Course IPCis: Image Processing with C#
Chapter1: More Professional: The BitmapWithButtons Project

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

This is a more professional version of the Bitmap project. On the left it has buttons that can be clicked in arbitrary order and on the right it has a Panel = space to display something. Otherwise it is identical to the original Bitmap project.

Copy all this code into an empty `Form1.cs` of a new `Windows Application C#`-project `bmpbuttons1` and clear `Form1.Designer.cs` and `Program.cs`.

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

public class Form1 : Form
{ static void Main()
  { Form form = new Form1();
    form.BackColor = Color.White;
    Application.Run( form );
  }
  Brush bbrush = SystemBrushes.ControlText;
  Brush rbrush = new SolidBrush( Color.Red );
  Bitmap bmp;
  const Int32 nButtons = 10;
  Button  [] button   = new Button[nButtons];
  Panel panel = new Panel();
  String filename = "http://www.miszalok.de/Images/Butterfly.jpg";
  Graphics g;

  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";
    for ( int i=0; i < nButtons; i++ )
    { button[i] = new Button(); Controls.Add( button[i] );
      button[i].Click += new EventHandler( button_handler );
      button[i].BackColor = SystemColors.Control;
    }
    button[0].Text = "Description";
    button[1].Text = "Original";
    button[2].Text = "Center";
    button[3].Text = "H-Stretch";
    button[4].Text = "V-Stretch";
    button[5].Text = "FullSize";
    button[6].Text = "Mirror";
    button[7].Text = "ZoomAnim";
    button[8].Text = "Rotation";
    button[9].Text = "Clear";
    Controls.Add( panel );
    panel.Paint += new PaintEventHandler( panel_paint );
    try //Delete this and the following 5 lines if you have no Internet connection running.
    { System.Net.WebRequest  webreq = System.Net.WebRequest.Create( filename );
      System.Net.WebResponse webres = webreq.GetResponse();
      System.IO.Stream       stream = webres.GetResponseStream();
      bmp = (Bitmap)Image.FromStream( stream );
    } catch {};
    //Butterfly.jpg as embedded resource:
    //bmp = new Bitmap( typeof( Form1 ), "bmpbuttons1.Butterfly.jpg" );
    Size = new Size( 800, 600 );
  }
```

```
void MenuFileRead( object obj, EventArgs ea )
{ OpenFileDialog dlg = new OpenFileDialog();
  if ( dlg.ShowDialog() != DialogResult.OK ) return;
  try { bmp = (Bitmap)Image.FromFile( filename = dlg.FileName ); } catch { return; }
  g.DrawImage( bmp, 0, 0 );
}

void MenuFileExit( object obj, EventArgs ea )
{ Application.Exit(); }

protected void button_handler( object sender, EventArgs e )
{ int line = 0;
  if ( bmp == null ) { g.DrawString( "Open an Image File !", Font, bbrush, 0, 0 ); return; }
  Size pSize = panel.Size;
  switch( ((Button)sender).Text )
  { case  "Description": //Information
      g.DrawString( "FileName  = " + filename, Font, rbrush, 0, line+=Font.Height );
      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);
      break;
    case  "Original"   : //Raw display
      g.Clear( Color.White ); g.DrawImage( bmp, 0, 100, bmp.Width, bmp.Height );
      break;
    case  "Center"     : //Center
      Int32 x = (pSize.Width  - bmp.Width ) / 2;
      Int32 y = (pSize.Height - bmp.Height) / 2;
      g.DrawImage(bmp, x, y, bmp.Width, bmp.Height);
      break;
    case  "H-Stretch"  : //Horizontal stretch
      x = 0;
      y = ( pSize.Height - bmp.Height / 2 ) / 2;
      g.DrawImage( bmp, x, y, pSize.Width, bmp.Height / 2 ); //full panel width, half bmp height
      break;
    case  "V-Stretch"  : //Vertical stretch
      x = ( pSize.Width - bmp.Width / 2 ) / 2;
      y = 0;
      g.DrawImage( bmp, x, y, bmp.Width / 2, pSize.Height ); //half bmp width, full panel height
      break;
    case  "FullSize"   : //Full size
      g.DrawImage( bmp, 0, 0, pSize.Width, pSize.Height );
      break;
   case  "Mirror"     : //Mirror
      g.DrawImage( bmp, pSize.Width/2, pSize.Height/2,  pSize.Width/2,  pSize.Height/2 );
      g.DrawImage( bmp, pSize.Width/2, pSize.Height/2, -pSize.Width/2,  pSize.Height/2 );
      g.DrawImage( bmp, pSize.Width/2, pSize.Height/2,  pSize.Width/2, -pSize.Height/2 );
      g.DrawImage( bmp, pSize.Width/2, pSize.Height/2, -pSize.Width/2, -pSize.Height/2 );
      break;
    case  "ZoomAnim"   : //Zoom animation
      x = pSize.Width  / 20;
      y = pSize.Height / 20;
      for ( Int32 i = 0; i < 20; i++ )
        g.DrawImage( bmp, 0, 0, pSize.Width - x*i, pSize.Height - y*i );
      break;
    case  "Rotation"   : //Rotation animation
      Single fx = panel.Width  / 100;
      Single fy = panel.Height / 100;
      PointF[] p = new PointF[3];
      p[1].X = panel.Width;
      p[2].Y = panel.Height;
      do
      { p[0].X += fx;
        p[1].Y += fy;
        p[2].Y -= fy;
        g.DrawImage( bmp, p );
      } while ( p[2].Y > 0 );
      break;
    case  "Clear"      : //Clear panel
      g.Clear( Color.White );
      break;
  }
}
```

```
  protected override void OnResize( EventArgs e )
  { Int32 w = ClientRectangle.Width / 5;
    Int32 h = ClientRectangle.Height / nButtons;
    Int32 i, top = 1;
    for ( i=0; i < nButtons; i++ )
    { button[i].Top    = top+2;
      button[i].Left   = 3;
      button[i].Width  = w;
      button[i].Height = h - 3;
      top += h;
    }
    panel.Location = new Point( w+4, 0 );
    panel.Size = new Size( ClientRectangle.Width-panel.Location.X, ClientRectangle.Height );
    g = panel.CreateGraphics();
  }

  protected void panel_paint( object sender, PaintEventArgs e )
  { g.DrawImage( bmp, 0, 100, bmp.Width, bmp.Height );
  }
}
```

Run the program. Drag the borders of the form and observe how the buttons and the panel adapt to their new sizes. Load other images via the `File`-menu.