About Fishhead Software

URL:

http://ww.fishware.com

Email: (for fastest reponse)

support@fishware.com

Voice:

630-892-6958

Fax:

630-892-6958

Write:
Fishhead Software
912 LaFayette Street
Aurora, IL 60505

Source Code

Remember: You can not distribute the source code or use any portion of it to create commercial, shareware, or freeware ActiveX controls or similar software.

Subclassing pitfalls

- When subclassing is turned on for the control at design time, GPFs can happen. They happened because the subclassing does not expect addresses to change. But, when you make a change to the control's source code, VB changes addresses. Therefore the control will reference an invalid address. To get around this, you can uncomment the code which test for UserMode in the routine Subclass of the control module. This only happens when you have a form opened with the control on it.

- When running an application, always use the application's exit option. Never use VB's end button. VB's end button will not notify the controls and loaded dlls that the application has ended. The net result, memory loss and/or VB to crash.

Modifying the source code to create an OCX for your company

Please do the following to the modified control:

- Change the project name and description;
- Change the class name of the control;
- Change the OCX name;
- Change version information to reflect your company;
- Use a different load address;

Note: other people have bought the control and may sell their applications to the same client. Taking these precautions will save you and our clients many headaches.

System Requirements

The system requirements are as follows:

Windows 95, 98 or Windows NT 4.0
Visual Basic 4.0 (32 bit) or Visual Basic 5.0 or higher
486 or higher processor
8 MB RAM
2 MB disk space

fsScroll ActiveX Control Version 2.00 A member of fsVBActiveX

About Fishhead Software   Copyright

fsScroll ActiveX Control is a container control that can automatically scroll the contained controls. fsScroll ActiveX Control can be used in two ways. fsScroll control can be sized to fit the entire form and work as a virtual form or it can be sized to fit a portion of the form to and scroll only the contained controls within it.

fsScroll5.ocx for use with Visual Basic 4.0 and 5.0
GUID = 0F7409B3-4543-11D2-A269-000000000000
Requires MSVBVM50.DLL

fsScroll6.ocx for use with Visual Basic 6.0
GUID = A2BF48B3-3D6A-11D2-A253-000000000000
Requires MSVBVM60.DLL

Getting Started
Frequently Asked Questions
Features and Uses

Properties
Methods
Events
Constants

System Requirements

Visit Fishhead Software on the WEB

www.fishware.com

Getting Started

fsScroll is easy to use.   To try it out, do the following:

1. Create a new project.   Select the "standard exe" project type.

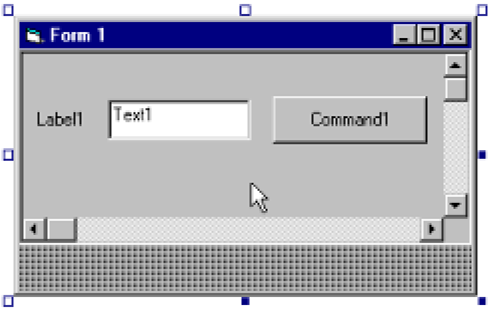
2. From the Component dialog, select "Fishhead Software fsScroll Control for VB 5" or "Fishhead Software fsScroll Control for VB 6" if using Visual Basic 6. If you are using the demo version, then select "Fishhead Software fsScroll Control Demo".

3. Select fsScroll control from the toolbar and draw it on the form.

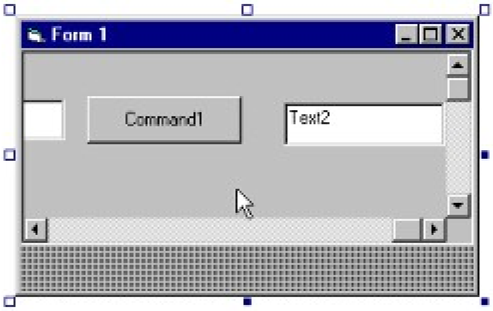
4. At this time you can change the properties of fsScroll control.   Lets change the Align property to vbAlignTop.   This will make the control resize to fit the top border and sides of the form.

