Help for JoyStk

Registration Information

Order Form

Getting Custom Controls Written

Description

JoyStk is a Visual Basic custom control that sends joystick information to your VB program.

File Name

JOYSTK1.VBX

Object Type

Joystick

Remarks

JoyStk requires that a Windows joystick driver installed.   If you do not have a driver, one has been provided in the SAMPLE.ZIP file (look for IBMJOY.ZIP).   This driver can be redistributed with your programs.

When using JoyStk, be sure to set the Enabled property to True at run-time.   If this is not done, the control won't function.   Merely making sure that the property is True at design time is not enough. Also, you should ensure that Enabled is set to false prior to exiting.

Compatibility

VB 2.0 and above

Distribution Note   When you develop and distribute an application that uses JoyStk, you should install the file JOYSTK1.VBX into the users Windows SYSTEM directory.   JoyStk has version information built into it.   So, during installation, you should ensure that you are not overwriting a newer version of JoyStk.

Properties

All of the properties that apply to this control are in this table.
Properties that have special meaning for this control or that only apply
to this control are marked with an asterisk (*).

Align	Left	Top
*Button1	*Manufacturer	*XMax
*Button2	Name	*XMin
*Button3	*Period	*XPos
*Button4	*PeriodMax	*YMax
*Buttons	*PeriodMin	*YMin
*Devices	*Port	*YPos
Enabled	*Product	*ZMax
HelpContextID	*ProductName	*ZMin
hWnd	Tag	*ZPos
Index	*Threshold	

Button1, Button2, Button3, and Button4 Properties

Description

Returns the current state of the joystick's buttons.

Usage

[*form.*][*control.*]**Button1**
[*form.*][*control.*]**Button2**
[*form.*][*control.*]**Button3**
[*form.*][*control.*]**Button4**

Remarks

These properties tell you which button(s) were pressed at the last time it was polled (see Period).
They are read-only at run-time only.   They are only valid after the joystick has been enabled.

Data Type

Integer (boolean)

See Also

Events:

ButtonDown

ButtonUp

Properties:

Buttons

Period

Buttons Property

Description

Returns the number of buttons on the joystick.

Usage

[*form.*][*control.*]**Buttons**

Remarks

This property returns the number of buttons on the joystick.

This property is read-only at run-time only. It is only valid after the joystick has been enabled.

Data Type

Integer

See Also

Events:

ButtonDown

ButtonUp

Properties:

Button1

Devices

Manufacturer

PeriodMax

PeriodMin

Port

Product

ProductName

Devices Property

Description

Returns the number of joysticks supported by the current driver.

Usage

[*form.*][*control.*]**Devices**

Remarks

This property tells you how many joysticks are supported in the system.

This property is read-only at run-time only. It is only valid after the joystick has been enabled.

Data Type

Integer

See Also

Properties:

Buttons

Manufacturer

PeriodMax

PeriodMin

Port

Product

ProductName

Manufacturer Property

Description

Returns the manufacturer's ID of the joystick.

Usage

[*form.*][*control.*]**Manufacturer**

Remarks

This represents the manufacturer of the driver.   The values returned are:

Value	Description
1	Microsoft

This property is read-only at run-time only.   It is only valid after the joystick has been enabled.

Data Type

Integer

See Also

Properties:
    Buttons
    Devices
    PeriodMax
    PeriodMin
    Port
    Product
    ProductName

Information Properties Example

In this example, the program will show the informational properties.   To try this example, paste the code into the Declarations section of a form that contains seven labels, and a JoyStk control.   Make sure that the <u>Port</u> property of the JoyStk control is set properly.   Press F5. You'll see the settings of all of the informational properties.

```
Sub Form_Load ()
    Joystick1.Enabled = True

    Label1.Caption = "Buttons: " & Joystick1.Buttons
    Label2.Caption = "Devices: " & Joystick1.Devices
    Label3.Caption = "Manufacturer: " & Joystick1.Manufacturer
    Label4.Caption = "PeriodMin: " & Joystick1.PeriodMin
    Label5.Caption = "PeriodMax: " & Joystick1.PeriodMax
    Label6.Caption = "Product: " & Joystick1.Product
    Label7.Caption = "ProductName: " & Joystick1.ProductName

    Joystick1.Enabled = False
End Sub
```

Period Property

Description

Sets the polling interval for the joystick.

