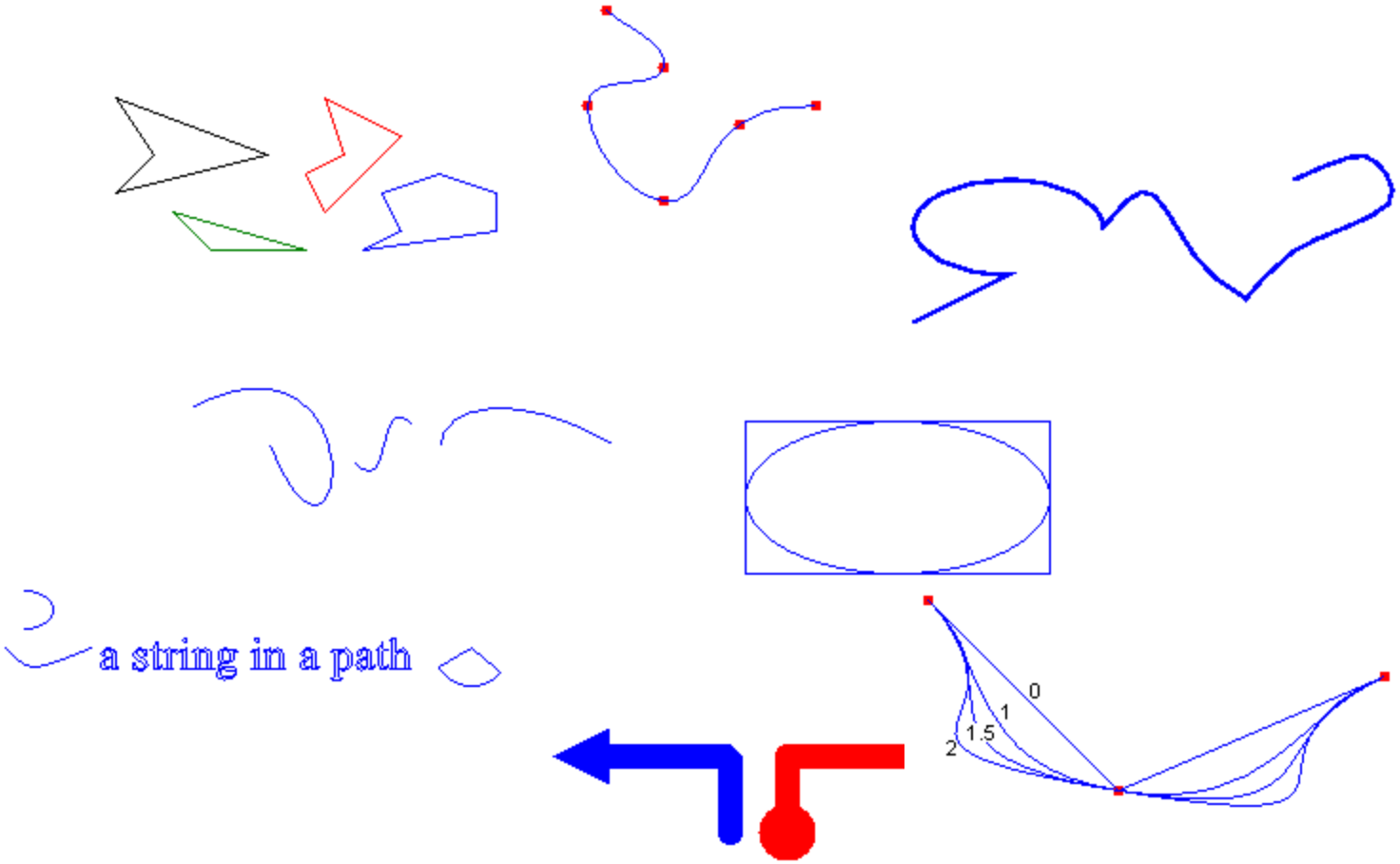


GDI+ Graphics Object

Encapsulates a GDI+ drawing surface.

System.Drawing.Graphics

- Encapsulates a GDI+ drawing surface.
- The Graphics class provides methods for drawing objects to the display device.
- A Graphics is associated with a specific device context.
- You can draw many different shapes and lines by using a Graphics object.
- You can also draw images and icons by using the **DrawImage** and **DrawIcon** methods

Obtaining a Graphics Object

- You can obtain a Graphics object by calling the **CreateGraphics** method on an object that inherits from System.Windows.Forms.**Control**.
- Or by handling a control's **Paint** event and accessing the Graphics property of the System.Windows.Forms.**PaintEventArgs** class.
- You can also create a Graphics object from an image by using the **FromImage** method.