5. Also, change the ScrollBars property to fsSBBoth.   This setting will cause the control to display the horizontal and vertical scroll bars all the time.   We could leave the setting as fsSBAuto and let the control determine which scroll bars need to be displayed at runtime.   Note: unlike Visual Basic's ScrollBars property, you change this setting at run time.

6. Now add a label, textbox, and command button and then move them to the left.

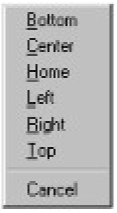


7. From the toolbar, add another text box and reposition the controls as in 6 above with textbox2 farthest right.   The easiest way to do this, hold down the Ctrl + right mouse button and move the mouse cursor. Notice how all the contained controls move at once.   This is a special feature of fsScroll to make designing large forms easier.



8. Select run and then press the tab key to tab to each contained control.   Notice how fsScroll will change the view to show the contained control that has the focus.

9. Like in design mode, hold down Ctrl + right mouse button and move the mouse cursor. Again, all the contained controls will move. This time press Ctrl + right mouse button and release. A popup menu should appear. Select the left option. This will move the controls, so the left most control will be in view. You can turn these features off by setting AllowCtrlRMouse and AllowPopupMenu properties to **false**.



10. So far we haven't had to add any code. This will be the normal case when using this control. To make fsScroll fit the form, add the following to the form's resize event, fsScroll1.Height = Me.ScaleHeight.

Frequently Asked Questions

Why are there two versions (fsScroll5.ocx and fsScroll6.ocx)?

fsScroll5.ocx was designed to be used in 32-bit Visual Basic 4 and Visual Basic 5 applications. It was created with Visual Basic 5. Therefore, fsScroll5.ocx requires MSVBVM50.DLL to be installed on the user's computer. Where as fsScroll6.ocx was designed to be used in Visual Basic 6 applications. It was created with Visual Basic 6. Therefore, fsScroll6.ocx requires MSVBVM60.DLL. to be installed on the users computer.

This does not mean you cannot use fsScroll5.ocx as part of a Visual Basic 6 application, you can. However, you will need MSVBVM60.DLL and MSVBVM50.DLL installed on the user's computer. When you run the application, both DLLs will be loaded. The Visual Basic 6 application will load MSVBVM60.DLL, while fsScroll5.ocx will load MSVBVM50.DLL.. This method will require more memory and will extend the load time.

How do I upgrade from fsScroll5.ocx or fsScroll6.ocx? Visual Basic gives me an error when I try to remove fsScroll.

To upgrade from fsScroll5.ocx to fsScroll6.ocx, you will need to open your Visual Basic project file with Notepad. Find the line that looks the one below:

```
    Object={0F7409B3-4543-11D2-A269-000000000000}#1.0#0; fsScroll5.ocx

and change it to

    Object={A2BF48B3-3D6A-11D2-A253-000000000000}#1.1#0; fsScroll6.ocx
```

Can I use fsScroll6.ocx in Visual Basic 4 or 5?

Yes you can, but not recommended.

How come when I add or remove a control at runtime, fsScroll does not recognize the change?

In most circumstances, Windows will send a WM_PARENTNOTIFY message, but not always.
To resolve this problem, after adding or removing the control, you need to call the Refresh method.
This will tell fsScroll to update its internal list.

How come some controls causes problems when I added it to fsScroll?

Depending on the method of subclassing, some controls will not work properly with fsScroll. To get around
this problem, add a frame control and then add the incompatible control onto the frame control. Or try a different control with the same functionality as the incompatible control.

Features and Uses

Features

- Small footprint;
- Uses standard Visual Basic runtime DLL, no additional DLLs or OCXs required;
- Visual Basic 5 and higher applications can compiled OCX into the application;
- Automatic scrolling when a contained control gets focus at run time;
- Supports MouseWheel in native environments, not available for Windows 95;
- Use of Ctrl + right mouse button to move contained controls at run time and design time for faster navigation;
- Appearance, BorderStyle and ScrollBars properties can be modified at design and run time;
- Win32 style scroll bars;

Uses

- To create a virtual form (no need to worry about screen sizes);
- To scroll a portion of the form;
- To create an item list;

