

# clsWindow

VB操作第三方窗口最好的框架

## 使用手册

sysdzw

# 目 录

## 关于clsWindow

### 一、得到窗口

GetWindowByTitle 函数  
GetWindowByTitleEx 函数  
GetWindowByClassName 函数  
GetWindowByClassNameEx 函数  
GetWindowByPoint 函数  
GetWindowByAppName 函数  
GetWindowByAppNameEx 函数  
GetWindowByCursorPos 函数  
GetWindowByPID 函数  
GetWindowByHwnd 函数  
Load 函数 ( 同GetWindowByHwnd )

### 二、操作窗口

Focus 方法  
SetFocus 方法 ( 同Focus )  
FadeIn 方法  
FadeOut 方法  
Transparent 函数  
SetTransparent 函数 ( 同Transparent )  
MakeTransparent 函数 ( 同Transparent )  
Shake 方法  
CloseWindow 函数  
ClassName 属性  
Caption 属性  
Text 属性 ( 同Caption )  
Title 属性 ( 同Caption )  
Value 属性 ( 同Caption )  
ZOrder 方法  
Left 属性  
Top 属性  
Width 属性  
Height 属性  
Visible 属性  
Enabled 属性  
WindowState 属性  
AlphaBlend 属性  
TransparentColor 属性  
CheckWindow 函数

Move 函数  
MoveToCenter 函数  
Show 函数  
Hide 函数  
Refresh 方法  
SetPosBottom 函数  
SetPosFront 函数  
SetPosNormal 函数  
SetTop 函数  
IsTopMost 函数  
IsForegroundWindow 属性  
Maximized 函数  
Minimized 函数  
Normal 函数

### 三、操作控件

GetElementHwndByText 函数  
GetElementHwndByClassName 函数  
GetElementTextByText 函数  
GetElementTextByClassName 函数  
GetClassNameByHwnd 函数  
GetTextByHwnd 函数  
GetCaptionByHwnd 函数 ( 同GetTextByHwnd )  
SetElementTextByText 函数  
SetElementTextByClassName 函数  
SelectComboBoxIndex 函数  
AppendElementTextByClassName 函数  
Click 函数  
ClickElementByText 函数  
ClickElementByClassName 函数

### 四、鼠标坐标相关

ClickCurrentPoint 方法  
ClickPoint 方法  
SetCursor 方法  
MoveCursor 方法 ( 同SetCursor )  
MoveCursorTo 方法 ( 同SetCursor )  
SetPoint 方法 ( 同SetCursor )  
GetCursorPoint 函数  
GetCursorPosCurrent 函数 ( 同GetCursorPoint )

### 五、应用程序相关

AppPath 函数  
AppName 函数  
IsWin64 函数

CommandLine 函数

AppCommandLine 函数 ( 同CommandLine )

CloseApp 函数

ProcessID 函数

PID 函数 ( 同ProcessID )

## 六、公有方法属性等

hWnd 公有变量

DebugMe 公有变量

Wait 方法

Version 方法

## 使用范例

01 控制微信发送消息

02 控制QQ发送消息

03 QQ登录器

04 花式调用计算器

05 用鼠标画圆

06 控制记事本输入并保存

07 随机移动鼠标防锁屏

08 根据ID进入腾讯课堂

09 限制程序重复打开

10 实战：微便签-窗口特效

11 实战：窗口图钉-窗口置顶

## 一些Q&A

01 目标窗体中存在多个相同类名的输入框该如何区分填写？

## 联系我们

# 关于clsWindow

---

clsWindow(VB6窗口操作类)是VB6环境下使用的一个操作外部程序窗口的类，当前最新版本为v2.2。

本框架封装了大量的api函数，通过这些函数可以方便地操作对外部程序，例如得到窗口句柄，得到窗口里某个文本框的内容，以及设置内容。非常的方便，使用它可以让您脱身于一堆api函数，功能强大使用简单！

clsWindow也可用于VBA，只要在vba中导入modWindow.bas和clsWindow.cls两个文件即可，这里作者已经做好了样例文件，可以在顶部github里面下载，不过只能适用于32位的office，而移植到64位的office需要修改不少地方，有精力的同学可以试下看看。

这个类作者很早就开始封装了，原本打算做成类似DOM对象那样，通过一堆getElementByXXX等方法实现对桌面程序下各个窗口以及里面各个控件对象的自由访问，但是具体要做的工作太多，目前只实现了一部分，期待大家一起加入更新维护。

目前该类封装了绝大部分对windows窗口的常用操作，例如：获取窗口句柄，设置窗口为活动窗口，设置窗口内文本框内容，点击窗口内的某些按钮等。

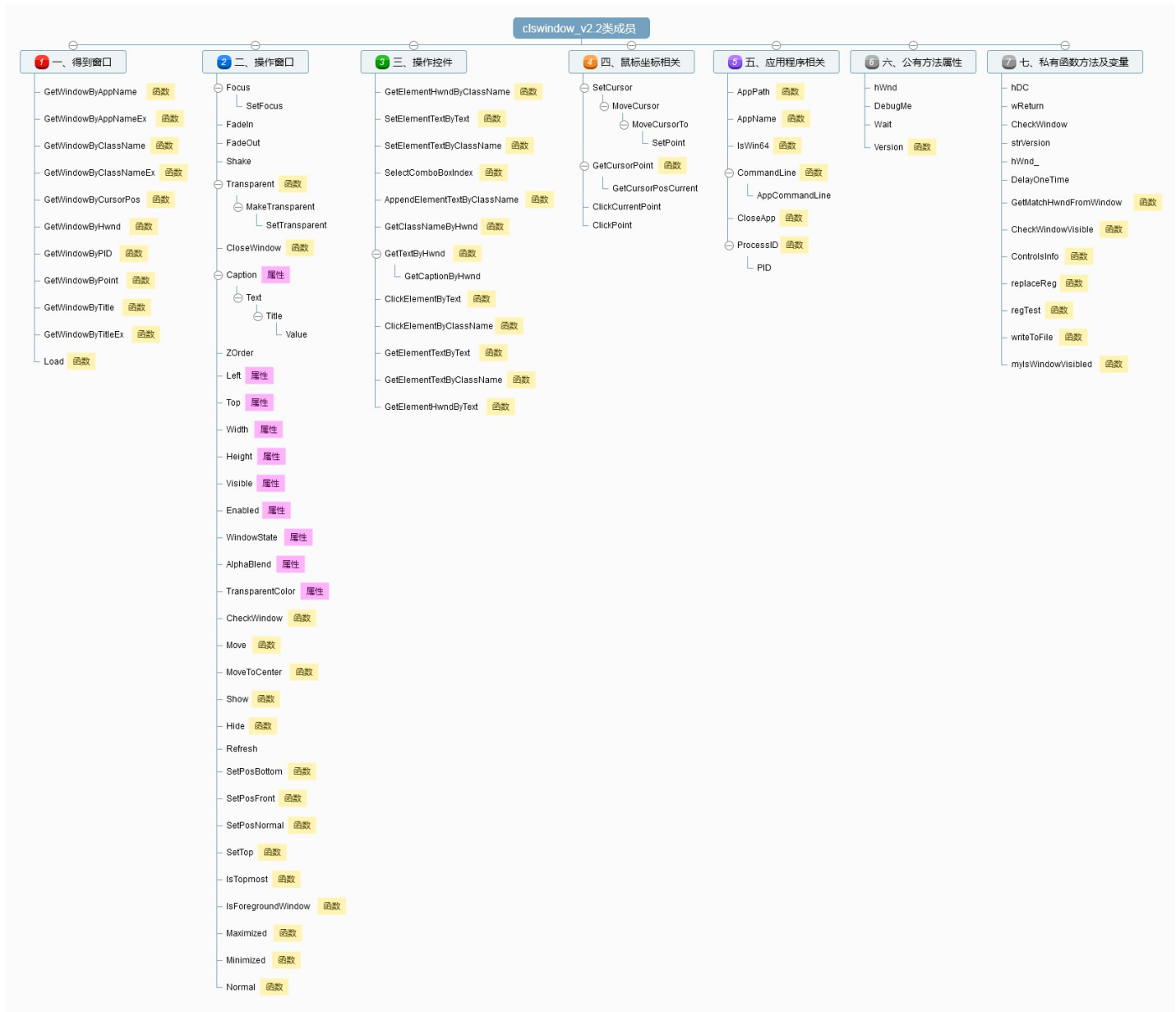
这个类现在还在一直不断地扩充，功能已经很强大很广泛，使用它可以轻而易举地设置窗口标题栏文字，移动窗体等等。以前要实现这些操作常常需要一大堆api函数，现在只需要一点点代码就可以了，完全让您脱身于api函数的海洋。当然您如果想知道里面的方法实现原理的话可以看一看源代码。

代码下载可以参考顶部的“github下载”和“码云下载”，本手册中的所有代码案例均测试通过，可以下载源代码后复制到工程中窗口代码处进行测试。

欢迎提供宝贵的建议和问题反馈。QQ171977759

脑图结构参考：

---



最新：<http://naotu.baidu.com/file/30e7a54048ea859337847d680d799d62?token=a4fee4bce5b5d700>

更新日志：

|   |       |
|---|-------|
| '版本：V1.0 初版                                   | 2     |
| 012/12/03                                     |       |
| ' V1.1 修正了几个正则相关的函数，调整了部分类结构                  | 2013/ |
| 05/28   |       |
| ' V1.2 增加属性Caption，可以获取或设置当前标题栏               | 2013  |
| /05/29  |       |
| ' V1.3 增加了方法Focus，可以激活当前窗口                    | 2013  |
| /06/01  |       |
| ' 增加了方法Left, Top, Width, Height, Move，处理窗口位置等 |       |
| ' V1.4 增加了窗口位置调整的几个函数                         | 2013  |
| /06/04  |       |
| ' 增加了得到应用程序路径的函数AppName                       |       |
| ' 增加了得到应用程序启动参数的函数AppCommandLine              |       |

|           |  |       |
|-----------|--|-------|
| '         | V1.5 增加了窗口最大最小化，隐藏显示正常的几个函数                          | 2013/ |
| 06/06     |  |       |
| '         | 增加了获取控件相关函数是否使用正则的参数UseRegExp默认F                     |       |
| '         | V1.6 将Left, Top函数改为属性，可获得可设置                         | 2013/ |
| 06/10     |  |       |
| '         | V1.7 增加函数：CloseApp 结束进程                              | 2013  |
| /06/13    |  |       |
| '         | 修正了部分跟正则匹配相关的函数                                      |       |
| '         | 增加函数：GetElementTextByText                            |       |
| '         | 增加函数：GetElementHwndByText                            |       |
| '         | V1.8 增加函数：GetWindowByClassName                       | 2     |
| 013/06/26 |  |       |
| '         | 增加函数：GetWindowByClassNameEx                          |       |
| '         | 增加函数：GetWindowByAppName                              |       |
| '         | 增加私有变量hwnd_  |       |
| '         | 增加属性hwnd，可设置，单设置时候会检查，非法则设置为0                        |       |
| '         | 更新GetWindowByTitleEx函数，使之可以选择性支持正则                   |       |
| '         | 删除GetWindowByTitleRegExp函数，合并到上面函数                   |       |
| '         | 增加SetFocus函数，调用Focus实现，为了是兼容VB习惯                     |       |
| '         | 扩了ProcessID、AppPath、AppName、AppCommandLine三个函数，可带参数  |       |
| '         | 网友wwb(wwbing@gmail.com)提供了一些函数和方法属性：                 |       |
| '         | CheckWindow, Load, WindowState, Visible, hDC, ZOrder |       |
| '         | AlphaBlend, Enabled, Refresh, TransparentColor       |       |
| '         | 采纳wwb网友的部分意见，将句柄变量改为hwnd_，但是hwnd作为公共属性               |       |
| '         | V1.9 修正函数：GetMatchHwndFromWindow 正则表达式的错误            | 2013  |
| /08/07    |  |       |
| '         | 修正函数：GetMatchHwndFromWindow 函数中的一些错误                 | 2014  |
| /09/23    |  |       |
| '         | 增加函数：GetWindowByClassNameEx                          |       |
| '         | 增加函数：GetWindowByPID 根据PID取窗口句柄                       |       |
| '         | 增加函数：GetCaptionByHwnd 根据句柄取得标题                       |       |
| '         | 增加函数：SetTop设置窗体置顶，传入参数false则取消                       | 2014/ |
| 09/24     |  |       |
| '         | 增加函数：Shake、FadeIn、FadeOut 抖动、淡入、淡出特效                 |       |
| '         | V2.0 修正函数：GetWindowByPID 遍历窗体Win7下有一些问题              | 2015/ |
| 09/29     |  |       |
| '         | 修正函数：GetWindowByAppName 遍历窗体Win7下有一些问题               |       |
| '         | 修正函数：GetWindowByAppNameEx 遍历窗体Win7下有一些问题             |       |
| '         | V2.1 修正函数：ClickPoint 增加位置模式参数相对和绝对，默认相对              | 2018/ |
| 06/05     |  |       |
| '         | 增加函数：SelectComboBoxIndex 根据指定的index选择下拉框中的项          |       |
| '         | 上述方法得到网友Chen8013的不少帮助，特此感谢                           |       |
| '         | 增加函数：GetWindowByHwnd 根据指定的句柄确定窗口                     | 2018/ |
| 07/22     |  |       |
| '         | 增加函数：GetWindowByCursorPos 根据当前光标获取窗口（控件）             |       |
| '         | 增加函数：GetWindowByPoint 根据指定的位置获取窗口（控件）                |       |
| '         | 升级ClickPoint函数，支持点击前后分别延时，默认延时为0                     | 2018/ |
| 07/23     |  |       |
| '         | V2.2 修正正则：网友小凡反应了句柄和id存在负数的情况                        | 2020/ |
| 01/08     |  |       |

|          |  |       |
|----------|--|-------|
| '        | 优化属性： <b>Caption</b> (Get)，根据网友小凡的建议改成可获得文本框内容               |       |
| '        | 增加方法：Wait 此方法原为clsWaitableTimer模块中，现集成进来                     | 2020/ |
| 01/09    |  |       |
| '        | 增加方法：ClickCurrentPoint 点击当前点                                 | 20    |
| 20/01/10 |  |       |
| '        | 增加方法： <b>SetCursor</b> (别名:SetPoint MoveCursor MoveCursorTo) |       |
| '        | 更新函数：将所有默认等待超时 <b>60</b> 秒的函数中默认等待时间都改为 <b>10</b> 秒          |       |
| '        | 增加属性：Text、Value、Title (均为Caption别名)                          | 2020  |
| /01/12   |  |       |
| '        | 优化代码：GetCaptionByHwnd采用原 <b>Caption</b> (Get)代码，后者也做了调整      |       |
| '        | 增加函数： <b>GetCursorPosCurrent</b> (别名：GetCursorPoint)得到当前坐标   |       |
| '        | 优化函数：所有窗口获取的函数增加了是否过滤可见的参数                                   | 2020/ |
| 01/16    |  |       |
| '        | 增加函数：GetTextByHwnd (同GetCaptionByHwnd)                       |       |
| '        | 优化代码结构。将模块中能移过来的都移到类模块中了                                     | 2020/ |
| 01/19    |  |       |
| '        | 增加函数：myIsWindowVisibled 判断窗体可见，长宽为 <b>0</b> 也认为不可见           | 2020/ |
| 01/31    |  |       |
| '        | 优化函数：GetTextByHwnd 网友小凡提供                                    | 20    |
| 20/02/03 |  |       |
| '        | 增加函数：CommandLine (同AppCommandLine)                           | 20    |
| 20/02/05 |  |       |
| '        | 增加函数：MakeTransparent 设置窗口透明度                                 | 2020  |
| /02/18   |  |       |
| '        | 增加函数：MoveToCenter 移动窗口到屏幕中心                                  |       |
| '        | 增加函数：IsTopmost 判断窗口是否为置顶                                     | 2020  |
| /02/20   |  |       |
| '        | 增加函数：GetWindowTextByHwnd 获得窗口标题，给窗口句柄专用                      | 2020/ |
| 02/28    |  |       |
| '        | 修正函数：Focus 旧方法使用后会改变置顶窗口属性                                   | 2020/ |
| 03/02    |  |       |
| '        | 增加函数：IsWin64 网友小凡提供  | 20    |
| 20/03/12 |  |       |
| '        | 修正函数：AppPath 网友小凡提供兼容 <b>64</b> 位系统的方法                       |       |
| '        | 修正函数：AppCommandLine 网友小凡做了兼容 <b>64</b> 位处理及其他代码优化            | 2020/ |
| 03/15    |  |       |
| '        | 增加函数：IsForegroundWindow 判断窗口是否为活动窗口                          | 2020/ |
| 03/17    |  |       |
| '        | 增加函数：GetClassNameByHwnd 根据句柄得到类名                             |       |
| '        | 增加属性： <b>ClassName</b> (Get) 返回窗口的类名                         |       |
| '        | 更新函数：CheckWindow 返回值由Long改成Boolean了，并且设为Public               |       |
| '        | 增加函数：Click 点击当前句柄或者指定句柄                                      | 2020/ |
| 06/29    |  |       |



# 一、得到窗口

---

## 简介

---

得到窗口是指获取目标窗口的句柄，然后准备进行一系列操作，本类提供了很多个函数来获得，基本都是由GetWindowBy这样的前缀组成，每个函数都有着它特定的用途，根据当前最适合的场景来决定使用哪个。

## 相关函数或方法

---

- GetWindowByTitle
- GetWindowByTitleEx
- GetWindowByAppName
- GetWindowByAppNameEx
- GetWindowByClassName
- GetWindowByClassNameEx
- GetWindowByCursorPos
- GetWindowByHwnd
- GetWindowByPID
- GetWindowByPoint
- Load

# GetWindowByTitle 函数

## 描述

设置根据窗体标题得到窗口句柄，可以指定等待几秒检测指定标题的窗体是否出现，默认是等待10秒。

## 语法

GetWindowByTitle(strTitle[, intWaitSeconds = 10[, windowVisible = AllWindow]])

该语法中有下列命名参数：

| 参数             | 描述   |
|----------------|--|
| strTitle       | 必须的。String类型，表示要查找的窗口的标题文字，需要写完整，如果要写一部分也能查找到可以使用该函数的扩展函数 <a href="#">GetWindowByTitleEx</a> 。   |
| intWaitSeconds | 可选的。Long 类型，表示等待超时时间。如果忽略，缺省值是 10，它表明最多等待10秒钟  |
| windowVisible  | 可选的。enumWindowVisible类型，表示指定所需要获得的窗口可见属性。如果忽略，缺省值是 AllWindow，它表明不管什么状态（可见&不可见）的窗口都获取。关于enumWindowVisible类型的介绍请参考下面“enumWindowVisible类型的设置值”表的介绍。 |

enumWindowVisible类型的设置值：

| 设置              | 描述        |
|-----------------|-----------|
| HiddenWindow    | 隐藏的窗口     |
| DisplayedWindow | 显示的窗口     |
| AllWindow       | 以上两种窗口都包含 |

## 返回值

|             |
|-------------|
| clsWindow   |
| 返回值为本身，方便连写 |

## 示例

说明：获得计算器窗口的句柄。

测试方式1：先打开计算器，然后运行下面代码，将会获得计算器的窗口句柄。

测试方式2：关闭所有计算器，运行下面代码，然后再打开计算器（注意在10秒内），请留意下和上面的比较下看看有什么区别。

```
Dim w As New clsWindow
MsgBox w.GetWindowByTitle("计算器").hWnd
```

注意，如果当前没有打开计算器那么程序将尝试寻找10秒钟然后放弃，如果在此期间你打开了计算器，函数会立马获得并返回。如果想自定义超时时间可以使用第2个参数，例如让它寻找3秒如果找不到目标窗口就放弃可以用下面代码：

```
Dim w As New clsWindow
MsgBox w.GetWindowByTitle("计算器", 3).hWnd
```

以上代码将在3秒内尝试寻找目标窗口，找不到就返回0，如果不想让函数尝试多次寻找窗口可以将第2个参数设置为0。

本函数一般适用于完整窗口标题匹配，如果要模糊匹配请用GetWindowByTitleEx函数。

# GetWindowByTitleEx 函数

## 描述

和GetWindowByTitle函数功能类似，不过功能相比要强大很多，支持很多功能，比如标题内容支持模糊匹配，甚至支持正则方式匹配。还可以结合pid控制窗口，以及搜集同类名的所有窗口句柄。

## 语法

GetWindowByTitleEx(strTitle[, intWaitSeconds = 10[, hWndAll[, isUseRegExp = False[, checkPid[, windowVisible = AllWindow]]]])

该语法中有下列命名参数：

| 参数             | 描述  |
|----------------|---|
| strTitle       | 必须的。String类型，表示要查找的窗口的标题文字，可以只写一部分，不过为了窗口查找准确建议还是写完整的标题文字，如果要对此参数进行正则控制可将第4个参数isUseRegExp设置为True。                                       |
| intWaitSeconds | 可选的。Long 类型，表示等待超时时间。如果忽略，缺省值是 10，它表明最多等待10秒钟   |
| hWndAll        | 可选的。String类型。注意，此函数是byRef（传地址）方式，由外部传入一个变量，函数得到的所有句柄都将存到这个变量中。  |
| isUseRegExp    | 可选的。Boolean类型，表示是否使用正则匹配模式。如果忽略，缺省值是False。  |
| checkPid       | 可选的。String 类型，表示要获取的窗口的pid是多少。如果忽略，缺省值是空，表示不检查checkPid值。  |
| windowVisible  | 可选的。enumWindowVisible类型，表示指定所需要获得的窗口可见属性。如果忽略，缺省值是 AllWindow，它表明不管什么状态（可见&不可见）的窗口都获取。关于enumWindowVisible类型的介绍请参考下面“enumWindowVisible类型的 |

|  |            |
|--|------------|
|  | 设置值” 表的介绍。 |
|--|------------|

enumWindowVisible类型的设置值：

| 设置              | 描述        |
|-----------------|-----------|
| HiddenWindow    | 隐藏的窗口     |
| DisplayedWindow | 显示的窗口     |
| AllWindow       | 以上两种窗口都包含 |

返回值

|             |
|-------------|
| clsWindow   |
| 返回值为本身，方便连写 |

示例

示例1

说明：根据窗口关键字模糊“计算”获得计算器窗口。建议先打开计算器然后获得执行下面代码。

```
Dim w As New clsWindow
If w.GetWindowByTitleEx("计算").hWnd <> 0 Then
    MsgBox "计算器窗口获取成功！它的句柄为：" & w.hWnd, vbInformation
    w.Caption = "获取成功 " & Time '改变计算器的标题
Else
    MsgBox "超时10秒。未发现包含“计算”二字的窗口标题", vbExclamation
End If
```

