
hPROC	流程句柄
hTOOL	工具句柄
hVIEW	视图句柄
h3DVIEW	3D 视图句柄
AppROI	ROI 类型定义
AppVIEW_WM	自定义鼠标消息
AppVIEW_MK	按键被按下的状态
AppVIEW_PARAM_ID	视图参数功能对应 ID
AppPROC_PARAM_ID	流程参数功能对应 ID
hVIEW_MSG_CALLBACK	视图消息回调
hVIEW_DRAW_CALLBACK	视图自定义绘制回调
CKApp	CKVisionBuilder3 开发版调用接口类

### AppROI

类型定义。

名称	值	描述
T_POINT	0	点
T_LINE	1	线段
T_ARC	2	圆弧
T_RECT	3	矩形
T_ROTRECT	4	旋转矩形
T_CIRCLE	5	圆形
T_RING	6	圆环
T_POLYGON	7	多边形

### AppVIEW\_WM

自定义鼠标消息。

名称	值	描述
----	---	----

MOUSEMOVE	WM_MOUSEMOVE	鼠标移动
MOUSEDRAW	WM_MOUSEMOVE	鼠标拖动
LBUTTONDOWN	WM_LBUTTONDOWN	鼠标左键点击
LBUTTONUP	WM_LBUTTONUP	鼠标左键弹起
LBUTTONDBLCLK	WM_LBUTTONDBLCLK	鼠标左键双击
RBUTTONDOWN	WM_RBUTTONDOWN	鼠标右键点击
RBUTTONUP	WM_RBUTTONUP	鼠标右键弹起
RBUTTONDBLCLK	WM_RBUTTONDBLCLK	鼠标右键双击
MBUTTONDOWN	WM_MBUTTONDOWN	当光标在客户区，且鼠标中键被按下时此消息会被发送
MBUTTONUP	WM_MBUTTONUP	当光标在客户区，且鼠标中键被释放时此消息会被发送
MBUTTONDBLCLK	WM_MBUTTONDBLCLK	当光标在客户区，且鼠标中键被双击时此消息会被发送
MOUSEWHEEL	WM_MOUSEWHEEL	当鼠标滑轮滑动时此消息会被发送到焦点窗口
XBUTTONDOWN	WM_XBUTTONDOWN	当光标在客户区，且鼠标第一个或第二个 X 键被按下时此消息会被发送
XBUTTONUP	WM_XBUTTONUP	当光标在客户区，且鼠标第一个或第二个 X 键被释放时此消息会被发送
XBUTTONDBLCLK	WM_XBUTTONDBLCLK	当光标在客户区，且鼠标第一个或第二个 X 键被双击时此消息会被发送
MOUSEHWHEEL	WM_MOUSEHWHEEL	当鼠标横向滑轮滑动时此消息会被发送到焦点窗口

## AppVIEW\_MK

按键被按下的状态。

名称	值	描述
CONTROL	MK_CONTROL	键盘 CTRL 键是按下状态
LBUTTON	MK_LBUTTON	鼠标左键是按下状态
MBUTTON	MK_MBUTTON	鼠标中键是按下状态
RBUTTON	MK_RBUTTON	鼠标右键是按下状态
SHIFT	MK_SHIFT	键盘 SHIFT 键是按下状态
_XBUTTON1	MK_XBUTTON1	鼠标第一个 X 按钮是按下状态
_XBUTTON2	MK_XBUTTON2	鼠标第二个 X 按钮是按下状态

## AppVIEW\_PARAM\_ID

SetViewParam/GetViewParam nID 定义

视图参数对应 ID

名称	值	描述
BKCOLOR	0	背景颜色
ZOOMPOS	1	是否指定缩放视图位置，0-默认视图中心，1-指定视图位置
ZOOMPOS_X	2	指定缩放视图位置点 X
ZOOMPOS_Y	3	指定缩放视图位置点 Y
ENABLE_RMENU	6	0-禁用右键菜单，1-启用右键菜单

## AppPROC\_PARAM\_ID

SetProcParam / GetProcParam nID 定义

流程参数参数对应 ID

名称	值	描述
EXCINDEX	0	编辑流程执行按钮对应执行的分支索引参数

## hVIEW\_MSG\_CALLBACK

视图消息回调。

```
public delegate void hVIEW_MSG_CALLBACK(UINT message, IntPtr wParam, IntPtr lParam);
```