CreateGraphics Method

```
private void AutoSizeControl(Control control, int textPadding)
{
    // Create a Graphics object for the Control.
    Graphics g = control.CreateGraphics();

    // Get the Size needed to accommodate the formatted Text.
    Size preferredSize = g.MeasureString(
       control.Text, control.Font).ToSize();

    // Pad the text and resize the control.
    control.ClientSize = new Size(
       preferredSize.Width + (textPadding * 2),
       preferredSize.Height + (textPadding * 2));

    // Clean up the Graphics object.
    g.Dispose();
}
```

PaintEventArgs

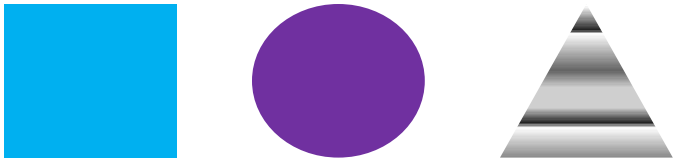
```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    // Create a local version of the graphics object for the
    Graphics g = e.Graphics;

    // Draw a string.
    g.DrawString("This is a diagonal line drawn on the control",
        new Font("Arial", 10), Brushes.Blue, new Point(30, 30));

    // Draw a line.
    g.DrawLine(Pens.Red, new Point(0,0), new Point(500,500));
}
```

Some Fill Methods

- Fills the given shape with a **Brush**.
  - FillEllipse Fills the interior of an ellipse defined by a bounding rectangle specified by a pair of coordinates, a width, and a height.
  - FillRectangle Fills the interior of a rectangle.
  - FillRectangles Fills the interiors of a series of rectangles.
  - FillPath Fills the interior of a **GraphicsPath**.



Some Draw Methods

- Outlines the given shape with a **Pen**.
  - DrawBezier Draws a Bézier spline defined by four **Point** structures.
  - DrawEllipse Draws an ellipse defined by a bounding rectangle.
  - DrawLine Draws a line connecting the two points.
  - DrawPath Draws a **GraphicsPath**.

DrawString Method

Draws the specified text string at the specified location with the specified **Brush** and **Font** objects.

```
public void DrawStringPointF(PaintEventArgs e)
{

    // Create string to draw.
    String drawString = "Sample Text";

    // Create font and brush.
    Font drawFont = new Font("Arial", 16);
    SolidBrush drawBrush = new SolidBrush(Color.Black);

    // Create point for upper-left corner of drawing.
    PointF drawPoint = new PointF(150.0F, 150.0F);

    // Draw string to screen.
    e.Graphics.DrawString(drawString, drawFont, drawBrush, drawPoint);
}
```

DrawImage Method

Draws the specified **Image** at the specified location. The Image can be rendered either smaller or larger than the original depending upon which override of the method you use .

```
private void DrawImageRectRect(PaintEventArgs e)
{

    // Create image.
    Image newImage = Image.FromFile("SampImag.jpg");

    // Create rectangle for displaying image.
    Rectangle destRect = new Rectangle(100, 100, 450, 150);

    // Create rectangle for source image.
    Rectangle srcRect = new Rectangle(50, 50, 150, 150);
    GraphicsUnit units = GraphicsUnit.Pixel;

    // Draw image to screen.
    e.Graphics.DrawImage(newImage, destRect, srcRect, units);
}
```

DrawImage Method

There are 30 different overrides of this function!

```
private void DrawImagePoint(PaintEventArgs e)
{
    // Create image.
    Image newImage = Image.FromFile("SampImag.jpg");

    // Create Point for upper-left corner of image.
    Point ulCorner = new Point(100, 100);

    // Draw image to screen.
    e.Graphics.DrawImage(newImage, ulCorner);
}
```

```
public void DrawImageRectF(PaintEventArgs e)
{
    // Create image.
    Image newImage = Image.FromFile("SampImag.jpg");

    // Create rectangle for displaying image.
    RectangleF rect = new RectangleF(100.0F, 100.0F, 450.0F, 150.0F);

    // Draw image to screen.
    e.Graphics.DrawImage(newImage, rect);
}
```

The Bitmap Class

- Encapsulates a GDI+ bitmap, which consists of the pixel data for a graphics image and its attributes.
- A **Bitmap** is an object used to work with images defined by pixel data.
- The **Bitmap** class is derived from the **Image** base class.
- GDI+ supports the following file formats: BMP, GIF, EXIG, JPG, PNG and TIFF.

The Bitmap Class

- You can create images from files, streams, and other sources by using one of the **Bitmap** constructors and save them to a stream or to the file system with the **Save** method.
- Images are drawn to the screen or to memory by using the **DrawImage** method of the **Graphics** object.

The End

GDI+: The Graphics Object