示例2

说明：如果当前有多个启动的计算器，我们自己程序用shell函数再启动一个计算器，那么怎样获得被我们启动的这个计算器的窗口呢？通过上述示例1显然无法获得，因为每个启动着的计算器窗口都是“计算器”，这时我们可以通过checkPid来校验，做进一步筛选，因为shell函数会返回pid的。请打开几个计算器后测试下面代码。

```
Dim w As New clsWindow
Dim lngPid As Long
lngPid = Shell("calc", 1)
If w.GetWindowByTitleEx("计算", , , , lngPid).hWnd <> 0 Then
    MsgBox "计算器窗口获取成功！它的句柄为：" & w.hWnd, vbInformation
    w.Caption = "获取成功 " & Time '改变计算器的标题
End If
```

## 示例3

说明：怎样获得所有符合条件的窗口句柄，然后逐一操作？下面代码将会获取所有记事本窗口并逐一关闭，测试前请多打开几个记事本文件。

```
Dim w As New clsWindow
Dim strHwndAll As String
Dim i As Integer, v
w.GetWindowByTitleEx "记事本", , strHwndAll
MsgBox "已获取所有记事本的句柄：" & strHwndAll, vbInformation
v = Split(strHwndAll, " ")
For i = 0 To UBound(v)
    w.hwnd = v(i)
    MsgBox "将关闭标题为"" & w.Caption & ""的记事本", vbInformation
    w.CloseWindow
Next
MsgBox "所有记事本窗口已被关闭！", vbInformation
```

## 示例4

说明：使用正则模糊匹配标题。下面代码将演示当前系统中标题包含数字的窗口。不用打开什么程序，直接运行下面代码即可

```
Dim w As New clsWindow
Dim strHwndAll As String
Dim i As Integer, v, s As String
w.GetWindowByTitleEx "\d+", , strHwndAll, True
v = Split(strHwndAll, " ")
MsgBox "系统当前标题含有数字的窗口共有" & UBound(v) + 1 & "个，它们的句柄为：" & strHwndAll, vbInformation
For i = 0 To UBound(v)
    s = s & v(i) & " 标题：" & w.GetWindowByHwnd(v(i)).Caption & vbCrLf
Next
MsgBox s
```

## 示例5

说明：阻止系统自带的小游戏启动，运行程序后尝试打开“扫雷”、“纸牌”等游戏看看。代码用了正则匹配，只要包含下面关键字的窗口都会被关闭。

```
Dim w As New clsWindow
Do
    If w.GetWindowByTitleEx("蜘蛛|纸牌|空当接龙|扫雷|红心大战|三维弹球", 10, , True).hwnd <> 0 Then
        w.CloseWindow '关闭窗口
    End If

```

```
w.Wait 100 '每100ms检测一次  
Loop
```

# GetWindowByClassName 函数

## 描述

设置根据窗体类名返回窗体句柄，可以指定等待几秒检测指定标题的窗体是否出现，默认是等待10秒。

## 语法

GetWindowByClassName(strClassName[, intWaitSeconds = 10[, windowVisible = AllWindow]])

该语法中有下列命名参数：

| 参数             | 描述   |
|----------------|--|
| strClassName   | 必须的。String类型，表示要查找的窗口的类名，需要写完整，如果要写一部分也能查找到可以使用该函数的扩展函数 <a href="#">GetWindowByClassNameEx</a> 。   |
| intWaitSeconds | 可选的。Long类型，表示xxx。如果忽略，缺省值是 10，它表明  |
| windowVisible  | 可选的。enumWindowVisible类型，表示指定所需要获得的窗口可见属性。如果忽略，缺省值是 AllWindow，它表明不管什么状态（可见&不可见）的窗口都获取。关于enumWindowVisible类型的介绍请参考下面“enumWindowVisible类型的设置值”表的介绍。 |

enumWindowVisible类型的设置值：

| 设置              | 描述        |
|-----------------|-----------|
| HiddenWindow    | 隐藏的窗口     |
| DisplayedWindow | 显示的窗口     |
| AllWindow       | 以上两种窗口都包含 |

## 返回值

|             |
|-------------|
| clsWindow   |
| 返回值为本身，方便连写 |

## 示例



说明：打开记事本，直接运行下面代码测试。

```
Dim w As New clsWindow  
MsgBox w.GetWindowByClassName("notepad").Caption
```

# GetWindowByClassNameEx 函数

## 描述

设置根据窗体类名得到窗口句柄，可以指定等待几秒检测指定类名的窗体是否出现，默认是等待10秒，允许使用正则表达式

。

## 语法

GetWindowByClassNameEx(strClassName[, intWaitSeconds= 10[, hWndAll[, isUseRegExp = False[, checkPid[, windowVisible = AllWindow]]]])

该语法中有下列命名参数：

| 参数             | 描述  |
|----------------|---|
| strClassName   | 必须的。String类型，表示要查找的窗口的类名，可以只写一部分，不过为了窗口查找准确建议还是写完整类名，如果要对此参数进行正则控制可将第4个参数isUseRegExp设置为True。  |
| intWaitSeconds | 可选的。Long 类型，表示等待超时时间。如果忽略，缺省值是 10，它表明最多等待10秒钟   |
| hWndAll        | 可选的。String类型。注意，此函数是byRef（传地址）方式，由外部传入一个变量，函数得到的所有句柄都将存到这个变量中。  |
| isUseRegExp    | 可选的。Boolean类型，表示是否使用正则匹配模式。如果忽略，缺省值是False。  |
| checkPid       | Optional 可选的。Variant类型，表示指定的进程pid。如果忽略，缺省值是""，表示未指定。  |
| windowVisible  | 可选的。enumWindowVisible类型，表示指定所需要获得的窗口可见属性。如果忽略，缺省值是 AllWindow，它表明不管什么状态（可见&不可见）的窗口都获取。关于enumWindowVisible类型的介绍请参考下面“enumWindowVisible类型的 |

|  |            |
|--|------------|
|  | 设置值” 表的介绍。 |
|--|------------|

enumWindowVisible类型的设置值：

| 设置              | 描述        |
|-----------------|-----------|
| HiddenWindow    | 隐藏的窗口     |
| DisplayedWindow | 显示的窗口     |
| AllWindow       | 以上两种窗口都包含 |

返回值

|             |
|-------------|
| clsWindow   |
| 返回值为本身，方便连写 |

示例

说明：打开记事本，直接运行下面代码测试。

```
Dim w As New clsWindow
MsgBox w.GetWindowByClassNameEx("Notepad").Caption
MsgBox w.GetWindowByClassNameEx("Notep.*?", , , True).Caption

Dim s As String
w.GetWindowByClassNameEx "\d", , s, True '将当前系统中所有含数字的窗口句柄都搜集放入s
MsgBox s
```

# GetWindowByPoint 函数

## 描述

根据指定的点获取窗口或控件的句柄

。

## 语法

GetWindowByPoint(x, y)

该语法中有下列命名参数：

| 参数 | 描述                    |
|----|-----------------------|
| x  | 必须的。 Long类型，表示鼠标的横坐标。 |
| y  | 必须的。 Long类型，表示鼠标的纵坐标。 |

## 返回值

|             |
|-------------|
| clsWindow   |
| 返回值为本身，方便连写 |

## 示例

说明：测试下面代码。试着将坐标改成其他值看看。

```
Dim w As New clsWindow
MsgBox w.GetWindowByPoint(100, 100).Caption
```

# GetWindowByAppName 函数

## 描述

根据进程名称返回窗体句柄，可以指定等待几秒检测指定进程名称的窗体是否出现，默认是等待10秒。

## 语法

GetWindowByAppName(strAppName[, intWaitSeconds = 10[, isWholeMatch = False[, windowVisible = AllWindow]]])

该语法中有下列命名参数：

| 参数             | 说明  |
|----------------|---|
| strAppName     | 必须的。String类型，表示要查找的窗口所属的进程名，例如：notepad.exe  |
| intWaitSeconds | 可选的。Long类型，设置查找窗口最多等待多时秒。默认为10秒。  |
| isWholeMatch   | 可选的。Boolean类型，表示是否进行完整匹配。默认为False，即非完整匹配。   |
| windowVisible  | 可选的。enumWindowVisible类型，表示指定所需要获得的窗口可见属性。如果忽略，缺省值是 AllWindow，它表明不管什么状态（可见&不可见）的窗口都获取。关于 enumWindowVisible类型的介绍请参考下面“enumWindowVisible类型的设置值”表的介绍。 |

enumWindowVisible类型的设置值：

| 设置              | 描述        |
|-----------------|-----------|
| HiddenWindow    | 隐藏的窗口     |
| DisplayedWindow | 显示的窗口     |
| AllWindow       | 以上两种窗口都包含 |

## 返回值

|             |
|-------------|
| clsWindow   |
| 返回值为本身，方便连写 |

## 示例

---

```
Dim w As New clsWindow  
MsgBox w.GetWindowByAppName("notepad").Caption
```

# GetWindowByAppNameEx 函数

## 描述

根据进程名称返回窗体句柄，可以指定等待几秒检测指定进程名称的窗体是否出现，默认是等待10秒。

## 语法

```
GetWindowByAppNameEx(strAppName[, intWaitSeconds = 10[, hWndAll[, isUseRegExp = False[, windowVisible = AllWindow]]]])
```

该语法中有下列命名参数：

| 参数             | 说明  |
|----------------|---|
| strAppName     | 必须的。String类型，表示要查找的窗口所属的进程名，例如：notepad.exe  |
| intWaitSeconds | 可选。Long类型，设置查找窗口最多等待多时秒。默认为10秒。   |
| hWndAll        | 可选的。String类型，函数获取的所有符合条件的句柄都放入到这个变量中。   |
| isUseRegExp    | 可选的。Boolean类型，表示是否正则进行匹配。默认为False，即不用正则匹配。  |
| windowVisible  | 可选的。enumWindowVisible类型，表示指定所需要获得的窗口可见属性。如果忽略，缺省值是 AllWindow，它表明不管什么状态（可见&不可见）的窗口都获取。关于 enumWindowVisible类型的介绍请参考下面“enumWindowVisible类型的设置值”表的介绍。 |

enumWindowVisible类型的设置值：

| 设置              | 描述        |
|-----------------|-----------|
| HiddenWindow    | 隐藏的窗口     |
| DisplayedWindow | 显示的窗口     |
| AllWindow       | 以上两种窗口都包含 |

## 返回值

|           |
|-----------|
| clsWindow |
|-----------|

## 返回值为本身，方便连写

### 示例

根据进程名关闭Notepad，为防止错关闭，可以加if结合窗口标题控制：（win7下好像无效，问题没仔细研究）

```
Dim w As New clsWindow
Dim sHwndAll$, v, i%
If w.GetWindowByAppNameEx("notepad.exe", 3, sHwndAll).hWnd > 0 Then 'xp下资源
浏览器进程名是explorer.exe
    v = Split(sHwndAll, " ") '所有符合条件的窗口句柄的集合
    For i = 0 To UBound(v)
        w.hWnd = v(i)
        If w.Caption <> "" And w.Caption <> "Program Manager" And w.Caption
<> "M" And w.Caption <> "Default IME" And w.Visible Then '必须可见并且内容不为空不
是系统特殊的窗口
            w.CloseWindow
        End If
    Next
End If
```



# GetWindowByCursorPos 函数

## 描述

直接获得当前鼠标下的窗口或控件的句柄

。

## 语法

GetWindowByCursorPos()

## 返回值

|           |
|-----------|
| clsWindow |
|-----------|

|             |
|-------------|
| 返回值为本身，方便连写 |
|-------------|

## 示例

说明：测试下面代码。试着鼠标放到不同的窗口上然后运行下面代码。

```
Dim w As New clsWindow
MsgBox w.GetWindowByCursorPos.Caption
```

# GetWindowByPID 函数

## 描述

根据pid获得窗体

。

## 语法

GetWindowByPID(PID [, intWaitSeconds = 10[, hWndAll[, windowVisible = AllWindow]]])

该语法中有下列命名参数：

| 参数             | 描述   |
|----------------|--|
| PID            | 必须的。 Long类型，表示经常标识。  |
| intWaitSeconds | 可选的。 Long 类型，表示等待超时时间。如果忽略，缺省值是 10，它表明最多等待10秒钟   |
| hWndAll        | 可选的。 String类型。注意，此函数是 byRef（传地址）方式，由外部传入一个变量，函数得到的所有句柄都将存到这个变量中，以便您在下面的程序中使用。  |
| windowVisible  | 可选的。 enumWindowVisible类型，表示指定所需要获得的窗口可见属性。如果忽略，缺省值是 AllWindow，它表明不管什么状态（可见&不可见）的窗口都获取。关于 enumWindowVisible类型的介绍请参考下面“enumWindowVisible类型的设置值”表的介绍。 |

enumWindowVisible类型的设置值：

| 设置              | 描述        |
|-----------------|-----------|
| HiddenWindow    | 隐藏的窗口     |
| DisplayedWindow | 显示的窗口     |
| AllWindow       | 以上两种窗口都包含 |

## 返回值

|           |
|-----------|
| clsWindow |
|-----------|

返回值为本身，方便连写

## 示例

说明：无需打开记事本，直接运行下面代码测试。

```
Dim w As New clsWindow
Dim lndPID As Long
lndPID = Shell("notepad.exe", 1) 'shell函数执行成功后会返回pid。你可以可以从任务管理器里找到记事本的pid然后传入
MsgBox w.GetWindowByPID(lndPID).Caption
```

# GetWindowByHwnd 函数

## 描述

直接由句柄指定窗口，与直接设置w.hwnd或者load方法同等效果。一般适用于已知句柄的情况下。比如vb自己窗口的句柄传入。

## 语法

GetWindowByHwnd(WindowHwnd)

该语法中有下列命名参数：

| 参数         | 描述                           |
|------------|------------------------------|
| WindowHwnd | 必须的。Variant类型，表示传入的句柄，一般为数字。 |

## 返回值

|             |
|-------------|
| clsWindow   |
| 返回值为本身，方便连写 |

## 示例

说明：打开一个记事本，然后测试下面代码。

```
Dim w As New clsWindow
Dim l As Long
l = w.GetWindowByClassName("Notepad").hwnd'句柄l我们这里我们是通过GetWindowByClassName来获取的，你也可以通过spy++等工具查看窗口句柄，然后直接复制给l进行测试。
MsgBox w.GetWindowByHwnd(l).Caption
```

# Load 函数 ( 同GetWindowByHwnd )

---

同GetWindowByHwnd , 请参考[GetWindowByHwnd](#)的说明。

## 二、操作窗口

---

### 简介

---

操作窗口是指在得到窗口句柄后对该窗口进行一系列的操作。为了方便这些操作我们封装了若干个方法和函数，以及属性。用户可根据自己的需要决定使用哪个方法和函数。

### 相关函数或方法

---

- Focus
- SetFocus
- FadeIn
- FadeOut
- Shake
- CloseWindow
- Caption
- Text ( 同Caption )
- Title ( 同Caption )
- Value ( 同Caption )
- ZOrder
- Left
- Top
- Width
- Height
- Visible
- Enabled
- WindowState
- AlphaBlend
- TransparentColor
- Move
- Show
- Hide
- Refresh
- SetPosBottom
- SetPosFront
- SetPosNormal

## 二、操作窗口

SetTop

Maximized

Minimized

Normal

# Focus 方法

## 描述

设置当前窗口为活动窗口

## 语法

Focus()

## 返回值

无

-

## 示例

说明：根据类名Notepad获得记事本窗口，然后调用Focus方法，会看到它变成活动窗口。建议先打开记事本并设置为非活动窗口，然后再运行下面代码。

```
Dim w As New clsWindow  
w.GetWindowByClassName("Notepad").Focus
```



# SetFocus 方法 ( 同Focus )

---

同Focus，请参考[Focus](#)的说明。

# FadeIn 方法

## 描述

淡入特效，执行后窗口会慢慢显示出来。

## 语法

FadeIn([speed = 10[, intPercentTansparentRemains]])

该语法中有下列命名参数：

| 参数                          | 描述  |
|-----------------------------|---|
| speed                       | 可选的。Long类型，表示淡入速度，数值越大淡入出来的越快，所允许值在1~255。如果忽略，缺省值是10，大概在1秒左右完成淡入，如果设置为100，窗体大概将在10ms左右完成淡入。具体情况设置速度 |
| intPercentTansparentRemains | 可选的。Integer类型，表示透明度截止到多少，允许值在1到100之间，比如设置为80，那么淡入到80%停止   |

## 示例

说明：根据类名Notepad获得记事本窗口，然后调用FadeIn方法，会看到它的淡入特效。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
If w.GetWindowByClassName("Notepad").hwnd <> 0 Then
    w.Normal '窗口正常
    w.Focus '设置为活动窗口
    w.FadeIn '默认淡入，默认速度为10
    w.FadeIn 2 '淡入速度改为2，相比默认值降低为5倍，测试会发现淡入速度明显变慢
    w.FadeIn 50 '淡入速度改为50，比默认值提高5倍，测试会发现淡入速度明显变快
End If
```

# FadeOut 方法

## 描述

淡出特效，执行后窗口透明度会越来越大直到完全消失。

## 语法

FadeOut([speed = 10])

该语法中有下列命名参数：

| 参数    | 描述   |
|-------|--|
| speed | 可选的。Long类型，表示淡出速度，数值越大淡出越快，所允许值在1~255。如果忽略，缺省值是10，大概在1秒左右完成淡出，如果设置为100，窗体大概将在10ms左右完成淡出。 |

## 示例

说明：根据类名Notepad获得记事本窗口，然后调用FadeOut方法，会看到它的淡出特效。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
If w.GetWindowByClassName("Notepad").hWnd <> 0 Then
    w.Normal '窗口正常
    w.Focus '设置为活动窗口
    w.FadeOut '默认淡入，默认速度为10
    w.FadeOut 2 '淡入速度改为2，相比默认值降低为5倍，测试会发现明显变慢
    w.FadeOut 50 '淡入速度改为50，比默认值提高10倍，测试会发现淡入淡入速度明显变快
    w.FadeIn
End If
```

# Transparent 函数

## 描述

窗体设置透明，Percent 只接受：0到100的整数、0%到100%、以及介于0到1的小数

## 语法

Transparent([Percent= "50"[, lngHwnd]])

该语法中有下列命名参数：

| 参数      | 描述   |
|---------|--|
| Percent | 可选的。Variant类型，表示设置的透明度，该参数只接受：0到100之间的数、字符串百分数形式“0%”到“100%”、以及介于0到1的小数，注意：Percent为1不等于100%，如果要全显示只有两种表达方式，即 Transparent 100 或 Transparent "100%" |
| lngHwnd | 可选的。Long类型，表示窗口句柄，如果缺省将会使用当前对象的句柄值，即hWnd_  |

## 示例

### 示例1

说明：设置记事本的透明度。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
If w.GetWindowByClassName("Notepad").hWnd <> 0 Then
    w.Transparent
    MsgBox "记事本已经设置透明度为50%，即默认值"

    w.Transparent 0.2
    MsgBox "记事本已经设置透明度为20%"

    w.Transparent "80%"
    MsgBox "记事本已经设置透明度为80%"

    w.Transparent 100
    MsgBox "记事本已经设置透明度为100%"
End If
```

## 示例2

说明：用spy++查看某个窗口的句柄，得到句柄值例如123456，然后编写下面代码运行。代码旨在说明可以通过句柄对指定窗口设置透明度。

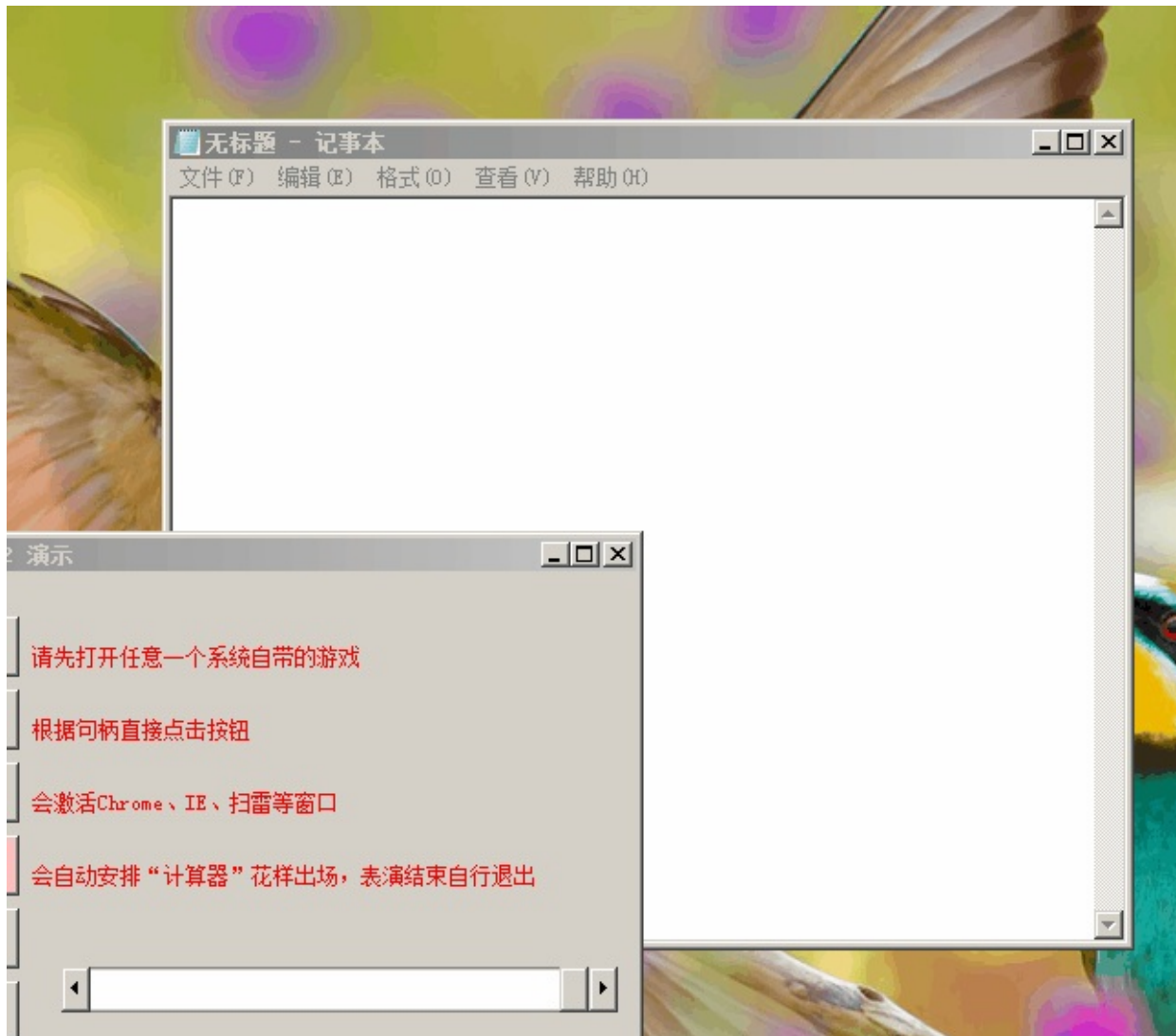
```
Dim w As New clsWindow
MsgBox w.MakeTransparent("60%", 123456) '这里的"60%"直接改成60或0.6都可以
```

## 示例3

说明：添加滚动条“HScroll1”，然后随便打开一个txt文件后，运行下面代码，拉动进度条看看记事本窗口的变化。