参数

**message**

当前响应的消息值。

**wParam**

消息附带的参数，参见 Windows 消息机制说明。

**lParam**

消息附带的参数，参见 Windows 消息机制说明。

**返回值**

无。

## hVIEW\_DRAW\_CALLBACK

视图自定义绘制回调。

```
public delegate void hVIEW_DRAW_CALLBACK(IntPtr hDC, int x, int y, int width, int height, double scale);
```

**参数**

**hDC**

视图的绘图 DC 句柄。

**x**

视图的左上角起始点 **x**。

**y**

视图的左上角起始点 **y**。

**width**

指定视图的宽度（以像素位单位）。

**height**

指定视图的高度（以像素位单位）。

**scale**

视图显示图形缩放比率。

**返回值**



无。

## 2 CKApp 主要操作函数

名称	描述
InitVisionApp	初始化 CKVision 库
ExitVisionApp	退出 CKVision 库
GetHaspKeyID	获取当前使用的加密锁 ID
SetLanguage	设置软件的语言标识
CreateProc	创建一个新的流程
FreeProc	释放流程句柄，删除流程中的所有工具并释放内存
LoadProc	从文件加载流程数据（自动删除流程的当前数据）
SaveProc	将流程数据保存到文件
GetProcName	获取流程名称
SetProcName	设置流程名称
GetProcParam	获取流程对应功能 ID 的参数
SetProcParam	设置流程对应功能 ID 的参数
ShowEditor	弹出流程编辑对话框
ExecuteProc	执行流程
AddTool	添加工具
DeleteTool	删除工具
MoveTool	移动工具位置
GetFirstTool	获取流程的第一个工具
GetNextTool	获取下一个工具
GetTool	根据工具名称获取工具句柄指针
GetToolType	获取工具类型 ID
GetToolStyle	获取工具风格选项
EnableTool	激活或禁用工具

GetToolName	获取工具名称
SetToolName	设置工具名称
GetToolNote	获取工具注释
SetToolNote	设置工具注释
ShowProperty	弹出工具属性对话框
StickEditor	将流程编辑对话框贴到指定区域
GetDataID	根据数据名称（不同语言名称不同）获取数据 ID
SetBoolValue	设置工具布尔值数据
GetBoolValue	获取工具布尔值数据
SetIntValue	设置工具整数值数据
GetIntValue	获取工具整数值数据
SetDbfValue	设置工具双精度浮点数数据
GetDbfValue	获取工具双精度浮点数数据
SetStrValue	设置工具字符串数据
GetStrValue	获取工具字符串数据
GetImageBits	获取工具图像数据地址
SetImageBits	设置工具图像数据地址
SaveImageFile	保存工具图像数据到文件
CreateView	创建显示图像和图形视图
MoveView	移动视图到指定位置
ShowView	显示或隐藏视图
SetViewProc	设置视图当前显示的流程图像（在编辑流程对话框中显示的图像）
SetViewTool	设置视图当前显示的图像来源工具
ZoomView	缩放视图操作
SetViewParam	设置视图对应参数
GetViewParam	获取视图对应参数
SetMsgCallback	设置视图消息回调函数
SetDrawCallback	设置视图用户自定义绘制回调函数
ViewToImage	视图坐标转换到图像坐标（视图当前显示的图像）

ImageToView	图像坐标（视图当前显示的图像）转换到视图坐标
RedrawView	立即刷新视图显示
FreeView	释放视图数据
Create3DView	创建显示 3D 视图
Move3DView	移动 3D 视图到指定位置
Show3DView	显示或者隐藏 3D 视图
Set3DViewProc	设置 3D 视图当前显示的流程图像（在编辑流程对话框中显示的图像）
Set3DViewTool	设置 3D 视图当前显示的图像来源工具
Redraw3DView	立即刷新 3D 视图显示
Free3DView	释放 3D 视图数据

## InitVisionApp

初始化 CKVision 库，进入程序时调用一次。

```
static void InitVisionApp(void);
```

### 参数

无。

### 返回值

无。

## ExitVisionApp

退出 CKVision 库，退出程序时调用一次。

```
static void ExitVisionApp(void);
```

### 参数

无。

### 返回值

无。

## GetHaspKeyID

获取当前使用的加密锁 ID。

```
static UINT GetHaspKeyID();
```

### 参数

无。

### 返回值

加密锁唯一 ID 标识。

## SetLanguage

设置软件的语言标识。

```
static void SetLanguage(String^ strLangKey);
```

### 参数

**strLangKey**

输入语言标识字符串,根据 Language 目录下的语言子文件夹名称设置。

### 返回值

空。

## CreateProc

创建一个新的流程。

```
static hPROC CreateProc(String^ strProcName);
```

### 参数

**strProcName**

输入流程名称标识。

### 返回值

如果成功，返回流程句柄指针；否则返回 0。

## FreeProc

释放流程句柄，删除流程中的所有工具并释放内存。

```
static void FreeProc(hPROC hProc);
```

### 参数

**hProc**

输入需要释放的流程句柄指针。

### 返回值

空。

## LoadProc

从文件加载流程数据（自动删除流程的当前数据）。

```
static bool LoadProc(hPROC hProc, String^ strFilePath);
```

