Mr. Giansante

Visual Basic

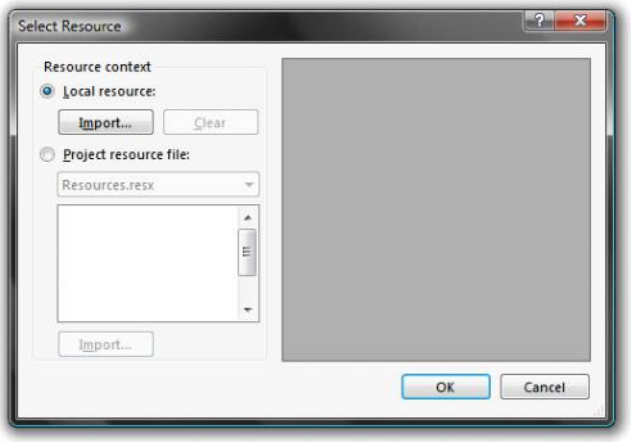
PictureBox Control & Graphics

August 2016

PictureBox Control

To load an existing image into a PictureBox (named PictureBox1) at design time, select the **Image** property of the PictureBox.

You will then be able to select a **Local Resource** and add it to your project.



To load an existing image into a PictureBox (named PictureBox1) ...

```
PictureBox1.Image = Image.FromFile("H:\Photos\Nico.jpg")
```

To clear the Picture in a PictureBox (named Picturebox1) ...

```
PictureBox1.Image = Nothing
```

To check if a PictureBox contains an image or not ...

```
If (PictureBox1.Image Is Nothing) Then
```

If you want the PictureBox's size to adjust to the size of the image it contains, set the **AutoSize** property to True.

Clipboard - Copy

```
My.Computer.Clipboard.SetImage(PictureBox1.Image)
```

Clipboard - Cut

```
My.Computer.Clipboard.SetImage(PictureBox1.Image)
PictureBox1.Image = Nothing
```

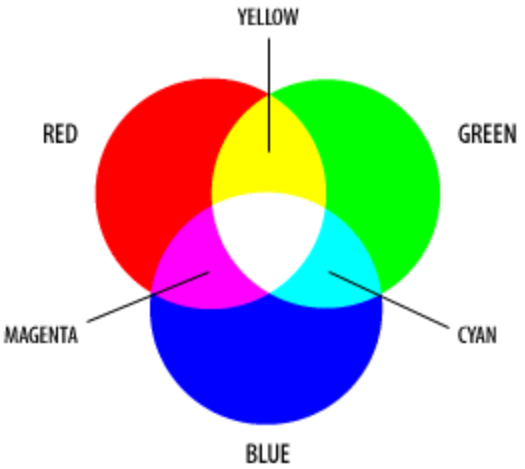
Clipboard - Paste

```
If My.Computer.Clipboard.ContainsImage() Then
   PictureBox1.Image = My.Computer.Clipboard.GetImage()
End If
```

Basics of Graphics

RGB Color

When programming with Visual Basic, colors can be specified using three numbers which describe the amount of Red, Green and Blue a certain color contains.



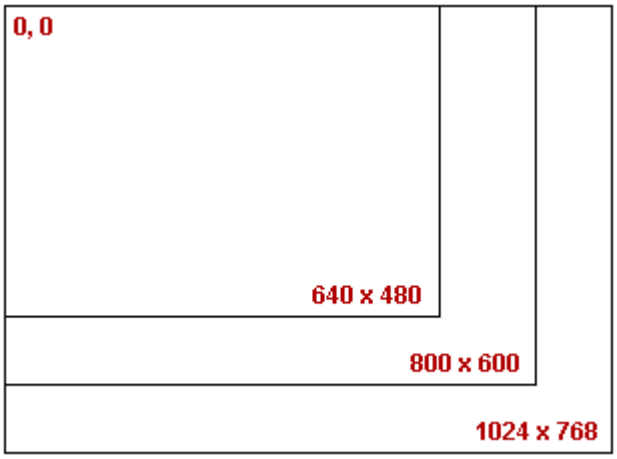
A value of 0 is the lowest possible and 255 is the highest. Using this system, there are approx. 16.7 million color choices available.

The following table lists some standard colors and their red, green, and blue values:

Black	RGB(0, 0, 0)
White	RGB(255, 255, 255)
Red	RGB(255, 0, 0)
Green	RGB(0, 255, 0)
Blue	RGB(0, 0, 255)
Cyan	RGB(0, 255, 255)
Magenta	RGB(255, 0, 255)
Yellow	RGB(255, 255, 0)

Coordinate System

When programming with Visual Basic, points can be specified using the x and y coordinate system. The point (0, 0) is found in the upper left-hand corner.



Graphics Commands

The Pen Class

The **Pen** class represents virtual pens which you use to draw on the Graphics object's surface. The Pen class is required when drawing lines, rectangles, ellipses, etc.