```
Private Sub HScroll1_Change()
    Dim w As New clsWindow
    If w.GetWindowByClassName("Notepad", 0).hWnd <> 0 Then
        w.MakeTransparent HScroll1.Value / HScroll1.Max * 100 & "%"
    Else
        MsgBox "请先打开一个记事本文件，然后再拉动滚动条，注意观察记事本窗口变化", vbInformation
    End If
End Sub

Private Sub HScroll1_Scroll()
    Call HScroll1_Change
End Sub
```



# SetTransparent 函数 ( 同Transparent )

---

同Transparent，请参考[Transparent](#)的说明。

# MakeTransparent 函数 ( 同Transparent )

---

同Transparent，请参考[Transparent](#)的说明。



# Shake 方法

## 描述

特效1：窗体抖动

## 语法

Shake([shakeShift = Both[, shakeRepeats = 30[, shakePads = 10[, shakeInterval = 25]]]])

该语法中有下列命名参数：

| 参数            | 描述   |
|---------------|--|
| shakeShift    | 可选的。enumShift类型，表示移动的方向。如果忽略，缺省值是 Both，它表明左右移动。关于enumShift类型的介绍请参考下面“enumShift类型的设置值”表的介绍。 |
| shakeRepeats  | 可选的。Long类型，表示移动次数。如果忽略，缺省值是 30，它表明默认移动30次。   |
| shakePads     | 可选的。Long类型，表示移动幅度。如果忽略，缺省值是 10，它表明移动 10像素，如果指定左右则表示左右移动 10像素                               |
| shakeInterval | 可选的。Long类型，表示移动时间（毫秒）间隔，也可以看做频率。如果忽略，缺省值是 25，它表明每25ms移动一下                                  |

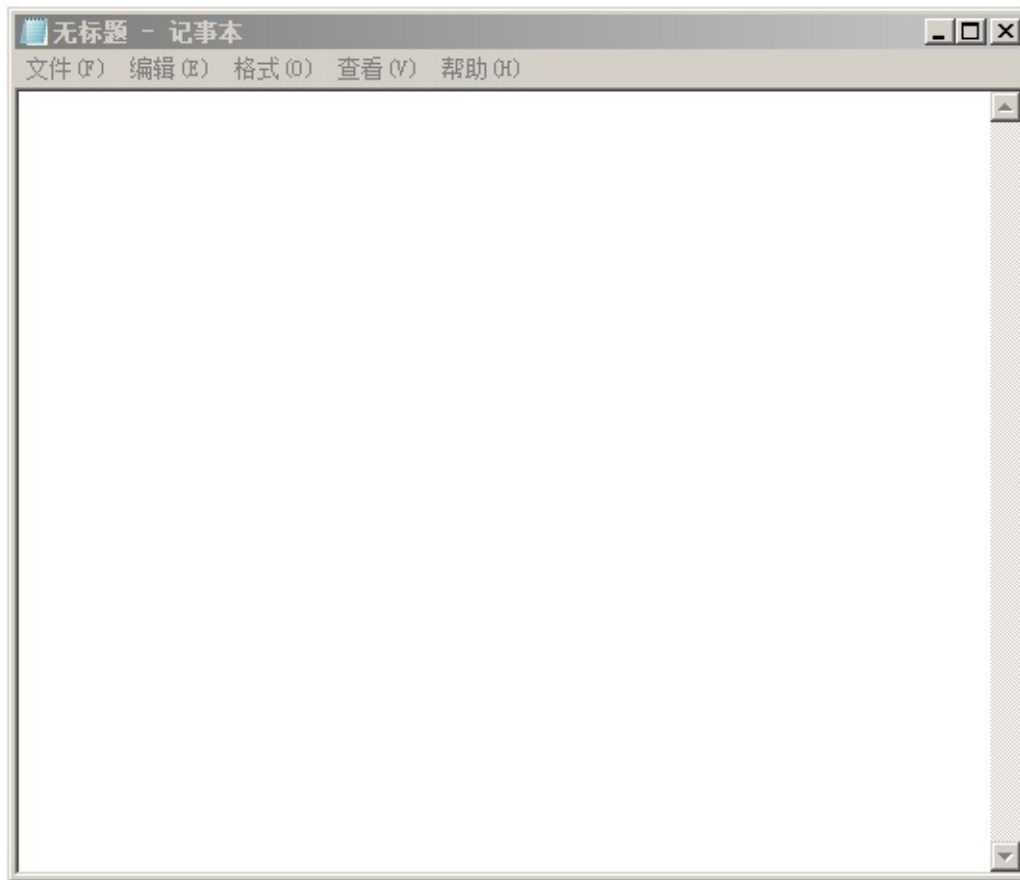
enumShift类型的设置值：

| 常数         | 值 | 描述       |
|------------|---|----------|
| Horizontal | 0 | 横向抖动     |
| Vertical   | 1 | 纵向抖动     |
| Both       | 2 | 纵向和横向都抖动 |

## 示例

说明：请先打开任意一个记事本，然后运行下面代码。注意比较每个抖动模式的区别。

```
Dim w As New clsWindow
If w.GetWindowByClassName("Notepad").hWnd <> 0 Then
    w.Normal '窗口正常
    w.Focus '设置为活动窗口
    w.Shake '默认抖动
    w.Shake Horizontal '横向抖动
    w.Shake Vertical '纵向抖动
    w.Shake , , 150 '抖动幅度为150像素
    w.Shake , 5, , 500 '抖动5次, 并且每隔500毫秒抖动一次
End If
```



# CloseWindow 函数

## 描述

关闭当前窗口，也可以根据句柄关闭指定的窗口。

## 语法

CloseWindow([hWnd1])

该语法中有下列命名参数：

| 参数    | 描述                                   |
|-------|--------------------------------------|
| hWnd1 | 可选的。Long类型。表示窗口句柄。如果缺省，则表示使用当前窗口的句柄。 |

## 返回值

|  |
|--|
| Long   |
| 操作成功点击则返回1，失败返回0，该返回值来自api函数PostMessage调用后的返回值 |

## 示例

说明：根据类名获得窗口，然后用CloseWindow关闭它。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
w.GetWindowByClassName("Notepad").CloseWindow
```

# ClassName 属性

## 描述

返回当前窗口的类名。注意，该属性仅有Get方法，无Set方法，也就是只能获得，没有设置的方法。

## 语法

variable = object. ClassName

## 返回值

|        |
|--------|
| String |
| 窗口类名   |

## 示例

说明：任意打开一个记事本，用spy++等工具获得其窗口句柄，例如123456，编写如下代码并运行测试。

```
Dim w As New clsWindow
w.hwnd = 123456
MsgBox w.ClassName '这个时候会看到显示类型Notepad
```

# Caption 属性

## 描述

设置或返回当前窗口的标题栏文字。

## 语法

object.Caption [= value]

该语法中有下列命名参数：

| 参数         | 描述                           |
|------------|------------------------------|
| strNewText | 必须的。 String类型，表示要设置的窗体标题栏文字。 |

## 返回值

|          |
|----------|
| String   |
| 窗口标题文字内容 |

## 示例

说明：根据类名获得窗口，然后显示该窗口的标题栏文字，接着再设置标题栏内容。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
MsgBox w.GetWindowByClassName("Notepad").Caption
w.GetWindowByClassName("Notepad").Caption = "新标题内容"
```

# Text 属性 ( 同Caption )

---

同Caption , 请参考[Caption](#)的说明。

# Title 属性 ( 同Caption )

---

同Caption , 请参考[Caption](#)的说明。

# Value 属性 ( 同Caption )

---

同Caption , 请参考[Caption](#)的说明。



# ZOrder 方法

## 描述

将当前窗口设置到顶部。

## 语法

object.ZOrder()

## 返回值

|   |
|---|
| 无 |
| - |

## 示例

# Left 属性

## 描述

得到或返回窗口的Left位置

## 语法

object. Left[= value]

该语法中有下列命名参数：

| 参数      | 描述                            |
|---------|-------------------------------|
| lngLeft | 必须的。 Long类型，表示要设置的窗体的Left坐标值。 |

## 返回值

|                 |
|-----------------|
| Long            |
| 返回值为当前窗体窗体的左坐标。 |

## 示例

说明：根据类名获得窗口，然后显示该窗口的Left坐标，接着再设置Left坐标为200。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
MsgBox w.GetWindowByClassName("Notepad").Left
w.GetWindowByClassName("Notepad").Left = 200
```

# Top 属性

## 描述

得到或返回窗口的Top位置

## 语法

object. Top[= value]

该语法中有下列命名参数：

| 参数     | 描述                           |
|--------|------------------------------|
| IngTop | 必须的。 Long类型，表示要设置的窗体的Top坐标值。 |

## 返回值

|                 |
|-----------------|
| Long            |
| 返回值为当前窗体窗体的左坐标。 |

## 示例

说明：根据类名获得窗口，然后显示该窗口的Top坐标，接着再设置Left坐标为200。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
MsgBox w.GetWindowByClassName("Notepad").Top
w.GetWindowByClassName("Notepad").Top= 200
```

# Width 属性

## 描述

得到或返回窗口的Width位置

## 语法

object.Width[= value]

该语法中有下列命名参数：

| 参数       | 描述                       |
|----------|--------------------------|
| lngWidth | 必须的。 Long类型，表示要设置的窗体的宽度。 |

## 返回值

|                 |
|-----------------|
| Long            |
| 返回值为当前窗体窗体的左坐标。 |

## 示例

说明：根据类名获得窗口，然后显示该窗口的宽度，接着设置宽度为200。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
MsgBox w.GetWindowByClassName("Notepad").Width
w.GetWindowByClassName("Notepad").Width= 200
```

# Height 属性

## 描述

得到或返回窗口的Height位置

## 语法

object.Height[= value]

该语法中有下列命名参数：

| 参数        | 描述                       |
|-----------|--------------------------|
| IngHeight | 必须的。 Long类型，表示要设置的窗体的高度。 |

## 返回值

|                 |
|-----------------|
| Long            |
| 返回值为当前窗体窗体的左坐标。 |

## 示例

说明：根据类名获得窗口，然后显示该窗口的高度，接着设置高度为200。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
MsgBox w.GetWindowByClassName("Notepad").Height
w.GetWindowByClassName("Notepad").Height= 200
```

# Visible 属性

## 描述

得到或返回窗口及控件的可见性

## 语法

object.Visible[= value]

该语法中有下列命名参数：

| 参数        | 描述                              |
|-----------|---------------------------------|
| vNewValue | 必须的。 Boolean类型，表示要设置的窗体或控件的可见性。 |

## 返回值

|                          |
|--------------------------|
| Long                     |
| 窗口或控件可见返回True，不可见返回False |

## 示例

说明：根据类名Notepad获得记事本窗口，然后获得可见性以及设置隐藏与显示。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
MsgBox w.GetWindowByClassName("Notepad").Visible
w.GetWindowByClassName("Notepad").Visible = False
MsgBox "窗口已经被设置为不可见"
w.GetWindowByClassName("Notepad").Visible = True
MsgBox "窗口被设置为可见"
```

# Enabled 属性

## 描述

得到窗口或控件的禁用状态

## 语法

object.Enabled[= value]

该语法中有下列命名参数：

| 参数        | 描述                               |
|-----------|----------------------------------|
| vNewValue | 必须的。 Boolean类型，表示要设置的窗口或者控件是否可用。 |

## 返回值

|                          |
|--------------------------|
| Boolean                  |
| 窗口或控件可用返回True，不可用返回False |

## 示例

说明：打开一个记事本，编辑框里面输入点内容，然后测试下面代码。

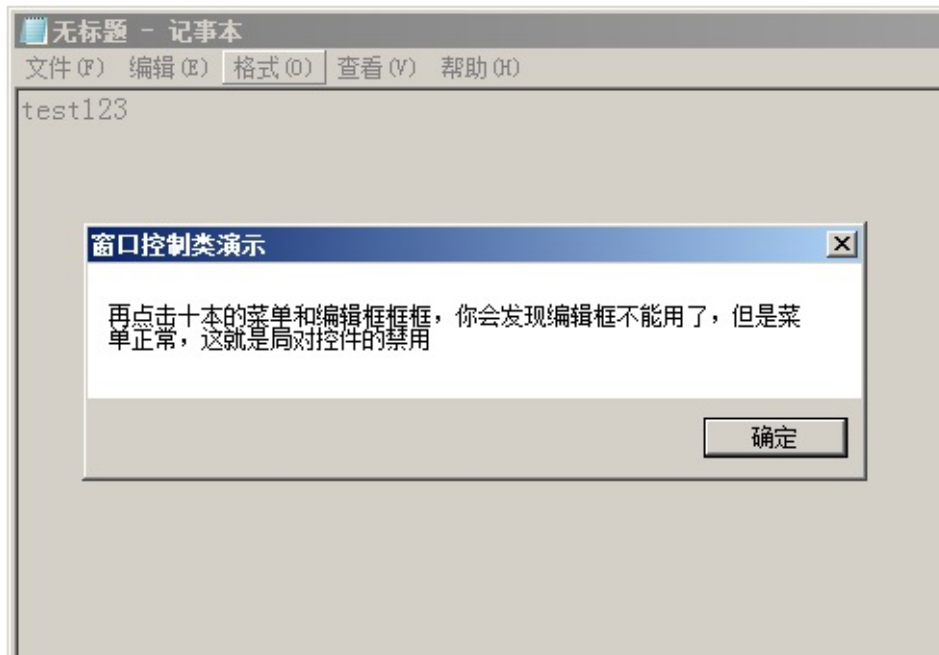
```
Dim w As New clsWindow
w.GetWindowByClassName("Notepad", 0).Enabled = False
MsgBox "点击记事本的菜单及编辑框框框，你会发现记事本没反应了", vbInformation

w.GetWindowByClassName("Notepad", 0).Enabled = True
MsgBox "恢复记事本可用，现在记事本正常了", vbInformation

w.GetWindowByHwnd(w.GetElementHwndByClassName("Edit")).Enabled = False
MsgBox "再点击十本的菜单和编辑框框框，你会发现编辑框不能用了，但是菜单正常，这就是局对控件的禁用", vbInformation

w.GetWindowByHwnd(w.GetElementHwndByClassName("Edit")).Enabled = True
MsgBox "恢复记事本编辑框可用，现在正常了", vbInformation
```

下图中编辑框内容被禁用了：





# WindowState 属性

## 描述

返回或设置一个值，该值用来指定目标窗口的可视状态。

## 语法

object.WindowState[= value]

该语法中有下列命名参数：

| 参数        | 描述  |
|-----------|---|
| vNewValue | 必须的。 enumWindowState类型，表示要设置的窗口的状态。关于 enumWindowState类型的介绍请参考下面 “enumWindowState类型的设置值” 表的介绍。 |

## 返回值

|  |
|--|
| enumWindowState  |
| 返回窗口状态的值。关于enumWindowState类型的介绍请参考下面 “enumWindowState类型的设置值” 表的介绍。 |

enumWindowState类型的设置值：

| 常数          | 值 | 描述  |
|-------------|---|-----|
| vbNormal    | 0 | 正常  |
| vbMinimized | 1 | 最小化 |
| vbMaximized | 2 | 最大化 |

## 示例

说明：根据类名Notepad获得记事本窗口，然后显示它的WindowState属性值，然后再设置它为最大化。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
MsgBox w.GetWindowByClassName("Notepad").WindowState
w.GetWindowByClassName("Notepad").WindowState = vbMaximized
```

# AlphaBlend 属性

## 描述

一个神秘的属性。设置指定颜色为透明，制作异形窗口使用，具体不得而知。由网友wwb提供。

## 语法

object.AlphaBlend[= value]

该语法中有下列命名参数：

| 参数        | 描述                 |
|-----------|--------------------|
| vNewValue | 必须的。 Long类型，我也不知道。 |

## 返回值

|            |
|------------|
| Long       |
| 返回值不知道是什么。 |

## 示例

# TransparentColor 属性

## 描述

一个神秘的属性。设置指定颜色为透明，制作异性窗口使用，具体不得而知。由网友wwb提供。

## 语法

object. TransparentColor[= value]

该语法中有下列命名参数：

| 参数        | 描述                 |
|-----------|--------------------|
| vNewValue | 必须的。 Long类型，我也不知道。 |

## 返回值

|            |
|------------|
| Long       |
| 返回值不知道是什么。 |

## 示例

# CheckWindow 函数

## 描述

判断窗口是否有效，是否存在。适用于一些检查指定句柄的窗口是否关闭。

## 语法

CheckWindow([lngHwnd])

该语法中有下列命名参数：

| 参数      | 描述  |
|---------|---|
| lngHwnd | 可选的。Long类型，表示判断指定句柄的窗口是否置顶。如果忽略，则使用类内部句柄。 |

## 返回值

|                               |
|-------------------------------|
| Boolean                       |
| True表示窗口有效或存在，False表示窗口不存在或无效 |

## 示例

### 示例1

说明：检测某个指定句柄的窗口是否存在。打开一个记事本，用spy++等工具查看窗口句柄值，例如123456，编写下面代码运行。

```
Dim w As New clsWindow
Msgbox w.CheckWindow(123456)
```

### 示例2

说明：检测某个已打开的窗口何时关闭。请打开系统自带的计算器程序然后运行下面代码，然后再关闭计算器看看程序反应。

```
Dim w As New clsWindow
If w.GetWindowByTitle("计算器", 0).hwnd <> 0 Then '注意这里的第2个参数0表示只尝试获取标题1次，默认是10秒超时
    Do
        If Not w.CheckWindow Then
            MsgBox "计算器已关闭"
            Exit Do
        End If
    Loop
```

```
        End If
        DoEvents
    Loop
Else
    MsgBox "计算器未运行"
End If
```

# Move 函数

## 描述

移动和设置窗口到指定位置，以及设置窗口的宽高。

## 语法

Move(x, y[, nWidth[, nHeight]])

该语法中有下列命名参数：

| 参数      | 描述                    |
|---------|-----------------------|
| x       | Long类型，表示窗口新的Left坐标值。 |
| y       | Long类型，表示窗口新的Top坐标值。  |
| nWidth  | 可选的。Long类型，表示窗口新的宽度。  |
| nHeight | 可选的。Long类型，表示窗口新的高度。  |

## 返回值

|                   |
|-------------------|
| Long              |
| 函数成功，返回非零值。失败返回0。 |

## 示例

说明：根据类名Notepad获得记事本窗口，设置它到坐标(100,100)，并且设置宽高都为500像素。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
w.GetWindowByClassName("Notepad").Move 100, 100, 500, 500
```

# MoveToCenter 函数

## 描述

移动窗口到屏幕中心

## 语法

MoveToCenter()

## 返回值

Long

函数成功，返回非零值。失败返回0。

## 示例

说明：设置记事本窗口为屏幕居中。建议先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow  
w.GetWindowByClassName("Notepad").MoveToCenter
```

# Show 函数

## 描述

设置当前窗口为活动窗口

## 语法

object. Show([ShowStatus=vbNormalFocus])

该语法中有下列命名参数：

| 参数         | 描述  |
|------------|---|
| ShowStatus | 可选的。VbAppWinStyle类型，表示窗口的状态。如果忽略，缺省值是vbNormalFocus，它表明默认设置窗体显示并获得焦点。关于VbAppWinStyle类型的介绍请参考下面“VbAppWinStyle类型的设置值”表的介绍。 |

VbAppWinStyle类型的设置值：

| 常数                 | 值 | 描述                                |
|--------------------|---|-----------------------------------|
| vbHide             | 0 | 窗口被隐藏，且焦点会移到隐式窗口。                 |
| VbNormalFocus      | 1 | 窗口具有焦点，且会还原到它原来的大小和位置。            |
| VbMinimizedFocus   | 2 | 窗口会以一个具有焦点的图标来显示。                 |
| VbMaximizedFocus   | 3 | 窗口是一个具有焦点的最大化窗口。                  |
| VbNormalNoFocus    | 4 | 窗口会被还原到最近使用的大小和位置，而当前活动的窗口仍然保持活动。 |
| VbMinimizedNoFocus | 6 | 窗口会以一个图标来显示。而当前活动的窗口仍然保持活动。       |

## 返回值

|   |
|---|
| 无 |
|---|



-

## 示例

说明：根据类名Notepad获得记事本窗口，然后调用Show方法，不带参数直接显示，带参数可以控制显示与隐藏以及最大化等。建议先打开记事本并设置为非活动窗口，然后再运行下面代码。

```
Dim w As New clsWindow
w.GetWindowByClassName("Notepad").Show vbHide
MsgBox "目标窗口隐藏了"
w.Show
MsgBox "目标窗口显示了"
w.Show vbMaximizedFocus
MsgBox "目标窗口最大化了"
```

# Hide 函数

## 描述

隐藏当前窗口。

## 语法

Hide()

## 返回值

|   |
|---|
| 无 |
| - |

## 示例

说明：根据类名Notepad获得记事本窗口，然后调用Hide方法隐藏它。建议先打开记事本并设置为非活动窗口，然后再运行下面代码。

```
Dim w As New clsWindow  
w.GetWindowByClassName("Notepad").Hide
```

# Refresh 方法

## 描述

更新当前窗口。一般用不到。

## 语法

Refresh()

## 返回值

|   |
|---|
| 无 |
| - |

## 示例

# SetPosBottom 函数

## 描述

窗口置后

## 语法

SetPosBottom()

## 返回值

|                             |
|-----------------------------|
| Long                        |
| 如果函数成功，返回值为非零；如果函数失败，返回值为零。 |

## 示例

说明：打开一个记事本，然后运行下面代码。

