C# Windows .NET Visual-Studio VS2010

A Simple Thumbnail Viewer Control

Nguyen.H.H.Dang

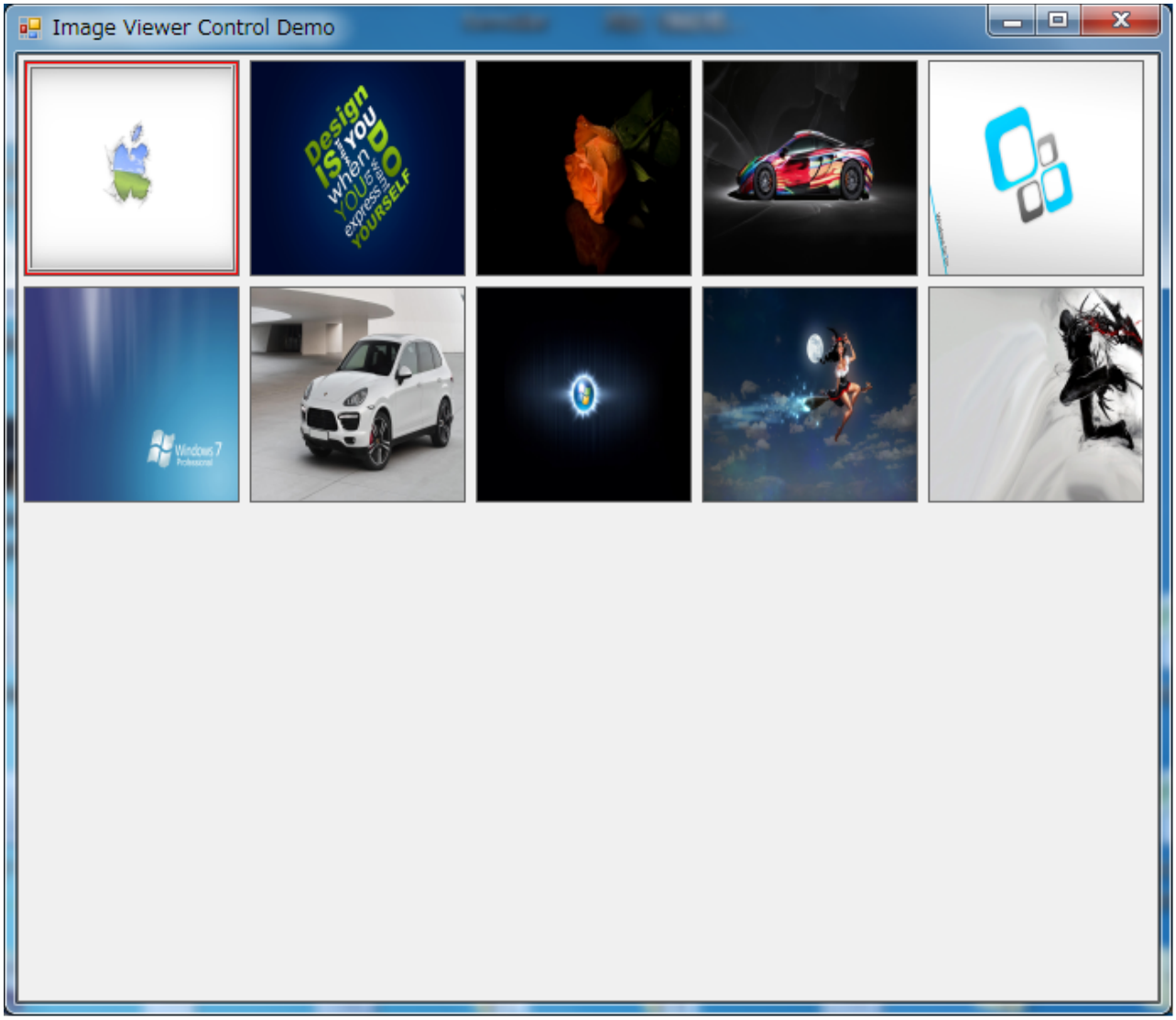
3.86/5 (3 votes)

3 Jul 2014 CPOL 1 min read 27.1K 1.4K

A very simple Thumbnail Viewer inherited from FlowLayoutPanel for beginners

Download ThumbnailViewer.rar - 45.6 KB

Download ThumbnailViewer.zip - 48.3 KB



Introduction

This control allows you to add Thumbnail by dragdrop image from desktop and preview the Image by double-clicking on Thumbnail.

Using the Code

At first, declare an "`ImageExtensions`" List to make sure all files added to Control are Image Type.

C#

```
/// <summary> Image Extensions accepted by this control
        /// </summary>
        private List<string> ImageExtensions = new List<string>
        { ".JPG", ".JPE", ".BMP", ".GIF", ".PNG" };
```

The `DragDrop` system is just simple, use `DragEnter` event to set an effect when user drags files to control and `DragDrop` event to deal with the files.

