

RTMOUSE.VBX Custom Control

Description

The RTMOUSE Control adds a new technique of providing help to the user that is not available any other way. This control allows the developer to provide the user with a dynamic popup help balloon for all the controls on each form in the VB application. The popup help text provided is especially useful to new user and is ideal for custom in-house applications where the user may not have a manual to refer to.

When enabled, the control provides a custom event that defines when the user moves the mouse cursor over a form's control and clicks the right mouse button. The VB application then passes a help text string to the RTMOUSE.VBX control which then pops up a help balloon in the vicinity of the selected control and displays the help text in the popup balloon. If help text for a control is not desired, then no return text should be provided by the VB application. The popup help balloon is removed when the user releases the right mouse button. Releasing the right mouse button terminates the help session.

A separate RTMOUSE Control is placed on each form that will have a right mouse button popup help session available.

A "popup help session" is defined from the time the RTMOUSE popup help balloon appears on-screen to the point at which the RTMOUSE popup help balloon is removed.

File Name

RTMOUSE.VBX

Remarks

When you create and distribute applications that use the RTMOUSE control you should install the file RTMOUSE.VBX in the customer's Microsoft 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 considerations 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

BackColor

Balloon Arrow

BalloonHorizontalOffset

BalloonLocationMethod

BalloonReferencePoint

Balloon Type

BalloonVerticalOffset

BalloonWidth

BalloonWidthMethod

CtlName

FontBold

FontName

FontSize

ForeColor

Index

Inverse Video

Left

MessageText

RTMKeyModify

Tag

Top

Events

RtmouseClick

Methods

The Action property is used as a pseudo method.

Utility Functions

Included control Utility Funtions are for accessing various control properties.

Tips and Techniques

Tips and Techniques

1. Controls Without hWnd Property:

Some Visual Basic controls do not have hWnd properties. The Visual Basic control types of Image, Shape and Label do not have hWnds. Use Text boxes and Picture controls as controls that perform the same functions and have hWnd properties. Any control that does not have a hWnd property will not be highlighted and will not trigger a RTMouseClick() event. If any controls without hWnd properties need to have help text displayed, include the control in an enclosing frame having a hWnd parameter. Use the hWnd of the container control in the Select Case statement of the RTMouseClick() event.

2. Screen Updates During RTMouse Session:

Once the popup RTMOUSE help message session has begun, the VISUAL BASIC application MUST NOT take any action or cause any action that would cause a screen update. For instance; You may NOT send a dialog box to the screen during a help message session. Doing so will cause the help session to immediately terminate, and may leave windows and the VISUAL BASIC application in an unstable state. To properly terminate a help message session, send a RTMouse.Action = STOP command to the control. This clears the help session and allows normal application processing to resume. Sending the RTMouse.Action = STOP command terminates the RTMOUSE control for this form only.

3. Terminating RTMouse:

Terminating the help message session from within the VISUAL BASIC application can cause the user to become confused. The VISUAL BASIC application should ALWAYS allow a help message session to terminate by the user. A mouse click anywhere on the screen or any keyboard key press ends the in progress RTMOUSE help session returning normal windows control to the user.

4. Setting Colors At Design Time:

It is important to note that during design time you must explicitly set the BackColor and the ForeColor for each RTMOUSE.VBX control used in a project. VISUAL BASIC does not have defaults for palette aware controls.

5. Setting Colors During Runtime:

If the application alters colors displayed by a RTMOUSE.VBX control's popup window during runtime, the application must assume all responsibility for the colors in all forms. The colors do not reset between popup window sessions. If the form containing a RTMOUSE control is unloaded, the color values for this form reset to the values design time values when the form reloads.

6. Disabled Form Controls:

Disabled controls return a hWnd to the RTMouseClick() event. The application can display help for disabled controls or choose not to. If the VISUAL BASIC application does not want to display help for a disabled control, use the Windows API call

```
IsWindowEnabled( hWndAtMouse)
```

to determine whether the control is disabled. Send MessageText to the control based on the results of this API call.