```
Dim w As New clsWindow

w.GetWindowByClassNameEx("Notepad").SetPosFront
MsgBox "设置置顶了"

w.GetWindowByClassNameEx("Notepad").SetPosBottom
MsgBox "设置窗口置后"

w.GetWindowByClassNameEx("Notepad").SetPosNormal
MsgBox "设置正常"

w.GetWindowByClassNameEx("Notepad").SetTop True '默认是true，为true时候和SetPosFront一样，区别是SetPosFront不带参数，SetTop可带参数，建议用SetPosFront
MsgBox "设置置顶了"

w.GetWindowByClassNameEx("Notepad").SetTop False '取消置顶，注意和前面的函数区别。
MsgBox "取消设置置顶"
```

# SetPosFront 函数

## 描述

窗口置前

## 语法

SetPosFront()

## 返回值

Long

如果函数成功，返回值为非零；如果函数失败，返回值为零。

## 示例

说明：打开一个记事本，然后运行下面代码。

```
Dim w As New clsWindow

w.GetWindowByClassNameEx("Notepad").SetPosFront
MsgBox "设置置顶了"

w.GetWindowByClassNameEx("Notepad").SetPosBottom
MsgBox "设置窗口置后"

w.GetWindowByClassNameEx("Notepad").SetPosNormal
MsgBox "设置正常"

w.GetWindowByClassNameEx("Notepad").SetTop True '默认是true，为true时候和SetPosFront一样，区别是SetPosFront不带参数，SetTop可带参数，建议用SetPosFront
MsgBox "设置置顶了"

w.GetWindowByClassNameEx("Notepad").SetTop False '取消置顶，注意和前面的函数区别。
MsgBox "取消设置置顶"
```

# SetPosNormal 函数

## 描述

设置窗口位置正常

## 语法

SetPosNormal()

## 返回值

|                             |
|-----------------------------|
| Long                        |
| 如果函数成功，返回值为非零；如果函数失败，返回值为零。 |

## 示例

说明：打开一个记事本，然后运行下面代码。

```
Dim w As New clsWindow

w.GetWindowByClassNameEx("Notepad").SetPosFront
MsgBox "设置置顶了"

w.GetWindowByClassNameEx("Notepad").SetPosBottom
MsgBox "设置窗口置后"

w.GetWindowByClassNameEx("Notepad").SetPosNormal
MsgBox "设置正常"

w.GetWindowByClassNameEx("Notepad").SetTop True '默认是true，为true时候和SetPosFront一样，区别是SetPosFront不带参数，SetTop可带参数，建议用SetPosFront
MsgBox "设置置顶了"

w.GetWindowByClassNameEx("Notepad").SetTop False '取消置顶，注意和前面的函数区别。
MsgBox "取消设置置顶"
```

# SetTop 函数

## 描述

设置窗体置顶

## 语法

SetTop([isSetTop = True])

该语法中有下列命名参数：

| 参数       | 描述  |
|----------|---|
| isSetTop | 可选的。Boolean 类型，表示是否设置置顶。如果忽略，缺省值是 True，它表明设置置顶。 |

## 返回值

|   |
|---|
| 无 |
| - |

## 示例

说明：打开一个记事本，然后运行下面代码。

```
Dim w As New clsWindow

w.GetWindowByClassNameEx("Notepad").SetPosFront
MsgBox "设置置顶了"

w.GetWindowByClassNameEx("Notepad").SetPosBottom
MsgBox "设置窗口置后"

w.GetWindowByClassNameEx("Notepad").SetPosNormal
MsgBox "设置正常"

w.GetWindowByClassNameEx("Notepad").SetTop True '默认是true，为true时候和SetPosFront一样，区别是SetPosFront不带参数，SetTop可带参数，建议用SetPosFront
MsgBox "设置置顶了"

w.GetWindowByClassNameEx("Notepad").SetTop False '取消置顶，注意和前面的函数区别。
MsgBox "取消设置置顶"
```

# IsTopMost 函数

## 描述

判断窗口当前是否置顶

## 语法

IsTopMost([Hwnd])

该语法中有下列命名参数：

| 参数   | 描述  |
|------|---|
| Hwnd | 可选的。Long类型，表示判断指定句柄的窗口是否置顶。如果忽略，则使用类内部句柄判断。 |

## 返回值

|                          |
|--------------------------|
| Boolean                  |
| True表示当前窗口为置顶，False表示非置顶 |

## 示例

说明：打开一个记事本，然后运行下面代码。

```
Dim w As New clsWindow
w.GetWindowByClassNameEx("Notepad").SetTop '设置为置顶
MsgBox w.IsTopmost '此时应该会返回True
w.SetPosNormal '取消置顶
MsgBox w.IsTopmost '此时应该返回False
```



# IsForegroundWindow 属性

## 描述

判断指定句柄的窗口是否为活动窗口

## 语法

IsForegroundWindow([hWnd])

该语法中有下列命名参数：

| 参数   | 描述   |
|------|--|
| Hwnd | 可选的。Long类型，表示判断指定句柄的窗口是否为活动窗口。如果忽略，则使用类内部句柄判断。 |

## 返回值

|                              |
|------------------------------|
| Boolean                      |
| True表示当前窗口为活动窗口，False表示非活动窗口 |

## 示例

说明：打开一个记事本，用spy++等工具查看窗口句柄值，例如123456，编写下面代码运行。

```
Dim w As New clsWindow
Msgbox w.IsForegroundWindow(123456)
```

测试会发现显示的是False，因为当前测试的程序是活动窗口，如果想要测试看到True的结果，可以将上面代码放到一个定时器中，设置3秒后运行，而你需要在3秒内把这个记事本窗口点击下让它变成活动窗口。

# Maximized 函数

## 描述

设置窗口最大化

## 语法

Maximized()

## 返回值

Long

如果窗口之前可见，则返回值为非零。如果窗口之前被隐藏，则返回值为零。

## 示例

说明：打开一个记事本，然后运行下面代码。

```
Dim w As New clsWindow
w.GetWindowByClassNameEx("Notepad").Maximized
MsgBox "最大化了"

w.GetWindowByClassNameEx("Notepad").Minimized
MsgBox "最小化了"

w.GetWindowByClassNameEx("Notepad").Normal
MsgBox "正常化"
```

# Minimized 函数

## 描述

设置窗口最小化

## 语法

Minimized()

## 返回值

Long

如果窗口之前可见，则返回值为非零。如果窗口之前被隐藏，则返回值为零。

## 示例

说明：打开一个记事本，然后运行下面代码。

```
Dim w As New clsWindow
w.GetWindowByClassNameEx("Notepad").Maximized
MsgBox "最大化了"

w.GetWindowByClassNameEx("Notepad").Minimized
MsgBox "最小化了"

w.GetWindowByClassNameEx("Notepad").Normal
MsgBox "正常化"
```

# Normal 函数

## 描述

设置窗口正常显示

## 语法

Normal()

## 返回值

Long

如果窗口之前可见，则返回值为非零。如果窗口之前被隐藏，则返回值为零。

## 示例

说明：打开一个记事本，然后运行下面代码。

```
Dim w As New clsWindow
w.GetWindowByClassNameEx("Notepad").Maximized
MsgBox "最大化了"

w.GetWindowByClassNameEx("Notepad").Minimized
MsgBox "最小化了"

w.GetWindowByClassNameEx("Notepad").Normal
MsgBox "正常化"
```

## 三、操作控件

---

### 简介

---

操作控件是指在得到窗口句柄后对该窗口内的控件进行一系列的操作。为了方便这些操作我们封装了若干个方法和函数，以及属性。用户可根据自己的需要决定使用哪个方法和函数。

### 相关函数或方法

---

- GetElementTextByText
- GetElementTextByClassName
- GetElementHwndByText
- GetElementHwndByClassName
- GetTextByHwnd
- GetCaptionByHwnd ( 同GetTextByHwnd )
- SetElementTextByText
- SetElementTextByClassName
- SelectComboBoxIndex
- AppendElementTextByClassName
- Click
- ClickElementByText
- ClickElementByClassName

# GetElementHwndByText 函数

## 描述

根据控件的文本内容得到控件的句柄

## 语法

GetElementHwndByText(strText[, intIndex = 1[, UseRegExp = False]])

该语法中有下列命名参数：

| 参数        | 描述   |
|-----------|--|
| strText   | 必须的。String类型，表示要查找的控件文字。   |
| intIndex  | 可选的。Integer类型，表示次序，如果有多个控件文字都一样，那么这个参数就有用了，依次1、2、3。。。用来区分它们，根据自己的需要使用对应的索引获取，默认是1表示符合条件的第一个控件。 |
| UseRegExp | 可选的。Boolean 类型，表示是否使用正则。如果忽略，缺省值是 False，它表明默认不使用正则。  |

## 返回值

|                              |
|------------------------------|
| Long                         |
| 根据控件内容获取控件成功则返回控件的句柄，获取失败返回0 |

## 示例

说明：请打开一个记事本，点开菜单“帮助” - “关于记事本”，本示例演示的是如何根据类名获得控件句柄，请注意“Button”修改为一个不存在的类名获取的结果。

```
Dim w As New clsWindow
If w.GetWindowByTitle("关于记事本", 0).hwnd <> 0 Then
    MsgBox w.GetElementHwndByClassName("Button") '将成功返回一个数字的句柄
    MsgBox w.GetElementHwndByClassName("Buttonxxx") '将会返回0，表示根据指定的类名未获取到句柄
End If
```

# GetElementHwndByClassName 函数

## 描述

根据控件的类名得到句柄

## 语法

GetElementHwndByClassName(strClassName[, intIndex = 1[, UseRegExp = False]])

该语法中有下列命名参数：

| 参数           | 描述   |
|--------------|--|
| strClassName | 必须的。String类型，表示要查找的控件文字。   |
| intIndex     | 可选的。Integer类型，表示次序，如果有多个类名相同的控件，那么这个参数就有用了，依次1、2、3。。。用来区分它们，根据自己的需要使用对应的索引获取，默认是1表示符合条件的第一个控件。这些输入框的索引可能并不是按照肉眼的顺序来的，如何区分它们请参考 <a href="#">目标窗体中存在多个相同类名的输入框该如何区分填写？</a> 。 |
| UseRegExp    | 可选的。Boolean 类型，表示是否使用正则。如果忽略，缺省值是 False，它表明默认不使用正则。  |

## 返回值

|                              |
|------------------------------|
| Long                         |
| 根据控件类名获取控件成功则返回控件的句柄，获取失败返回0 |

## 示例

### 示例1

说明：请打开一个记事本，点开菜单“帮助” - “关于记事本”，本示例演示的是如何根据类名获得控件句柄，请留意“Button”修改为一个不存在的类名获取的结果。

```
Dim w As New clsWindow
If w.GetWindowByTitle("关于记事本", 0).hwnd <> 0 Then
    MsgBox w.GetElementHwndByClassName("Button") '将成功返回一个数字的句柄
```

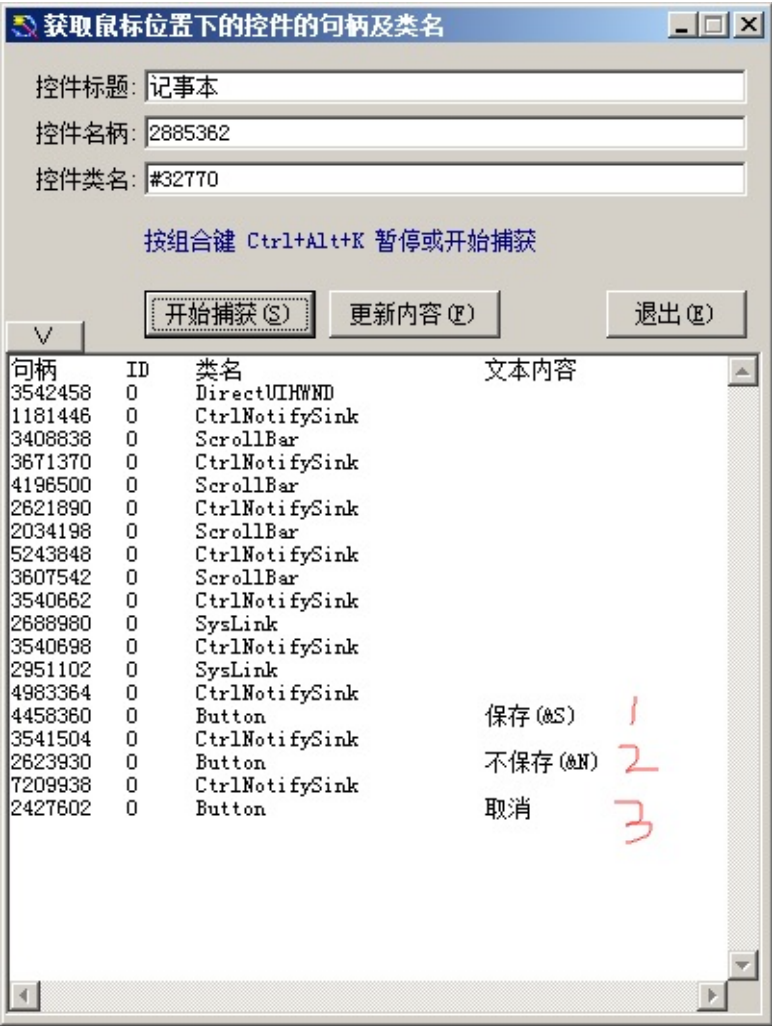
```
MsgBox w.GetElementHwndByClassName("Buttonxxx") '将会返回0，表示根据指定的类名未  
获取到句柄  
End If
```

## 示例2

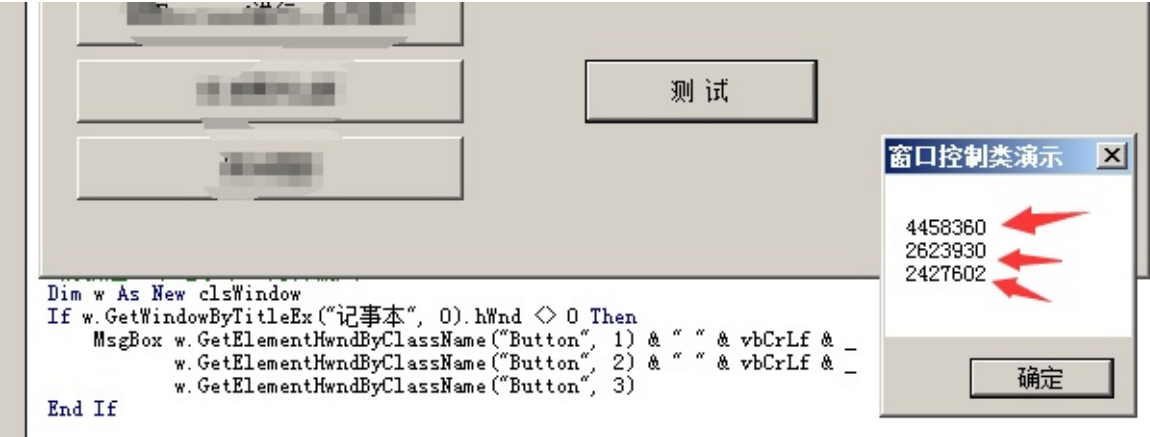
说明：打开记事本随便输入点内容，然后点击右上角关闭，这时我们会看到是否保存的对话框，里面有3个按钮，通过工具查看发现它们的类名都是“Button”，并且次序正好是依次1、2、3，这时我们测试获取和3个按钮的句柄。







```
Dim w As New clsWindow
If w.GetWindowByTitleEx("记事本", 0).hwnd <> 0 Then
    MsgBox w.GetElementHwndByClassName("Button", 1) & " " & vbCrLf & _
        w.GetElementHwndByClassName("Button", 2) & " " & vbCrLf & _
        w.GetElementHwndByClassName("Button", 3)
End If
```



# GetElementTextByText 函数

## 描述

根据控件的内容得到控件的显示文字

## 语法

GetElementTextByText(strText[, intIndex = 1[, UseRegExp = False]])

该语法中有下列命名参数：

| 参数        | 描述  |
|-----------|---|
| strText   | 必须的。String类型，表示要查找的控件文字。  |
| intIndex  | 可选的。Integer类型，表示次序，如果有多个控件文字都一样，那么这个参数就有用了，依次1、2、3。。。用来区分它们，根据自己的需要使用对应的索引获取。如果忽略，缺省值是1，表示符合条件的第1个控件。 |
| UseRegExp | 可选的。Boolean 类型，表示是否使用正则。如果忽略，缺省值是 False，它表明默认不使用正则。   |

## 返回值

|                              |
|------------------------------|
| String                       |
| 根据控件内容获取控件成功则返回控件的内容，获取失败返回空 |

## 示例

说明：请新建一个记事本，内容输入12345，然后运行下面代码测试

```
'请新建一个记事本，内容输入12345
Dim w As New clsWindow
If w.GetWindowByClassName("Notepad", 0).hWnd <> 0 Then
    MsgBox w.GetElementTextByText("1234") '根据内容完全匹配失败，获取为空
    MsgBox w.GetElementTextByText("12345") '根据内容完全匹配成功，成功获取到内容
    MsgBox w.GetElementTextByText("1234", , True) '根据内容模糊匹配获取
    MsgBox w.GetElementTextByText("\d+", , True) '根据正则获取
End If
```

# GetElementTextByClassName 函数

## 描述

根据控件的类名得到控件的显示文字

## 语法

GetElementTextByClassName(strText[, intIndex = 1[, UseRegExp = False]])

该语法中有下列命名参数：

| 参数        | 描述   |
|-----------|--|
| strText   | 必须的。String类型，表示要查找的控件文字。   |
| intIndex  | 可选的。Integer类型，表示次序，如果有多个类名相同的控件，那么这个参数就有用了，依次1、2、3。。。用来区分它们，根据自己的需要使用对应的索引获取，默认是1表示符合条件的第一个控件。这些输入框的索引可能并不是按照肉眼的顺序来的，如何区分它们请参考 <a href="#">目标窗体中存在多个相同类名的输入框该如何区分填写？</a> 。 |
| UseRegExp | 可选的。Boolean 类型，表示是否使用正则。如果忽略，缺省值是 False，它表明默认不使用正则。  |

## 返回值

|                              |
|------------------------------|
| String                       |
| 根据控件类名获取控件成功则返回控件的内容，获取失败返回空 |

## 示例

说明：请新建一个记事本，随意输入一些内容，然后运行下面代码测试

```
Dim w As New clsWindow
If w.GetWindowByClassName("Notepad", 0).hwnd <> 0 Then
    MsgBox w.GetElementTextByClassName("Edit") '根据类名获得记事本框框中内容
End If
```

# GetClassNameByHwnd 函数

## 描述

根据句柄获得窗口或句柄的类名

## 语法

GetClassNameByHwnd(hWnd)

该语法中有下列命名参数：

| 参数   | 描述                       |
|------|--------------------------|
| hWnd | 必须的。 Long类型，表示窗口或者控件的句柄。 |

## 返回值

|              |
|--------------|
| String       |
| 返回值为窗口或控件的类名 |

## 示例

说明：打开一个记事本，用spy++等工具查看窗口句柄值，例如123456，编写下面代码运行。

```
Dim w As New clsWindow
MsgBox w.GetClassNameByHwnd(123456)
```

# GetTextByHwnd 函数

## 描述

根据句柄获得窗口或控件的内容

## 语法

GetTextByHwnd(hWnd)

该语法中有下列命名参数：

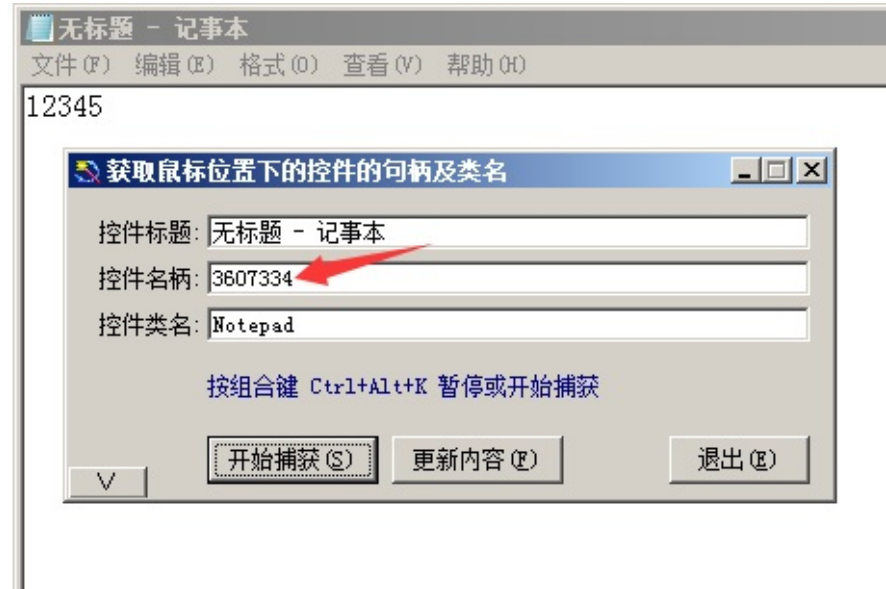
| 参数   | 描述                       |
|------|--------------------------|
| hWnd | 必须的。 Long类型，表示窗口或者控件的句柄。 |

## 返回值

|              |
|--------------|
| String       |
| 返回值为窗口或控件的内容 |

## 示例

说明：随便打开一个记事本，然后用工具获取它的窗口句柄，然后编写代码测试。



## GetTextByHwnd 函数



测试代码：

```
Dim w As New clsWindow
MsgBox w.GetTextByHwnd(3607334)
```

# GetCaptionByHwnd 函数 ( 同GetTextByHwnd )

---

同GetTextByHwnd，请参考[GetTextByHwnd](#)的说明。

# SetElementTextByText 函数

## 描述

根据控件的显示文字设置新的显示文字

## 语法

SetElementTextByText(strText, strNewText[, intIndex = 0[, UseRegExp = False]])

该语法中有下列命名参数：

| 参数         | 描述   |
|------------|--|
| strText    | 必须的。String类型，表示要查找的控件文字。   |
| strNewText | 必须的。String类型，表示对控件设置的新内容。  |
| intIndex   | 可选的。Integer类型，表示次序，如果有多个控件文字都一样，那么这个参数就有用了，依次1、2、3。。。用来区分它们，根据自己的需要使用对应的索引获取，默认是1表示符合条件的第一个控件。 |
| UseRegExp  | 可选的。Boolean 类型，表示是否使用正则。如果忽略，缺省值是 False，它表明默认不使用正则。  |

## 返回值

|  |
|--|
| Boolean                                |
| 返回True表示根据文字查找到控件并修改成功，返回False表示未查找到控件 |

## 示例

### 示例1

说明：请新建一个记事本，内容输入12345，然后运行下面代码测试