### 参数

**hProc**

输入已创建流程句柄指针。

**strFilePath**

输入流程文件路径名称。

### 返回值

如果成功，返回 **true**；否则返回 **false**。

## SaveProc

将流程数据保存到文件。

```
static bool SaveProc(hPROC hProc, String^ strFilePath);
```

### 参数

**hProc**

输入需要保存的流程句柄指针。

**strFilePath**

输入流程文件路径名称。

### 返回值

如果成功，返回 **true**；否则返回 **false**。

## GetProcName

获取流程名称。

```
static String^ GetProcName(hPROC hProc);
```

### 参数

**hProc**

输入已创建流程句柄指针。

### 返回值

如果成功，返回流程名称字符串；否则返回 0。

## SetProcName

设置流程名称。

```
static bool SetProcName(hPROC hProc, String^ strProcName);
```

### 参数

**hProc**

输入已创建流程句柄指针。

**strProcName**

输入新的流程名称字符串。

### 返回值

如果成功，返回 **true**；否则返回 **false**。

## GetProcParam

获取流程参数。

```
static int GetProcParam(hPROC hProc, int nParam);
```

### 参数

**hProc**

输入已创建流程句柄指针。

nParam

输入流程参数标识。

### 返回值

如果成功，回流程参数值；失败返回-1 。

## SetProcParam

设置流程参数。

```
static bool SetProcParam(hPROC hProc, int nParam, int nValue)
```

### 参数

hProc

输入已创建流程句柄指针。

nParam

输入流程参数对应功能标识 ID。

nValue

输入流程参数值。

### 返回值

如果成功，返回 **true**； 否则返回 **false**。



## ShowEditor

弹出流程编辑对话框。

```
static bool ShowEditor(hPROC hProc, IntPtr hWnd, UINT nStyle);
```

### 参数

**hProc**

输入流程句柄指针。

**hWnd**

输入父级窗口句柄。

**nStyle**

输入窗口风格标记，未使用。

### 返回值

如果成功，返回 **true**；否则返回 **false**。

## StickEditor

弹出流程编辑对话框。

```
static bool StickEditor( HPROC hProc, HWND hWnd, int left, int top, int width, int height );
```

### 参数

**hProc**

输入流程句柄指针。

**hWnd**

输入父级窗口句柄。

**left**

输入相对于父窗口的左边位置。

**top**

输入相对于父窗口的上边位置。

**width**

输入程编辑对话框宽度。

**height**

输入程编辑对话框高度。

### 返回值

如果成功，返回 **true**；否则返回 **false**。

## ExecuteProc

执行流程。

```
static bool ExecuteProc(hPROC hProc, int nMsgID);
```

### 参数

**hProc**

输入已创建流程句柄指针。

**nMsgID**

输入用于流程内部分支的索引值。

### 返回值

如果成功，返回 **true**；否则返回 **false**。

## AddTool

添加工具

```
static bool AddTool(hPROC hProc, UINT nToolID, String^ strToolName);
```