Copyright and Trademarks

Constants

```
Public Enum fsAppearanceConstants
    fsAFlat = 0
    fsA3D = 1
End Enum

Public Enum fsBorderStyleConstants
    fsBSNone = 0
    fsBSFixedSingle = 1
    fsBSAuto = 2
End Enum

Public Enum fsSCError
            fsSCErrInvalidPropertyValue = -2147191503
End Enum

Public Enum fsOLEDropModeConstants
    fsOLEDropNone = 0
    fsOLEDropManual = 1
End Enum

Public Enum fsScrollBarsConstants
    fsSBNone = 0
    fsSBHorizontal = 1
    fsSBVertical = 2
    fsSBBoth = 3
    fsSBAuto = 4
End Enum
```

Properties

About	Standard Property
AccessKeys	Standard Property
ActiveControl	Standard Property
AdjustBottom	
AdjustLeft	
AdjustRight	
AdjustTop	
Allign	Standard Property
AllowClipCursor	
AllowCtrlRMouse	
AllowPopupMenu	
Appearance	Standard Property
AutoRedraw	Standard Property
AutoScroll	
BackColor	Standard Property
BorderStyle	Standard Property
DragIcon	Standard Property
DragMode	Standard Property
Enabled	Standard Property
Height	Standard Property
hWnd	Standard Property
Indexed	Standard Property
LargeChangeX	
LargeChangeY	
Left	Standard Property
MouseIcon	Standard Property
MousePointer	Standard Property
Name	Standard Property
Object	Standard Property
OLEDropMode	Standard Property
Parent	Standard Property
ScrollBars	
ScrollBorderX	
ScrollBorderY	
SmallChangeX	
SmallChangeY	
TabIndex	Standard Property
TabStop	Standard Property
Tag	Standard Property
ToolTipText	Standard Property
Top	Standard Property
Visible	Standard Property
WhatThisHelpId	Standard Property
Width	Standard Property

AdjustBottom, AdjustLeft, AdjustRight and AdjustTop Properties

Adjusts the left, top, right, and bottom values by *n* when scrolling the contained control with the focus into view.

Syntax

object.**AdjustBottom** [= *single*]
object.**AdjustLeft** [= *single*]
object.**AdjustRight** [= *single*]
object.**AdjustTop** [= *single*]

The Adjust properties syntax has these parts:

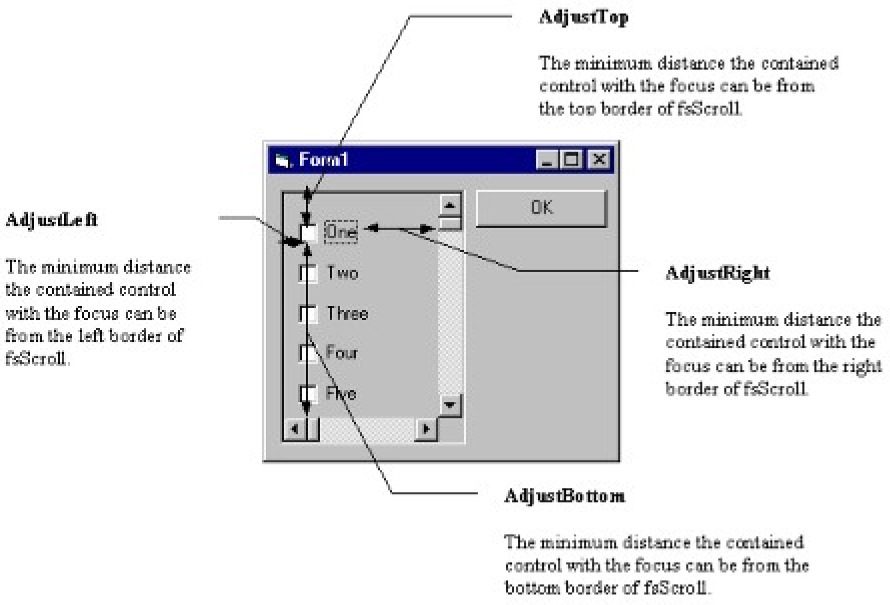
Part	Description
object	An object expression that evaluates to fsScroll object.
single	The amount to adjust the contained control values when it comes into view.