```
'请新建一个记事本，内容输入12345
Dim w As New clsWindow
If w.GetWindowByClassName("Notepad", 0).hwnd <> 0 Then
    MsgBox w.SetElementTextByText("1234", "hello world!") '返回False，表示替换失败
    , 因为记事本中内容12345和当前要判断的内容1234不一致，所以未找到
```



```

    MsgBox w.SetElementTextByText("12345", "hello world!") '返回True, 表示替换成功
, 要查找的内容和当前记事本中的内容12345一致即可设置
    MsgBox w.SetElementTextByText("hello", "hello world!") '返回False, 表示替换失
败, 因为内容不一致
    MsgBox w.SetElementTextByText("hello", "666", , True) '返回True, 表示替换成功
, 因为第4个参数表示是否启用正则, 设置为True表示支持正则匹配成功即可设置
    MsgBox w.SetElementTextByText("\d+", "ok", , True) '返回True, 表示替换成功, 原
因同上, \d+表示有数字, 匹配成功即可设置
End If

```

## 示例2

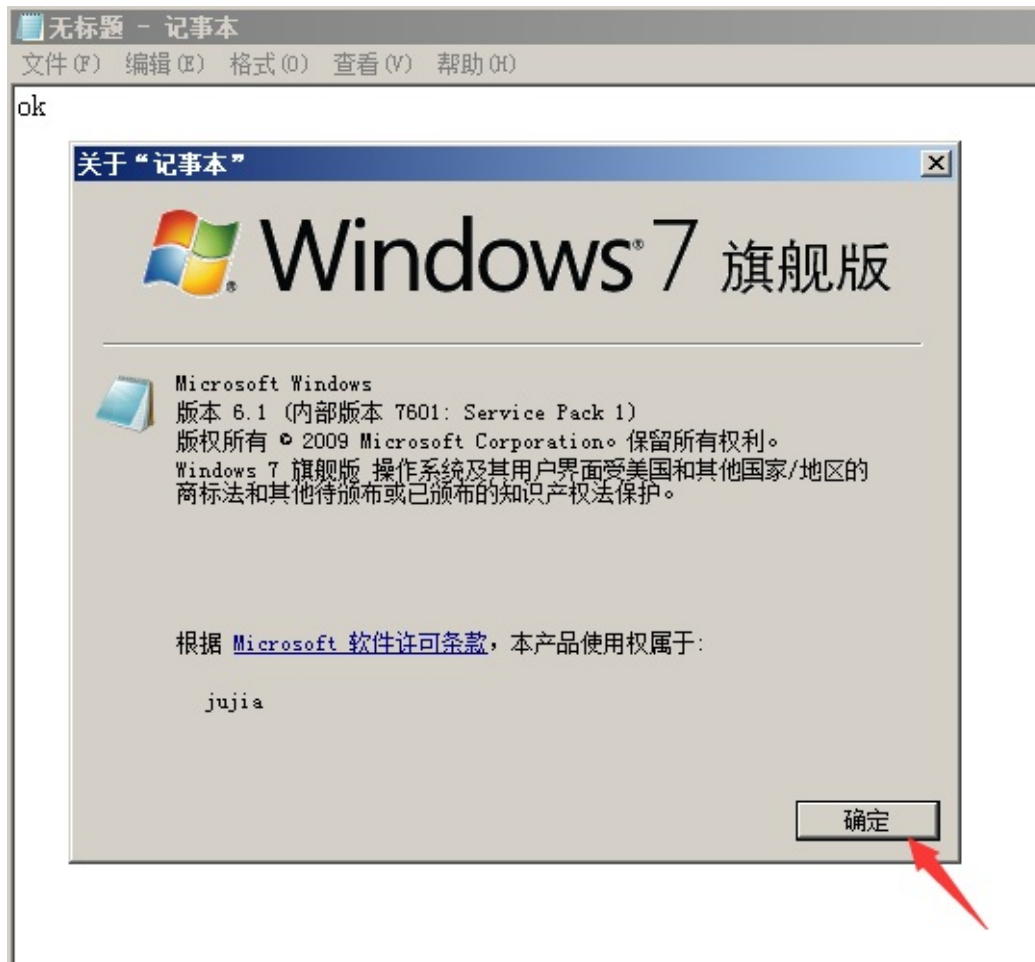
说明：请打开一个记事本，点开菜单“帮助” - “关于记事本”，本示例演示的是如何根据控件文字内容找到按钮并修改内容。

```

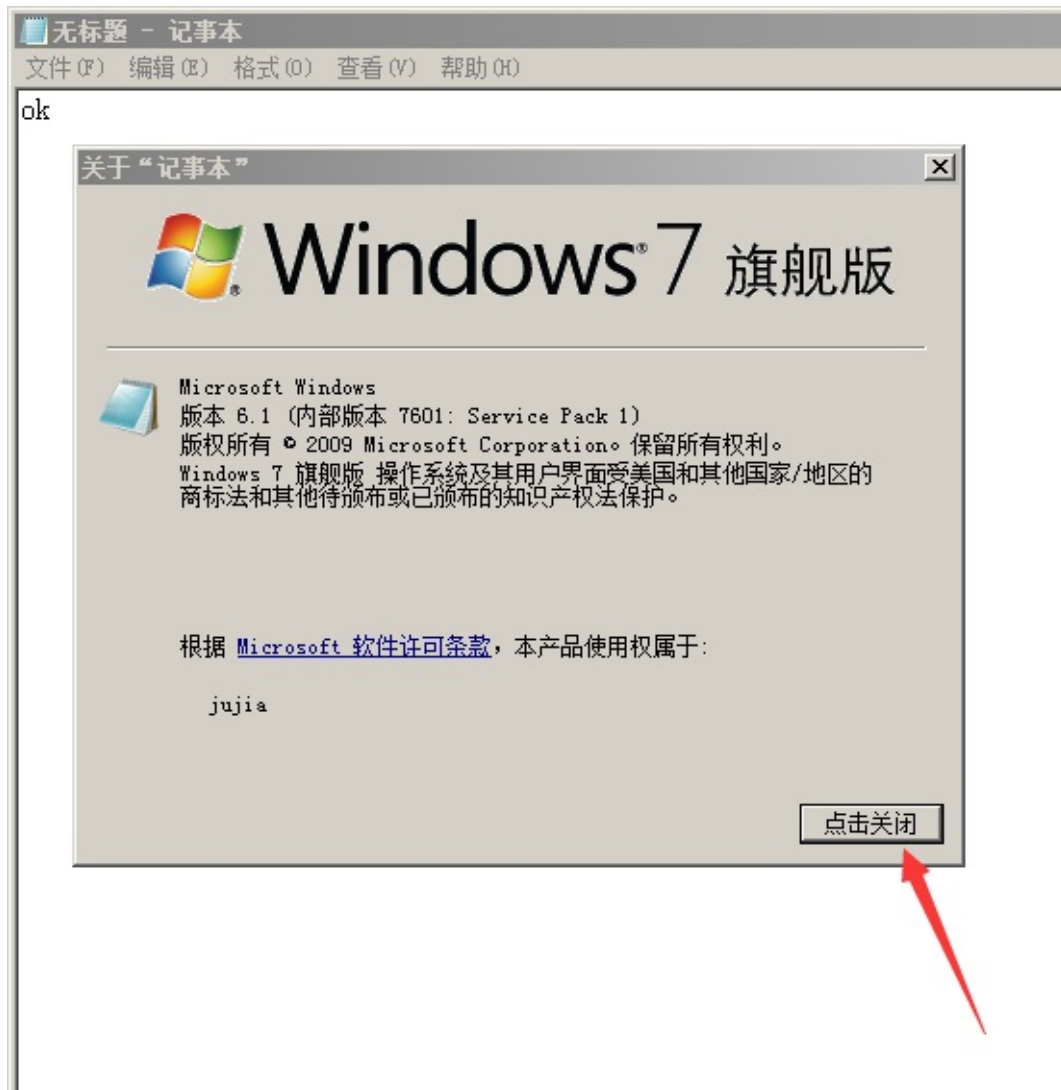
Dim w As New clsWindow
If w.GetWindowByTitle("关于“记事本”", 0).hWnd <> 0 Then
    If w.SetElementTextByText("确定", "点击关闭") Then
        MsgBox "按钮内容修改成功！", vbInformation
    Else
        MsgBox "未发现目标按钮！", vbExclamation
    End If
End If

```

案例2执行前：



案例2执行后：



# SetElementTextByClassName 函数

## 描述

根据控件的类名设置控件的显示文字

## 语法

SetElementTextByClassName(strClassName, strNewText[, intIndex = 1[, UseRegExp = False]])

该语法中有下列命名参数：

| 参数           | 描述   |
|--------------|--|
| strClassName | 必须的。String类型，表示要查找的控件类名。   |
| strNewText   | 必须的。String类型，表示对控件设置的新内容。  |
| intIndex     | 可选的。Integer类型，表示次序，如果有多个类名相同的控件，那么这个参数就有用了，依次1、2、3。。。用来区分它们，根据自己的需要使用对应的索引获取，默认是1表示指定类名的第一个控件。这些输入框的索引可能并不是按照肉眼的顺序来的，如何区分它们请参考 <a href="#">目标窗体中存在多个相同类名的输入框该如何区分填写？</a> 。 |
| UseRegExp    | 可选的。Boolean 类型，表示是否使用正则。如果忽略，缺省值是 False，它表明默认不使用正则。  |

## 返回值

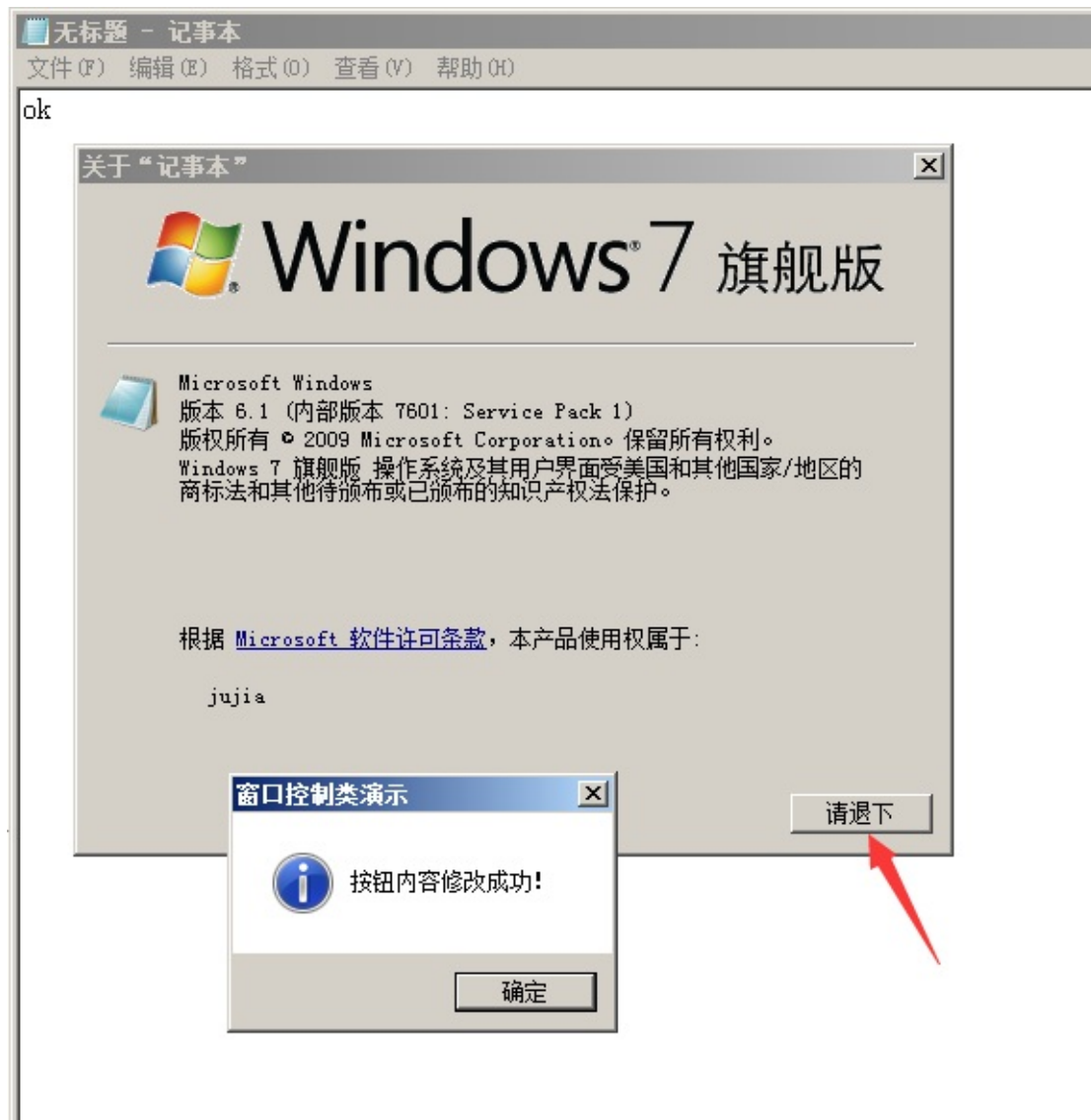
|  |
|--|
| Boolean                                |
| 返回True表示根据类名查找到控件并修改成功，返回False表示未查找到控件 |

## 示例

说明：请打开一个记事本，点开菜单“帮助” - “关于记事本”，本示例演示的是如何根据控件类名找到按钮并修改内容。

```
Dim w As New clsWindow
If w.GetWindowByTitle("关于记事本", 0).hwnd <> 0 Then
```

```
If w.SetElementTextByClassName("Button", "请退下") Then
    MsgBox "按钮内容修改成功!", vbInformation
Else
    MsgBox "未发现目标按钮!", vbExclamation
End If
End If
```



# SelectComboBoxIndex 函数

## 描述

根据指定的index选择下拉框中内容

## 语法

SelectComboBoxIndex(hWnd, intIndex)

该语法中有下列命名参数：

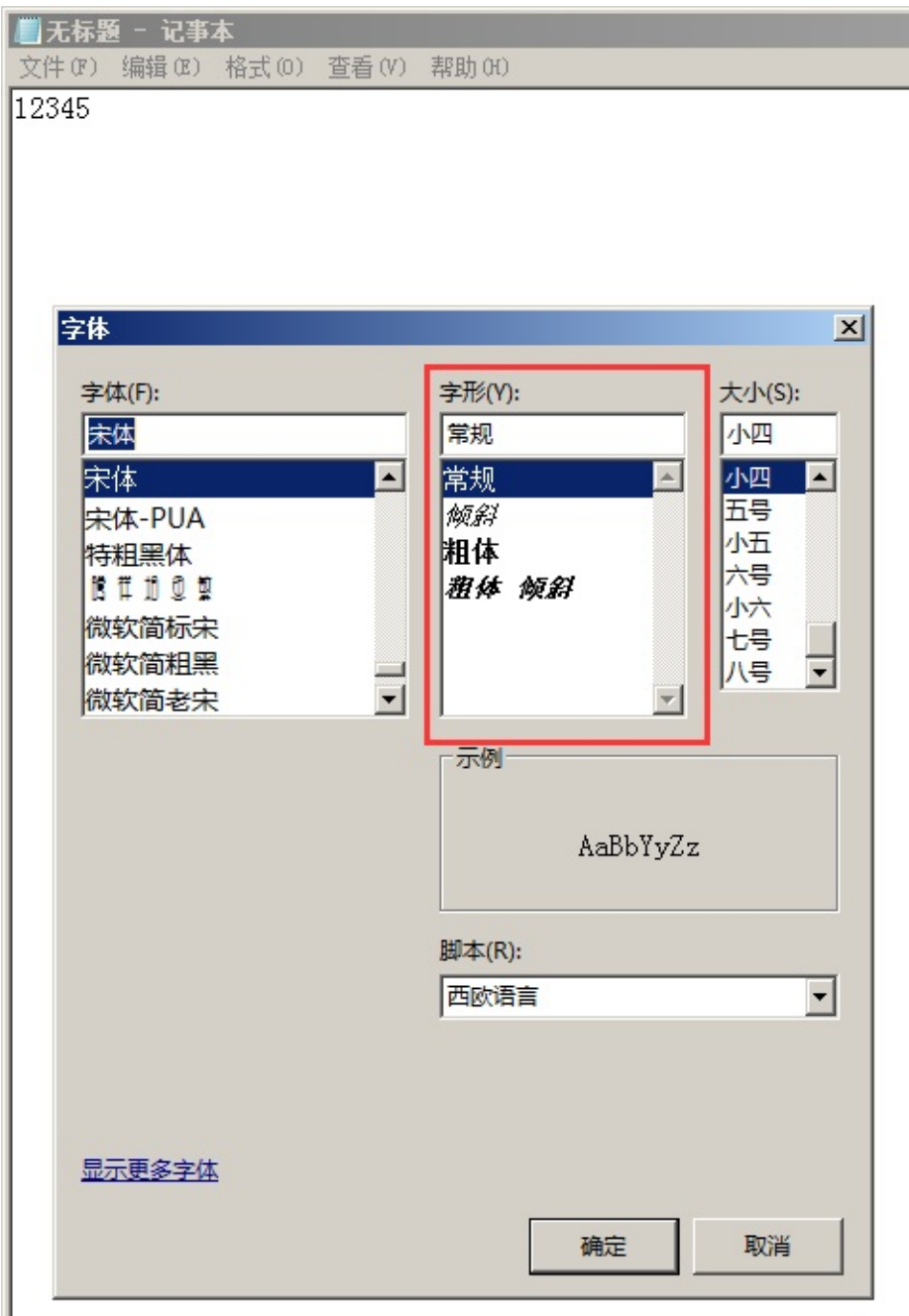
| 参数       | 描述                                  |
|----------|-------------------------------------|
| hWnd     | 必须的。 Long类型，表示要操作的下拉框的句柄。           |
| intIndex | 必须的。 Integer类型，表示要选择的下拉列表中的索引，从0开始。 |

## 返回值

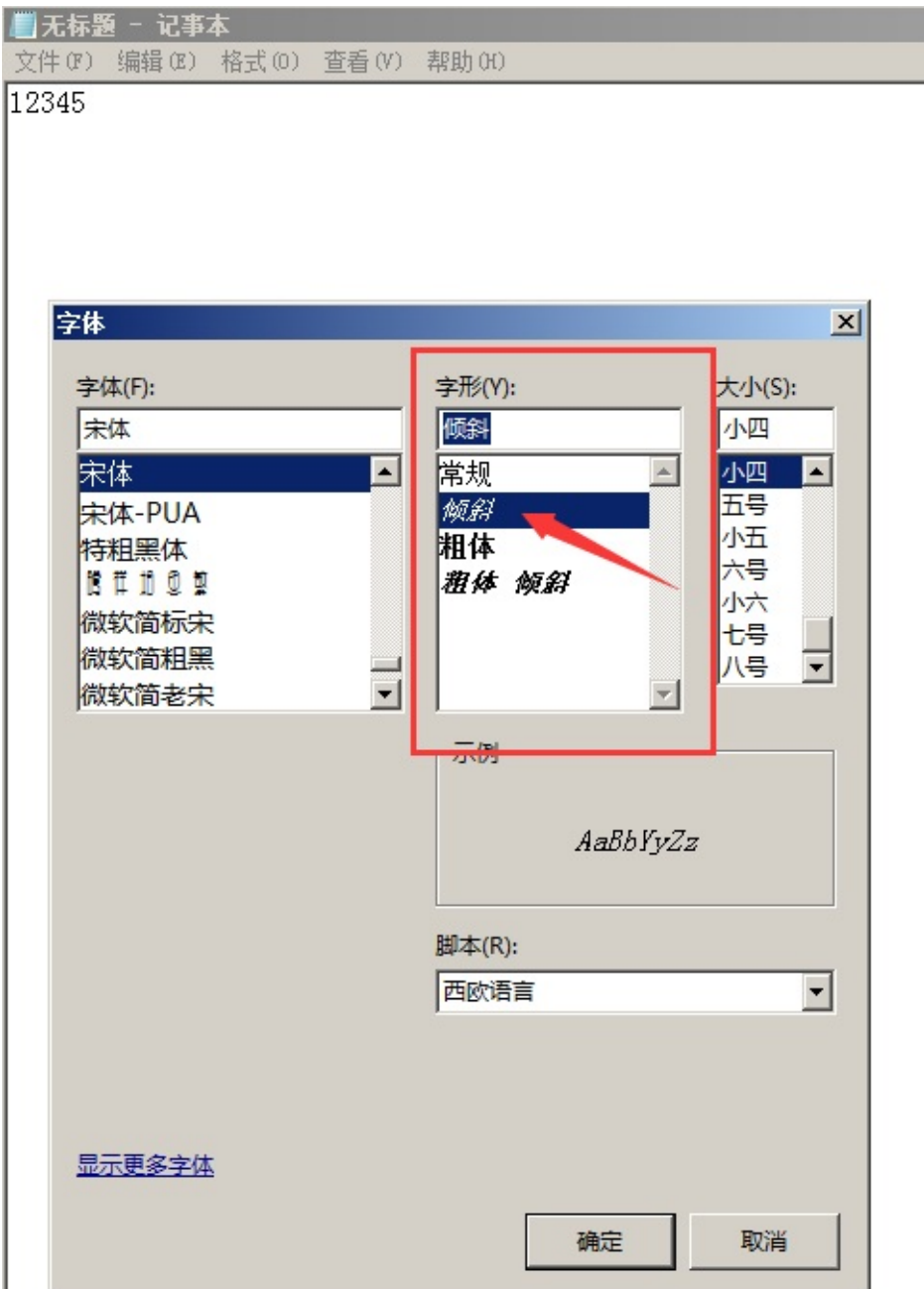
|   |
|---|
| 无 |
| - |

## 示例

说明：请打开记事本，然后点菜单“格式” - “字体”，运行中的每一步注意对照字体对话框界面，看看代码对应的变化情况



运行后：



代码：