C#

```
 void ThumbnailViewerControl_DragEnter(object sender, DragEventArgs e)
        {
            if (e.Data.GetDataPresent(DataFormats.FileDrop))
                e.Effect = DragDropEffects.Copy;
            else
                e.Effect = DragDropEffects.None;
        }

void ThumbnailViewerControl_DragDrop(object sender, DragEventArgs e)
        {
            if (e.Data.GetDataPresent(DataFormats.FileDrop))
            {
                string[] files = (string[])e.Data.GetData(DataFormats.FileDrop);
                AddImage(files);
            }
        }
```

When dropped files to control, the `AddImage` method will add image to `List` as binary and then show a thumbnail on `Panel` by `MakeThumbnail` method.

C# Shrink ▲

```
public void AddImage(string[] files)
        {
            this.Cursor = Cursors.WaitCursor;

            byte[] binary;
            for (int i = 0; i < files.Count(); i++)
            {
                // Only accept Image files
                if
(ImageExtensions.Contains(Path.GetExtension(files[i]).ToUpperInvariant()))
                {
                    // Convert Image File to Binary
                    binary = File.ReadAllBytes(files[i]);

                    // Add binary data to List
                    ImageList.Add(binary);

                    // Create a Thumnail of Image and add Thumbnail to Panel
                    MakeThumbnail(binary);

                    GC.GetTotalMemory(true);
                }
            }

            this.Cursor = Cursors.Default;
        }

public void MakeThumbnail(byte[] binary)
        {
            // Create a Picture Box for showing thumbnail image
```

```
            PictureBox thumb = new PictureBox();
            thumb.MaximumSize = new Size(128, 128);
            thumb.MinimumSize = new Size(128, 128);
            thumb.Size = new Size(128, 128);
            thumb.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
            thumb.SizeMode = PictureBoxSizeMode.Zoom;

            // Create a border when Mouse entered
            thumb.MouseEnter += new EventHandler(thumb_MouseEnter);

            // Clear border when Mouse leaved
            thumb.MouseLeave += new EventHandler(thumb_MouseLeave);

            // Preview image when Mouse Double Clicked
            thumb.DoubleClick += new EventHandler(thumb_DoubleClick);

            // Set thumbnail image
            MemoryStream ms = new MemoryStream();
            thumb.Image = Image.FromStream(new MemoryStream(binary))
                .GetThumbnailImage(thumb.Width - 2, thumb.Height - 2, null, new
IntPtr());
            ms.Close();

            // Add to Panel
            this.Controls.Add(thumb);
        }
```

Now, thumbnails are added to `Panel`, but to make it more prettier when user moves mouse cursor over it, use `MouseEnter` event to draw a border around thumbnail and clear border when `Mouse` leaves by `MouseLeave` event.

C#

```
void thumb_MouseLeave(object sender, EventArgs e)
        {
            ((PictureBox)sender).Invalidate();
        }

void thumb_MouseEnter(object sender, EventArgs e)
        {
            var rc = ((PictureBox)sender).ClientRectangle;
            rc.Inflate(-2, -2);
            ControlPaint.DrawBorder(((PictureBox)sender).CreateGraphics()
                , ((PictureBox)sender).ClientRectangle, Color.Red,
ButtonBorderStyle.Solid);
            ControlPaint.DrawBorder3D(((PictureBox)sender).CreateGraphics()
                , rc, Border3DStyle.Bump);
        }
```

Thumbnail is small, right? So when user wants to "preview" full size image, let's show them a preview form created on thumbnail's mouse doubleclick event.

C#

```
void thumb_DoubleClick(object sender, EventArgs e)
        {
```

```
            Form previewForm = new Form();
            previewForm.FormBorderStyle = FormBorderStyle.SizableToolWindow;
            previewForm.MinimizeBox = false;
            previewForm.Size = new System.Drawing.Size(800, 600);
            previewForm.StartPosition = FormStartPosition.CenterScreen;

            PictureBox view = new PictureBox();
            view.Dock = DockStyle.Fill;

            int index = this.Controls.GetChildIndex((PictureBox)sender);
            view.Image = BinaryToImage(ImageList[index]);

            view.SizeMode = PictureBoxSizeMode.Zoom;

            previewForm.Controls.Add(view);
            previewForm.ShowDialog();
        }
```

On the event above, when doubleclicked on thumbnail, we must convert the binary data back to an `Image`, this method will do that job.

C#

```
public static Image BinaryToImage(byte[] binaryData)
       {
           if (binaryData == null) return null;
           byte[] buffer = binaryData.ToArray();
           MemoryStream memStream = new MemoryStream();
           memStream.Write(buffer, 0, buffer.Length);
           return Image.FromStream(memStream);
       }
```

That's all !!! This is just a tip for beginners, I recommend you add more methods to it like:

- Delete Image method: Delete a selecting thumbnail, be careful with `Panel` control index and `List` item index.
- `BackgroundWorker`: The UI will freeze when you add a large number of images, so a backgroundworker will handle it.
- Threading: Large size image will consuming time. Try something like Parallel looping to make the process faster.

Thanks for reading and feel free to comment.

License