### 参数

**hProc**

输入流程句柄指针。

**nToolID**

输入工具类型 ID。

**strToolName**

输入工具名称。

### 返回值

如果成功，返回 **true**；否则返回 **false**。

## DeleteTool

删除工具。

```
static bool DeleteTool(hPROC hProc, String^ strToolName);
```

### 参数

**hProc**

输入流程句柄指针。

**strToolName**

输入工具名称。

### 返回值

如果成功，返回 **true**；否则返回 **false**。

## MoveTool

移动工具位置。

```
static bool MoveTool(hTOOL hTool, int direction);
```

## 参数

**hTool**

输入工具句柄指针。

**direction**

移动方向，0-向上；1-向下。

## 返回值

如果成功，返回 **true**；否则返回 **false**。

# GetFirstTool

获取流程的第一个工具。

```
static hTOOL GetFirstTool(hPROC hProc);
```

## 参数

**hProc**

输入流程句柄指针。

## 返回值

如果成功，返回第一个工具句柄指针；否则返回 **0**。

# GetNextTool

获取下一个工具。

```
static hTOOL GetNextTool(hPROC hProc, hTOOL hTool);
```

## 参数

**hProc**

输入流程句柄指针。

**hTool**

输入当前工具句柄指针。

### 返回值

如果成功，返回下一个工具句柄指针；否则返回 0。

## GetTool

根据工具名称获取工具句柄指针。

```
static hTOOL GetTool(hPROC hProc, String^ strToolName);
```

### 参数

**hProc**

输入流程句柄指针。

**strToolName**

输入工具名称。

### 返回值

如果成功，返回工具句柄指针；否则返回 0 。

## GetToolType

获取工具类型 ID；创建工具时定义的 ID，同一种工具添加多次到流程中的 ID 值一样。

```
static UINT GetToolType(hTOOL hTool);
```

### 参数

**hTool**

输入工具句柄指针。

### 返回值

如果成功，返回工具类型 ID；否则返回 0 。

## GetToolStyle

获取工具风格选项。

```
static UINT GetToolStyle(hTOOL hTool);
```

### 参数

**hTool**

输入工具句柄指针。

### 返回值

如果成功，返回工具风格（0x01 激活，0x02 快捷）；否则返回 0 。

## EnableTool

激活或禁用工具。

```
static bool EnableTool(hTOOL hTool, bool bEnable);
```

### 参数

**hTool**

输入工具句柄指针。

**bEnable**

输入 true 为激活工具，false 为禁用工具。

### 返回值

如果成功，返回 true；否则返回 false。

## GetToolName

获取工具名称。

```
static String^ GetToolName(hTOOL hTool);
```

### 参数

**hTool**

输入工具句柄指针。

### 返回值

如果成功，返回工具名称，失败返回 0 。

## SetToolName

设置工具名称。

```
static bool SetToolName(hTOOL hTool, String^ strToolName);
```

### 参数

**hTool**

输入工具句柄指针。

**strToolName**

输入新的工具名称。

### 返回值

如果成功，返回 **true**；否则返回 **false** 。

## GetToolNote

获取工具注释。

```
static String^ GetToolNote(hTOOL hTool);
```

### 参数

**hTool**

输入工具句柄指针。

### 返回值

如果成功，返回工具注释；否则返回 0 。

## SetToolNote

设置工具注释。

```
static bool SetToolNote(hTOOL hTool, String^ strToolNote);
```

### 参数

**hTool**

输入工具句柄指针。

**strToolNote**

输入新的工具注释。

### 返回值

如果成功，返回 true；否则返回 false 。

## ShowProperty

弹出工具属性对话框。

```
static bool ShowProperty(hTOOL hTool, IntPtr hWnd);
```



## 参数

**hTool**

输入工具句柄指针。

**hWnd**

输入父级窗口句柄指针。

## 返回值

点击“确认”返回 **true**，点击“取消”返回 **false** 。

# GetDataID

根据数据名称（不同语言名称不同）获取数据 ID 。

```
static int GetDataID(hTOOL hTool, String^ strDataName);
```

## 参数

**hTool**

输入工具句柄指针。

**strDataName**

输入数据名称。

## 返回值

如果成功，返回数据 ID；否则返回 0 。

# SetBoolValue

设置工具布尔值数据。

```
static bool SetBoolValue(hTOOL hTool, int dataID, int index, bool srcVal);
```

## 参数

**hTool**

输入工具句柄指针。

**dataID**

输入工具的数据 ID，可在流程编辑界面中查看。

**index**

输入数组数据的索引，非数组数据填 0 。

**srcVal**

输入布尔值数据的值。

## 返回值

如果成功，返回 **true**；否则返回 **false**；失败原因可能是工具句柄或者数据 ID 不正确，或者该数据不支持设置操作，或者数组数据的索引超出范围。

## GetBoolValue

获取工具布尔值数据。

```
static bool GetBoolValue(hTOOL hTool, int dataID, int index, bool% retVal);
```

## 参数

**hTool**

输入工具句柄指针。

**dataID**

输入工具的数据 ID，可在流程编辑界面中查看。

**index**

输入数组数据的索引，非数组数据填 0 。

**retVal**

返回布尔值数据。

## 返回值

如果成功，返回 **true**；否则返回 **false** 。

## SetIntValue

设置工具整数值数据。

```
static bool SetIntValue(hTOOL hTool, int dataID, int index, int srcVal);
```

### 参数

**hTool**

输入工具句柄指针。

**dataID**

输入工具的数据 ID，可在流程编辑界面中查看。

**index**

输入数组数据的索引，非数组数据填 0 。

**srcVal**

输入整数数据的值。

### 返回值

如果成功，返回 **true**；否则返回 **false**；失败原因可能是工具句柄或者数据 ID 不正确，或者该数据不支持设置操作，或者数组数据的索引超出范围 。

