

Contents

[Overview](#)

[Properties Reference](#)

[Events Reference](#)

[If You Get Stuck](#)

[System Requirements](#)

Overview

MouseKeeper Mouse Hook Custom Control for Visual Basic is a VBX that triggers MouseDown, MouseMove, and MouseUp events for all controls in your application or in other applications. With MouseKeeper, you can maintain a centralized location to execute code based on mouse events such as displaying a status line or providing context sensitive help. MouseKeeper also gives you the option to return a property (i.e., Tag, Caption, Name, etc.) of the control that the mouse is over!

[Properties, Events, & Methods](#)

Using MouseKeeper

All Visual Basic custom controls are loaded into your project from within the Visual Basic environment. Once a custom control is added to your project and your project is saved, it will always load whenever you load that project.

To add the MouseKeeper custom control to your project, choose '**Add File**' from the '**File**' menu in Visual Basic. Then, enter the name of the MouseKeeper custom control (MKEEPER.VBX), and its fully qualified path if necessary. The MouseKeeper icon should now appear in your Visual Basic Toolbox.

To use MouseKeeper, you must first place it in a form in your project. When you add the MouseKeeper custom control to your form, it will appear on your form as an icon similar to the one that appears in the Toolbox. Although you can reposition the icon anywhere you like on the form, the position of the icon is not important. When you run your application, the custom control becomes *invisible* and therefore cannot be seen by the user.

Refer to the '*Loading Custom Controls*' section in the *Visual Basic Programmer's Guide* for a detailed explanation of adding controls to your project.

If You Get Stuck

Customer support is available for MouseKeeper by Compuserve, account # **71222,3352**. Please leave a message detailing your question or problem and a representative will respond promptly. If you wish, you may leave a return phone number if you prefer to be contacted by phone.

We at JOSWare, Inc. wish to serve our user community as best as we can. We believe that any product, no matter how well constructed and tested, may work differently in different situations and environments. Also, each user of a product has a much different perception of how software should look and behave. It is our intention to listen to all of our users' requests and suggestions very seriously, because after all, it is you who must be happy with the product. We hope you enjoy using this product and we are looking forward to provide you with even more products in the very near future.

MouseKeeper Custom Control Reference

Description The MouseKeeper Custom Control allows you to subclass a form or control to receive and/or intercept its messages.

File Name MKEEPER.VBX

Object Type MouseKeeper

Properties, Events, & Methods

All of the properties, events, and methods for MouseKeeper are listed in the tables below. All standard Visual Basic properties, events, and methods are denoted with an asterisk(*) and can be found documented in the *Visual Basic Language Reference* that comes with Visual Basic.

Properties

(About)	*Index	ReturnProperty
*Height	*Left	*Tag
HookType	*Name	*Top
*hWnd	*Parent	*Width

Events

MouseDown	MouseMove	MouseUp
---------------------------	---------------------------	-------------------------

Methods

MouseKeeper does not support any methods.

Properties Reference

The following is a detailed reference of all the properties supported by MouseKeeper.

[HookType Property](#)

[ReturnProperty Property](#)

(About) Property

Description	Displays version information about the MouseKeeper Custom Control.
Usage	Double click on the ellipses ('...') button next to the property text to activate the about dialog box.
Remarks	Available only at design time.
Data Type	N/A

HookType Property

Description	Determines if the control should report mouse activity for all tasks of the current task.						
Usage	<code>[form.]MouseKeeper.HookType[= setting]</code>						
Remarks	<table><thead><tr><th>Setting</th><th>Description</th></tr></thead><tbody><tr><td>0</td><td>Within Same Task - The control will only fire mouse events for mouse activity that occurs within the application or task.</td></tr><tr><td>1</td><td>Across All Tasks - The control will fire mouse events for all mouse activity for all tasks.</td></tr></tbody></table>	Setting	Description	0	Within Same Task - The control will only fire mouse events for mouse activity that occurs within the application or task.	1	Across All Tasks - The control will fire mouse events for all mouse activity for all tasks.
Setting	Description						
0	Within Same Task - The control will only fire mouse events for mouse activity that occurs within the application or task.						
1	Across All Tasks - The control will fire mouse events for all mouse activity for all tasks.						
Data Type	Integer (Enumerated)						