```
Dim w As New clsWindow
Dim lngHwnd As Long
If w.GetWindowByTitle("字体", 0).hwnd <> 0 Then
    lngHwnd = w.GetElementHwndByClassName("ComboBox", 2) '根据类名获得“字型”组合框
    的句柄，第1个是“字体”，索引2位“字型”

    w.SelectComboBoxIndex lngHwnd, 1
    MsgBox "选中“倾斜”"

    w.SelectComboBoxIndex lngHwnd, 3
    MsgBox "选中了“粗体倾斜”"
```



```
w.SelectComboBoxIndex lngHwnd, 0  
MsgBox "选中“常规”"  
  
w.SelectComboBoxIndex lngHwnd, 2  
MsgBox "选中“粗体”"  
End If
```

# AppendElementTextByClassName 函数

## 描述

根据控件的类名追加显示控件的文字

## 语法

AppendElementTextByClassName(strClassName, strAppendText[, intIndex = 1[, UseRegExp = False]])

该语法中有下列命名参数：

| 参数            | 描述   |
|---------------|--|
| strClassName  | 必须的。String类型，表示要查找的控件类名。   |
| strAppendText | 必须的。String类型，表示要追加的文字内容。   |
| intIndex      | 可选的。Integer类型，表示次序，如果有多个类名相同的控件，那么这个参数就有用了，依次1、2、3。。。用来区分它们，根据自己的需要使用对应的索引获取，默认是1表示指定类名的第一个控件。这些输入框的索引可能并不是按照肉眼的顺序来的，如何区分它们请参考 <a href="#">目标窗体中存在多个相同类名的输入框该如何区分填写？</a> 。 |
| UseRegExp     | 可选的。Boolean 类型，表示是否使用正则。如果忽略，缺省值是 False，它表明默认不使用正则。  |

## 返回值

|   |
|---|
| Long                                    |
| 根据控件类名获取控件成功并且成功设置值则返回1，未获取到控件或设置值失败返回0 |

## 示例

说明：无需打开记事本，直接运行下面代码即可，不过当前有打开的记事本也没关系，不会产生干扰。

```
Dim w As New clsWindow
w.GetWindowByPID(Shell("notepad", 1)).Focus '根据pid直接获取窗口并设置为当前活动窗口
```

```
w.SetElementTextByClassName "Edit", "clsWindow" & w.Version & "发布 " & Now()  
    '设置输入框文字  
w.FadeIn '淡入  
w.AppendElementTextByClassName "Edit", vbCrLf & "功能全面" '向输入框追加内容  
w.FadeIn '淡入  
w.AppendElementTextByClassName "Edit", vbCrLf & "欢迎测试反馈" '向输入框追加内容
```

# Click 函数

## 描述

根据句柄点击目标窗口或控件，如果参数缺省则使用类对象的句柄。

## 语法

Click([hWnd1])

该语法中有下列命名参数：

| 参数    | 描述                                   |
|-------|--------------------------------------|
| hWnd1 | 可选的。Long类型。表示窗口句柄。如果缺省，则表示使用当前窗口的句柄。 |

## 返回值

| Long   |
|--|
| 操作成功点击则返回1，失败返回0，该返回值来自api函数PostMessage调用后的返回值 |

## 示例

说明：用spy++等工具获取某个按钮的句柄，比如12345，那么使用下面的方法点击这个按钮

```
Dim w As New clsWindow
w.Click(12345) '方法一
'w.GetWindowByHwnd(12345).Click '方法二
```

# ClickElementByText 函数

## 描述

根据控件的显示文字点击控件，这个使用场景非常多，比如点击“确定”按钮等。

## 语法

ClickElementByText(strText[, intIndex = 1[, UseRegExp = False]])

该语法中有下列命名参数：

| 参数        | 描述   |
|-----------|--|
| strText   | 必须的。String类型，表示要点击的控件的文字内容。  |
| intIndex  | 可选的。Integer类型，表示次序，如果有多个控件文字都一样，那么这个参数就有用了，依次1、2、3。。。用来区分它们，根据自己的需要使用对应的索引获取，默认是1表示符合条件的第一个控件。 |
| UseRegExp | 可选的。Boolean 类型，表示是否使用正则。如果忽略，缺省值是 False，它表明默认不使用正则。  |

## 返回值

|                                       |
|---------------------------------------|
| Long                                  |
| 根据控件内容获取控件成功并且成功点击则返回1，未获取到控件或点击失败返回0 |

## 示例



### 示例1

说明：打开记事本随便输入点内容，然后点击右上角关闭，这时我们会看到是否保存的对话框，我们用这个函数来点击“不保存”按钮，运行如下代码：

```
Dim w As New clsWindow
MsgBox w.GetWindowByTitle("记事本", 0).ClickElementByText("不保存(&N)")
```

示例1运行后我们会发现窗口关闭了，说明点击“不保存”按钮成功了。

### 示例2

说明：打开记事本随便输入点内容，然后点击右上角关闭，这时我们会看到是否保存的对话框，我们用这个函数来点击“保存”按钮，运行如下代码：

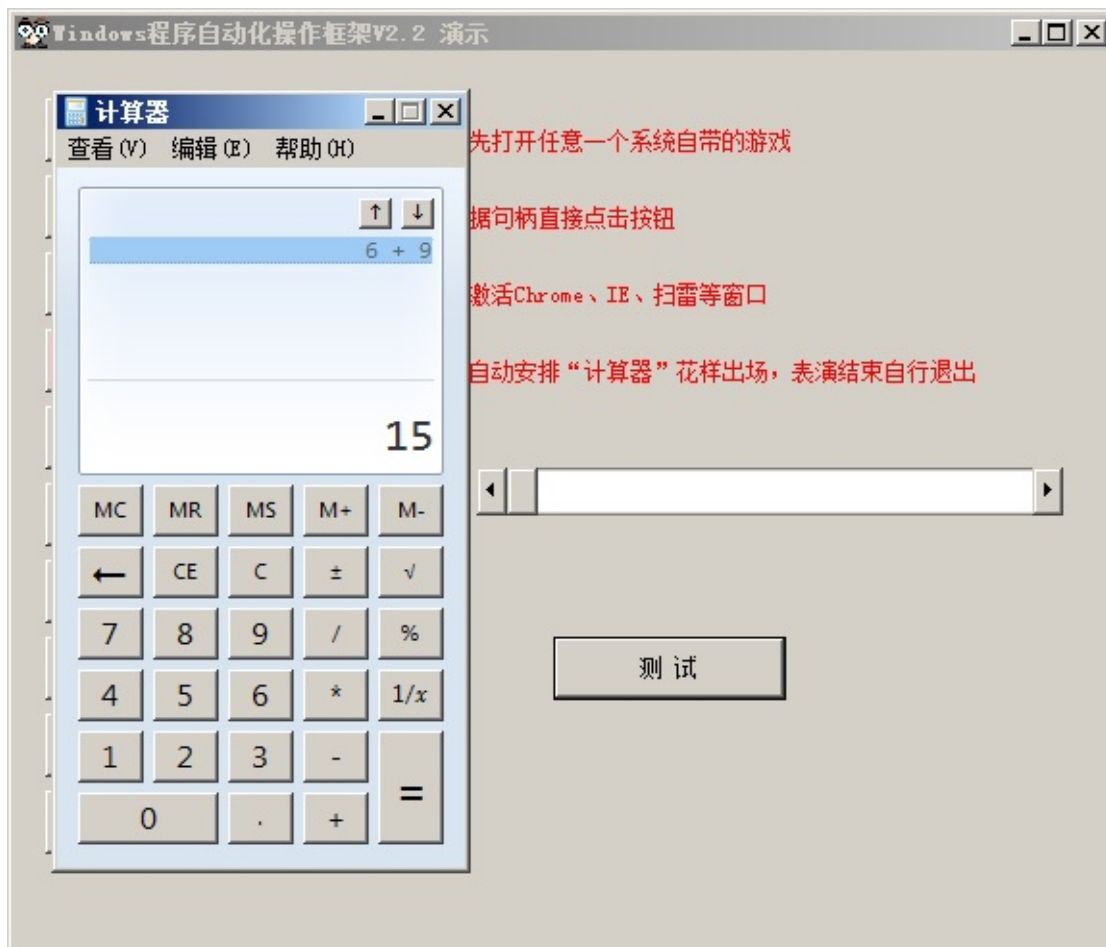
```
Dim w As New clsWindow
w.GetWindowByTitle("记事本", 0).ClickElementByText "保存(&S)"
```

### 示例3

说明：直接运行下面代码即可，效果为软件自动调用计算器，然后计算6+9的结果：

```
Private Sub Command1_Click()
    Dim w As New clsWindow
    w.GetWindowByPID (Shell("calc" , 1)) '打开计算器并获得它的句柄
    w.ClickElementByText "6"
    w.ClickElementByText "+"
End Sub
```

```
w.ClickElementByText "9"  
w.ClickElementByText "="  
End Sub
```



# ClickElementByClassName 函数

## 描述

根据控件的类名点击控件，这个使用场景非常多，比如点击“确定”按钮等。

## 语法

ClickElementByClassName(strClassName[, intIndex = 1[, UseRegExp = False]])

该语法中有下列命名参数：

| 参数           | 描述   |
|--------------|--|
| strClassName | 必须的。String类型，表示要点击的控件的类名。  |
| intIndex     | 可选的。Integer类型，表示次序，如果有多个类名相同的控件，那么这个参数就有用了，依次1、2、3。。。用来区分它们，根据自己的需要使用对应的索引获取，默认是1表示指定类名的第一个控件。这些输入框的索引可能并不是按照肉眼的顺序来的，如何区分它们请参考 <a href="#">目标窗体中存在多个相同类名的输入框该如何区分填写？</a> 。 |
| UseRegExp    | 可选的。Boolean 类型，表示是否使用正则。如果忽略，缺省值是 False，它表明默认不使用正则。  |

## 返回值

|                                       |
|---------------------------------------|
| Long                                  |
| 根据控件类名获取控件成功并且成功点击则返回1，未获取到控件或点击失败返回0 |

## 示例

### 示例1

说明：打开记事本，然后点击菜单“编辑 - 转到”，然后运行下面代码：

```
Private Sub Command1_Click()  
    Dim w As New clsWindow  
    If w.GetWindowByTitle("转到指定行", 0).hwnd <> 0 Then
```



```
w.SetElementTextByClassName "Edit", 999 '将行号改成999
w.ClickElementByClassName "Button" '点击“转到”按钮，如果要点击“取消”按钮可以
用w.ClickElementByClassName "Button", 1
End If
End Sub
```



## 四、鼠标坐标相关

---

### 简介

---

鼠标坐标相关部分主要是操作鼠标移动点击等。

### 相关函数或方法

---

SetCursor

MoveCursor ( 同SetCursor )

MoveCursorTo ( 同SetCursor )

SetPoint ( 同SetCursor )

GetCursorPoint

GetCursorPosCurrent ( 同GetCursorPoint )

ClickCurrentPoint

ClickPoint

# ClickCurrentPoint 方法

## 描述

点击当前坐标

## 语法

ClickCurrentPoint()

## 返回值

|   |
|---|
| 无 |
| - |

## 示例

说明：运行后会执行点击当前点，测试时注意，如果您放在按钮过程中，那么执行后代码点击当前的点就是当前按钮了，这样会陷入死循环，为了避免这个情况可以在加入一行代码，移开鼠标再执行点击当前点。

```
Dim w As New clsWindow
w.SetCursor 100, 100 '移动鼠标到要点击的坐标
w.ClickCurrentPoint
```

# ClickPoint 方法

## 描述

点击指定坐标，可以指定相对于本窗口还是相对于屏幕，还可以指定点击次数以及每次点击的前后延时。

## 语法

```
ClickPoint(x1, y1[, PositionMod = relative[, clickTimes = 1[, delayMillisecondsBefore = 0[,  
delayMillisecondsAfter = 0]]]])
```

该语法中有下列命名参数：

| 参数                      | 描述   |
|-------------------------|--|
| x1                      | Long类型，表示要点击的Left坐标值。  |
| y2                      | Long类型，表示要点击的Top坐标值。   |
| PositionMod             | 可选的。enumPositionMode 类型，表示坐标使用绝对还是相对方式。如果忽略，缺省值是relative，它表明坐标使用相对方式，例如点击100,100这个点，默认会点击相对于窗口左顶点的位置100,100，如果要相对于屏幕可以设置本参数值为absolute。关于enumWindowVisible类型的介绍请参考下面“enumPositionMode类型的设置值”表的介绍。 |
| clickTimes              | 可选的。Integer 类型，表示点击次数。如果忽略，缺省值是 1，它表明点击1次。   |
| delayMillisecondsBefore | 可选的。Long 类型，表示点击前延时多久，单位为毫秒(ms)。如果忽略，缺省值是 0，它表明点击前不延时。   |
| delayMillisecondsAfter  | 可选的。Long 类型，表示点击后延时多久，单位为毫秒(ms)。如果忽略，缺省值是 0，它表明点击后不延时。   |

enumPositionMode类型的设置值：

| 设置       | 描述       |
|----------|----------|
| absolute | 坐标使用绝对方式 |
| relative | 坐标使用相对方式 |

## 返回值

无

-

## 示例

说明：请向窗口添加按钮Command1，然后复制下面代码后点击按钮运行。运行前先打开记事本

```
Private Sub Command1_Click()
    Dim w As New clsWindow
    w.ClickPoint 200, 200 '点击坐标200,200
    w.GetWindowByClassName "Notepad" '获得记事本窗口
    w.Wait 500
    w.Focus '设置记事本窗口为活动窗口
    w.Wait 500
    w.ClickPoint 200, 200 '注意这里和上上行的区别，点击的坐标是相对于记事本左上坐标
    的，相当于(记事本做顶点+100,记事本上顶点+100)，未获取句柄之前是实际默认句柄是0，即桌面的窗口
    w.Caption = "当前鼠标移动到相对记事本左上的200,200处了"
    w.Wait 3000 '延时3秒以便观看效果
    w.ClickPoint 200, 200, relative '和上面一样，默认就是relative，即相对模式
    w.Wait 3000 '延时3秒以便观看效果
    w.ClickPoint 200, 200, absolute '绝对模式，是相对于桌面左上顶点开始计算的
    w.Caption = "当前鼠标移动到相对屏幕左上的200,200处了"
    w.Wait 3000 '延时3秒以便观看效果
    w.Caption = "以下延时根据菜单坐标点击相应菜单"
    w.Focus '设置记事本窗口为活动窗口
    w.ClickPoint 240, 30, , , 200, 300 '点击相对于记事本的(240,30)坐标处，实际是
    “帮助”菜单的位置，并且点击前延时200ms，点击后延时300ms
    w.ClickPoint 240, 80, , , 200, 300 '点击坐标(240,80)坐标，实际正好是二级菜单
    “关于记事本”的位置
End Sub
```

# SetCursor 方法

## 描述

移动鼠标到指定点

## 语法

SetCursor(x1, y1[, PositionMod = absolute[, delayMillisecondsBefore = 0[, delayMillisecondsAfter = 0]]])

该语法中有下列命名参数：

| 参数                      | 描述  |
|-------------------------|---|
| x1                      | Long类型，表示光标要移动到的Left坐标值。  |
| y2                      | Long类型，表示光标要移动到的Top坐标值。   |
| PositionMod             | 可选的。enumPositionMode 类型，表示坐标使用绝对还是相对方式。如果忽略，缺省值是absolute，它表明坐标使用绝对方式，例如点击100,100这个点，默认会点击相对于屏幕左顶点的位置100,100，如果要相对于当前窗口可以指定改参数为relative。至于为什么这里默认值和ClickPoint不一样，主要是考虑到移动光标更多的需求是绝对定位。关于enumWindowVisible类型的介绍请参考下面“enumPositionMode类型的设置值”表的介绍。 |
| clickTimes              | 可选的。Integer 类型，表示点击次数。如果忽略，缺省值是 1，它表明点击1次。  |
| delayMillisecondsBefore | 可选的。Long 类型，表示点击前延时多久，单位为毫秒(ms)。如果忽略，缺省值是 0，它表明点击前不延时。  |
| delayMillisecondsAfter  | 可选的。Long 类型，表示点击后延时多久，单位为毫秒(ms)。如果忽略，缺省值是 0，它表明点击后不延时。  |

enumPositionMode类型的设置值：

| 设置       | 描述       |
|----------|----------|
| absolute | 坐标使用绝对方式 |
| relative | 坐标使用相对方式 |

返回值

|   |
|---|
| 无 |
| - |

示例

说明：先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
w.SetCursor 100, 100 '再试试加上参数,relative 看看区别
```

# MoveCursor 方法 ( 同SetCursor )

---

同SetCursor , 请参考[SetCursor](#)的说明。



# MoveCursorTo 方法 ( 同SetCursor )

---

同SetCursor , 请参考[SetCursor](#)的说明。

# SetPoint 方法 ( 同SetCursor )

---

同SetCursor , 请参考[SetCursor](#)的说明。

# GetCursorPoint 函数

## 描述

直接获得当前鼠标下的窗口或控件的句柄

## 语法

GetCursorPoint()

该语法中有下列命名参数：

| 参数 | 描述 |
|----|----|
|----|----|

## 返回值

|                             |
|-----------------------------|
| String                      |
| 返回值为当前坐标值，用逗号间隔，例如"100,100" |

## 示例

说明：获得当前坐标

```
Dim w As New clsWindow
MsgBox w.GetCursorPoint
```

# GetCursorPosCurrent 函数 ( 同GetCursorPoint )

---

同GetCursorPoint , 请参考[GetCursorPoint](#)的说明。

## 五、应用程序相关

---

### 简介

---

应用程序相关部分主要是获得窗口所属的应用程序的一些信息，以及关闭等操作。

### 相关函数或方法

---

AppPath

AppName

CommandLine

AppCommandLine

CloseApp

ProcessID

PID ( 同ProcessID )

# AppPath 函数

## 描述

得到当前窗口所属的进程的全路径，也可以得到指定的窗口句柄所属进程的全路径。

## 语法

AppPath([hwn1])

该语法中有下列命名参数：

| 参数   | 描述   |
|------|--|
| hwn1 | 可选的。Variant类型，表示要获得进程路径的窗口句柄。如果缺省则使用当前clsWindow对象设置的句柄值。 |

## 返回值

|          |
|----------|
| String   |
| 路径的字符串内容 |

## 示例

说明：先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
MsgBox w.GetWindowByClassName("Notepad").AppPath
```

# AppName 函数

## 描述

得到当前窗口所属的进程的exe名，也可以得到指定的窗口句柄所属进程的exe名。

## 语法

AppName([hwnd1])

该语法中有下列命名参数：

| 参数   | 描述  |
|------|---|
| hwn1 | 可选的。Variant类型，表示要获得进程名的窗口句柄。如果缺省则使用当前clsWindow对象设置的句柄值。 |

## 返回值

|         |
|---------|
| String  |
| 返回值为进程名 |

## 示例

说明：先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
MsgBox w.GetWindowByClassName("Notepad").AppName
```

# IsWin64 函数

## 描述

判断当前是否是64位系统。

## 语法

IsWin64()

## 返回值

| Boolean                      |
|------------------------------|
| 如果当前是64位系统返回True , 否则返回False |

## 示例

说明：运行下面代码测试：

```
Dim w As New clsWindow
MsgBox w.IsWin64
```



# CommandLine 函数

## 描述

得到当前窗口所属的进程的启动参数，也可以得到指定的进程pid的启动参数。

## 语法

CommandLine([IPid ])

该语法中有下列命名参数：

| 参数   | 描述                                 |
|------|------------------------------------|
| IPid | 可选的。Long类型，表示进程ID。如果缺省则使用当前存储的PID。 |

## 返回值

|             |
|-------------|
| String      |
| 返回值为本身，方便连写 |

## 示例

说明：先打开记事本，从任务管理器里找到它的进程pid（例如12345），然后再运行下面代码。

```
Dim w As New clsWindow
MsgBox w.GetWindowByClassName(12345).CommandLine
```

# AppCommandLine 函数 ( 同CommandLine )

---

同CommandLine , 请参考[CommandLine](#)的说明。

# CloseApp 函数

## 描述

关闭当前窗口所属的进程，也可以关闭指定窗口所属的进程。

## 语法

CloseApp([hWnd1])

该语法中有下列命名参数：

| 参数    | 描述                                   |
|-------|--------------------------------------|
| hWnd1 | 可选的。Long类型。表示窗口句柄。如果缺省，则表示使用当前窗口的句柄。 |

## 返回值

| Long  |
|---|
| 操作成功点击则返回1，失败返回0，该返回值来自api函数TerminateProcess调用后的返回值 |

## 示例

说明：先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
w.GetWindowByClassName("Notepad").CloseApp
```

# ProcessID 函数

## 描述

得到当前窗口所属的进程的标识（也叫进程id、PID），也可以得到指定的窗口句柄所属进程的标识。

## 语法

ProcessID([hWnd1])

该语法中有下列命名参数：

| 参数    | 描述                                   |
|-------|--------------------------------------|
| hWnd1 | 可选的。Long类型。表示窗口句柄。如果缺省，则表示使用当前窗口的句柄。 |

## 返回值

|   |
|---|
| Long  |
| 操作成功点击则返回1，失败返回0，该返回值来自api函数<br>GetWindowThreadProcessId调用后的返回值 |

## 示例

说明：先打开记事本，然后再运行下面代码。

```
Dim w As New clsWindow
MsgBox w.GetWindowByClassName("Notepad").ProcessID
```

# PID 函数 ( 同ProcessID )

---

同ProcessID，请参考[ProcessID](#)的说明。

## 六、公有方法属性等

---

### 简介

---

公有方法属性等部分主要是类本身一些的操作，例如直接操作句柄，获得版本等等，除了wait方法一般不常用。

### 相关函数或方法

---

- hWnd
- DebugMe
- Wait
- Version

# hWnd 公有变量

## 描述

hWnd为当前类对应的窗口的句柄，一般获得窗口成功后会对这个全局变量赋值。如果您知道目标窗口的句柄可以直接对此公有变量进行赋值。一般用来查看当前窗口的句柄。

## 示例

说明：根据窗口标题“记事本”获得窗口，然后显示该窗口的句柄。

```
Dim w As New clsWindow  
MsgBox w.GetWindowByTitleEx("记事本").hWnd
```

# DebugMe 公有变量

## 描述


DebugMe为开发人员设置的调试用的。默认为False，如果设置为True，类将在每调用一个函数都会生成一个日志文件，名称为DEBUG.txt，如果查询窗体内元素还会生成一个controls.txt文件，便于开发人员分析使用。

## 示例

说明：设置DebugMe为True，然后尝试调用一些函数看看生成的文件内容。运行前请先随便打开一个记事本，记事本内随便放一些文字内容。

```
Dim w As New clsWindow
w.DebugMe = True
MsgBox w.GetWindowByTitleEx("记事本").hWnd '此时会看到有文件DEBUG.txt
MsgBox w.GetElementTextByClassName("Edit") '此时会看到有文件controls.txt
```

我们看到程序旁生成了两个日志文件。

| 名称   | 修改日期 ▾          | 类型   | 大小   |
|--|-----------------|------|------|
|  controls.txt | 2020-02-02 2:20 | 文本文档 | 1 KB |
|  DEBUG.txt    | 2020-02-02 2:20 | 文本文档 | 1 KB |

DEBUG.txt 文件内容为：