## GetIntValue

获取工具整数值数据。

```
static bool GetIntValue(hTOOL hTool, int dataID, int index, int% retVal);
```

### 参数

**hTool**

输入工具句柄指针。

**dataID**

输入工具的数据 ID，可在流程编辑界面中查看。

**index**

输入数组数据的索引，非数组数据填 0 。

**retVal**

返回整数数据。

### 返回值

如果成功，返回 **true**；否则返回 **false** 。

## SetDbfValue

设置工具双精度浮点数数据。

```
static bool SetDbfValue(hTOOL hTool, int dataID, int index, double srcVal);
```

### 参数

**hTool**

输入工具句柄指针。

**dataID**

输入工具的数据 ID，可在流程编辑界面中查看。

**index**

输入数组数据的索引，非数组数据填 0 。

**srcVal**

输入双精度浮点数数据的值。

### 返回值

如果成功，返回 **true**；否则返回 **false**；失败原因可能是工具句柄或者数据 ID 不正确，或者该数据不支持设置操作，或者数组数据的索引超出范围。

## GetDbIValue

获取工具双精度浮点数数据。

```
static bool GetDbIValue(hTOOL hTool, int dataID, int index, double% retVal);
```

### 参数

**hTool**

输入工具句柄指针。

**dataID**

输入工具的数据 ID，可用在流程编辑界面中查看。

**index**

输入数组数据的索引，非数组数据填 0 。

**retVal**

返回双精度浮点数数据。

### 返回值

如果成功，返回 **true**；否则返回 **false** 。

## SetStrValue

设置工具字符串数据

```
static bool SetStrValue(hTOOL hTool, int dataID, int index, String^ srcVal);
```

### 参数

**hTool**

输入工具句柄指针。

**dataID**

输入工具的数据 ID，可在流程编辑界面中查看。

**index**

输入数组数据的索引，非数组数据填 0 。

srcVal

输入字符串数据。

## 返回值

如果成功，返回 **true**；否则返回 **false**；失败原因可能是工具句柄或者数据 ID 不正确，或者该数据不支持设置操作，或者数组数据的索引超出范围。

## GetStrValue

获取工具字符串数据。

```
static String^ GetStrValue(hTOOL hTool, int dataID, int index);
```

### 参数

hTool

输入工具句柄指针。

dataID

输入工具的数据 ID，可在流程编辑界面中查看。

index

输入数组数据的索引，非数组数据填 0 。

## 返回值

如果成功，返回字符串数据；否则返回空 。

## GetImageBits

获取工具图像数据地址。

```
static IntPtr GetImageBits(hTOOL hTool, int% format, int% width, int% height);
```

### 参数

**hTool**

输入工具句柄指针。

**format**

返回图像格式，(8 位灰度图)，3 (24 位彩色图)。

**width**

返回图像宽度（单位像素）。

**height**

返回图像高度（单位像素）。

## 返回值

如果成功，返回图像数据首地址；否则返回 NULL 。

## SetImageBits

设置工具图像数据地址。

```
static bool SetImageBits(hTOOL hTool, int format, int width, int height, IntPtr buf);
```

## 参数

**hTool**

输入工具句柄指针，该工具必须有输出图像。

**format**

输入图像格式，1 (8 位灰度图)，3 (24 位彩色图)。

**width**

输入图像宽度（单位像素）。

**height**

输入图像高度（单位像素）。

**buf**

输入图像数据首地址(.Scan0)。

## 返回值

如果成功，返回 **true**；否则返回 **false** 。

## SaveImageFile

保存工具图像数据到文件。

```
static bool SaveImageFile(hTOOL hTool, String^ strFilePath, int flag);
```

### 参数

#### hTool

输入工具句柄指针，该工具必须有输出图像。

#### strFilePath

输入文件全路径名称。

#### flag

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，带有 0x200 标记为保存 JPG 格式图像。

### 返回值

如果成功，返回 **true**；否则返回 **false** 。

## CreateView

创建显示图像和图形视图。

```
static hVIEW CreateView(IntPtr hWnd, int left, int top, int right, int bottom);
```

### 参数

#### hWnd

输入父级窗口句柄指针。



**left**

输入视图相对于父窗口的左边位置。

**top**

输入视图相对于父窗口的上边位置。

**right**

输入视图相对于父窗口的右边位置。

**bottom**

输入视图相对于父窗口的下边位置。

**返回值**

如果成功，返回新的视图句柄；否则返回空。

## MoveView

移动视图到指定位置。

```
static bool MoveView(hVIEW hView, int left, int top, int right, int bottom);
```