Default Values in twips

AdjustBottom	300
AdjustLeft	1200
AdjustRight	300
AdjustTop	300

Remarks

The Adjust properties apply to all contained controls that can receive focus. fsScroll uses these properties to position the contained control upon focus within view. The Adjust values are useful to help display associated fields such as labels into view when the contained control gets the focus. To set the Adjust values for a specific contained control, call the method SetControlAdjustValues.



AdjustTop

The minimum distance the contained control with the focus can be from the top border of fsScroll.

AdjustLeft

The minimum distance the contained control with the focus can be from the left border of fsScroll.

AdjustRight

The minimum distance the contained control with the focus can be from the right border of fsScroll.

AdjustBottom

The minimum distance the contained control with the focus can be from the bottom border of fsScroll.

Note: fsScroll can only adjust the amount set in the Adjust properties if there is enough distance between the contained control with the focus and fsScroll border. This happens when the Adjust value is greater than the original positional value for the control. For example: if the AdjustLeft was set to 2000 and the left property of the contained control was set to 1500, then fsScroll can only position the contained control 1500 twips from the left border.

AllowClipCursor Property

Toggles restricting the mouse cursor within fsScroll window area.

Syntax

object.**AllowClipCursor** [= *boolean*]

The AllowClipCursor property syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
boolean	Determines if the mouse cursor shoud be restricted to fsScroll window area. The property default value is **false**.

Remarks

When set to true, the mouse cursor will be restricted to fsScroll window area. When set to false, the mouse cursor can move anywhere on the screen. When restricting the mouse cursor, remember to allow the user a way to close the current form or application.

Note

1) If the form or fsScroll gets resized or moved, the developer should reset AllowClipCursor.
2) Setting this property has no effect during design time.

AllowCtrlRMouse Property

Toggles the Ctrl + right mouse button activation to scroll the contained controls.

Syntax

object.**AllowCtrlRMouse** [= *boolean*]

The AllowCtrlRMouse property syntax has these parts:

Part	**Description**
object	An object expression that evaluates to fsScroll object.
boolean	Determines if the Ctrl + right mouse is active when scrolling is available. The property default value is **true**.

Remarks

When set to true and a scroll bar is present, the user can use Ctrl + right mouse button click to scroll the contained controls. This is useful to quickly navigate a large virtual window.

Note

Setting this property has no effect during design time. Ctrl + right mouse button will always be available during design time.

AllowPopupMenu Property

Determines if the popup menu will be displayed after the user has pressed Ctrl + right mouse button.

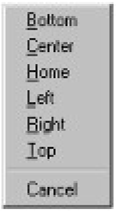
Syntax

object.**AllowPopupMenu** [= *boolean*]

The AllowPopupMenu property syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
boolean	Allows showing the popup menu. The property default value is **true**.

Remarks



When set to true and a scroll bar is present and AllowCtrlRMouse property is true, the user can use Ctrl + right mouse click to popup a menu.   This is useful to quickly navigate a large virtual window.

PopupMenu	Action
Bottom	Moves the view to the bottom;
Center	Centers the view;
Home	Returns the view to original position;
Left	Changes the view to leftmost position;
Right	Changes the view to rightmost position;
Top	Moves the view to the top;
Cancel	Closes the popup menu;

AutoScroll Property

Determines if the contained control with the focus should be scrolled into view when gaining focus.

Syntax

object.**AutoScroll** [= *boolean*]

The AutoScroll property syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
boolean	The property default value is **True**.

Remarks

Most applications will want this property be set to **True**. However, in special cases, you may want to set this to **False**. By setting this property to **False**, it will be up to the application to scroll the hidden contained controls into view.

LargeChangeX, LargeChangeY Properties

Returns or sets the amount of change to the view area when the user clicks the area between the scroll box and scroll arrow.

Syntax

object.**LargeChangeX** [= *single*]
object.**LargeChangeY** [= *single*]

The LargeChange properties syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
single	Specsifies the amount of change.