ReturnProperty Property

Description	Determines the property value of the control that the mouse is over that will be passed to theMouseDown, MouseMove and MouseUp events in the PropValue parameter.												
Usage	<code>[form.]MouseKeeper.ReturnProperty[= setting]</code>												
Remarks	<table><thead><tr><th>Setting</th><th>Description</th></tr></thead><tbody><tr><td>0</td><td>None - No property will be passed to the events.</td></tr><tr><td>1</td><td>Caption - The Caption property will be passed to the events.</td></tr><tr><td>2</td><td>Name - The name of the control will be passed to the events.</td></tr><tr><td>3</td><td>Tag - The Tag property will be passed to the events.</td></tr><tr><td>4</td><td>HelpContextID - The HelpContextID property will be passed to the events.</td></tr></tbody></table>	Setting	Description	0	None - No property will be passed to the events.	1	Caption - The Caption property will be passed to the events.	2	Name - The name of the control will be passed to the events.	3	Tag - The Tag property will be passed to the events.	4	HelpContextID - The HelpContextID property will be passed to the events.
Setting	Description												
0	None - No property will be passed to the events.												
1	Caption - The Caption property will be passed to the events.												
2	Name - The name of the control will be passed to the events.												
3	Tag - The Tag property will be passed to the events.												
4	HelpContextID - The HelpContextID property will be passed to the events.												

Data Type Integer (Enumerated)

Events Reference

The following is a detailed reference of all the events supported by MouseKeeper.

[MouseDown, MouseMove, MouseUp Events](#)

MouseDown, MouseMove, MouseUp Events

Description These events are triggered to notify you when and what type of mouse activity occurred.

Syntax **Sub MouseKeeper_MouseDown** (*hWindow As Integer, PropValue As String, Button As Integer, Shift As Integer, X As Integer, Y As Integer*)

Sub MouseKeeper_MouseMove (*hWindow As Integer, PropValue As String, Button As Integer, Shift As Integer, X As Integer, Y As Integer*)

Sub MouseKeeper_MouseUp (*hWindow As Integer, PropValue As String, Button As Integer, Shift As Integer, X As Integer, Y As Integer*)

Remarks If the mouse is pressed, the **MouseDown** event is fired. When the mouse moves, the **MouseMove** event is fired. When the mouse button is released, the **MouseUp** event is fired. In each of these cases, parameters are passed to the event so that you can identify which window the mouse is over when the event occurs.

Parameter	Description
------------------	--------------------

hWindow	Identifies the window handle that is associated with the window that the mouse is over.
---------	---

PropValue	This string contains the value of the property of the window if it is a Visual Basic custom control. The property passed depends on the setting of the ReturnProperty property.
-----------	--

Note: This value is not sent when the mouse activity occurs over a window within a different task.

Button	The state of the mouse buttons, in which a bit is set if the button is down. The Button argument is a bit field, with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2) values 1, 2, and 4, respectively. It indicates the complete state of the mouse buttons; some, all, or none of these three bits can be set, indicating that some, all, or none of the buttons is pressed.
--------	---

Shift	The state of the Shift, Ctrl and Alt keys. Bit is set if the key is down. The Shift argument is a bit field, with the least-significant bits corresponding to the Shift key (bit 0), the Ctrl key (bit 1), and the Alt key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Shift indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys is pressed. For example, if both Ctrl and Alt were pressed, the value of Shift would be 6.
-------	--

X, Y	The current location of the mouse pointer.
------	--

Appendix - Technical Specifications

All of the following is subject to change. See the README.TXT on the MouseKeeper installation diskette for any changes.

System Requirements

- Microsoft Windows 3.x Operating Environment
- Microsoft Visual Basic Version 2.0 or higher.