7. MDI Applications:

For MDI applications, use a separate control for the MDI form and for each Child form that is to be monitored. The control cannot be placed directly on a MDI form. Place the MDI form's control on a

toolbar or status bar.

MessageText Property, RTMOUSE Control

Description

Sets the text to be displayed in the popup window.

The initial text to be displayed is set prior to the issuance of the `Rtmouse1.Action` property command. If no default text is supplied, then the control uses the internal message to start a RTMOUSE help session.

Usage

`[form.]RTMOUSE.MessageText[= setting %]`

Data Type

String, maximum of 255 characters.

How to use

The `MessageText` property moves the popup help RTMOUSE window to the location of the control the mouse is over and puts the text in the popup RTMOUSE window.

If the mouse is over a control that help text is NOT desired, then do not set the `MessageText` property. This "do nothing" default does not move the popup window and does not update the text. The next active control the mouse is placed over will continue the move-update sequence. As an example, in the `Case Select` statement the entry for the window to skip can be omitted, left blank (i.e. "") or use an "exit sub" program statement.

Use the included utility functions to retrieve the help text information as required.

Action Property, RTMOUSE Control

Description

Setting the Action property to ID_START (50) enables the RTMOUSE.VBX control to accept right button mouse clicks from the user for the form containing the RTMOUSE control. Setting the Action property to ID_STOP (51) disables the RTMOUSE control and no right button mouse clicks will be processed for the form containing the RTMOUSE control.

Each form that is to provide right button mouse click event information needs to have a separate RTMOUSE control placed on it during application design.

Usage

[*form.*]RTMOUSE.Action[= *setting* %]

Settings

The Action property settings are:

Setting	Description
ID_START = 50	Start a popup RTMOUSE help session.
ID_STOP = 51	Stop a popup RTMOUSE help session.

Remarks

The code fragment to begin monitoring the form for right mouse button click events is:

```
Form.Rtmouse1.Action = ID_START
```

The user terminates a popup RTMOUSE help session by releasing the right mouse button.

The RTMOUSE control is disabled by issuing the following command:

```
Form.Rtmouse1.Action = ID_STOP
```

Typically, enable the RTMOUSE control on when the form is loaded, turn the disable the RTMOUSE control when the form is unloaded.

Data Type

Integer

BackColor Property, RTMOUSE Control

Description

The background color of the popup window is set by the color palette attached to the control's BackColor property.

The BackColor must be explicitly set to a color; the default is not pre-set when the control is loaded. Each RTMOUSE.VBX control in a project may have different BackColor properties set.

Usage

`[form.]RTMOUSE.BackColor[= setting %]`

Settings

The BackColor property is a color selection palette to choose the popup window's background color. Select the desired color with the mouse, or type in an unsigned long color value.

Data Type

Unsigned long color value. For SDK users, ForeColor uses the same format as the WINDOWS.H COLORREF parameter.

Balloon Arrow Property, RTMOUSE Control

The Balloon Arrow Property of the popup balloon help window is set by selecting true/false from the Properties at design time. The Balloon Arrow Property is a screen graphic which points to the control the help text is generated for. This property allows the designer to customize the screen view of the help balloon.

The properties Balloon Arrow, Balloon Type and Inverse Video may be combined to achieve the desired screen effect.

Usage

Not applicable

Settings

True turns pointer on.

False turns pointer off

Data Type

Boolean

BalloonHorizontalOffset Property, RTMOUSE Control

The BalloonHorizontalOffset property is the horizontal offset value from the form reference point in pixels. The form reference point for the origin of the offset value (i.e., the 0 value) determined by the setting selected in the BalloonCustomOffset property.

If the BalloonLocationMethod property is set to "Default" this property has no effect.

Usage

[*form.*]RTMOUSE1.**BalloonHorizontalOffset** [= *setting* %]

Settings

User supplied value in screen pixels.

Data Type

INTEGER

See Also: [BalloonReferencePoint](#), [BalloonLocationMethod](#), [BalloonVerticalOffset](#)

BalloonReferencePoint Property, RTMOUSE Control

The BalloonReferencePoint property determines the balloon reference location for positioning the balloon.