Default Values in twips

LargeChangeX	300
LargeChangeY	300

See Also
SmallChangeX
SmallChangeY

SmallChangeX, SmallChangeY Properties

Returns or sets the amount of change to the view area when the user clicks a scroll arrow.

Syntax

object.**SmallChangeX** [= *single*]
object.**SmallChangeY** [= *single*]

The SmallChange properties syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
single	Specsifies the amount of change.

Default Values in twips

SmallChangeX	75
SmallChangeY	75

See Also
LargeChangeX
LargeChangeY

ScrollBars Property

Returns or sets a value indicating whether a fsScroll has horizontal or vertical scroll bars.

Syntax

object.**ScrollBars** [= *setting*]

The ScrollBars property syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
setting	Determines which scroll bars gets activated.

Settings of fsScrollBarsConstants

Constant	Setting	Description
fsSBNone	0	None;
fsSBHorizontal	1	Horizontal;
fsSBVertical	2	Vertical;
fsSBBoth	3	Both;
fsSBAuto	4	(Default) Scroll bars are displayed when the contained controls extend beyond fsScroll borders;

Remarks

When set to a value other than fsSBNone and fsSBAuto, scrolling and the scroll bars will be available. If the scroll bars are not visible, then Ctrl + Right Mouse button will not be available, regardless what the property setting is set to for AllowCtrlRMouse.

ScrollBorderX, ScrollBorderY Properties

Returns or sets the distance from the border to cursor position for auto mouse scrolling.

Syntax

object.**ScrollBorderX** [= *single*]
object.**ScrollBorderY** [= *single*]

The ScrollBorder properties syntax has these parts:

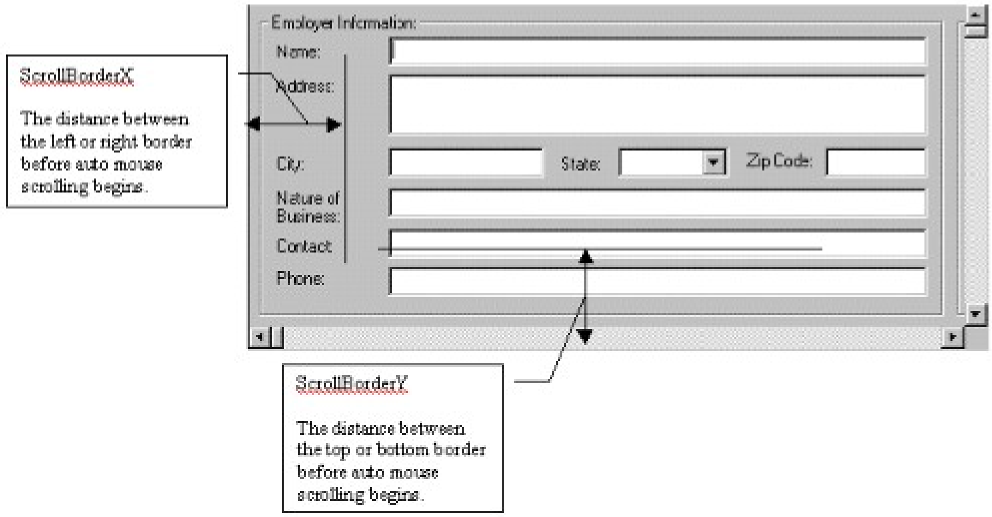
Part	**Description**
object	An object expression that evaluates to fsScroll object.
single	The distance between the border and the mouse cursor for auto mouse scrolling.

Default Values in twips

ScrollBorderX	0
ScrollBorderY	0

Remarks

The ScrollBorderX and ScrollBorderY properties allows the developer to set auto mouse scrolling. The ScrollBorderX sets the horizontal distance between the left or right border and the mouse cursor, while ScrollBorderY sets the vertical distance between the top or bottom border and the mouse cursor. Once the cursor is in this area, fsScroll will begin scrolling it contents.



ScrollBorderX

The distance between the left or right border before auto mouse scrolling begins.

ScrollBorderY

The distance between the top or bottom border before auto mouse scrolling begins.

