

Bitmap Scrolling

Description

The following demonstrates how to import and scroll a single bitmap. It also explains how to offset mouse input to take into account scrolling. The Paint and MouseClick events are handled in a panel that is derived from ScrollableControl. The existing Panel class in .NET is already derived from this control. There is also a default bitmap that has been added to the project resources.

ScrollableControl

You do not typically use the ScrollableControl class directly. The Panel class inherits from this class. The ScrollableControl class acts as a base class for controls that require the ability to scroll. To enable a control to display scroll bars as needed, set the AutoScrollMinSize property to the desired size. When the control is sized smaller than the specified minimum size the appropriate scroll bars are displayed.

AutoScrollMinSize

Gets or sets the minimum size of the automatic scroll bars. The AutoScrollMinSize property is used to manage the screen size allocated to the automatic scroll bars.

AutoScrollPosition

Gets or sets the location of the auto-scroll position. The AutoScrollPosition property is used to adjust the position of items rendered in the scrollable control.

Example

```
using System;
using System.Drawing;
using System.Windows.Forms;

namespace ScrollingBitmapTutorialApplication
{
    public partial class Form1 : Form
    {
        // By default the bitmap will point a bitmap in the resources file.
        Bitmap bitmap = Properties.Resources._default;

        public Form1()
        {
            InitializeComponent();

            // The scroll bars need to be adjusted to the current
            // size of the default bitmap. This will cause the bars
            // to appear and disappear depending on whether the panel
            // is smaller or larger than the bitmap. The panel must
            // be docked in order to change size.
            graphicsPanel1.AutoScrollMinSize = bitmap.Size;
        }
    }
}
```

```

private void graphicsPanel11_Paint(object sender, PaintEventArgs e)
{
    // The location at which the bitmap is rendered must be offset
    // by the amount the user has scrolled. The amount the user has
    // scrolled is available through the panel's AutoScrollPosition property.
    // It's X and Y value should be added to location the bitmap would
    // be rendered to if it was not scrolled.

    // Default location of the bitmap would be 0,0.
    Point defaultLocation = new Point(0, 0);

    Point offsetLocation = Point.Empty;

    // Add AutoScrollPosition to the default location.
    offsetLocation.X = defaultLocation.X + graphicsPanel11.AutoScrollPosition.X;
    offsetLocation.Y = defaultLocation.Y + graphicsPanel11.AutoScrollPosition.Y;

    // Render the bitmap to the offset location.
    e.Graphics.DrawImage(bitmap, offsetLocation);
}

private void newToolStripMenuItem_Click(object sender, EventArgs e)
{
    // Set the bitmap back to the default.
    bitmap = Properties.Resources._default;

    // Readjust the scroll bars.
    graphicsPanel11.AutoScrollMinSize = bitmap.Size;

    // Invalidate the panel.
    graphicsPanel11.Invalidate();
}

private void importToolStripMenuItem_Click(object sender, EventArgs e)
{
    OpenFileDialog dlg = new OpenFileDialog();

    // Set the dialog filters.
    dlg.Filter = "All Files|*.*|BMP Files|.bmp|JPG Files|.jpg";

    // Display the dialog box
    if (DialogResult.OK == dlg.ShowDialog())
    {
        // Import the new bitmap.
        bitmap = new Bitmap(dlg.FileName);

        // Readjust the scroll bars.
        graphicsPanel11.AutoScrollMinSize = bitmap.Size;

        // Invalidate the panel.
        graphicsPanel11.Invalidate();
    }
}

private void graphicsPanel11_MouseClick(object sender, MouseEventArgs e)
{
    // Any mouse input must take into account the current scroll position.
    // This is done by subtracting AutoScrollPosition from the default
    // mouse location.

    // Default mouse location.
    Point defaultMouseLocation = e.Location;

    Point offsetMouseLocation = Point.Empty;

    // Subtract AutoScrollPosition from the default mouse location.
    offsetMouseLocation.X = defaultMouseLocation.X - graphicsPanel11.AutoScrollPosition.X;
    offsetMouseLocation.Y = defaultMouseLocation.Y - graphicsPanel11.AutoScrollPosition.Y;
}
}
}

```