

ScrollBars Demo



```
/*
Professional Windows GUI Programming Using C#
by Jay Glynn, Csaba Torok, Richard Conway, Wahid Choudhury,
Zach Greenvoss, Shripad Kulkarni, Neil Whitlow

Publisher: Peer Information
ISBN: 1861007663
*/
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace ScrollBars
{
    ///
    /// Summary description for ScrollBars.
    ///
    public class ScrollBars : System.Windows.Forms.Form
    {
        private System.Windows.Forms.TextBox textBox1;
        private System.Windows.Forms.VScrollBar vScrollBar1;
        ///
        /// Required designer variable.
        ///
        int counter=0;
        private System.Windows.Forms.Label label1;
        private System.ComponentModel.Container components = null;

        public ScrollBars()
        {
```

```
//
// Required for Windows Form Designer support
//
InitializeComponent();

//
// TODO: Add any constructor code after InitializeComponent
call
//
}

///

/// Clean up any resources being used.
///
protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if (components != null)
        {
            components.Dispose();
        }
    }
    base.Dispose( disposing );
}

#region Windows Form Designer generated code
///

/// Required method for Designer support – do not modify
/// the contents of this method with the code editor.
///
private void InitializeComponent()
{
    this.textBox1 = new System.Windows.Forms.TextBox();
    this.vScrollBar1 = new System.Windows.Forms.VScrollBar();
    this.label1 = new System.Windows.Forms.Label();
    this.SuspendLayout();
}
```

```

//
// textBox1
//
this.textBox1.Font = new System.Drawing.Font("Microsoft Sans
Serif", 12F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
this.textBox1.ForeColor = System.Drawing.Color.Transparent;
this.textBox1.Location = new System.Drawing.Point(24, 56);
this.textBox1.Multiline = true;
this.textBox1.Name = "textBox1";
this.textBox1.ReadOnly = true;
this.textBox1.Size = new System.Drawing.Size(144, 32);
this.textBox1.TabIndex = 4;
this.textBox1.Text = "";
this.textBox1.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center;
//
// vScrollBar1
//
this.vScrollBar1.Location = new System.Drawing.Point(168, 56);
this.vScrollBar1.Name = "vScrollBar1";
this.vScrollBar1.Size = new System.Drawing.Size(16, 32);
this.vScrollBar1.TabIndex = 7;
this.vScrollBar1.Scroll +=
new
System.Windows.Forms.ScrollEventHandler(this.vScrollBar1_Scrol
l);
//
// label1
//
this.label1.Location = new System.Drawing.Point(8, 16);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(192, 16);
this.label1.TabIndex = 6;
this.label1.Text = "Numeric Scrolling using VScroll Bars";
//
// ScrollBars
//

```

```

this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.ClientSize = new System.Drawing.Size(208, 109);
this.Controls.AddRange(new System.Windows.Forms.Control[] {
this.label1,
this.vScrollBar1,
this.textBox1});
this.Name = "ScrollBars";
this.Text = "Numeric Scroll";
this.Load += new System.EventHandler(this.ScrollBars_Load);
this.ResumeLayout(false);

}
#endregion

///

/// The main entry point for the application.
///
[STAThread]
static void Main()
{
Application.Run(new ScrollBars());
}

private void textBox1_TextChanged(object sender,
System.EventArgs e)
{
}

private void ScrollBars_Load(object sender, System.EventArgs
e)
{
// Set the maximum range for the scrollbar
vScrollBar1.Maximum = 100;
// Set the minimum range for the scrollbar
vScrollBar1.Minimum = 0 ;

// Set the SmallChange factor
vScrollBar1.SmallChange = 1;

```

```

}

private void vScrollBar1_Scroll(object sender,
System.Windows.Forms.ScrollEventArgs e)
{
// Check if the increment is Small
if ( e.Type == ScrollEventType.Last )
counter = 100 ;
else
// Check if the scroll is moved to minimum pos
if ( e.Type == ScrollEventType.First)
counter = 0 ;
else
// Check if the scroll is moved small distance
if ( e.Type == ScrollEventType.SmallDecrement )
counter - ;
else
// Check if the scroll is moved small distance
if ( e.Type == ScrollEventType.SmallIncrement )
{
counter++;
MessageBox.Show("Small increment");
}
else
// Check if the scroll is moved large distance
if ( e.Type == ScrollEventType.LargeDecrement )
counter-=5;
else
// Check if the scroll is moved large distance
if ( e.Type == ScrollEventType.LargeIncrement )
{
MessageBox.Show("Large increment");
counter+=5;
}
else
// Check if the scroll is moved to the Min position
if ( e.Type == ScrollEventType.First )

```

```
counter = 0 ;  
else  
// Check if the scroll to the Max position  
if ( e.Type == ScrollEventType.Last)  
counter = 100 ;  
  
Console.WriteLine(e.NewValue+  
");  
// Check if the scroll is moved large distance  
if ( counter > 100 ) counter = 100 ;  
if ( counter < 0 ) counter = 0 ; textBox1.Text =  
counter.ToString() ; } } } [/csharp]
```