

Course IPCis: Image Processing with C#

Chapter C1: The Bitmap Project

Copyright © by V. Miszalok, last update: 08-01-2008

- ↓ [An empty window](#)
- ↓ [Read and display an image](#)
- ↓ [Center](#)
- ↓ [Horizontal stretch](#)
- ↓ [Vertical stretch](#)
- ↓ [Maximal size](#)
- ↓ [Mirror](#)
- ↓ [Zoom animation](#)
- ↓ [Rotation animation](#)
- ↓ [Sample images](#)
- ↓ [Exercises](#)

An empty window

Guidance for **Visual Studio 2008**:

1) Main Menu after start of VS 2008: `File -> New Project... -> Visual Studio installed templates: Windows Forms Application`
 Name: `bitmap1` -> Location: `C:\temp` -> Create directory for solution: `switch off` -> `OK`
`Form1.cs[Design]` appears.

2) Two superfluous files must be deleted: `Form1.Designer.cs` and `Program.cs`.

You reach these files via the `Solution Explorer` - `bitmap1-window`:

Click the plus-sign in front of branch `bitmap1` and the plus-sign in front of branch `Form1.cs`.

Right-click the branch `Program.cs`. A context menu opens. Click `Delete`.

A message box appears: '`Program.cs`' will be deleted permanently. Quit with `OK`.

Right-click the branch `Form1.Designer.cs` and delete this file too.

3) Right-click the gray window `Form1`. A small context menu opens. Click `View Code`.

You see now the pre programmed code of `Form1.cs`. Erase this code completely.

4) Write the following three lines into the empty `Form1.cs`:

```
public class Form1 : System.Windows.Forms.Form
{
    static void Main() { System.Windows.Forms.Application.Run( new Form1() ); }
}
```

5) Click `Debug` in the main menu of VS 2008.

A submenu opens. Click `Start Without Debugging Ctrl F5`.

Important: Always finish all instances of `bitmap1` before writing new code and starting it !

Start the `Task Manager` with `Ctrl+Alt+Del` and check if any `bitmap1.exe`-process is still running. If yes, kill it.

Important Tip: In case of mistype the compiler presents a `Message Box: There were build errors. ...`

You quit with **no**. An `Error List`-window with warnings and errors will appear in Visual Studio below Your program.

In this error list scroll up to the first error (ignore the warnings !). Double click the line with the first error. The cursor jumps automatically into Your code into the line where the error was detected. Look for mistypes in this line and remove them.

(Sometimes You will not find the error in this line but above, where You forgot a comma or a semicolon.) Ignore all errors below the first error (in most cases they are just followers of the first one) and compile. Repeat this procedure until further error message boxes disappear and Your program compiles, links and starts as expected.

If You use Visual Studio 2008 Professional You should switch off the vexatious automatic format- and indent- mechanism of the code editor before You copy the following code to `Form1.cs`

(otherwise all the code will be reformatted into chaos):

1. Main menu of Visual Studio 2008 Professional: Click menu "`Tools`".

2. A drop-down-menu appears. Click "`Options...`".

3. An `Options` dialog box appears.

4. Click the branch "`Projects and Solutions`". Click "`General`". Change all three paths to `C:\temp`.

5. Click the branch "`Text Editor`", then "`C#`".

6. A sub tree appears with the following branches: "`General`", "`Tabs`", "`Advanced`", "`Formatting`", "`IntelliSense`".

7. Click "`Tabs`". Change "`Indenting`" to `None`, "`Tab size`" and "`Indent size`" to `1` and switch on the option "`Insert spaces`".

8. Within the branch "`C#`" click the plus-sign in front of "`Formatting`" and change all "`Formatting`" branches:

"`General`": switch off all check boxes, "`Indentation`": switch off all check boxes, "`New Lines`": switch off all check boxes, "`Spacing`": switch off all check boxes, "`Wrapping`": switch **on** both check boxes.

