

# MhFormScroll



Version 1.0

[Introduction](#)

[Getting Started](#)

[Reference](#)

[Other MicroHelp Products for Visual Basic](#)

# ***Introduction***

MhFormScroll is a FREE custom control that allows you to have "virtual" forms in Visual Basic. This control adds horizontal and/or vertical scroll bars to a form, allowing you to create a client area that is much larger than the visible area. Any child controls that are placed on the form are automatically scrolled for you in both design and run modes.

MhFormScroll will also work on most other "container" type controls, including the standard Visual Basic picture control, and even the MDI control in the Microsoft Professional Toolkit. [VBTools 2 also has a MDI control with additional features, including built in scroll bars.]

## ***Technical Support***

Since MhFormScroll is being released as "freeware", we cannot provide free telephone technical support like we do for the rest of our products. However, if you find a bug or have a problem with MhFormScroll, you can leave us a message on our BBS. We will do our best to fix any problems you might find and answer your questions through the BBS.

The latest version of MhFormScroll can be downloaded from our BBS. The filename is MHFS200.ZIP.

The telephone numbers for our BBS are:

(404) 516-1397      2400 baud  
(404) 516-1497      9600 baud (USR dual standard)

# ***Getting Started***

Using MhFormScroll is very easy. Place the MHFS200.VBX in either your Windows or Windows\System directory, or anywhere in your executable path. Start Visual Basic and select File-Add File (ALT-F, D) from the VB menu. From the file open dialog box, change to the directory where you copied MHFS200.VBX and double click on the its file name. The icon for MhScrollForm will appear in the VB toolbar and the control ready to use just like any of the other controls in the toolbar.

## ***Accessing Help***

You can access this Windows Help file either directly through a "File Open" in WinHelp, or directly from the custom control. Place the MHFS200.HLP file in your \WINDOWS, \WINDOWS\SYSTEM, or any directory in your executable path. Place a single instance of MhFormScroll on a form and then double-click the right mouse button on the control. WINHELP.EXE will be started with the MHFS200.HLP file indexed to the reference section.

All of MicroHelp's VBTools 2 controls have this automatic Windows help access feature built in!

# MhFormScroll

**Description:** The MhFormScroll control allows you to add scroll bars to a Visual Basic form or container control, allowing scrolling of the viewable area within the defined "virtual" form.

**Class Name:** MhFormScroll

**Remarks:** The MhFormScroll control is visible only at design time.

When you enable the horizontal and/or vertical scroll bar properties for the control, it modifies the window style of its parent to have the selected types of scroll bars.

The scroll bars provide "instant feedback" and will scroll the parent window in both design and run modes. Any child controls placed on the parent will be automatically scrolled with the client area of the parent.

While you can place a MhFormScroll control on forms and several different types of controls (including VB picture controls), it may not always work as you expect. If you place a MhFormScroll control on anything other than a form or picture control, you should test it thoroughly to make sure it will give the desired effect.

If you are using a MhFormScroll control on a VB form and want to set the .Picture property, you must set the MhFormScroll.HVal and .VVal properties to zero before selecting the image. Otherwise, the image selected will not be painted on the form correctly.

**Properties** (Property names without asterisks are standard VB properties and are not defined here)

CtlName	Height	*HLargeChange	*HMax	
*HMin	HSmallChange	*HValue	Left	
Scalemode	*Scrollbars	Tag	Top	
*VirtualHeight		*VirtualWidth	*VLargeChange	*VMax
*VMin	*VSmallChange	*VValue	Width	

## Events:

Change

## Custom Properties

### HLargeChange (*Integer*)

**Description:** The .HLargeChange property sets or returns the amount to increase or decrease the .HValue property when the scrollbar is clicked between the thumb and the ends of the horizontal scrollbar.

**Usage:** `[form].MhFormScroll.HLargeChange = increment%`

**Remarks:** The default value of the `.HLargeChange` property is 10. The valid range is 0 - 32,767.

**HMax** (*Integer*)

**Description:** The `.HMax` property sets or returns the maximum value of the horizontal scrollbar when the thumb is moved to the rightmost position.

**Usage:** `[form].MhFormScroll.HMax = limit%`

**Remarks:** The default value of the `.HMax` property is 100. The valid range is 0 - 32,767.

**HMin** (*Integer*)

**Description:** The `.HMin` property sets or returns the minimum value of the horizontal scrollbar when the thumb is moved to the leftmost position.

**Usage:** `[form].MhFormScroll.HMin = limit%`

**Remarks:** The default value of the `.HMin` property is 0. The valid range is 0 - 32,767. The `.HMin` property must be less than the `.HMax` property or an error will occur.

**HSmallChange** (*Integer*)

**Description:** The `.HSmallChangeValue` property sets or returns the amount to increase or decrease the `.HValue` property when the scrollbar is clicked on the arrows at each end of the control.

**Usage:** `[form].MhFormScroll.HSmallChangeValue = increment%`

**Remarks:** The default value of the `.HSmallChangeValue` property is 1. The valid range is 0 - 32,767. Normally, the minimum value for this property would be 1. If you set it to zero, the horizontal scroll arrows will be disabled.

**HValue** (*Integer*)

**Description:** The `.HValue` property sets or returns the current position of the horizontal scrollbar. Setting this value in your code will cause the `_Change` event to be fired.

