

NUMWIN.VBX Custom Control

Description

The NUMWIN.VBX custom control allows you to make use of all the hWnds of all applications running in the windows environment. When issued a IDC_REFRESH action command, NUMWIN searches the desktop for all windows and returns three arrays that can be searched for the appropriate data desired.

hWnds include the main windows, MDI windows, MDI child windows, text boxes, buttons etc. In VB some controls do not have hWnds, ie. VB label controls, and they will not show up in the return arrays.

File Name

NUMWIN.VBX

Remarks

When you create and distribute applications that use the NUMWIN control you should install the file NUMWIN.VBX in the customer's Microsoft Windows \WINDOWS\SYSTEM sub directory. All of the properties, events, and methods for this control are listed below. Properties and events that apply only to this control, or require special consideration when used with it, are underlined. They are documented in this help file. See the Visual Basic *Language Reference* or on-line Help for documentation of the remaining properties, events, and methods.

Properties

Action

CtlName

Index

Left

Top

Tag

Events

NumWinClick

Methods

The Action property is used as a pseudo method. To cause the control to refresh the window data arrays, an IDC_REFRESH action command is issued.

Typical Problems

Typical Problems

The debugging version of NUMWIN.VBX will most likely resolve any errors encountered when developing your application.

1. When a refresh action command is issued, the arrays returned are for the instance in time that the action command is issued. One of the problems that can be encountered is with stale data, that is a hWnd that is no longer valid. This typically happens when a window is closing, or the environment has been altered by opening additional documents. As a precaution, always use the values returned immediately after issuing a IDC_REFRESH action. When it is desired to make use of the values again, re-issue the IDC_REFRESH action and then immediately use the values.

Action Property, NUMWIN Control

Description

Setting the Action property to IDC_REFRESH causes the custom control to search the desktop, load the arrays with all hWnds found and return a click event signaling the arrays are filled. The VB application can then query the arrays and take out the data the application wishes to use.

Since Windows is a dynamic environment, the data returned by NUMWIN.VBX should be used immediately, and before using the data again a new IDC_REFRESH should be issued. This insures that the arrays contain only the most current hWnd parameters. You should refrain from issuing a VB DoEvents() function while making use of the data to prevent Windows from changing the hWnds, ie. closing any open Window, and invalidating the hWnd.

Usage

[form.]CALCKER.Action[= setting %]

Settings

The Action property settings are:

Setting	Description
IDC_REFRESH = 10	Refresh the hWnd arrays.

Remarks

The data returned is valid only at the time the action command is issued. Use the results immediately, and re-issue the IDC_REFRESH action if the hWnd data is needed later.

Data Type

Integer

NumWinClick Event, NUMWIN Control

Description

This event is generated when the user issues the IDC_REFRESH action and the arrays contain valid data. The arrays can be queried immediately upon receipt of this event.

Syntax

Sub *NUMWIN_NumWinClick* (Count As Integer)

Remarks

The "Count" returned is the number of total windows found.

The three data arrays returned are invisible to the VB application and need no declaration from VB.

They are accessed as follows;

1. The hWnds are returned in *HwndArray*.
2. The Parent hWnd are returned in *HwndParArray*.
3. The Window Text, that is what appears in the title bar or the caption property are returned in the *WinTextArray*.

The arrays are accessed from VB as follows:

```
MyHwnd = NumWin1.HwndArray(ndex)
MyParent = NumWin1.HwndParArray(ndex)
MyWindowText = NumWin1.WinTextArray(ndex)
where "ndex" is the array index of the desired hWnd
```

You can use the returned data to find out what the structure of the DeskTop look like. For instance, a Window with a *HwndParArray(ndex) = 0* is a top level window, that is it has no parent. The Window Text obtained for the same index is the title bar text of the main window for that application. Thus to build a picture of all top level windows, search for the value *NumWin1.HwndParArray(ndex) = 0* and use all these windows as top level windows

Usage

The returned parameters are usable by VB or any Windows API call which takes a parameter of one of the returned types. See the demo program for tips on what additional aspects of windows might be accessed.