```
2020-02-02 02:20:44 函数GetWindowByTitleEx被调用
参数为 strTitle=记事本      intWaitSeconds=10
2020-02-02 02:20:45 函数GetElementTextByClassName被调用
参数为 strClassName=Edit   intIndex=1
2020-02-02 02:20:45 函数GetElementTextByClassName中调用函数getMatchHwndFromWindow
2020-02-02 02:20:45 函数getMatchHwndFromWindow被调用
参数为 Pattern=^([-\\d]+)\\s+([-\\d]+)\\sEdit\\s.*?$      intIndex=1
2020-02-02 02:20:45 函数getMatchHwndFromWindow中调用函数ControlsInfo, 参数为 hWnd_=
1573918
2020-02-02 02:20:45 函数getMatchHwndFromWindow执行完毕
匹配数量为=1
```

controls.txt 文件内容为：

```
1573918 0   Notepad 新建文本文档.txt - 记事本
1901478 15  Edit      test
```



|        |      |                    |
|--------|------|--------------------|
| 722488 | 1025 | msctls_statusbar32 |
|--------|------|--------------------|

# Wait 方法

## 描述

延时指定毫秒数

## 语法

Wait(Milliseconds)

该语法中有下列命名参数：

| 参数           | 描述                     |
|--------------|------------------------|
| Milliseconds | 必须的。Long类型，表示需要延时的毫秒数。 |

## 示例

说明：延时指定的毫秒数

```
Dim w As New clsWindow
w.Wait 500
MsgBox "延时500毫秒完成"
w.Wait 1000
MsgBox "延时1秒完成"
```

# Version 方法

## 描述

得到版本信息

## 语法

Version(\*\*)

## 返回值

|                     |
|---------------------|
| String              |
| 返回值为当前clsWindow类版本号 |

## 示例

说明：获取当前clsWindow类版本号

```
Dim w As New clsWindow
MsgBox w.Version
```

# 使用范例

---

- 01 控制微信发送消息
- 02 控制QQ发送消息
- 03 QQ登录器
- 04 花式调用计算器
- 05 用鼠标画圆
- 06 控制记事本输入并保存
- 07 随机移动鼠标防锁屏
- 08 根据ID进入腾讯课堂
- 09 限制程序重复打开
- 10 实战：微便签-窗口特效
- 11 实战：窗口图钉-窗口置顶

# 01 控制微信发送消息

说明：

本案例是通过clsWindow2.2控制微信桌面版来发送消息的。原理是模拟人工操作：进入微信桌面版 -> 点击搜索框 -> 输入对应人员名 -> 点击搜索结果 -> 输入框中输入消息 -> 按回车，代码注释很详细，看下就能明白，有一定vb基础的人可以改成群发，定时发送，群中@发送。

测试环境：

win7x64 + 微信桌面版2.6.2.22 + clsWindow2.2

提示：如果不能用可能是以下原因：

- 微信版本太旧不兼容当前代码。对应措施：修改相应坐标
- 微信更新导致一些控件位置大小改变。对应措施：修改相应坐标位置
- 操作系统性能反应太慢。对应措施：可在需要的步骤加延时

如果有以上问题，请仔细检查并修改相应代码。本范例代码2020-02-05测试通过，不保证更新。

声明：

本代码免费，可用于任何商业用途，但请勿做非法用途，所引起的一切后果由使用者本人承担。

代码：

用法参考： `sendWeixinMsg "小张","你好"`

```
Private Sub sendWeixinMsg(ByVal strName$, ByVal strMsg$)
    Dim w As New clsWindow
    If w.GetWindowByClassName("WeChatMainWndForPC").hWnd <> 0 Then
        w.Focus
        w.ClickPoint 30, 100, , , 200 '延时200ms后点击聊天板块
        w.ClickPoint 130, 40, , , 200, 500 '延时200ms后点击搜索框，点击后再延时500ms
        SendKeys strName
        w.ClickPoint 150, 130, , , 500, 500 '点击搜索结果
        SendKeys strMsg & "{ENTER}"
    Else
        MsgBox "未发现微信桌面版，请打开微信桌面版再测试！", vbExclamation
    End If
    Set w = Nothing
End Sub
```

以下专门为一些性能较低的机器修改的慢速版本：

```

Private Sub sendWeixinMsg(ByVal strName$, ByVal strMsg$)
    Dim w As New clsWindow
    If w.GetWindowByClassName("WeChatMainWndForPC").hwnd <> 0 Then
        w.SetPosNormal
        w.Focus
        w.ClickPoint 30, 100, , , 1200 '延时200ms后点击聊天板块

        w.Focus

        w.ClickPoint 130, 40, , , 1200, 1500 '延时200ms后点击搜索框, 点击后再延时500
ms
        w.Focus
        SendKeys strName

        w.ClickPoint 150, 130, , , 1500, 1500 '点击搜索结果
        w.Focus
        SendKeys strMsg & "{ENTER}"
        w.Wait 500
    Else
        MsgBox "未发现微信桌面版, 请打开微信桌面版再测试!", vbExclamation
    End If
    Set w = Nothing
End Sub

```

完整工程下载：<https://pan.baidu.com/s/1JxXC9XA87cGaKqgD9qjQ7Q> 提取码: n2rh



## 02 控制QQ发送消息

说明：

本案例是使用clsWindow2.2控制QQ桌面版来发送消息的。原理是模拟人工操作：选择对应的人员或群 -> 点击消息框 -> 输入框中输入消息 -> 按回车，代码注释很详细，看下就能明白，有一定vb基础的人可以改成群发，定时发送，群中@发送。

测试环境：

win7x64 + QQ9.2.2 + clsWindow2.2

说明：如果不能用可能是您的版本太旧，或者QQ软件更新导致一些控件位置大小改变，请仔细检查并修改相应代码。以下代码2020-02-05测试通过，本代码不保证更新。

声明：

本代码免费，可用于任何商业用途，但请勿做非法用途，所引起的一切后果由使用者本人承担。

代码：

用法参考：`sendQQMsg "clswindow交流群", "大家好, clswindow框架交流群QQ: 788028734"`

```
Private Sub sendQQMsg(ByVal strName$, ByVal strMsg$)
    Dim w As New clsWindow
    If w.GetWindowByTitle(strName).hWnd <> 0 Then
        w.Normal '设置窗口正常，防止当前是最小化到任务栏的
        w.Focus
        w.ClickPoint w.Left + 35, w.Top + w.Height - 100, absolute, , 300, 500
        SendKeys strMsg & "{ENTER}"
    Else
        MsgBox "未发现包含“strName”的QQ聊天窗口，请打开对应的窗口再测试，注意请在面板上取消勾选“合并勾选窗口”， vbExclamation
    End If
    Set w = Nothing
End Sub
```



## 03 QQ登录器

说明：

本案例是使用clsWindow2.2通过程序自动登录QQ的，使得登录QQ的过程变得简单，修改下可以变成批量登录QQ的工具，方便一些拥有多个QQ的人定时批量登录QQ防回收。

测试环境：

win7x64 + QQ9.2.2 + clsWindow2.2

说明：如果不能用可能是您的版本太旧，或者QQ软件更新导致一些控件位置大小改变，请仔细检查并修改相应代码。以下代码2020-02-05测试通过，本代码不保证更新。

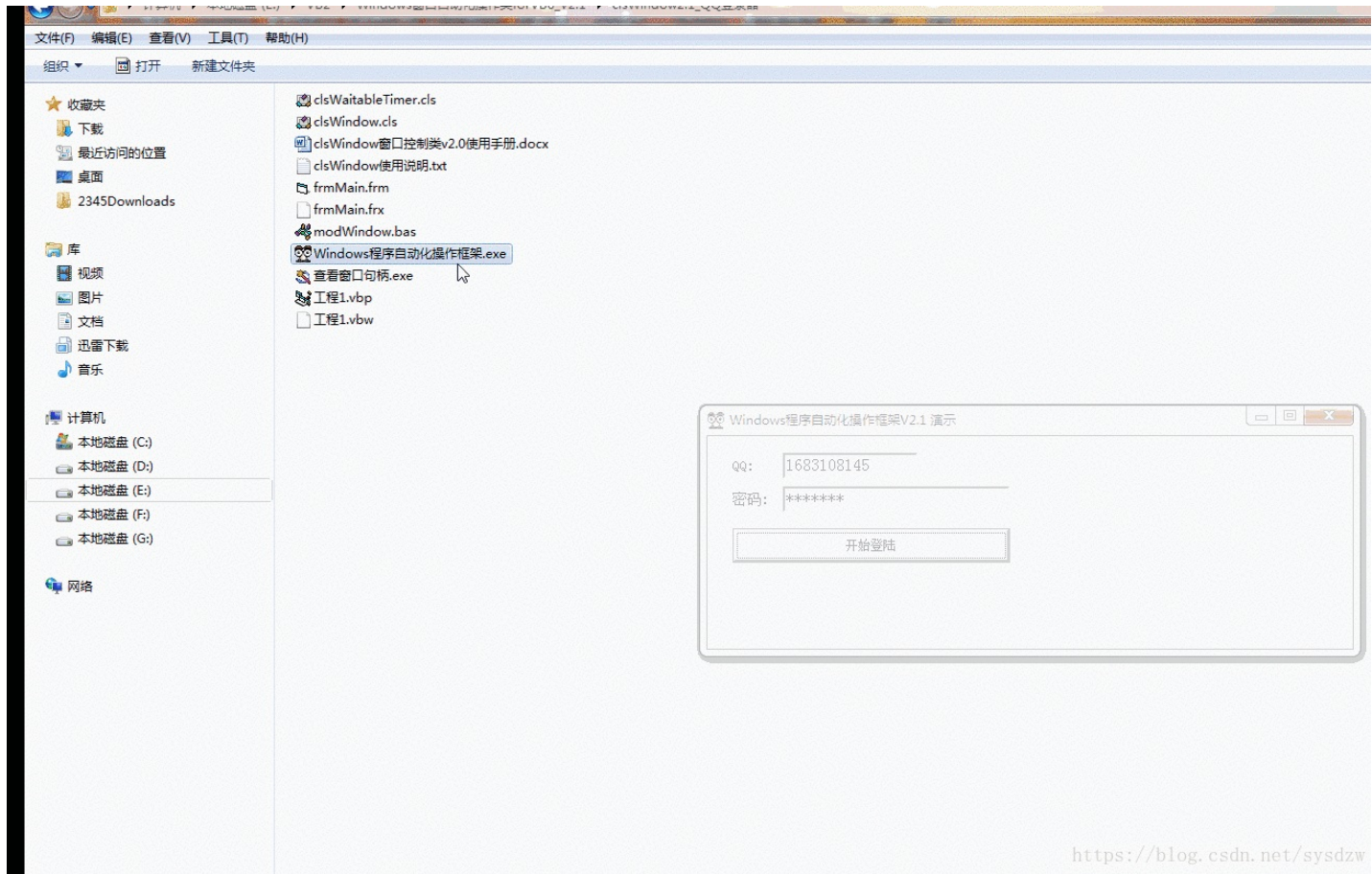
声明：

本代码免费，可用于任何商业用途，但请勿做非法用途，所引起的一切后果由使用者本人承担。

代码：

用法参考：`loginQQ "12345", "pass123"`

```
Private Sub loginQQ(ByVal strQQ$, ByVal strPass$)
    Dim w As New clsWindow
    Const strQQPath = "D:\Program Files (x86)\Tencent\QQ\Bin\qq.exe"
    Shell strQQPath, 1
    Do While w.Width <> 495
        w.GetWindowByTitle "QQ", 1
    Loop
    w.Focus
    w.ClickPoint 163, 247, , , 500, 700
    SendKeys "{DEL}"
    SendKeys strQQ
    w.WAIT 500
    SendKeys "{TAB}"
    w.WAIT 500
    SendKeys strPass
    SendKeys "{ENTER}"
    Set w = Nothing
End Sub
```



<https://blog.csdn.net/sysdzw>

## 04 花式调用计算器

说明：

本案例是使用clsWindow2.2框架调用系统计算机做一些操作，旨在熟悉框架的控件操作部分的一些函数用法。

测试环境：

win7x64 + clsWindow2.2

说明： 如果不能用可能是您的版本太旧，或者操作系统版本不一样，请仔细检查并修改相应代码。以下代码2020-02-05测试通过，本代码不保证更新。

声明：

本代码免费，可用于任何商业用途，但请勿做非法用途，所引起的一切后果由使用者本人承担。

代码：

请向窗口添加按钮Command1，然后复制下面代码后点击按钮运行。

用法参考：`loginQQ "12345", "pass123"`

```
'设置窗口位置
Private Sub Command1_Click()
    Dim i&, j&, lngLeft&, lngTop&, strHwnd$, v
    Command1.Enabled = False
    Dim w As New clsWindow

    '横排依次打开4个计算器
    For i = 1 To 4
        w.GetWindowByTitleEx "计算器", 2, Shell("calc.exe", 1)
        w.Caption = "句柄为:" & w.hWnd
        w.Move lngLeft, 300
        lngLeft = lngLeft + w.Width
        strHwnd = strHwnd & w.hWnd & ", "
    Next

    w.GetWindowByTitleEx "计算器", 2, Shell("calc.exe", 1)
    strHwnd = strHwnd & w.hWnd & ", "
    w.Wait 20
    '向下移动
    For i = 0 To Screen.Height / 15 - w.Height Step 10
        w.Wait 5
        w.Move lngLeft, i
    Next
```

```

'再回头
For i = Screen.Height / 15 - w.Height To 300 Step -10
    w.Wait 5
    w.Move lngLeft, i
Next
w.Move lngLeft, 300

'竖排依次打开3个计算器，并且处理为渐渐移动到右边的效果
For i = 1 To 3
    w.GetWindowByTitleEx "计算器", 2, Shell("calc.exe", 1)
    w.Caption = "句柄为:" & w.hwnd
    For j = 0 To Screen.Width / 15 - w.Width Step 10
        w.Wait 10
        w.Caption = "坐标(" & j & "." & lngTop & ")"
        w.Move j, lngTop
    Next
    lngTop = lngTop + w.Height
    strHwnd = strHwnd & w.hwnd & ","
Next

v = Split(strHwnd, ",")

'激活测试，窗口全部激活一遍，注意标题会闪动一下
w.Wait 1000
For i = 0 To UBound(v) - 1
    w.hwnd = v(i)
    w.Focus
    w.FadeIn 20
Next

'1秒后逐步关闭所有刚刚打开的窗口
w.Wait 1000
For i = UBound(v) - 1 To 0 Step -1
    w.hwnd = v(i)
    w.Wait 100 '延时是为了看到逐步关闭的进程，不然一下子关了看不到效果
    w.CloseWindow
Next
Command1.Enabled = True
End Sub

```

## 05 用鼠标画圆

说明：

本案例是使用clsWindow2.2框架调用系统计算机做一些操作，旨在熟悉框架的控件操作部分的一些函数用法。

测试环境：

win7x64 + clsWindow2.2

说明： 如果不能用可能是您的版本太旧，或者操作系统版本不一样，请仔细检查并修改相应代码。以下代码2020-02-05测试通过，本代码不保证更新。

声明：

本代码免费，可用于任何商业用途，但请勿做非法用途，所引起的一切后果由使用者本人承担。

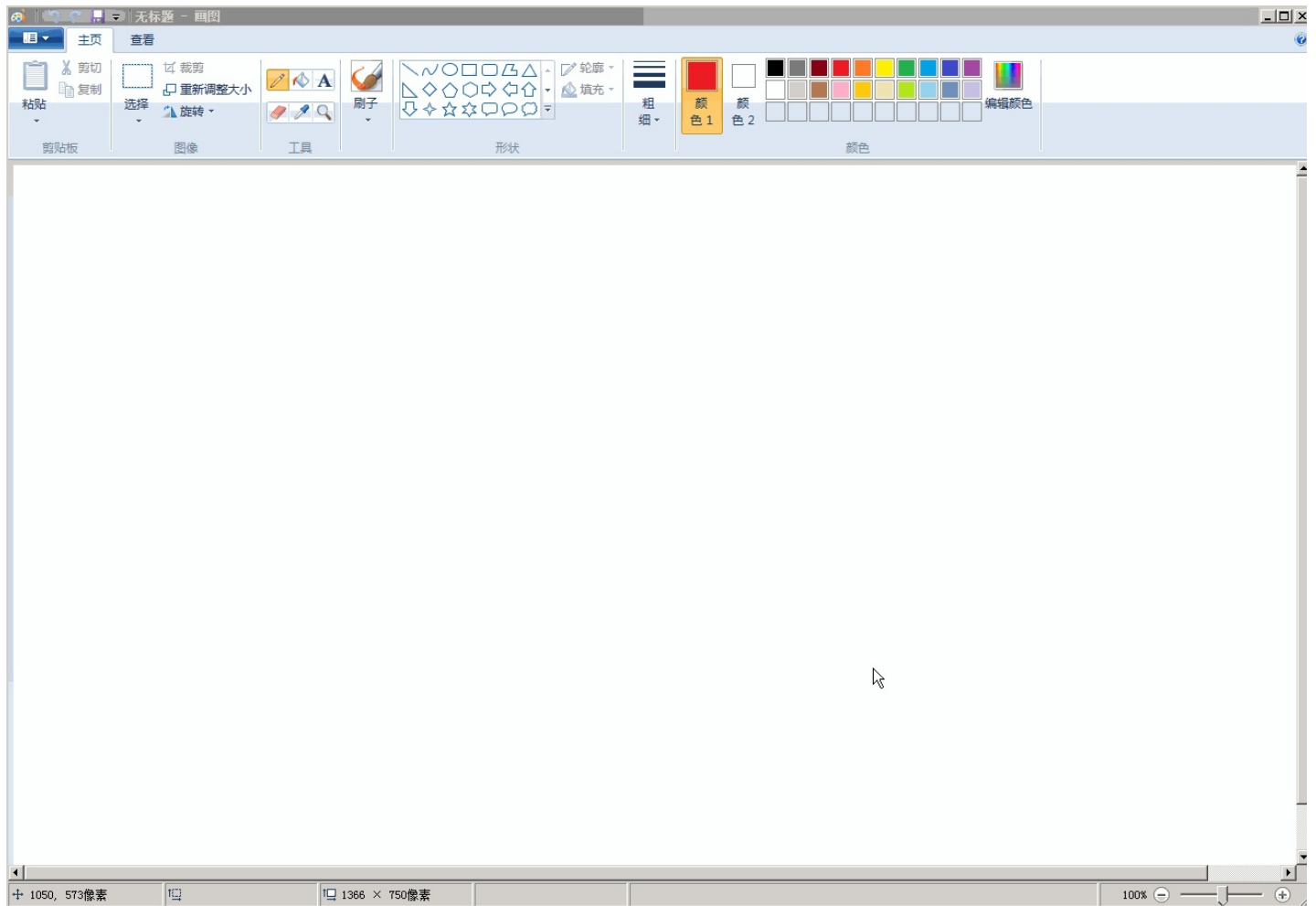
代码：

请向窗口添加按钮Command1，然后复制下面代码后点击按钮运行。

```
Private Sub Command1_Click()  
    Dim i%, w As New clsWindow  
    Dim k As Single  
    w.Wait 5000 '留5秒钟把鼠标放到画图工具上，一定要在5秒内放过去，不然鼠标在桌面上乱点击的  
    !  
    Const pi As Single = 3.14159  
    Do While k < 2 * pi  
        w.ClickPoint Cos(k) * 300 / 4 + 500, Sin(k) * 500 / 4 + 300, , , 10  
        k = k + pi / 180  
        DoEvents  
    Loop  
End Sub
```

运行后我们会看到鼠标在桌面上移动，并且移动轨迹是一个圆，如果把SetCursor改成ClickPoint，然后打开Windows画图工具，然后运行代码会看到它画了一个标准的点阵状的圆，是不是很有意思呢，如果你的水平够高可以空中写字画爱心，O(∩\_∩)O

## 05 用鼠标画圆



工程打包下载地址：[https://pan.baidu.com/s/1eR1CPwwGH830\\_Yl6U\\_kz2Q](https://pan.baidu.com/s/1eR1CPwwGH830_Yl6U_kz2Q) 提取码: kg69

## 06 控制记事本输入并保存

说明：

本案例是使用clsWindow2.2框架调用记事本做一些操作，旨在熟悉框架的控件操作部分的一些函数用法。

测试环境：

win7x64 + clsWindow2.2

说明： 如果不能用可能是您的版本太旧，或者操作系统版本不一样，请仔细检查并修改相应代码。以下代码2020-02-06测试通过，本代码不保证更新。

声明：

本代码免费，可用于任何商业用途，但请勿做非法用途，所引起的一切后果由使用者本人承担。

代码：

请向窗口添加按钮Command1，然后复制下面代码后点击按钮运行。

```
'调用记事本然后写入一些内容后保存到c:\test日期.txt
Private Sub Command1_Click()
    Dim w As New clsWindow
    w.GetWindowByPID(Shell("notepad", 1)).Focus '根据pid直接获取窗口并设置为当前活动窗口
    w.Caption = "看到记事本打开了吗？" & Now '设置应用程序标题内容
    w.Shake '抖动窗口，可以通过参数调节抖动方向、速度、幅度、次数

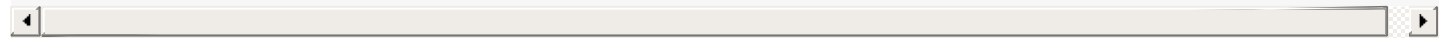
    w.SetElementTextByClassName "Edit", "clsWindow" & w.Version & "发布 " & Now(
) '设置输入框文字
    w.Wait 1000
    w.FadeOut '淡出
    w.AppendElementTextByClassName "Edit", vbCrLf & "功能全面" '向输入框追加内容
    w.FadeIn '淡入
    w.AppendElementTextByClassName "Edit", vbCrLf & "欢迎测试反馈" '向输入框追加内容
    w.Wait 1000

    w.Focus
    w.Wait 500
    SendKeys "^{s}" '设置焦点后按快捷键保存
    Dim w2 As New clsWindow
    w2.GetWindowByTitleEx("另存为").SetElementTextByClassName "Edit", "c:\test" &
Format(Now, "yyyymmddhhnnss") & ".txt"
```

