



Popup Menu Custom Control Contents

Description

The Popup Menu custom control allows the creation of popup menus within a Visual Basic application. The Popup can be activated in response to an event on the parent form or another control, such as the MouseDown event when the right button is clicked. The control can emulate items within the main menu simply by setting its MenuCaption property in which case no extra coding is required, or a custom menu can be created using the AddItem method and MenuReturnID property of the control.

File Name

Popup.Vbx

About Popup Menu Custom Control

Registering

Custom Properties

Activate

MenuAlignment

MenuCaption

MenuReturnID

TrackingButton

Methods

AddItem

Clear

Errors

Error Messages

Popup Menu - Activate

Description

When the activate property is set to a non-zero value at run time, the popup menu is activated and displays its current contents, if any. This property is available only at run time. It is not necessary to reset the activate property before re-activating the menu. The assignment of a non zero value to the property suffices. The Visual Basic code following the property assignment statement resumes only when an entry is selected from the menu or the mouse is clicked outside the area of the menu, thus in either case causing the menu to terminate.

Syntax

PopupMenu.**Activate** [= *activate%*] .

Data Type

Integer

Popup Menu - MenuAlignment

Description

This property takes one of three values : 0 (Right), 1 (Centre) and 2 (Left). The effect of setting this property is to change the position that the popup menu appears in relation to the position of the mousepointer. Setting this property to 0 causes the menu to pop up on the right hand side of the mousepointer. Similarly, setting this property to 2 causes the menu to appear to the left side of the mousepointer. Finally setting this property to 1 causes the popup menu to appear with the mousepointer at its centre.

Syntax

PopupMenu.MenuAlignment [= *setting%*] .

Data Type

Integer (Enumerated)

Popup Menu - MenuCaption

Description

The MenuCaption property is set to a string variable which represents the caption of the form's main menu item or any submenu item which the popup control mimics. The popup menu is only capable of mimicking the menu of its parent form, so it is not possible to use one popup control to mimic the menus of several forms. However, in the case of a MDI application, a popup control placed on a child form can mimic the menu of the MDI container form. The popup menu will exactly duplicate the menu item and its associated submenu, including check marks, shortcut keys, and event procedures executed in response to an item being selected. It is important to remember that in order for the popup menu control to identify the target menu item, the MenuCaption property must be set exactly to the caption of the target menu item, in which case the string must contain the correct cases and also any additional characters. For example, to mimic the menu item with the caption "Edit", it is necessary to use the exact string "&Edit". The strings "Edit" or "&edit" will not suffice.

Syntax

PopupMenu.MenuCaption [= CaptionString\$] .

Data Type

String

Popup Menu - MenuReturnID

Description

The MenuReturnID property returns an ID number which identifies the selected menu item when a custom menu is used rather than an item from the parent form's menu is being mimicked. To see how to create a custom menu and associate ID numbers with menu entries, see the [Additem](#) method.

Syntax

[*ReturnID%* =] *PopupMenu.MenuAlignment* .

Data Type

Integer

Popup Menu - TrackingButton

Description

The use of this property is to select which mouse button is used to track the popup menu. This property can be set to one of two values : 0 (Left) or 1 (Either). The default setting for this property is 'Left', in which case the menu is tracked using the left mouse button. Setting this property to 'Either' allows the menu to be tracked using either the Left or Right mouse buttons.

Syntax

PopupMenu.**TrackingButton** [= *Setting%*] .

Data Type

Integer (Enumerated)

Popup Menu - AddItem Method

Description

The AddItem method allows the Visual Basic programmer to create customised menus using the Popup Menu control. This facility is useful when the main menu does not contain a submenu whose contents matches the popup's desired contents. The use of this method together with the Clear Method and the MenuReturnID property allows for a simple and effective way of creating menus and responding to the selection.

The AddItem Method appends an item to either an already existing popup menu or creates a new empty menu and appends an entry to it. Subsequent calls to the AddItem method will append further entries to the menu. The AddItem method allows one level of submenu within the popup menu, i.e., an entry within the popup menu can have a submenu associated with it but entries within that submenu may not have further submenus associated with them.

The string which is passed to the AddItem method is of a defined format where entries within the string are separated with tab characters (ASCII 9). All entries within the string are optional with the exception of the menu item caption. If the string consists of just one item, this is taken to be the caption.