To turn on auto mouse scrolling, set ScrollBorderX or ScrollBorderY to a non-zero value. To turn off auto mouse scrolling, set ScrollBorderX and ScrollBorderY to zero.

Methods

Drag	Standard Method
GetControlAdjustValues	
GetScrollPos	
GetScrollPosMax	
GetScrollPosMin	
Move	Standard Method
OLEDrag	Standard Method
Refresh	Standard Method
RemoveControlAdjustValues	
SetControlAdjustValues	
SetScrollPos	
ShowControl	
ShowWhatsThis	Standard Method
Version	
ZOrder	Standard Method

GetControlAdjustValues Method

Gets the adjust values for a contained control.

Syntax

object.**GetControlAdjustValues** (hWnd As Long, AdjustLeft As Single, AdjustTop As Single, AdjustRight As Single, AdjustBottom As Single)

The GetControlAdjustValues method syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
hWnd	A window handle.
AdjustLeft	The distance to adjust the contained control from the left border.
AdjustTop	The distance to adjust the contained control from the top border.
AdjustRight	The distance to adjust the contained control from the right border.
AdjustBottomt	The distance to adjust the contained control from the bottom border.

Remarks

This method is useful when you need to know the adjust values for a contain control.   Under most circumstances these values will be zero.

Note: fsScroll will not generate an error message if the hWnd value is not a valid contained control hWnd.

See Also

RemoveControlAdjustValues, SetControlAdjustValues, Adjust Properties

Example

fsScroll1.GetControlAdjustValues Text1.hWnd, aleft, atop, aright, abottom

GetScrollPos Method

Returns the thumb position for each scroll bar;

Syntax

object.**GetScrollPos** (xPos As Single, yPos As Single*)*

The GetScrollPos method syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
xPos	The thumb position for the horizontal scroll bar.
yPos	The thumb position for the vertical scroll bar.

Remarks

This method is useful when you need to know where the thumb positionfor each scroll bar.

Example

```
fsScroll1.GetScrollPos xPos, yPos
```

GetScrollPosMax Method

Returns the maximum thumb position for each scroll bar;

Syntax

object.**GetScrollPosMax** (xMax As Single, yMax As Single)

The GetScrollPosMax method syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
xMax	The maximum thumb position for the horizontal scroll bar.
yMax	The maximum thumb position for the vertical scroll bar.

Remarks

This method is useful when you need to know the maximum thumb position for each scroll bar.

Example

```
fsScroll1.GetScrollPosMax xMax, yMax
```

GetScrollPosMin Method

Returns the minimum thumb position for each scroll bar;

Syntax

object.**GetScrollPosMin** (xMin As Single, yMin As Single)

The GetScrollPosMin method syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
xMin	The minimum thumb position for the horizontal scroll bar.
yMin	The minimum thumb position for the vertical scroll bar.

Remarks

This method is useful when you need to know the minimum thumb position for each scroll bar.

Example

```
fsScroll1.GetScrollPosMin xMin, yMin
```

RemoveControlAdjustValues Method

Removes the adjust values for a specified contain control.

Syntax

object.**RemoveControlAdjustValues** (hWnd As Long**)**

The RemoveControlAdjustValues method syntax has these parts:

Part	**Description**
object	An object expression that evaluates to fsScroll object.
hWnd	A window handle.

Remarks

This method will remove the adjust values that were previously set by SetControlAdjustValues.

Note: fsScroll will not generate an error message if the hWnd value is not a valid contained control hWnd.

See Also

GetControlAdjustValues, SetControlAdjustValues, Adjust Properties

Example

```
fsScroll1.RemoveControlAdjustValues Text1.hWnd
```

SetControlAdjustValues Method

Sets the Adjust values for specified contained control.

Syntax

object.S**etControlAdjustValues** (hWnd As Long, New_AdjustLeft As Single, New_AdjustTop As Single, New_AdjustRight As Single, New_AdjustBottom As Single*)*

The SetControlAdjustValues method syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
hWnd	A window handle.
New_AdjustLeft	The distance to adjust the contained control from the left border.
New_AdjustTop	The distance to adjust the contained control from the top border.
New_AdjustRight	The distance to adjust the contained control from the right border.
New_AdjustBottomt	The distance to adjust the contained control from the bottom border.

