GDI+ Coordinate Structures

Point, Size and Rectangle

Point Structure

- Represents an ordered pair of integer x- and y-coordinates that defines a point in a two-dimensional plane.

Point Properties

- **IsEmpty** Gets a value indicating whether this Point is empty.
- **X** Gets or sets the x-coordinate of this Point.
- **Y** Gets or sets the y-coordinate of this Point.

Point Equality

```
private void Button1_Click(System.Object sender, System.EventArgs e)
{

    // Construct a new Point with integers.
    Point Point1 = new Point(100, 100);

    // Create a Graphics object.
    Graphics formGraphics = this.CreateGraphics();

    // Construct another Point, this time using a Size.
    Point Point2 = new Point(new Size(100, 100));

    // Call the equality operator to see if the points are equal,
    // and if so print out their x and y values.
    if (Point1 == Point2)
    {
        formGraphics.DrawString(String.Format("Point1.X: " +
            "{0},Point2.X: {1}, Point1.Y: {2}, Point2.Y {3}",
            new object[] { Point1.X, Point2.X, Point1.Y, Point2.Y }),
            this.Font, Brushes.Black, new PointF(10, 70));
    }

}
```

Size Structure

- Stores an ordered pair of integers, typically the width and height of a rectangle.
- The unit for the **Height** and **Width** of the Size structure depend on the PageUnit and PageScale settings for the Graphics object used to draw.

Size Properties

- Height Gets or sets the vertical component of this Size.
- IsEmpty Tests whether this Size has width and height of 0.
- Width Gets or sets the horizontal component of this Size.

Size Constructor

```
private void InitializeLabel1()
{
    // Set a border.
    Label1.BorderStyle = BorderStyle.FixedSingle;

    // Set the size, constructing a size from two integers.
    Label1.Size = new Size(100, 50);

    // Set the location, constructing a point from a 32-bit integer
    // (using hexadecimal).
    Label1.Location = new Point(0x280028);

    // Set and align the text on the lower-right side of the label.
    Label1.TextAlign = ContentAlignment.BottomRight;
    Label1.Text = "Bottom Right Alignment";
}
```

Rectangle Structure

- Stores a set of four integers that represent the location and size of a rectangle.
- A rectangle is defined by its width, height, and upper-left corner.
- Top, Left, Right and Bottom properties are read only.

Rectangle Properties

- Bottom Gets the y-coordinate that is the sum of the Y and Height property values.
- Height Gets or sets the height.
- IsEmpty Tests whether all numeric properties of this Rectangle have values of zero.
- Left Gets the x-coordinate of the left edge.
- Location Gets or sets the coordinates of the upper-left corner.
- Right Gets the x-coordinate that is the sum of X and Width property value.
- Size Gets or sets the size of this Rectangle.
- Top Gets the y-coordinate of the top edge.
- Width Gets or sets the width.
- X Gets or sets the x-coordinate of the upper-left corner.
- Y Gets or sets the y-coordinate of the upper-left corner.

Rectangle Methods

- Contains Determines if the specified point is contained within the rectangle.
- Inflate Inflates a Rectangle structure by the specified amount.
- Intersect Determines the Rectangle structure that represents the intersection of two rectangles.
- Offset Adjusts the location of this rectangle by the specified amount.
- Union Gets a Rectangle structure that contains the union of two Rectangle structures.

Rectangle Intersection

```
private void InstanceRectangleIntersection(PaintEventArgs e)
{

    Rectangle rectangle1 = new Rectangle(50, 50, 200, 100);
    Rectangle rectangle2 = new Rectangle(70, 20, 100, 200);

    e.Graphics.DrawRectangle(Pens.Black, rectangle1);
    e.Graphics.DrawRectangle(Pens.Red, rectangle2);

    if (rectangle1.IntersectsWith(rectangle2))
    {
        rectangle1.Intersect(rectangle2);
        if (!rectangle1.IsEmpty)
        {
            e.Graphics.FillRectangle(Brushes.Green, rectangle1);
        }
    }
}
```

Rectangle Inflate

```
public void RectangleInflateTest2(PaintEventArgs e)
{

    // Create a rectangle.
    Rectangle rect = new Rectangle(100, 100, 50, 50);

    // Draw the uninflated rectangle to screen.
    e.Graphics.DrawRectangle(Pens.Black, rect);

    // Set up the inflate size.
    Size inflateSize = new Size(50, 50);

    // Call Inflate.
    rect.Inflate(inflateSize);

    // Draw the inflated rectangle to screen.
    e.Graphics.DrawRectangle(Pens.Red, rect);
}
```

Floating Point Structures

- **PointF** - Represents an ordered pair of floating-point x- and y-coordinates that defines a point in a two-dimensional plane.
- **SizeF** - Stores an ordered pair of floating-point numbers, typically the width and height of a rectangle.
- **RectangleF** - Stores a set of four floating-point numbers that represent the location and size of a rectangle.

The End

GDI+ Coordinate Structures