**参数**

**hView**

输入视图句柄指针。

**left**

输入视图相对于父窗口的左边位置。

**top**

输入视图相对于父窗口的上边位置。

**right**

输入视图相对于父窗口的右边位置。

**bottom**

输入视图相对于父窗口的下边位置。

**返回值**

如果成功返回 **true**； 否则返回 **false** 。

## ShowView

显示或者隐藏视图。

```
bool ShowView( HVIEW hView, bool bShow );
```

### 参数

**hView**

输入视图句柄。

**bShow**

true为显示，false为隐藏。

如果成功，返回 **true**； 否则返回 **false** 。

## SetViewProc

设置视图当前显示的流程图像（在编辑流程对话框中显示的图像）。

```
static bool SetViewProc(HVIEW hView, HPROC hProc);
```

### 参数

**hView**

输入视图句柄指针。

**hProc**

输入流程句柄指针。

### 返回值

如果成功，返回 **true**； 否则返回 **false** 。

## SetViewTool

设置视图当前显示的图像来源工具。

```
static bool SetViewTool(hVIEW hView, hTOOL hTool);
```

### 参数

**hView**

输入视图句柄指针。

**hTool**

输入工具句柄，该工具必须有输出图像。

### 返回值

如果成功，返回 **true**；否则返回 **false**。

## ZoomView

缩放视图操作

```
static bool ZoomView(hVIEW hView, int nMode );
```

### 参数

**hView**

输入视图句柄指针。

**nMode**

缩放模式，0-放大，1-缩小，2-自适应，3-还原，4-居中。

## 返回值

如果成功，返回 **true**； 否则返回 **false**。

## SetViewParam

设置视图对应参数

```
static bool SetViewParam(hVIEW hView, int nID, int nValue);
```

### 参数

#### hView

输入视图句柄指针。

#### nID

输入参数 ID，0-背景颜色，1-指定缩放视图位置，2-指定缩放位置点 X，3-指定缩放位置点 Y。

#### nValue

输入参数值。

## 返回值

如果成功，返回 **true**； 否则返回 **false**。

## GetViewParam

获取视图对应参数

```
static bool GetViewParam(hVIEW hView, int nID, int% retVal);
```

### 参数

**hView**

输入视图句柄指针。

**nID**

输入参数 ID，0-背景颜色，1-指定缩放视图位置，2-指定缩放位置点 X，3-指定缩放位置点 Y。

**retVal**

返回参数值。

## 返回值

如果成功，返回 **true**； 否则返回 **false**。

## SetMsgCallback

设置视图消息回调函数。

```
static bool SetMsgCallback(hVIEW hView, hVIEW_MSG_CALLBACK^ lpMsgCallback);
```

## 参数

**hView**

输入视图句柄指针。

**lpMsgCallback**

输入消息回调函数指针。

## 返回值

如果成功，返回 **true**； 否则返回 **false**。

## SetDrawCallback

设置视图用户自定义绘制回调函数

```
static bool SetDrawCallback(HVIEW hView, HVIEW_DRAW_CALLBACK^ lpDrawCallback);
```

### 参数

**hView**

输入视图句柄指针。

**lpDrawCallback**

输入自定义绘制回调函数指针。

### 返回值

如果成功，返回 **true**； 否则返回 **false**。

## ViewToImage

视图坐标转换到图像坐标（视图当前显示的图像）。

```
static bool ViewToImage(HVIEW hView, double% x, double% y);
```

### 参数

**hView**

输入视图句柄指针。

**x**

输入需要转换的 **x** 坐标数据，成功直接修改该坐标数据。

**y**

输入需要转换的 **y** 坐标数据，成功直接修改该坐标数据。

### 返回值

如果成功，返回 **true**； 否则返回 **false** 。

## ImageToView

图像坐标（视图当前显示的图像）转换到视图坐标。

```
static bool ImageToView(hVIEW hView, double% x, double% y);
```

### 参数

**hView**

输入视图句柄。

**x**

输入需要转换的 x 坐标数据，成功直接修改该坐标数据。

**y**

输入需要转换的 y 坐标数据，成功直接修改该坐标数据。

### 返回值

如果成功，返回 **true**；否则返回 **false** 。

## RedrawView

立即刷新视图显示。

```
static void RedrawView(hVIEW hView);
```

### 参数

**hView**

输入视图句柄指针。

### 返回值

空。

## FreeView

释放视图数据。

```
static void FreeView(hVIEW hView);
```

### 参数

**hView**

输入视图句柄指针。

### 返回值

空。

## Create3DView

创建显示 3D 视图。

```
static h3DVIEW Create3DView (IntPtr hWnd, int left, int top, int right, int bottom);
```

