

# MENUWATCH Control

## About the Author

## Description

The "MenuWatch" custom control allows you to easily add a full-featured menu help status bar to your application. By placing this custom control directly on your main form, you can display help "snippets" in a help status bar at the bottom of the window as the user highlights menu items.

This control supports MDI windows, The MDI "WindowList" property, and it does not interfere with the normal activation of menu-related context-sensitive help.

## File Name

MENUWATCH.VBX

## Remarks

**Distribution Note** When you create and distribute applications that use the MENUWATCH control you should install the file MENUWATCH.VBX in the customer's Microsoft Windows \SYSTEM subdirectory. The Visual Basic Setup Kit included with the Professional VB product provides tools to help you write setup programs that install your applications correctly.

---

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 online Help for documentation of the remaining properties, events, and methods.

## Properties

<u>Changed</u>	<u>NumPulldowns</u>
<u>NumOptions</u>	Name
Index	Parent
Tag	hWnd
Align	

## Events

<u>Open</u>	<u>Select</u>
<u>Close</u>	

## Methods

Refresh

## Changed Property, MENUWATCH Control

### Description

Returns -1 if the menu has been changed since the application started or the Refresh method was last invoked, 0 otherwise.

This property is read-only.

### Usage

*[MenuHasChanged %= ] [form.]MenuWatch.Changed*

### Data Type

Integer

## **NumPulldowns Property, MENUWATCH Control**

### **Description**

Returns the number of pulldown menus currently associated with the application. For example, an application which provides only "File" and "Edit" menus has 2 pulldown menus.

This property is read-only.

### **Usage**

`[NumPulldowns %= ] [form.]MenuWatch.NumPulldowns`

### **Data Type**

Integer

## NumOptions Property, MENUWATCH Control

### Description

Returns the total number of options under the pulldown menu identified. For example, an application which provides only "File" and "Edit" menus has 2 pulldown menus, and if the "Edit" menu has only "Cut", "Copy", and "Paste" options, **NumOptions** with *PulldownID%* = 2 returns 3.

This property is read-only.

### Usage

[*NumOptions* %= ] [*form.*]MenuWatch.**NumOptions**(*PulldownID%*)

### Data Type

Integer

## Open Event, MENUWATCH Control

### Description

Occurs when the user first begins to interact with the menu, but before a Select event occurs.

### Syntax

**Sub** *MenuWatch\_Open* ( )

### Remarks

This event is the best place to put code which initializes your menu help status bar and Refreshes the MenuWatch control (if necessary).

## Select Event, MENUWATCH Control

### Description

Occurs when the user highlights a menu option.

### Syntax

**Sub** MenuWatch\_**Select** (ItemID **As Integer**, fSeparator **As Integer**)

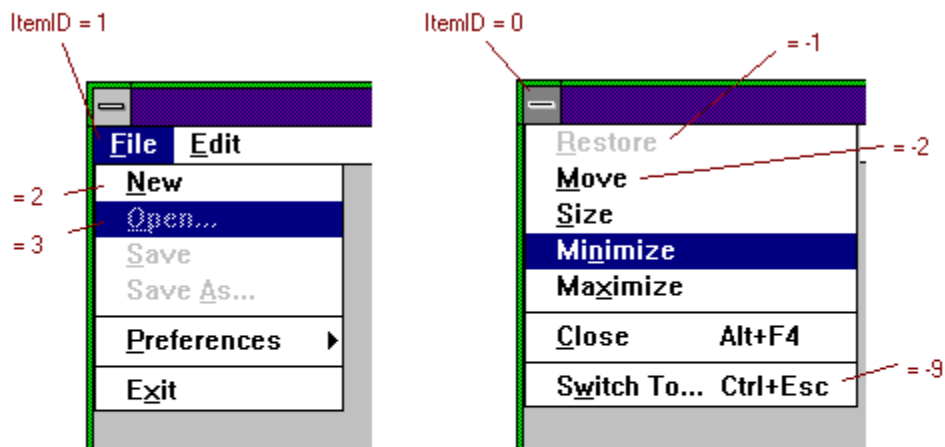
### Remarks

The fSeparator parameter is -1 if the menu option is a separator and zero otherwise.

ItemID is 1 for the first option in your menu, 2 for the second, etc. ItemID is 0 for the Control menu itself, -1 for the first control menu option, -2 for the second control menu option, etc.

Note: if the currently selected menu item is not recognized by MenuWatch (probably because the menu has changed and the Refresh method has not been called), then ItemID is -32768. The value of fSeparator is accurate regardless of the value of ItemID.

The following picture illustrates how the ItemID parameter relates to your menu:



## **Close Event, MENUWATCH Control**

### **Description**

Occurs when the user completes their interaction with the menu.

### **Syntax**

**Sub** *MenuWatch\_Close* ( )

### **Remarks**

This event is the best place to put code which de-initializes your menu help status bar.

## Refresh Method, MENUWATCH Control

### Description

Use this method to update the MenuWatch control when the menu changes. If this method is not invoked, any changes to the menu will not be reflected when a Select event occurs.

### Syntax

*MenuWatch.Refresh*

### Remarks

The Open event is a good place to invoke this method when necessary.



The "MenuWatch" custom control was developed by:

Steve Lathrop  
19101 Willow Spring Drive  
Germantown, MD 20874

CIS ID: 70540,370

Support for a registered copy of "MenuWatch" may be obtained via CompuServe e-mail to the ID above. Suggestions for improvements to "MenuWatch" and for the development of other custom controls is encouraged.