```
w2.ClickElementByText "保存(&S)"
```

```
MsgBox w.GetElementTextByClassName("Edit")
```

```
End Sub
```





## 07 随机移动鼠标防锁屏

说明：

本案例是使用clsWindow2.2框架简单调用鼠标随机移动，简单一招防止电脑休眠。

测试环境：

win7x64 + clsWindow2.2

说明： 如果不能用可能是您的版本太旧，或者操作系统版本不一样，请仔细检查并修改相应代码。以下代码2020-02-22测试通过，本代码不保证更新。

声明：

本代码免费，可用于任何商业用途，但请勿做非法用途，所引起的一切后果由使用者本人承担。

代码：

请向窗口添加按钮Command1，然后复制下面代码后点击按钮运行。

```
Private Sub Command1_Click()  
    Dim w As New clsWindow  
    Randomize  
    Do  
        x1 = Screen.Width / 15 * Rnd  
        y1 = Screen.Height / 15 * Rnd  
        w.SetCursor x1, y1 '随机移动到屏幕内任意一个点  
        w.Wait 5000 '等待5秒钟  
    Loop  
End Sub
```

运行后我们会看到鼠标每隔5秒便会动一次，这样电脑就不会进入到休眠状态了。虽然电脑可以设置防止休眠，但是很多人反映按照百度搜的教程设置了结果电脑还是休眠了，用这个模拟鼠标随机移动让电脑一直以为有人在使用，这样就不会休眠或者待机了。运行后建议关闭显示器，省一点电 o(￣▽￣)o

工程打包下载地址：<https://pan.baidu.com/s/1tnehLHIjqxiCvYj27-4Ssg> 提取码: bsy9

以上工程代码比较全，包含画圆画线段、不同大小、不同长度方向等动作，让电脑不再感到单调无聊：)

# 08 根据ID进入腾讯课堂

说明：

本案例是使用clsWindow2.2框架自动化操作腾讯课堂，根据指定课堂ID进入直播。

测试环境：

win7x64 + clsWindow2.2 + 腾讯课堂客户端V1.5.4.35

说明： 如果不能用可能是您的版本太旧，或者操作系统版本不一样，请仔细检查并修改相应代码。以下代码2020-02-28测试通过，本代码不保证更新。

声明：

本代码免费，可用于任何商业用途，但请勿做非法用途，所引起的一切后果由使用者本人承担。

代码：

请向窗口添加按钮Command1，复制下面代码后，打开腾讯课堂并登录，然后运行。

```
'点击相应菜单打开窗口“打开课堂”
Private Sub Command1_Click()
    Dim w As New clsWindow
    If w.GetWindowByTitleEx("腾讯课堂", 1).Hwnd <> 0 Then
        w.Focus
        w.ClickPoint 30, 30, , , 100, 500 '点击左上角坐标拉出菜单，也就是图中1处
        w.ClickPoint 30, 150, , , 100, 500 '点击菜单“更多”坐标，也就是图中2处
        w.ClickPoint 230, 150, , , 100, 500 '点击菜单“上课ID进入课堂”坐标，也就是图中3
    处
    End If
End Sub
'输入ID进入课堂
Private Sub Command2_Click()
    Dim w As New clsWindow
    If w.GetWindowByTitle("打开课堂", 1).Hwnd <> 0 Then
        w.Focus
        SendKeys "100140792" & vbTab & vbCrLf '制表符表示切换到按钮，然后发送回车表示
    点击确定
    End If
End Sub
```

第一个按钮是找到腾讯课堂窗口然后点击一些列菜单打开课堂窗口，第二个按钮是自动输入id然后进入的。



按钮1操作如下图所示：



按钮2操作如下图所示：



工程打包下载地址：[https://pan.baidu.com/s/1p6e8drFmSriuyHS\\_uZw4HA](https://pan.baidu.com/s/1p6e8drFmSriuyHS_uZw4HA) 提取码: pjyf

## 09 限制程序重复打开

说明：

本案例是使用clsWindow2.2框架，限制程序重复打开，让相同进程重复打开仅保留最早实例并激活窗口。

测试环境：

win7x64 + clsWindow2.2

说明： 如果不能用可能是您的版本太旧，或者操作系统版本不一样，请仔细检查并修改相应代码。以下代码2020-10-21测试通过，本代码不保证更新。

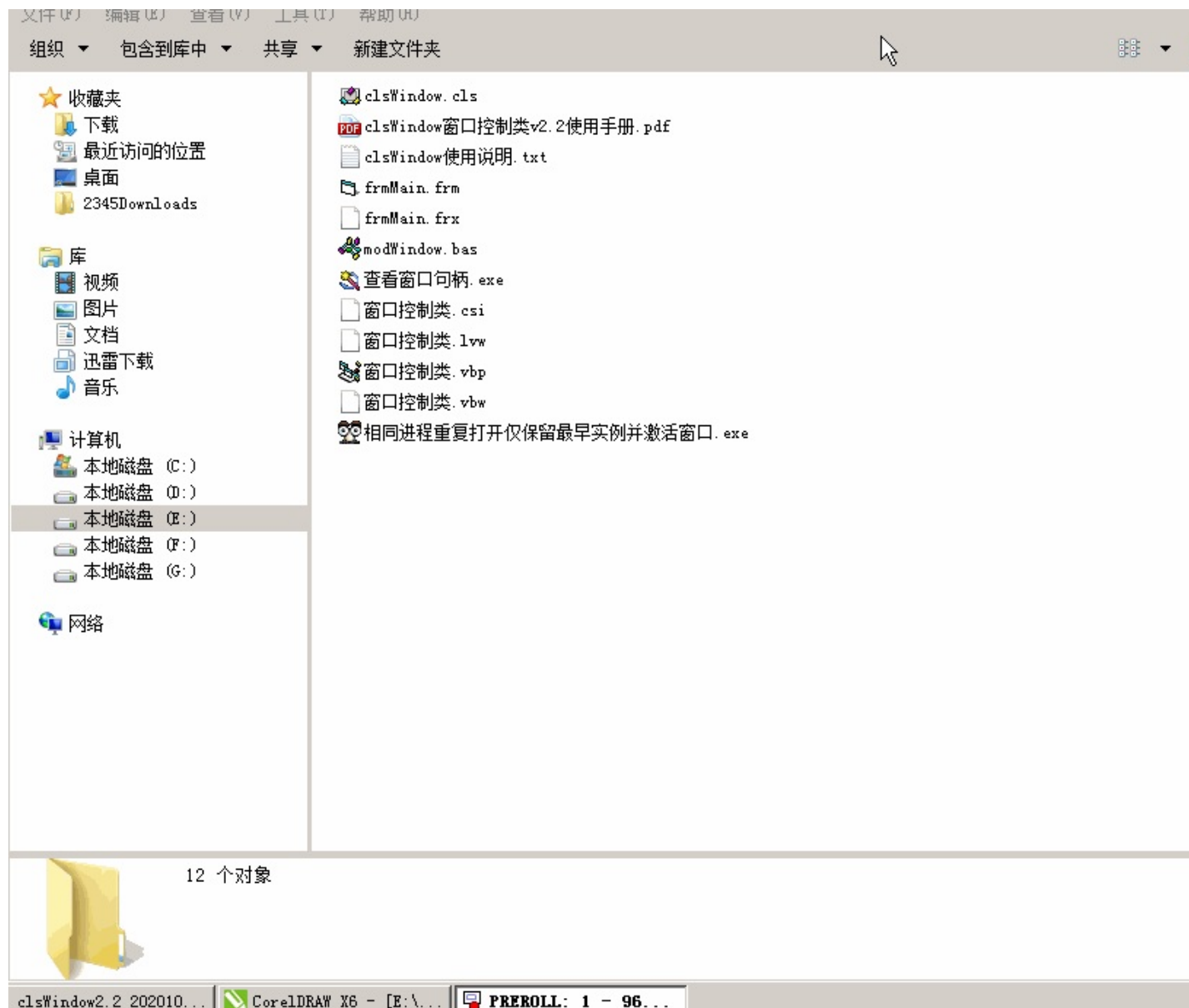
声明：

本代码免费，可用于任何商业用途，但请勿做非法用途，所引起的一切后果由使用者本人承担。

代码：

请向添加Form1窗口，然后复制下面代码进去，编译好exe程序后双击测试，请勿直接在VB开发环境测试。

```
Private Sub Form_Load()  
    Dim w As New clsWindow  
    If w.GetWindowByAppName(App.EXENAME, 0, , DisplayedWindow).hWnd <> 0 Then  
        w.Focus  
        w.Shake '抖动  
    End  
End If  
End Sub
```



工程打包下载地址：链接: <https://pan.baidu.com/s/1BV9ZiJ2VFRcGinnREjRI3w> 提取码: v4nr

如果目标窗体是隐藏的，或者工程中有多个窗体该怎么办？可以下载参考，里面提供了多种方法。

# 10 实战：微便签-窗口特效

说明：

本案例和clsWindow框架关系不大，主要是使用clsWindow2.2中的一些特效和窗口管理，使得软件开发进程加快，事半功倍。

软件叫做微便签，是一款windows操作系统下的便签软件，完全可以替代系统自带的便笺，每个便签可单独设置提醒，可设置半透明。便签平常都是在系统右下角托盘区，不占用任务栏。在底部有完整的源码下载。

测试环境：win7x64

提示：如果在xp等较低版本的操作系统上运行可能会发生错误，可以更换图标编译，或者直接下载我编译好的适合xp下运行的版本。

部分代码摘录：

说明：这是Form\_Load过程中的部分代码，可以看到大量使用了clsWindow框架的方法，例如fadein淡入，设置窗口置顶，以及半透明等等。

```
Dim w As New clsWindow
w.hwnd = Me.hwnd
picMain.BackColor = vbWhite
If strInitData <> "" Then '说明是需要初始化的
    w.MakeTransparent 0 '防止闪动
    Me.Show

    w.FadeIn 50
    If v(6) <> 100 Then '如果指定透明度了的那么还要设置下
        w.Transparent v(6)
        HScroll1.Value = v(6)
    End If

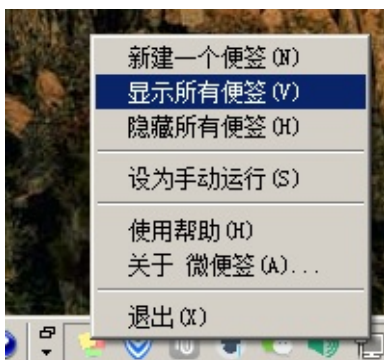
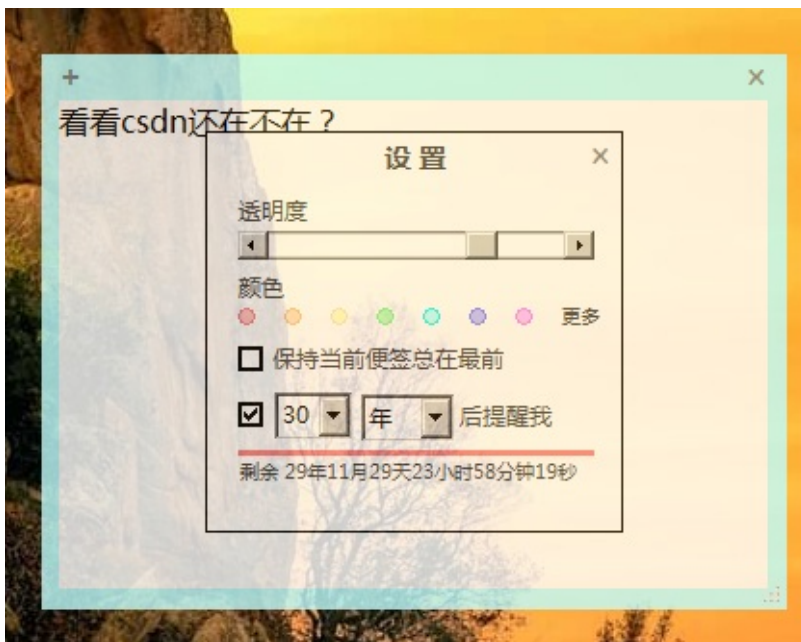
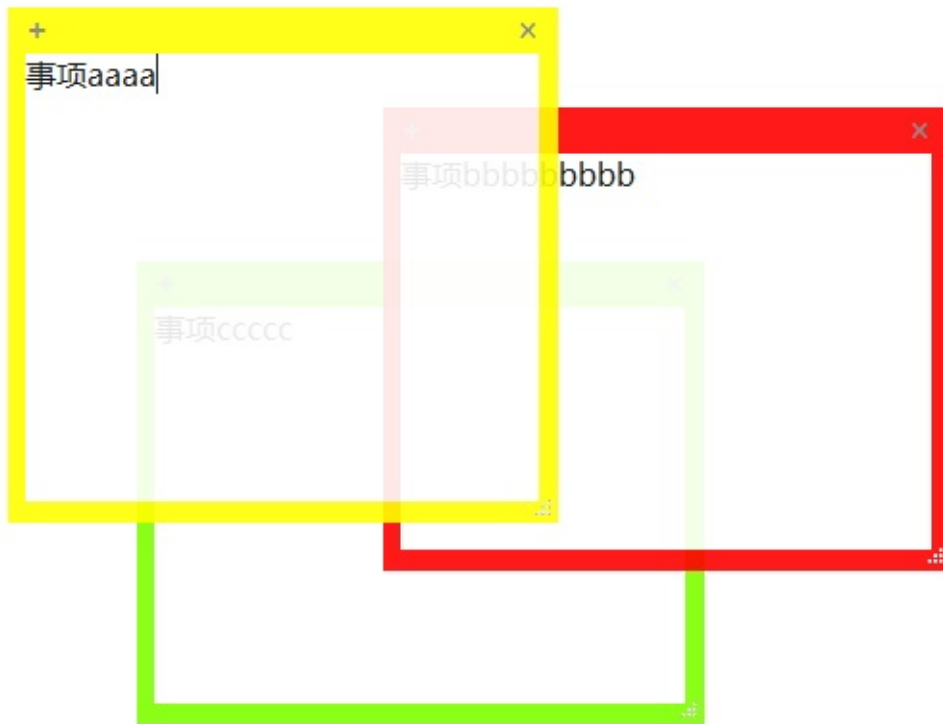
    .
    .
    .

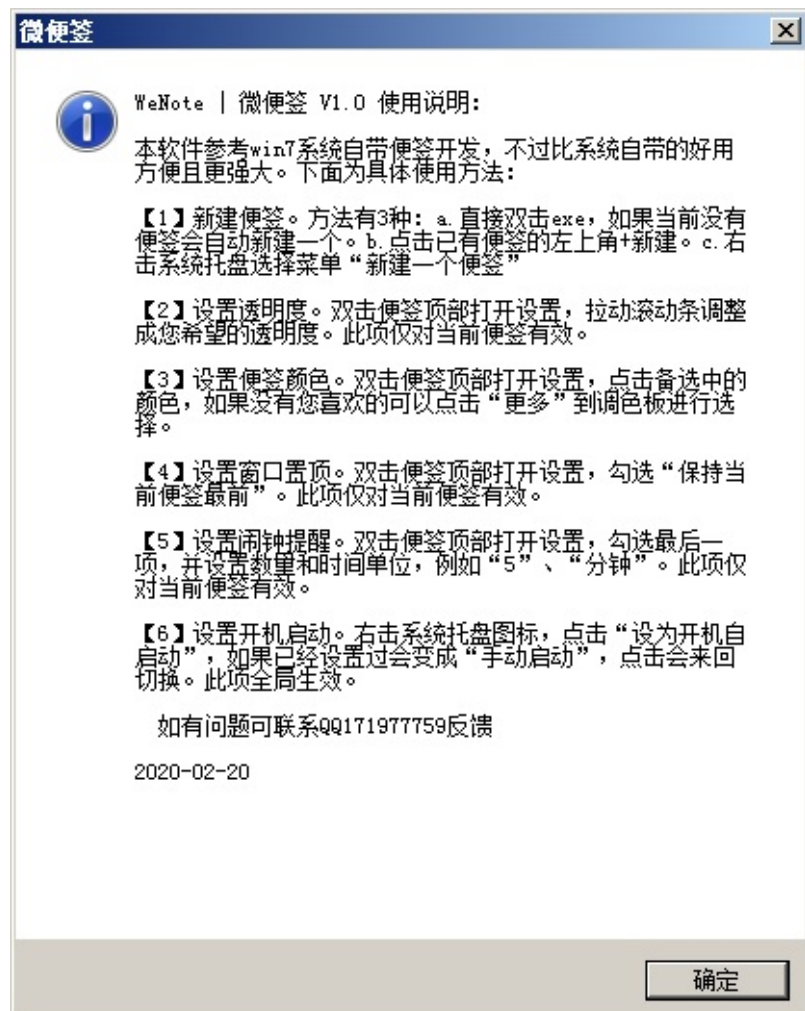
    If v(8) = "是" Then
        w.SetTop
        Check1.Value = 1
    End If
Else
    w.MakeTransparent 0
    Me.Show
    w.FadeIn 50
```

```

HScroll1.Value = 90
w.Transparent HScroll1.Value
End If

```





## 软件及源码下载

软件下载：<https://github.com/sysdzw/WeNote/raw/master/WeNote.exe>

源码(github)：<https://github.com/sysdzw/WeNote>

源码(gitee)：<https://gitee.com/sysdzw/WeNote>

xp下的成品软件如果不能运行请下载：[https://github.com/sysdzw/WeNote/raw/master/WeNote\\_for\\_xp.exe](https://github.com/sysdzw/WeNote/raw/master/WeNote_for_xp.exe)



# 11 实战：窗口图钉-窗口置顶

说明：

本案例基于clsWindow开发的一个小软件，功能是方便的对目标窗口设置置顶还是取消置顶。软件启动后会在每个窗口的标题栏右端加一个置顶功能的小控制块。通常情况下点击它会使得对应的窗口置顶，如果要取消置顶可以再次点击一下。

测试环境：win7x64

提示：在win10上可能有一些位置遮挡本身的最小化等按钮，后期会调整。

部分代码摘录：

说明：以下是部分代码摘录，您会看到软件大量使用了SetTop等方法。完整代码可以从底部github获得，里面有直接的编译好的exe可以直接使用。

```
Private Sub addControlBox()
    Dim w As New clsWindow, s$, v, i%
    w.GetWindowByTitleEx ".+?", 0, s, True, , DisplayedWindow
    v = Split(s, " ")
    For i = 0 To UBound(v)
        If v(i) <> Me.Hwnd Then
            If Not isHasAddControlBox(v(i)) Then
                w.Hwnd = v(i)
                If InStr("|SetWindowTop|Program Manager|", "|" & w.Caption & "|")
                    = 0 Then
                        lngHandleHwnd = w.Hwnd
                        Call createControlWindow
                    End If
                End If
            End If
        End If
    Next
End Sub

.
.
.

Private Sub setTagPos()
    isClickHandle = True
    wTag.SetTop Not wTag.IsTopmost

    Set Image1.Picture = IIf(wTag.IsTopmost, imgAfter.Picture, imgBefore.Picture)
End Sub
```

```

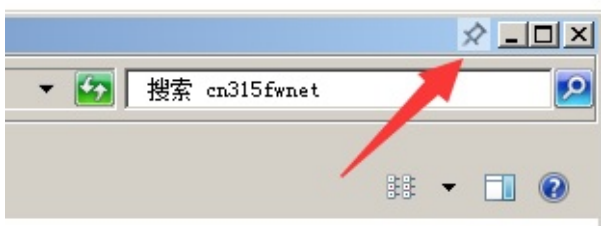
If wTag.IsTopmost Then
    wTag.SetTop
    wTag.Focus
    wMe.SetTop
End If
wTag.Focus
End Sub

Private Sub Timer1_Timer()
    If Not wTag.CheckWindow Then '如果窗口不存在就关闭
        Unload Me
    End If

    If wTag.IsForegroundWindow Then '如果当前是活动窗口，那么需要显示并移动控制窗口
        lngLeft = (wTag.Left + wTag.Width) * 15 - Me.Width - 60 * 15
        lngTop = wTag.Top * 15 + 60
        If Me.Left <> lngLeft Or Me.Top <> lngTop Then '位置需要更新时再移动
            Me.Move lngLeft, lngTop
        End If
        Me.Visible = True
        wMe.SetTop
    ElseIf Not wMe.IsForegroundWindow Then
        Me.Visible = False
    End If

    If wTag.IsTopmost <> isTop Then
        isTop = Not isTop
        Set Image1.Picture = IIf(isTop, imgAfter.Picture, imgBefore.Picture)
    End If
End Sub

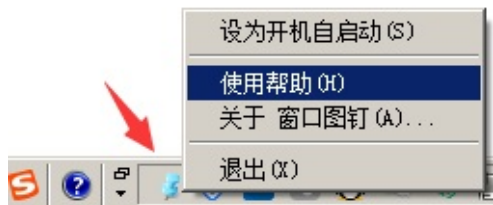
```



点击下变成置顶，并且图标发生变化：



如果要退出可以点击右下角托盘，选择菜单“退出”



## 软件及源码下载

---

软件下载：<https://github.com/sysdzw/SetWindowTop/blob/master/SetWindowTop.exe>

源码(github)：<https://github.com/sysdzw/SetWindowTop>

# 一些Q&A

---

01 目标窗体中存在多个相同类名的输入框该如何区分填写？

# 01 目标窗体中存在多个相同类名的输入框该如何区分填写？

我们经常会遇到一个窗口中有多个输入框，且输入框的类名一样，那么该如何区分填写呢？

要解决这个问题需要先明白一个原则，多个输入框用遍历控件的方法访问，不管访问多少次，它的先后顺序是一致的，除非做软件的故意将每次显示时控件顺序打乱，例如动态生成的控件随意摆放位置，一般这个情况下极少，目前还没遇到过。这样就可以根据次序来进行填写。

那么我们的思路是这样的：1.测试输入框的次序， 2.根据次序填写到目标输入框

测试次序的代码：

```
Dim i%
Dim w As New clsWindow
w.GetWindowByTitleEx "微标电脑收发存统计系统", 1
i = 0
Do
    i = i + 1
    If Not w.SetElementTextByClassName("ThunderRT6TextBox", "次序" & i, i) Then
        Exit Do
    Loop
MsgBox "次序填写完毕！"
```

这样子的话我们就可以区分输入框了，例如窗口里有5个输入框，你要填写的输入框通过上面的测试是“次序4”，想把它填写为xxxxxx，那么你就可以编写代码：

```
w.SetElementTextByClassName "ThunderRT6TextBox", "xxxxxx", 4
```

同样的如果要读取第“次序4”的输入框的内容，用下面代码：

```
MsgBox w.GetElementTextByClassName("ThunderRT6TextBox", 4)
```

# 联系我们

---

如果有问题或者建议以下途径反馈：

---

1. 在github上提交 Issue
2. 看云文档每个页面底部都有评论入口，可以相关的页面底部留言
3. 当然也可以加我扣扣联系，QQ：171977759