### 参数

**hWnd**

输入父级窗口句柄指针。

**left**

输入视图相对于父窗口的左边位置。

**top**

输入视图相对于父窗口的上边位置。

**right**

输入视图相对于父窗口的右边位置。



**bottom**

输入视图相对于父窗口的下边位置。

## 返回值

如果成功，返回新的视图句柄；否则返回空。

**Move3DView**

```
static bool Move3DView (hVIEW h3DView, int left, int top, int right, int bottom);
```

## 参数

**hView**

输入视图句柄指针。

**left**

输入视图相对于父窗口的左边位置。

**top**

输入视图相对于父窗口的上边位置。

**right**

输入视图相对于父窗口的右边位置。

**bottom**

输入视图相对于父窗口的下边位置。

## 返回值

如果成功返回 **true**； 否则返回 **false** 。

### 3 附加视图 ROI 绘制函数(与图像处理无关单独添加显示)

名称	描述
GetRoiMode	获取视图是否为 ROI 交互模式
SetRoiMode	将视图设置成 ROI 交互模式
AddRoi	添加 ROI 图形到视图
GetRoiCount	获取视图中 ROI 图形的数量
GetActiveRoi	获取视图中当前选取 ROI 图形的索引
SetActiveRoi	设置视图中当前选取的 ROI 图形
GetRoiColor	获取视图中指定 ROI 图形的颜色
SetRoiColor	设置视图中指定 ROI 图形的颜色
GetRoiType	获取视图中 ROI 图形的类型
DeleteRoi	删除指定 ROI 图形
DeleteAllRoi	删除所有 ROI 图形
GetRoiData	获取 ROI 图形数据
SetRoiData	设置 ROI 图形数据
SaveRoIs	保存所有 ROI 图形数据到文件
LoadRoIs	从文件加载 ROI 图形数据

#### GetRoiMode

获取图是否为 ROI 交互模式。

```
static bool GetRoiMode(hVIEW hView);
```

#### 参数

hView

输入视图句柄指针。

### 返回值

ROI 交互模式，返回 **true**；否则返回 **false** 。

## SetRoiMode

将视图设置成 ROI 交互模式,ROI 是否可操作（拖动调节移动等）。

```
static bool SetRoiMode(hVIEW hView, bool bVal);
```

### 参数

**hView**

输入视图句柄指针。

**bVal**

**true** 为 ROI 交互模式（可操作），鼠标左键拖动图像功能被禁止；**false** 为禁止操作 ROI，鼠标左键可以拖动放大后的图。

### 返回值

如果成功，返回 **true**，否则返回 **false**。

## AddRoi

添加 ROI 图形到视图。

```
static bool AddRoi(hVIEW hView, int nRoiType);
```

### 参数

**hView**

输入视图句柄指针。

**nRoiType**

需要添加的 ROI 类型（参加 ROI 类型定义）。

### 返回值

如果成功，返回 **true**，否则返回 **false**。

## GetRoiCount

获取视图中 ROI 图形的数量。

```
static int GetRoiCount(hVIEW hView);
```

### 参数

**hView**

输入视图句柄指针。

### 返回值

返回 ROI 图形数量。

## GetActiveRoi

获取视图中当前选取 ROI 图形索引。

```
static int GetActiveRoi(hVIEW hView);
```

### 参数

**hView**

输入视图句柄指针。

### 返回值

返回当前选取 ROI 图形索引。

## SetActiveRoi

设置视图中当前选取的 ROI 图形。

```
static bool SetActiveRoi(hVIEW hView, int nIndex);
```

### 参数

**hView**

输入视图句柄指针。

**nIndex**

设置需要激活的 ROI 图形索引。

### 返回值

如果成功，返回 **true**，否则返回 **false**。

## GetRoiColor

获取视图中指定 ROI 图形的颜色。

```
static int GetRoiColor(hVIEW hView, int nIndex);
```

### 参数

**hView**

输入视图句柄指针。

**nIndex**

输入需要获取颜色的 ROI 图形索引。

### 返回值

如果成功，返回 ROI 图形的颜色 RGB 值；否则返回 -1 。

## SetRoiColor

设置视图中指定 ROI 图形的颜色。

```
static bool SetRoiColor(hVIEW hView, int nIndex, int nColor);
```