Selection is made by choosing the appropriate value from the enumeration list.

The choices are:

Top/Left uses the balloon top-left for the reference point. Selecting Top/Left also positions the balloon BELOW the point determined by BalloonHorizontalOffset and the BalloonVerticalOffset property values. The arrow, if selected, will be located at the balloon top left.

Bottom/Left uses the balloon bottom-left as the reference point. Selecting Bottom/Left also positions the balloon ABOVE the point determined by BalloonHorizontalOffset and BalloonVerticalOffset property values. The arrow, if selected, will be located at the balloon bottom left.

Top/Right uses the balloon top-right as the reference point. Selecting Top/Right also positions the balloon BELOW the point determined by BalloonHorizontalOffset and BalloonVerticalOffset property values. The arrow, if selected, will be located at the balloon top right.

Bottom/Right uses the balloon bottom-right as the reference point. Selecting Bottom/Right also positions the balloon ABOVE the point determined by BalloonHorizontalOffset and BalloonVerticalOffset property values. The arrow, if selected, will be located at the balloon bottom right.

The BalloonReferencePoint location is the apex of the square balloon window at the Top/Left or Bottom/Left corner.

To modify the balloon reference point for different balloon features:

For a balloon with a shadow, the BalloonReferencePoint includes the shadow height of 10 pixels.

For a rounded balloon, the BalloonReferencePoint is the same location as for a square balloon.

For a balloon with the BalloonArrow feature the BalloonReferencePoint is 10 pixels horizontal from the arrow base and 18 pixels vertical from the arrow point.

To calculate the arrow point's horizontal location: If the BalloonReferencePoint property is set to Top/Left or Top/Right add 10 pixels to the horizontal position. If the BalloonReferencePoint property is set to Top/Right or Bottom/Right subtract 10 pixels from the horizontal position.

To calculate the arrow point's vertical location: If the BalloonReferencePoint property is set to Top/Left or Top/Right subtract 18 pixels from the vertical position. If the BalloonReferencePoint property is set to Bottom/Left or Bottom/Right add 18 pixels to the vertical position.

The above reference position modification values are automatically taken into account by the control when setting the final on screen balloon position. The Visual Basic designer should take these values into account when determining the property values for BalloonHorizontalOffset and BalloonVerticalOffset.

If the BalloonLocationMethod property is set to "Default" this property has no effect.

Usage

[form.]RTMOUSE1.BalloonReferencePoint [= setting %]

Settings

- 0 - Top/Left ...the balloon Top/Left is used for positioning reference.
- 1 - Bottom/Left...the balloon Bottom/Left is used for positioning reference.
- 2 - Top/Right ...the balloon Top/Right is used for positioning reference.
- 3 - Bottom/Right ...the balloon Bottom/Right is used for positioning reference.

Data Type

INTEGER

See Also: [BalloonLocationMethod](#), [BalloonHorizontalOffset](#), [BalloonVerticalOffset](#)

BalloonLocationMethod Property, RTMOUSE Control

The BalloonLocationMethod property determines how the balloon location is determined. Select the desired property value by choosing from the enumeration list.

Selecting a BalloonLocationMethod other than "Default Location" determines the reference location used to apply the BalloonHorizontalOffset and BalloonVerticalOffset offset value to. The offset values are applied in a linear fashion, both positive and negative. The reference location is defined as the (0,0) point.

If the BalloonLocationMethod property is set to "Default Location" these properties have no effect:

BalloonHorizontalOffset

BalloonVerticalOffset

BalloonReferencePoint

Usage

[form.]RTMOUSE1.BalloonLocationMethod [= setting %]

Settings

0 - Default Location	Use the control's default internal algorithm.
1 - Reference To Control	Offset is referenced to the target controls top, left. Control top left is assumed to be 0,0
2 - Reference To Cursor	Offset is referenced to the cursor's position when the balloon is put on screen.
3 - Reference To Window	Offset is referenced to the parent form's top-left. Form window top-left is assumed to be 0,0.

