

# Chapter 40: GDI+

## Section 40.1: Draw Shapes

To start drawing a shape you need to define a pen object The Pen accepts two parameters:

1. Pen Color or Brush
2. Pen Width

The Pen Object is used to create an **outline** of the object you want to draw

After Defining the Pen you can set specific Pen Properties

```
Dim pens As New Pen(Color.Purple)
pens.DashStyle = DashStyle.Dash 'pen will draw with a dashed line
pens.EndCap = LineCap.ArrowAnchor 'the line will end in an arrow
pens.StartCap = LineCap.Round 'The line draw will start rounded
'*Notice* - the Start and End Caps will not show if you draw a closed shape
```

Then use the graphics object you created to draw the shape

```
Private Sub GraphicForm_Paint(sender As Object, e As PaintEventArgs) Handles MyBase.Paint
    Dim pen As New Pen(Color.Blue, 15) 'Use a blue pen with a width of 15
    Dim point1 As New Point(5, 15) 'starting point of the line
    Dim point2 As New Point(30, 100) 'ending point of the line
    e.Graphics.DrawLine(pen, point1, point2)

    e.Graphics.DrawRectangle(pen, 60, 90, 200, 300) 'draw an outline of the rectangle
```

By default, the pen's width is equal to 1

```
Dim pen2 as New Pen(Color.Orange) 'Use an orange pen with width of 1
Dim origRect As New Rectangle(90, 30, 50, 60) 'Define bounds of arc
e.Graphics.DrawArc(pen2, origRect, 20, 180) 'Draw arc in the rectangle bounds
```

End Sub

## Section 40.2: Fill Shapes

Graphics.FillShapes draws a shape and fills it in with the color given. Fill Shapes can use

1. Brush Tool - to fill shape with a solid color

```
Dim rect As New Rectangle(50, 50, 50, 50)
e.Graphics.FillRectangle(Brushes.Green, rect) 'draws a rectangle that is filled with green

e.Graphics.FillPie(Brushes.Silver, rect, 0, 180) 'draws a half circle that is filled with silver
```

2. HatchBrush Tool - to fill shape with a pattern

```
Dim hBrush As New HatchBrush(HatchStyle.ZigZag, Color.SkyBlue, Color.Gray)
'creates a HatchBrush Tool with a background color of blue, foreground color of gray,
'and will fill with a zigzag pattern
Dim rectan As New Rectangle(100, 100, 100, 100)
```

```
e.Graphics.FillRectangle(hBrush, rectan)
```

3. LinearGradientBrush - to fill shape with a gradient

```
Dim lBrush As New LinearGradientBrush(point1, point2, Color.MediumVioletRed, Color.PaleGreen)
Dim rect As New Rectangle(50, 50, 200, 200)
e.Graphics.FillRectangle(lBrush, rect)
```

4. TextureBrush - to fill shape with a picture

You can choose a picture from resources, an already defined Bitmap, or from a file name

```
Dim textBrush As New TextureBrush(New Bitmap("C:\ColorPic.jpg"))
Dim rect As New Rectangle(400, 400, 100, 100)
e.Graphics.FillPie(textBrush, rect, 0, 360)
```

Both the Hatch Brush Tool and LinearGradientBrush import the following statement : **Imports System.Drawing.Drawing2D**

## Section 40.3: Text

To draw text onto the form use the DrawString Method

When you draw a string you can use any of the 4 brushes listed above

```
Dim lBrush As New LinearGradientBrush(point1, point2, Color.MediumVioletRed, Color.PaleGreen)
e.Graphics.DrawString("HELLO", New Font("Impact", 60, FontStyle.Bold), lBrush, New Point(40, 400))
'this will draw the word "Hello" at the given point, with a linearGradient Brush
```

Since you can't define the width or height of the text use Measure Text to check text size

```
Dim lBrush As New LinearGradientBrush(point1, point2, Color.MediumVioletRed, Color.PaleGreen)
Dim textSize = e.Graphics.MeasureString("HELLO", New Font("Impact", 60, FontStyle.Bold), lBrush)
'Use the textSize to determine where to place the string, or if the font needs to be smaller
```

Ex: You need to draw the word "Test" on top of the form. The form's width is 120. Use this loop to decrease the font size till it will fit into the forms width

```
Dim FontSize as Integer = 80
Dim textSize = e.graphics.measeString("Test", New Font("Impact",FontSize, FontStyle.Bold), new
Brush(colors.Blue, 10)
Do while textSize.Width >120
FontSize = FontSize -1
textSize = e.graphics.measeString("Test", New Font("Impact",FontSize, FontStyle.Bold), new
Brush(colors.Blue, 10)
Loop
```

## Section 40.4: Create Graphic Object

There are three ways to create a graphics object

1. From the **Paint Event**

Every time the control is redrawn (resized, refreshed...) this event is called, use this way if you want the control to

---

consistently draw on the control

```
'this will work on any object's paint event, not just the form  
Private Sub Form1_Paint(sender as Object, e as PaintEventArgs) Handles Me.Paint  
    Dim gra as Graphics  
    gra = e.Graphics  
End Sub
```

## 2. Create Graphic

This is most often used when you want to create a one time graphic on the control, or you don't want the control to repaint itself

```
Dim btn as New Button  
Dim g As Graphics = btn.CreateGraphics
```

## 3. From an Existing Graphic

Use this method when you want to draw and change an existing graphic

```
'The existing image can be from a filename, stream or Drawing.Graphic  
Dim image = New Bitmap("C:\TempBit.bmp")  
Dim gr As Graphics = Graphics.FromImage(image)
```