### 参数

**hView**

输入视图句柄指针。

**nIndex**

输入需要设置的 ROI 图形索引。

**nColor**

输入 ROI 图形的颜色 RGB 值。

### 返回值

如果成功，返回 **true**；否则返回 **false**。

## GetRoiType

获取视图中 ROI 图形的类型。

```
static int GetRoiType(hVIEW hView, int nIndex);
```

### 参数

**hView**

输入视图句柄指针。

**nIndex**

输入需要获取的 ROI 图形索引。

### 返回值

返回 ROI 图形类型（参见 ROI 类型定义）。

## DeleteRoi

删除指定 ROI 图形。

```
static bool DeleteRoi(hVIEW hView, int nIndex);
```

### 参数

**hView**

输入视图句柄指针。

**nIndex**

输入需要删除 ROI 图形的索引。

### 返回值

如果成功，返回 **true**；否则返回 **false** 。

## DeleteAllRoi

删除所有 ROI 图形。

```
static bool DeleteAllRoi(hVIEW hView);
```

### 参数

**hView**

输入视图句柄指针。

### 返回值

如果成功，返回 **true**；否则返回 **false** 。

## GetRoiData

获取 ROI 图形数据。

```
static bool GetRoiData(hVIEW hView, int nIndex, int dataID, double% retVal);
```

### 参数

**hView**

输入视图句柄指针。

**nIndex**

输入 ROI 图形的索引。

**dataID**

输入需要获取 ROI 图形数据的索引，根据 ROI 类型不同而不同，参见附录 1。

**retVal**

返回 ROI 图形对于 dataID 的数据。

### 返回值

如果成功，返回 true；否则返回 false 。

## SetRoiData

设置 ROI 图形数据。

```
static bool SetRoiData(hVIEW hView, int nIndex, int dataID, double srcVal);
```

### 参数

**hView**

输入视图句柄指针。

**nIndex**

输入 ROI 图形的索引。

**dataID**

输入需要获取 ROI 图形数据的索引，根据 ROI 类型不同而不同，参见附录 1。

**srcVal**

输入 ROI 图形对于 dataID 的数据。



## 返回值

如果成功，返回 **true**； 否则返回 **false** 。

## SaveRois

保存所有 ROI 图形数据到文件。

```
static bool SaveRois(hVIEW hView, String^ strPathName);
```

### 参数

**hView**

输入视图句柄指针。

**strPathName**

文件全路径名称。

## 返回值

如果成功，返回 **true**； 否则返回 **false** 。

## LoadRois

从文件加载 ROI 图形数据。

```
static bool LoadRois(hVIEW hView, String^ strPathName);
```

### 参数

**hView**

输入视图句柄指针。

**strPathName**

文件全路径名称。

## 返回值

如果成功，返回 **true**；否则返回 **false** 。

## 附录 1.

例如

**WM\_LBUTTONDOWN**

产生：单击。

参数

**wParam**

包含一个整数值，以标识鼠标键的按下状态。

鼠标的按下标识及说明。

数值	描述
----	----

<b>MK_LBUTTON</b>	标识按下鼠标左键。
-------------------	-----------

<b>MK_MBUTTON</b>	标识按下鼠标中键。
-------------------	-----------

<b>MK_RBUTTON</b>	标识按下鼠标右键。
-------------------	-----------

**lParam**

长参数 **lParam** 的低字位包含当前坐标的 **x** 坐标,在高字节包含当前坐标的 **y** 坐标。

相似的消息有

**WM\_LBUTTONUP**

**WM\_RBUTTONDOWN**

**WM\_RBUTTONUP**

**WM\_LBUTTONDBLCLK**

**WM\_RBUTTONDBLCLK**

## ROI 数据 ID 定义

类型	dataID 值	描述
点	0	坐标X
	1	坐标Y
线段	0	起点X
	1	起点Y
	2	终点X
	3	终点Y
圆弧	0	中心X
	1	中心Y
	2	半径
	3	起始角度
	4	角度范围
矩形	0	左边
	1	上边
	2	右边
	3	下边
旋转矩形	0	中心X
	1	中心Y
	2	宽度
	3	高度
	4	角度
圆形	0	中心X
	1	中心Y
	2	半径
圆环	0	中心X
	1	中心Y
	2	内圆半径
	3	半径范围
	4	起始角度
	5	角度范围
多边形	0	顶点1X
	1	顶点1Y

	2	顶点2X
	3	顶点2Y
	4	顶点3X
	5	顶点3Y
	...	...
	...	...