Usage

[*form.*][*control.*]**Period**[= *integer*]

Remarks

This property determines the polling rate (in milliseconds) of the joystick.

When changing this property at run-time, the control should be disabled prior to changing it, then re-enabled afterwards.

Data Type

Integer

See Also

Properties:
[PeriodMax](#)
[PeriodMin](#)

Period Property Example

In this example, the program will show the effects of different settings for the Period property.   To try this example, paste the code into the Declarations section of a form that contains a horizontal scroll bar, two labels, and a JoyStk control.   Press F5.   Play with the horizontal scroll bar, then try the joystick, then the scroll bar again, etc.

```
Sub Form_Load ()
    Joystick1.Enabled = True

    HScroll1.Min = Joystick1.PeriodMin
    HScroll1.Max = Joystick1.PeriodMax
    HScroll1.Value = (Joystick1.PeriodMax + Joystick1.PeriodMin) / 2

    Call HScroll1_Change
End Sub

Sub Form_Unload (Cancel As Integer)
    Joystick1.Enabled = False
End Sub

Sub HScroll1_Change ()
    Label1.Caption = "Period: " & HScroll1.Value & " milliseconds"

    ' need to disable the joystick while doing this
    Joystick1.Enabled = False
    Joystick1.Period = HScroll1.Value
    Joystick1.Enabled = True
End Sub

Sub HScroll1_Scroll ()
    Call HScroll1_Change
End Sub

Sub Joystick1_Move (X As Integer, Y As Integer, Z As Integer)
    Label2.Caption = "Position: " & X & "," & Y
End Sub
```

PeriodMin and PeriodMax Properties

Description

Returns the minimum and maximum times between polling that this joystick supports.

Usage

[*form.*][*control.*]**PeriodMax**
[*form.*][*control.*]**PeriodMin**

Remarks

These properties specify the minimum and maximum times between polling that the joystick driver supports (see Period).

This property is read-only at run-time only.   It is only valid after the joystick has been enabled.

Data Type

Integer

See Also

Properties:

Port Property

Description

Determines which joystick port to use.

Usage

[*form.*][*control.*]**Port**[= *integer*]

Remarks

This property determines which joystick to use.   The allowable values are:

Value	Description
1	Joystick 1, 3-D joystick, or 4-button joystick
2	Joystick 2

This property must be set at design-time.

Data Type

Integer

Product Property

Description

Returns the product ID fo the joystick driver.

Usage

[*form.*][*control.*]**Product**

Remarks

This represents the product ID of the driver.   The values returned are:

Value	Description
1	Microsoft MIDI Mapper
2	Microsoft Wave Mapper
3	Sound Blaster MIDI output port
4	Sound Blaster MIDI input port
5	Sound Blaster internal synthesizer
6	Sound Blaster waveform output port
7	Sound Blaster waveform input port
9	AdLib-compatible synthesizer
10	MPU401 MIDI output port
11	MPU401 MIDI input port
12	IBM Game Control Adapter

This property is read-only at run-time only.   It is only valid after the joystick has been enabled.

Data Type

Integer

See Also

Properties:

[Buttons](#)

[Devices](#)

[Manufacturer](#)

[PeriodMax](#)

[PeriodMin](#)

[Port](#)

[ProductName](#)

ProductName Property

Description

Returns the product name of the joystick driver.

Usage

[*form.*][*control.*]**ProductName**

Remarks

This property specifies the product name of the joystick driver.

This property is read-only at run-time only.   It is only valid after the joystick has been enabled.

Data Type

String

See Also

Properties:
  [Buttons](#)
  [Devices](#)
  [Manufacturer](#)
  [PeriodMax](#)
  [PeriodMin](#)
  [Port](#)
  [Product](#)

Threshold Property

Description

Determines the amount the joystick must move for an event to be fired.

Usage

[*form.*][*control.*]**Threshold**[= *integer*]

Remarks

This property determines the amount the joystick must move for a Move eventMove event to be fired.

When changing this property at run-time, the control should be disabled prior to changing it, then re-enabled afterwards.

