

## 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 which 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 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

BackColor

Balloon Type

Balloon Arrow

CtlName

FontBold

FontName

FontSize

ForeColor

Index

Inverse Video

Left

MessageText

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.

## Typical Problems

1. Some Visual Basic controls do not have hwnd properties.

The Visual Basic control types of Image, Shape and Label do not have hwnds and are not intended to be highlighted and have help text displayed. 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 which has a hwnd parameter. Use the hwnd of the container control in the Select Case statement of the RtmouseClick() event.

2. WARNING ... Once the popup RTMOUSE help message session has begun, the VB application MUST NOT take any action or cause any action which 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 VB 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. When the Rtmouse.Action = STOP command is given, the right mouse button will no longer initiate a help session.

3. Terminating the help message session from within the VB application can cause the user to become confused. The VB 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. 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. VB does not have defaults for palette aware controls.

5. IMPORTANT ... If the colors displayed by a RTMOUSE.VBX control's popup window are altered during runtime, the program must assume all responsibility for the colors. The colors do not reset between popup window sessions. If the window containing the RTMOUSE.VBX control is unloaded, then the colors will reset to the value set at design time.

## 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 which 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 a 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

## **Balloon Type Property, RTMOUSE Control**

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

These types of popup balloons available:

- Type 0: Square balloon with drop shadow
- Type 1: Square balloon with NO drop shadow
- Type 2: Square balloon 3D with NO drop shadow
- Type 3: Rounded balloon with NO drop shadow.

The properties Balloon Arrow, Balloon Type and Inverse Video 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 backcolor to light gray. The text color can be set to any value.

### **Usage**

Not applicable.

### **Settings**

Choose the appropriate type from the list.

### **Data Type**

Not applicable.



## **Inverse Video 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 which 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 a 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.**ForeColor**[ = *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.ForeColor[ = 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.

## 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 with 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.