Remarks

Use this method to set a specific contained control Adjust values to something different than the Adjust properties. Under normal circumstances, this method is not needed. But, can be useful to give a specific contained control a different action when getting the focus. Such as positioning the bottommost contained control an additional amount to show the associated text beneath.

Note: fsScroll will not generate an error message if the hWnd value is not a valid contained control hWnd.

See Also

GetControlAdjustValues, RemoveControlAdjustValues, Adjust Properties

Example

fsScroll1.SetControlAdjustValues Text1.hWnd, 3000, 200, 1000, 1000

SetScrollPos Method

Sets the thumb position for each scroll bar;

Syntax

object.**SetScrollPos** (xPos As Single, yPos As Single)

The SetScrollPos method syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
xPos	The new thumb position for the horizontal scroll bar.
yPos	The new thumb position for the vertical scroll bar.

Remarks

This method is useful when you need to update thumb position for each scroll bar. This can be used with a timer to information across the screen. If xPos is outside the range of the horizontal scrollbar, then the maximum or minimum value will be used. If yPos is outside the range of the vertical scrollbar, then the maximum or minimum value will be used.

Example

```
fsScroll1.SetScrollPos xPos, yPos
```

ShowControl Method

Brings a contained control into view.

Syntax

object.**ShowControl** (hWnd As Long)

The ShowControl method syntax has these parts:

Part	**Description**
object	An object expression that evaluates to fsScroll object.
hwnd	A window handle to a contained control.

Remarks

This method is useful when a contained control needs to be in view.

Example

```
fsScroll1.ShowControl = text1.hWnd   ' This will force text1 to be in the view area
```

Version Method

Returns the version number for fsScroll.

Syntax

object.**Version** () As String

The Version method syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object;

Remarks

The Version method will return a the saved version information in the form of major.minor.revision.

Example

```
MsgBox sScroll1.Version    ' Will display "1.40.0000" for the 1.40 release
```

Events

Click	Standard Event
DblClick	Standard Event
DragDrop	Standard Event
DragOver	Standard Event
Error	
MouseDown	Standard Event
MouseMove	Standard Event
MouseUp	Standard Event
OLECompleteDrag	Standard Event
OLEDragDrop	Standard Event
OLEDragOver	Standard Event
OLEGiveFeedback	Standard Event
OLESetData	Standard Event
OLEStartDrag	Standard Event
Scroll	
WithFocus	

Error Event

The Error event gets fired when an error occurs.

Syntax

Private Sub object_**Error** (ByVal nError As Long, ByVal Description As String,   bCancel As Boolean)

The Error event syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
nError	The error number being generated
Description	A string value representing the generated error.
bCancel	Determines if an error message dialog displays. Default is False.

Remarks

This event is useful if you want to centralize your error handling or prevent fsScroll from displaying an error (Set bCancel = True).

Scroll Event

Returns the thumb position for each scroll bar during scrolling;

Syntax

object.**Scroll** ([index As Integer], xPos As Single, yPos As Single)

The Scroll event syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
index	An integer that uniquely identifies a control if it's in a control array.
xPos	The thumb position for the horizontal scroll bar during scroll.
yPos	The thumb position for the vertical scroll bar during scroll.

Remarks

This event will be fired when the thumb gets moved for either scrollbar.

Example

```
Private Sub fsScroll1_ Scroll (xPos As Single, yPos As Single)

End Sub
```

WithFocus Event

This event fires whenever a contained control gets focus;

Syntax

object.**WithFocus** ([index As Integer], hWnd As Long)

The WithFocus event syntax has these parts:

Part	Description
object	An object expression that evaluates to fsScroll object.
index	An integer that uniquely identifies a control if it's in a control array.
hWnd	The Window handle to the control with the focus.

Remarks

Use this event to track when a particular contained control gets the focus.   Can be useful for controls that don't fire the GotFocus event.

Example

```
Private Sub fsScroll1_WithFocus(hWnd As Long)

End Sub
```