Data Type

Integer

See Also

Events:
    Move
Properties:
    Port

Threshold Property Example

In this example, the program will show the effects of different settings for the Threshold property.   To try this example, paste the code into the Declarations section of a form that contains a horizontal scroll bar, two labels, and a JoyStk control.   Press F5.   Play with the horizontal scroll bar, then try the joystick, then the scroll bar again, etc.

```
Sub Form_Load ()
    Joystick1.Enabled = True

    HScroll1.Min = 1
    HScroll1.Max = 32000
    HScroll1.Value = 100

    Call HScroll1_Change
End Sub

Sub Form_Unload (Cancel As Integer)
    Joystick1.Enabled = False
End Sub

Sub HScroll1_Change ()
    Label1.Caption = "Threshold: " & HScroll1.Value & " units"

    ' need to disable the joystick while doing this
    Joystick1.Enabled = False
    Joystick1.Threshold = HScroll1.Value
    Joystick1.Enabled = True
End Sub

Sub HScroll1_Scroll ()
    Call HScroll1_Change
End Sub

Sub Joystick1_Move (X As Integer, Y As Integer, Z As Integer)
    Label2.Caption = "Position: " & X & "," & Y
End Sub
```

XMin, XMax, YMin, YMax, ZMin, and ZMax Properties

Description

Returns the full range of the joystick's position.

Usage

[*form.*][*control.*]**XMin**
[*form.*][*control.*]**XMax**
[*form.*][*control.*]**YMin**
[*form.*][*control.*]**YMax**
[*form.*][*control.*]**ZMin**
[*form.*][*control.*]**ZMax**

Remarks

These properties tell you the full range of the joystick's position.   They are read-only at run-time only. They are only valid after the joystick has been enabled.

Data Type

Integer (long)

See Also

Properties:

Period

Port

XPos

YPos

ZPos

XPos, YPos, and ZPos Properties

Description

Returns the current position of the joystick.

Usage

[*form.*][*control.*]**XPos**
[*form.*][*control.*]**YPos**
[*form.*][*control.*]**ZPos**

Remarks

These properties tell you where the joystick was at the last time it was polled (see Period).   They are read-only at run-time only.   They are only valid after the joystick has been enabled.

Data Type

Integer (long)

See Also

Events:
  Move
Properties:
  Period
  Port
  XMax
  XMin
  YMax
  YMin
  ZMax
  ZMin

Positional Properties Example

In this example, the program will show the changes in the position of the joystick when run.   It will also show the relative amount (as compared to <u>XMin</u>, <u>XMax</u>, <u>YMin</u>, and <u>YMax</u>).   To try this example, paste the code into the Declarations section of a form that contains two labesl and a JoyStk control.   Press F5.   Then, play with the joystick.

```
Sub Form_Load ()
    Joystick1.Enabled = True
End Sub

Sub Form_Unload (Cancel As Integer)
    Joystick1.Enabled = False
End Sub

Sub Joystick1_Move (X As Integer, Y As Integer, Z As Integer)
    Dim XP As Integer
    Dim YP As Integer

    Label1.Caption = "Position: " & X & "," & Y

    XP = (100 * (X - Joystick1.XMin)) / (Joystick1.XMax - Joystick1.XMin)
    YP = (100 * (Y - Joystick1.YMin)) / (Joystick1.YMax - Joystick1.YMin)
    Label2.Caption = "Relative: X " & XP & "%  Y " & YP & "%"
End Sub
```

Events

All of the events that apply to this control are in this table. Events that have special meaning for this control or that only apply to this control are marked with an asterisk (*).

[*ButtonDown](#)
[*ButtonUp](#)
[*Move](#)

ButtonDown Event

Description

Occurs when the user pushes on one or more of the joystick's buttons.

Syntax

Sub *ctlname*_**ButtonDown (***Button* **As Integer)**

Remarks

This event only occurs when a button is pressed.   The joystick must be enabled for this event to happen.

Button represents the button pressed.

See Also

Events:

    ButtonUp

Properties:

    Button1

    Button2

    Button3

    Button4

    Buttons

ButtonUp Event

Description

Occurs when the user releases one or more of the joystick's buttons.

Syntax

Sub *ctlname_***ButtonUp (***Button* **As Integer)**

Remarks

This event only occurs when a button is released.   The joystick must be enabled for this event to happen.

Button represents the button pressed.