**Usage:** `[form].MhFormScroll.HValue = value%`

**Remarks:** The default value of the .HValue property is 0. The valid range is .HMin to HMax.

### **ScrollBars** (*Integer*)

**Description:** The .ScrollBars property sets or returns the type and presence of scrollbars on the parent object.

**Usage:** `[form].MhFormScroll.ScrollBars = value%`

**Remarks:** The default value of the .ScrollBars property is 0. The valid range is zero to three.

<b>Setting</b>	<b>Description</b>
0	None (no scrollbars)
1	Vertical
2	Horizontal
3	Both

### **VirtualHeight** (*Integer*)

**Description:** The .VirtualHeight property sets or returns the "virtual height" of the parent object. By default, MhFormScroll uses pixels for the scalemode. If you set the .ScaleMode property value for MhFormScroll to any value other than one, the control will use pixels for the scale mode. The value placed in .VirtualHeight defines the size of the "virtual form" where the form's actual .Height and .Width properties define the viewable area within the virtual form.

**Usage:** `[form].MhFormScroll.VirtualHeight = value%`

**Remarks:** The default value of the .VirtualHeight property is 2 times the "container" control's height at the time that MhFormScroll is created. The valid range is zero to 32,767, but by definition this value should never be less than the parent containers .Height property.

### **VirtualWidth** (*Integer*)

**Description:** The .VirtualWidth property sets or returns the "virtual width" of the parent object. By default, MhFormScroll uses pixels for the scalemode. If you set the .ScaleMode property value for MhFormScroll to any value other than one, the control will use pixels for the scale mode. The value placed in .VirtualWidth defines the size of the "virtual form" where the form's actual .Height and .Width properties define the viewable area within the virtual form.

**Usage:** `[form].MhFormScroll.VirtualWidth = value%`

**Remarks:** The default value of the .VirtualWidth property is 2 times the

"container" control's height at the time that MhFormScroll is created. The valid range is zero to 32,767, but by definition this value should never be less than the parent containers .Height property.

#### **VLargeChange** (*Integer*)

**Description:** The .VLargeChange property sets or returns the amount to increase or decrease the .HValue property when the scrollbar is clicked between the thumb and the ends of the horizontal scrollbar.

**Usage:** `[form].MhFormScroll.VLargeChange = increment%`

**Remarks:** The default value of the .VLargeChange property is 10. The valid range is 0 - 32,767.

#### **VMax** (*Integer*)

**Description:** The .VMax property sets or returns the maximum value of the vertical scrollbar when the thumb is moved to the bottommost position.

**Usage:** `[form].MhFormScroll.VMax = limit%`

**Remarks:** The default value of the .VMax property is 100. The valid range is 0 - 32,767.

#### **VMin** (*Integer*)

**Description:** The .VMin property sets or returns the minimum value of the vertical scrollbar when the thumb is moved to the topmost position.

**Usage:** `[form].MhFormScroll.VMin = limit%`

**Remarks:** The default value of the .VMin property is 0. The valid range is 0 - 32,767. The .VMin property must be less than the .VMax property or an error will occur.

#### **VSmallChangeValue** (*Integer*)

**Description:** The .VSmallChangeValue property sets or returns the amount to increase or decrease the .VValue property when the scrollbar is clicked on the arrows at each end of the control.

**Usage:** `[form].MhFormScroll.VSmallChangeValue = increment%`

**Remarks:** The default value of the .VSmallChangeValue property is 1. The valid range is 0 - 32,767. Normally, the minimum value for this property would be 1. If you set it to zero, the vertical scroll arrows will be disabled.

**VValue** (*Integer*)

**Description:** The .VValue property sets or returns the current position of the vertical scrollbar. Setting this value in your code will cause the \_Change event to be fired.

**Usage:** *[form].MhFormScroll.VValue = value%*

**Remarks:** The default value of the .VValue property is 0. The valid range is .VMin to VMax.



# *Other MicroHelp Products for Visual Basic*

**VBTools version 2-** Contains **36** custom controls, video "special effects", examples of how to use some of the more popular Windows API functions, access to the Windows 3.1 "Common Dialog Routines" and much MORE!

**MicroHelp Muscle** - The only Visual Basic add-on to win Computer Language Magazine's 1991 Productivity Award. Muscle contains over 430 assembly language routines that provide you with the speed and functionality that is not available anywhere else. DOS and Windows low level services, string and array manipulation, date and time functions, bit manipulation and routines that will allow you to create arrays that can use all available Windows memory (including temporary and permanent swap files).

**MicroHelp Communications Library** - Since Visual Basic 1.0 doesn't have any native communications support, this library provides you with everything you need, including source code for a working Windows terminal program. Seven different file transfer protocols that work even when your application is minimized, ANSI terminal emulation, and the ability to have upto eight simultaneous communications sessions at ONE TIME!

**MicroHelp Network Library** - 100% assembly language routines that support Novell, Lantastic and NetBIOS API functions. Now you can access and manipulate all objects in the bindery, add, change and delete items in the print queues as well as redirect all shared resources.

**MicroHelp VBXRef** - The "ultimate" Project Management tool for Visual Basic. VBXRef is the **ONLY** utility that can actually read the binary VB form file (.FRM). Create reports that contain ALL form and control properties and current values, or only those properties that have values that have been changed from their defaults. VBXRef also creates a cross reference listing of your entire project, listing all functions, procedures and variables. It even tells you how many times each of these items was referenced in your project as well as where in the project any variable was assigned a value.

For more information and free DEMOS call:



4359 Shallowford Ind Pkwy  
Marietta, GA 30066

**\*In Georgia or Outside the U.S. (404) 516-0899**