Data Type

INTEGER

See Also: [BalloonHorizontalOffset](#), [BalloonVerticalOffset](#), [BalloonReferencePoint](#)

BalloonVerticalOffset Property, RTMOUSE Control

The BalloonVerticalOffset property is the vertical offset value from the form reference point in pixels. The form reference point for the origin of the offset value (i.e., the 0 value) determined by the setting selected in the BalloonCustomOffset property.

If the BalloonLocationMethod property is set to "Default" this property has no effect.

Usage

[*form.*]RTMOUSE1.**BalloonVerticalOffset** [= *setting* %]

Settings

User supplied value in screen pixels.

Data Type

INTEGER

See Also: [BalloonReferencePoint](#), [BalloonLocationMethod](#), [BalloonHorizontalOffset](#)

BalloonWidth Property, RTMOUSE Control

The BalloonWidth property sets the width of the balloon in screen pixels. The width value entered does not include the margin around the text and the shadow. Add approximately 10 pixels for the margin, and 10 pixels for the shadow to arrive at the final balloon width.

The BalloonWidth minimum value is 40 pixels. The maximum value is 32000 pixels.

If the BalloonWidthMethod property is set to "Default" this property has no effect.

Usage

[form.]RTMOUSE1.**BalloonWidth** [= setting %]

Settings

User supplied value in screen pixels.

Data Type

INTEGER

See Also: [BalloonWidthMethod](#)

BalloonWidthMethod Property, RTMOUSE Control

The BalloonWidthMethod property selects the method to use to determine the balloon width. Selection is made by choosing the appropriate value from the enumeration list.

The choices are:

Use the control's default width calculation algorithm.

Use the balloon width set in the BalloonWidth property.

If the BalloonWidthMethod property is set to "Default", this property has no effect.

Usage

[*form.*]RTMOUSE1.**BalloonWidthMethod** [= *setting* %]

Settings

0 - Default Use the control's default algorithm.

1 - Custom Use the pixel value in BalloonWidth to set Balloon Width

Data Type

INTEGER

See Also: [BalloonWidth](#)

BalloonType Property, RTMOUSE Control

The BalloonType Property of the balloon help window is set by choosing the desired type from the enumeration list in the properties window at design time.

The properties BalloonArrow, BalloonType and InverseVideo may be combined to achieve the desired screen effect.

When using the "Type 2: Square balloon 3D with NO drop shadow" style balloon, set the bgcolor to light gray. The text color can be set to any value.

Usage

[form.]WSHELP1.**BalloonType** [= setting %]

Settings

0 - Square, With Shadow	square balloon with drop shadow.
1 - Square, No Shadow	square balloon with NO drop shadow.
2 - Square, 3D, No Shadow	balloon 3D with NO drop shadow.
3 - Rounded	rounded balloon with NO drop shadow.

Data Type

INTEGER

InverseVideo Property, RTMOUSE Control

The Inverse Video Property of the popup balloon help window is set by selecting true/false from the properties window at design time. The Inverse Video Property is a screen graphic control that determines whether the selected control is displayed in Inverse Video when the selection is made.

The properties Balloon Arrow, Balloon Type and Inverse Video may be combined to achieve the desired screen effect.

Usage

Not applicable

Settings

True turns Inverse Video on.

False turns Inverse Video off

Data Type

Boolean

ForeColor Property, RTMOUSE Control

Description

The text color of the popup window is set by the color palette attached to the control's ForeColor property.

The ForeColor must be explicitly set to a color; the default is not pre-set when the control is loaded. Each RTMOUSE.VBX control in a project may have different ForeColor properties set.

Usage

[form.]RTMOUSE.ForeColor[= setting %]

Settings

The ForeColor property is a color selection palette to choose the popup window's text color. Select the desired color with the mouse, or type in an unsigned long color (i.e., COLORREF) value.

Data Type

Unsigned long color value. For SDK users, ForeColor uses the same format as the WINDOWS.H COLORREF parameter.

FontBold Property, RTMOUSE Control

Description

The FontBold property is a standard Visual Basic property. It sets the font bold attribute of the text which appears in the popup window.