See Also

Events:

Properties:

Button Event Example

In this example, the labels show which buttons are down and which are up. To try this example, paste the code into the Declarations section of a form that contains two labels and a JoyStk control.   Press F5.

```
Sub Form_Load ()
    Joystick1.Enabled = True

    Call JoystickCheck
End Sub

Sub Form_Unload (Cancel As Integer)
    Joystick1.Enabled = False
End Sub

Sub Joystick1_ButtonDown (Button As Integer)
    Call JoystickCheck
End Sub

Sub Joystick1_ButtonUp (Button As Integer)
    Call JoystickCheck
End Sub

Sub JoystickCheck ()
    Dim Up As String
    Dim Down As String

    If Not Joystick1.Button1 Then Up = Up + "1" Else
            Down = Down + "1"
    If Joystick1.Buttons >= 2 Then If Not Joystick1.Button2 Then Up = Up +
            "2" Else Down = Down + "2"
    If Joystick1.Buttons >= 3 Then If Not Joystick1.Button3 Then Up = Up +
            "3" Else Down = Down + "3"
    If Joystick1.Buttons >= 4 Then If Not Joystick1.Button4 Then Up = Up +
            "4" Else Down = Down + "4"

    Label1.Caption = "Up: " & Up
    Label2.Caption = "Down: " & Down
End Sub
```

Move Event

Description

Occurs when the user moves the joystick.

Syntax

Sub *ctlname*_**Move (***X* **As Integer,** *Y* **As Integer,** *Z* **As Integer)**

Remarks

This event occurs when the joystick's position changes.   The change must be larger than the current setting of the Threshold property.   The joystick must be enabled for this event to happen.

X, *Y*, and *Z* represent the joystick's new position (Z is only valid for 3-D joysticks).

See Also

Properties:

Registration Information

Credits

JoyStk was written by Zane Thomas.   He can be reached on CompuServe at 72060,3327 or on Internet at 72060.3327@compuserve.com.   His mailing address is:

Zane Thomas
Post Office Box 300
Indianola, WA   98342

Inquiries, tech support, comments should be sent to James Shields.   His address is 71231,2066 on CompuServe, or mabry@halcyon.com on Internet.   If you must send something via U.S. Mail, the address is:

Mabry Software
Post Office Box 31926
Seattle, WA   98103-1926

Registration

You can register this program by sending $15 ($20 for international orders) and your address.   You can register JoyStk **and** its C source code by sending $35 ($40 for international orders).   With your order, you will receive a copy of our manual documenting all of our controls.

For your convenience, an <u>order form</u> has been provided that you can print out directly from this help file.

E-mail Discount

You may take a $5 discount for e-mail delivery of this package (CompuServe or Internet).   If you choose this option, please note: a printed manual is not included.   Be sure to include your full mailing address with your order.   Sometimes (on the Internet) the package cannot be e-mailed.   So, we are forced to send it through the normal mails.

CompuServe members may also take the $5 e-mail discount by registering this package in the software registration forum (GO SWREG).   JoyStks SWREG ID number is 1340.   The source code version's ID number is 1341.

Credit Card Orders

We accept VISA and Mastercard.   If you e-mail your order to us, please be sure to include your card number, expiration date, complete mailing address, and your phone number (in case we have any questions about your order).

© Copyright 1993-1995 by Zane Thomas and Mabry Software

JoyStk Order Form

Use the Print Topic.. command from the File menu to print this order form.

Mail this Mabry Software
form to: Post Office Box 31926
           Seattle, WA   98103-1926

           Phone: 206-634-1443
           Fax: 206-632-0272
           BBS: WinDev BBS 206-634-0783
           CompuServe: 71231,2066
           Internet: mabry@halcyon.com

Where did you get this copy of JoyStk?

_____

Ship to: _____

           _____

           _____

           _____

           _____

Phone: _____

Fax: _____

E-Mail: _____

MC/VISA: _____ exp. _____

Disk Size: (circle one)   3½   5¼

   qty ordered ____   REGISTRATION
                          $15 each, postpaid (check or money order in hard currency). Outside of North
                          America add $5.00 shipping.

   qty ordered ____   SOURCE CODE AND REGISTRATION
                          $35 each, postpaid (check or money order in hard currency). Outside of North
                          America add $5.00 shipping.

Getting Custom Controls Written

If you or your organization would like to have custom controls written, you can contact me at the following:

James Shields
Mabry Software
Post Office Box 31926
Seattle, WA   98103-1926

Phone: 206-634-1443
Fax: 206-632-0272
BBS: WinDev BBS 206-634-0783

CompuServe: 71231,2066
Internet: mabry@halcyon.com

You can also contact Zane Thomas.   He can be reached at:

Zane Thomas
Post Office Box 300
Indianola, WA   98342

CompuServe: 72060,3327