STMOUSE.VBX Custom Control

Description

The STMOUSE 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 status line help system for all the controls on each form in the VB application. The status line help text information 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. The VB application then supplies a status line help text to a text display on the form.

A separate STMOUSE Control must be placed on each form that will have a status line help message available.

File Name

STMOUSE.VBX

Remarks

When you create and distribute applications that use the STMOUSE control you should install the file STMOUSE.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

CtlName

Index

Left

Tag

Top

Events

STMouseClick

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 STMouseClick() 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 STMouseClick() event.

Action Property, STMOUSE Control

Description

Setting the Action property cause the STMOUSE.VBX control to return the STMouseClick event when the mouse is moved over a forms control.

During the STMouseClick event the VB application can take what ever action is appropriate for the control under the mouse. The VB application displays and removes all text placed in the on form status line.

Typically the status line displayed will be a textbox, a label, or in the case of a MDI application a MDI form status label. See the various VB sample applications for information on how to implement status line help text. To hide the status text when the mouse moves off of a control and over the form, use the form's Form1_MouseMove() event and set the status text to a null string.

Usage

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

Settings

The Action property settings are:

Setting		Description
ID_START	= 60	Start a STMOUSE help session.
ID_STOP	= 61	Stop a STMOUSE help session.

Remarks

The code fragment to begin monitoring the form for mouse move events is

```
Form.STMOUSE1.Action = ID_START
```

To disable further monitoring action use the command:

```
Form.STMOUSE1.Action = ID_STOP
```

Data Type

Integer

STMouseClick Event, STMOUSE 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 STMouseClick Event to the application and awaits the help text to be applied to the popup help window.

Syntax

Sub *STMOUSE*_**STMouseClick** (hwndAtMouse **As Integer**)

Remarks

Code fragments to illustrate the use of STMOUSE Control.

```
Start up the control;
     Form.STMOUSE.Action = ID_START

The following SUB processes the returned hwndAtMouse and sets the desired
help text in the VB form's StatusBox text;

SUB STMOUSE1_STMouseClick( hwndAtMouse As Integer )
     Select Case hwndAtMouse
           Case Control.hwnd-X1
                 Form.StatusBox.Text = "status message text desired"
           Case Control.hwnd-X2
                 Form.StatusBox.Text = "status message text desired"
           .
           .
           .
           Case Control.hwnd-Xn
                 ' No text desired
                 Exit Sub
           Case Else
                 Exit Sub
End Sub
```

Utility Functions

The STMOUSE 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.

Only works with VB3.

Hwnd2HelpContextID:

Retrieves a control's HelpContextID.

Returns -1 if property not found.

Only works with VB2 and above.

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 "STMOUSE.VBX" (ByVal hwndControl%)
As Integer
```

Hwnd2CtlName

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

Hwnd2HelpContextID

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

Hwnd2TagText

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

Place all of the declare statement on a single line.