9. Quit the dialog with button "`OK`".

Read and display an image

If You don't see Your code anymore, click the tab **Form1.cs** of Visual Studio. Delete everything (including the last brace). No code remains. Write the following lines into the empty window of **Form1.cs**:

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

public class Form1 : Form
{
    static void Main() { Application.Run( new Form1() ); }
    Brush bbrush = SystemBrushes.ControlText;
    Brush rbrush = new SolidBrush( Color.Red );
    Bitmap bmp;
    int nClicks;
    public Form1()
    {
        MenuItem miRead = new MenuItem( "&Read", new EventHandler( MenuFileRead ) );
        MenuItem miExit = new MenuItem( "&Exit", new EventHandler( MenuFileExit ) );
        MenuItem miFile = new MenuItem( "&File", new MenuItem[] { miRead, miExit } );
        Menu = new System.Windows.Forms.MainMenu( new MenuItem[] { miFile } );
        Text = "Bitmap1";
        SetStyle( ControlStyles.ResizeRedraw, true );
        Width = 1024;
        Height = 800;
        try { //Delete this and the following 6 lines if you have no Internet connection running.
            System.Net.WebRequest webreq = System.Net.WebRequest.Create(
"http://www.miszalok.de/Images/Madonna.bmp" );
            System.Net.WebResponse webres = webreq.GetResponse();
            System.IO.Stream stream = webres.GetResponseStream();
            bmp = (Bitmap)Image.FromStream( stream );
            Invalidate();
        } catch { }
    }
    void MenuFileRead( object obj, EventArgs ea )
    {
        OpenFileDialog dlg = new OpenFileDialog();
        if ( dlg.ShowDialog() != DialogResult.OK ) return;
        try { bmp = (Bitmap)Image.FromFile( dlg.FileName ); } catch { return; }
        nClicks = 0;
        Invalidate();
    }
    void MenuFileExit( object obj, EventArgs ea )
    {
        Application.Exit(); }

    protected override void OnMouseDown( MouseEventArgs e )
    {
        nClicks++;
        Invalidate();
    }
    protected override void OnPaint( PaintEventArgs e )
    {
        Graphics g = e.Graphics;
        if ( bmp == null ) { g.DrawString( "Open an Image File !", Font, bbrush, 0, 0 ); return; }
        Rectangle cr = ClientRectangle;
        int line = 0;
        switch ( nClicks % 9 )
        {
            case 0: //Information
                g.DrawString( "RawFormat = " + bmp.RawFormat.ToString(), Font, bbrush, 0, line+=Font.Height );
                if ( bmp.RawFormat.Guid == ImageFormat.Bmp.Guid )
                    g.DrawString( "BMP", Font, bbrush, 0, line+=Font.Height );
                if ( bmp.RawFormat.Guid == ImageFormat.Jpeg.Guid )
                    g.DrawString( "JPG", Font, bbrush, 0, line+=Font.Height );
                g.DrawString( "Width = " + bmp.Width.ToString(), Font, bbrush, 0, line+=Font.Height );
                g.DrawString( "Height = " + bmp.Height.ToString(), Font, bbrush, 0, line+=Font.Height );
                g.DrawString( "PixelFormat = " + bmp.PixelFormat.ToString(), Font, bbrush, 0, line+=Font.Height );
                g.DrawString( "Click on left mouse button !", Font, rbrush, 0, line+=Font.Height );
                break;
            case 1: //Raw display
                g.DrawImage( bmp, 0, Font.Height );
                g.DrawString( "Click on left mouse button !", Font, rbrush, 0, 0 );
                break;
            case 2: //Center
                g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
                break;
            case 3: //Horizontal stretch
                g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
                break;
            case 4: //Vertical stretch
                g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
                break;
        }
    }
}
```

```

        case 5: //Full size
            g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
            break;
        case 6: //Mirror
            g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
            break;
        case 7: //Zoom animation
            g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
            break;
        case 8: //Rotation animation
            g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
            break;
    }
}
}

```

Click Debug -> Start Without Debugging Ctrl F5. Try out bitmap1 by reading any sort of image format: BMP, ICO, GIF, JPG, PNG, TIFF.

Center

Version2: Finish bitmap1.

Write the following lines into protected override void OnPaint(PaintEventArgs e) below the existing line case 2: //Center:

```

        Int32 x = ( cr.Width - bmp.Width ) / 2;
        Int32 y = ( cr.Height - bmp.Height ) / 2;
        g.DrawImage( bmp, x, y, bmp.Width, bmp.Height );

```

Click Debug -> Start Without Debugging Ctrl F5. Try out how the image is centered in the middle of the client area of Form1.

Horizontal stretch

Version3: Finish bitmap1.

Write the following lines into protected override void OnPaint(PaintEventArgs e) below the existing line case 3: //Horizontal stretch:

```

        x = 0;
        y = ( cr.Height - bmp.Height / 2 ) / 2;
        g.DrawImage( bmp, x, y, cr.Width, bmp.Height / 2 ); //full form width, half bmp height

```

Click Debug -> Start Without Debugging Ctrl F5. Try out how the image stretches.

Vertical stretch

Version4: Finish bitmap1.

Write the following lines into protected override void OnPaint(PaintEventArgs e) below the existing line case 4: //Vertical stretch:

```

        x = ( cr.Width - bmp.Width / 2 ) / 2;
        y = 0;
        g.DrawImage( bmp, x, y, bmp.Width / 2, cr.Height ); //half bmp width, full form height

```

Click Debug -> Start Without Debugging Ctrl F5. Try out how the image stretches.

Maximal size

Version5: Finish bitmap1.

Write the following lines into protected override void OnPaint(PaintEventArgs e) below the existing line case 5: //Full size:

```

        g.DrawImage( bmp, cr );

```

Click Debug -> Start Without Debugging Ctrl F5. Draw the window borders, to try out the sizing.

Mirror

Version6: Finish bitmap1.

Write the following lines into protected override void OnPaint(PaintEventArgs e) below the existing line case 6: //Mirror:

```

        g.DrawImage( bmp, cr.Width/2, cr.Height/2, cr.Width/2, cr.Height/2 );
        g.DrawImage( bmp, cr.Width/2, cr.Height/2, -cr.Width/2, cr.Height/2 );
        g.DrawImage( bmp, cr.Width/2, cr.Height/2, cr.Width/2, -cr.Height/2 );
        g.DrawImage( bmp, cr.Width/2, cr.Height/2, -cr.Width/2, -cr.Height/2 );

```

Click Debug -> Start Without Debugging Ctrl F5.

Zoom animation

Version7: Finish `bitmap1`.

Write the following lines into protected override `void OnPaint(PaintEventArgs e)` below the existing line `case 7: //Zoom animation:`

```
x = cr.Width / 20;
y = cr.Height / 20;
for ( Int32 i = 0; i < 20; i++ )
    g.DrawImage( bmp, 0, 0, cr.Width - x*i, cr.Height - y*i );
```

Click Debug -> Start Without Debugging Ctrl F5.

Rotation animation

Version8: Finish `bitmap1`.

Write the following lines into protected override `void OnPaint(PaintEventArgs e)` below the existing line `case 8: //Rotation animation:`

```
Single fx = cr.Width / 100;
Single fy = cr.Height / 100;
PointF[] p = new PointF[3];
p[1].X = cr.Width;
p[2].Y = cr.Height;
do
{
    p[0].X += fx;
    p[1].Y += fy;
    p[2].Y -= fy;
    g.DrawImage( bmp, p );
} while ( p[2].Y > 0 );
```

Click Debug -> Start Without Debugging Ctrl F5.

Sample images

The program should read and display a broad range of image formats: BMP, ICO, GIF, JPG, PNG, TIFF.

If You use an old 8-bit graphics board or if You set Your desktop to 256 colors, the colors may look strange.

If You find no sample images on Your hard disk, use the following ones:

Download: [Butterfly.bmp 217 kB 24bit-true-color image](#)

Download: [Madonna.bmp 18 kB 8bit-gray-value image](#)

Download: [Lena256.bmp 66 kB 8bit-gray-value image](#)

Download: [Lena512.bmp 258 kB 8bit-gray-value image](#)

Download: [Angiography.bmp 66 kB 8bit-gray-value image](#)

Evercises

Click `Help` in the main menu of Visual Studio. Click the sub-menu `Index`.

Choose `Filtered by: .NET Framework`. Type into the `Look for:-field` the following key words:

`Image.FromFile`, `Bitmap.RawFormat`, `Bitmap.RawFormat.Guid`, `Bitmap.PixelFormat`,

`Graphics.DrawImage`. Read the help texts. If the help texts cover Your code, You can remove them using the X-Button in the upper right window corner.

Change the image position in `case 2: //Center`.

Change the amount of stretch in `case 3` and `case 4`.

Change the step width and the zooming center in `case 5: //Zoom animation`.

Change the step width and the rotation direction in `case 6: //Rotation animation`.

Invent and try out new variants of the program in form of new projects: `bitmap2`, `bitmap3`.