The syntax of the AddItem statement is as follows:

`Popup1Control.AddItem [tab][IDNUMBER][tab]CAPTION[tab][CHECKED][tab][DISABLED]`

A leading tab character directs the additem method to add the following entry as a submenu of the last item which was added to the menu. If this character is absent, the entry is added to the top level popup. Subsequent entries to the same submenu should also have a leading tab character.

The *IDNUMBER* is the number which is returned through the MenuReturnID property when an item is selected from the popup menu. This field is optional but if it is not present, the value returned in the MenuReturnID property is undefined. Any integer number can be used for an *IDNUMBER* but the value 0 is reserved for internal use by the control and should not be used.

The *CHECKED* field determines if the menu item has a check mark beside it. This field is optional , the default being no check mark. To set the check mark, a non zero number should be placed in the *CHECKED* position.

The *DISABLED* field determines if the menu item is disabled and grayed. As with the *CHECKED* field, this is an optional field, the default being enabled. To disable the entry, a non zero number should be placed in the *DISABLED* position.

The AddItem method can also be used to place a separator bar in the popup menu simply by passing the string argument "-".

NOTE: It is not possible to use the AddItem method to add entries to the popup menu if the popup menu is mimicking part of the main menu. The AddItem method when first applied will simply create a new menu and add an entry to that menu.

Syntax

PopupMenu.AddItem String

Popup Menu - Clear Method

Description

The Clear method removes all entries from the popup menu which were appended using the AddItem method. If the MenuCaption property of the control is set to a valid menu caption, any subsequent activations of the popup control will mimic the main menu item, until such time as the AddItem Method is again invoked and a new custom menu loaded.

Syntax

PopupMenu.Clear

Popup Menu - Error Messages

Separator Error - It is not possible to add a submenu to a separator bar.

This error occurs if an attempt is made to add a submenu to a menu where the last entry is a separator bar. For instance, the following code would cause this error:

```
Popup1.AddItem "-" ' add a separator bar to the menu
Popup1.AddItem Chr$(9) & "1" & Chr$(9) & "&Cut" 'Add first element of submenu
```

Empty Error - It is not possible to add a submenu to an empty menu.

This error occurs if an attempt is made to add a submenu to an empty menu. The following code would cause this error:

```
Popup1.Clear ' Clear out contents of menu
Popup1.AddItem Chr$(9) & "1" & Chr$(9) & "&Cut" 'Add first element of submenu
```

Empty Error - Invalid character in string.

This error occurs if an invalid character occurs in the string passed to the AddItem method. The only valid characters which can occur in the string are the Tab character and characters between ASCII 32 and 126 inclusive.

Tabs Error - Two tabs in succession found.

Self Explanatory!! Tabs must be separated by at least one other valid character.

Registering the Popup Menu Control

This custom control is Shareware. You may freely distribute it to other Visual Basic developers. If, after a evaluationary period of twenty eight days, you decide to continue using it, you should register the control by printing and filling out the [registration form](#), and enclosing a cheque, bank draft or postal order. The source code for the control, in C, is also available.

The prices are as follows:

Registration : IR£20.00, Stg£20.00 or US\$30.00.

Registration and source code IR£35.00, Stg£35.00 or US\$50.00.

Cheques or Bank Drafts in other hard currencies equivalent in value to the above are also acceptable. These should be made payable to and sent to the following:

Robert Mullins,
Ballyknockane,
Clonmel,
Co. Tipperary,
Rep. of Ireland.

In return, you will be sent a registered version of the popup control in which the popup window does not appear on loading the control, and which is personally registered to the user, and if specified, the source code. A copy of printed documentation on the control will also be sent to the user at no extra cost. Any comments or suggestions for improvements would also be welcome.

Disclaimer of Warranty

This custom control is distributed as is. I have done my best to ensure that this control is bug free. However, it is the responsibility of the developer who uses the folder control to ensure that it operates correctly and any applications which use it are thoroughly tested with non-critical data before being relied upon. The author can accept no liability for loss of data or any other problems which may arise through use of this control.

Registration Form

To register, print the following form and send it to the given address:

Popup Menu Custom Control Version 1.00

I enclose a cheque / bank draft for _____ in payment for _____ copies of the Popup Menu custom control / Popup Menu custom control and source code.

Name(s) _____

Company _____

Address _____

Comments _____