Usage

`[form.]RTMOUSE.FontColor[= setting %]`

Settings

The FontBold property is a standard Visual Basic property. See the manuals for the appropriate values

Data Type

Boolean (integer).

FontName Property, RTMOUSE Control

Description

The FontName property is a standard Visual Basic property. It sets the font name of the text which appears in the popup window.

Usage

`[form.]RTMOUSE.FontName [= font]`

Settings

The FontName property is a standard Visual Basic property. See the manuals for the appropriate values

Data Type

String

FontSize Property, RTMOUSE Control

Description

The FontSize property is a standard Visual Basic property. It sets the size of the text which appears in the popup window.

Usage

[form.]RTMOUSE.FontSize [= points]

Settings

The FontSize property is a standard Visual Basic property. See the manuals for the appropriate values

Data Type

Integer.

RTMKeyModify Property, RTMOUSE Control

Description

The RTMKeyModify property modifies the right mouse action. Since there are other uses for the right mouse button, these modifier keys allow the selection of unique right mouse activation properties.

Usage

[*form.*]WSHELP1.**BalloonArrow** [= *setting* %]

Settings

- 0 - No Modifer the right mouse key down action will initiate a balloon sequence.
- 1 - CTRL Key the right mouse click notifies the control only if the CTRL key is depressed when the right mouse button is pressed.
- 2 - SHIFT Key the right mouse notifies the control only if the SHIFT key is depressed when the right mouse button is pressed.

Data Type

INTEGER

RtmouseClick Event, RTMOUSE Control

Description

This event is generated only after a help session is initiated. A popup help window will appear on-screen. When the user moves the mouse over any control on a form, the control will be displayed in inverse video and the popup window moves to the control windows location, sends the RtmouseClick Event to the application and awaits the help text to be applied to the popup help window.

Syntax

Sub *RTMOUSE_RtmouseClick* (*hwndAtMouse* **As Integer**)

Remarks

Code fragments to illustrate the use of RTMOUSE Control.

Start up the control;

```
Form.Rtmouse.MessageText = "Initial message text to display"  
Form.Rtmouse.Action = ID_START
```

The following SUB processes the returned hwndAtMouse and returns the desired help text to the RTMOUSE control;

```
SUB RTMOUSE1_RtmouseClick( hwndAtMouse As Integer )  
    Select Case hwndAtMouse  
        Case Control.hwnd-X1  
            Form.Rtmouse.MessageText = "message text desired"  
        Case Control.hwnd-X2  
            Form.Rtmouse.MessageText = "message text desired"  
        .  
        .  
        .  
        Case Control.hwnd-Xn  
            ' No text desired  
            Exit Sub  
        Case Else  
            Exit Sub  
    End Sub
```

Utility Functions

The Rtmouse control includes the following utility functions;

Hwnd2ControlTabIndex:

Retrieves a control's TabIndex property.

Returns a zero length string if property not found.

Hwnd2CtlName:

Retrieves a control's name.

Returns a zero length string if property not found.

Hwnd2HelpContextID:

Retrieves a control's HelpContextID.

Returns -1 if property not found.

Hwnd2TagText:

Retrieves a control's Tag property.

Returns -1 if property not found.

Each utility function takes a single parameter, the hWnd of the control, for which the property is desired. The declares for using these functions are as follows.

Hwnd2ControlTabIndex

```
Declare Function Hwnd2ControlTabIndex Lib "RTMOUSE.VBX" (ByVal hwndControl%) As Integer
```

Hwnd2CtlName

```
Declare Function Hwnd2CtlName Lib "RTMOUSE.VBX" (ByVal hwndControl%) As String
```

Hwnd2HelpContextID

```
Declare Function Hwnd2HelpContextID Lib "RTMOUSE.VBX" (ByVal hwndControl%) As Long
```

Hwnd2TagText

```
Declare Function Hwnd2TagText Lib "RTMOUSE.VBX" (ByVal hwndControl%) As String
```

Place all of the declare statement on a single line.