To construct a new Pen object, you must specify the pen's color and width in pixels. The following statements declare three Pen objects with the same color and different widths:

```
Dim thinPen, mediumPen, thickPen As Pen

thinPen = New Pen(Color.Black, 1)
mediumPen = New Pen(Color.Black, 3)
thickPen = New Pen(Color.Black, 5)
```

You can also define a Pen object using an RGB Color Code rather than a built-in color.

```
Dim myPen As Pen

myPen = Pen(Color.FromArgb(255, 0, 255), 4)
```

Drawing Lines

To draw a line, you need to specify the Pen to be used as well as a starting point (x1, y1) and an ending point (x2, y2)

```
PictureBox1.CreateGraphics.DrawLine(Pen to be Used, x1, y1, x2, y2)
```

Examples:

```
PictureBox1.CreateGraphics.DrawLine(New Pen(New Pen(Color.Blue, 2), 20, 20, 120, 120)

PictureBox1.CreateGraphics.DrawLine(New Pen(Color.FromArgb(255, 0, 255), 2), 20, 20, 80, 120)

PictureBox1.CreateGraphics.DrawLine(mediumPen, 20, 20, 120, 120)
```

Note: The last example works only if you have defined mediumPen as shown in the section above entitled "The Pen Class".

Clearing Graphics

To clear a PictureBox control of drawn content ...

```
PictureBox1.CreateGraphics.Clear(PictureBox1.BackColor)
```

Graphics Commands

Drawing Rectangles and Squares

To draw a rectangle, you need to specify the Pen to be used, the coordinates of the top left-hand corner and the desired width and height.

```
PictureBox1.CreateGraphics.DrawRectangle(New Pen(Color.Red, 4), 20, 20, 100, 100)
```

Top, left-hand corner | width | height

Note: If width and height are the same, you will create a square.

Drawing Ellipses and Circles

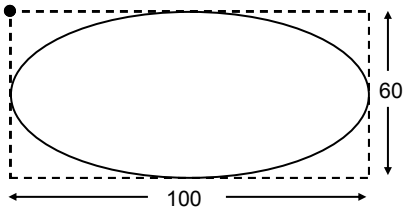
To draw an ellipse, you need to specify the Pen to be used and the bounding rectangle for the ellipse.

Note: If horizontal and vertical radius are the same, you will create a circle.

```
PictureBox1.CreateGraphics.DrawEllipse(Pens.Black, 20, 20, 100, 60)
```

upper lefhand corner of bounding rectangle | horizontal radius | vertical radius

(20, 20)



Drawing Dashed Lines

The **DashStyle** property can be used to control how a drawn line (or rectangle, or ellipse) appears.

The choices include: Dash, DashDot, DashDotDot, Dot, Solid

Example:

```
Dim myPen As New System.Drawing.Pen(System.Drawing.Color.Blue, 3)
myPen.DashStyle = System.Drawing.Drawing2D.DashStyle.DashDotDot
PictureBox1.CreateGraphics.DrawLine(myPen, 20, 20, 20, 200)
```

Graphics Commands

The Brush Class

The **Brush** class is required when drawing filled rectangles, filled ellipses, etc.

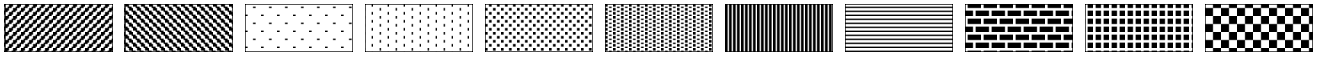
Drawing Filled Rectangles

```
PictureBox1.CreateGraphics.FillRectangle(New SolidBrush(Color.Red), 20, 20, 100, 100)
```

Drawing Filled Ellipses

```
PictureBox1.CreateGraphics.FillEllipse(New SolidBrush(Color.Blue), 20, 20, 60, 100)
```

Using HatchBrush to Fill Objects



HatchBrush allows you to fill in objects using a pattern rather than simply a solid color. Some examples of different HatchBrush styles are shown above. A list of some of the HatchBrush styles are listed below.

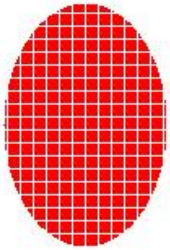
BackwardDiagonal	ForwardDiagonal	OutlinedDiamond	Percent05
Cross	Horizontal	Plaid	Percent10
DarkDownwardDiagonal	HorizontalBrick	Shingle	Percent20
DarkHorizontal	LargeCheckerBoard	SmallCheckerBoard	Percent25
DarkUpwardDiagonal	SmallCheckerBoard	SmallConfetti	Percent30
DarkVertical	LargeConfetti	SmallGrid	Percent40
DashedDownwardDiagonal	LargeGrid	SolidDiamond	Percent50
DashedHorizontal	LightDownwardDiagonal	Sphere	Percent60
DashedUpwardDiagonal	LightHorizontal	Trellis	Percent70
DashedVertical	LightUpwardDiagonal	Vertical	Percent75
DiagonalBrick	LightVertical	Wave	Percent80
DiagonalCross	Max	Weave	Percent90
Divot	Min	WideDownwardDiagonal	
DottedDiamond	NarrowHorizontal	WideUpwardDiagonal	
DottedGrid	NarrowVertical	ZigZag	

Example: To create the image shown to the right, use the code listed below.

```
myBrush = New System.Drawing.Drawing2D.HatchBrush
          (Drawing.Drawing2D.HatchStyle.Cross, Color.White, Color.Red)

PictureBox1.CreateGraphics.FillEllipse(myBrush, 70, 10, 100, 150)
```

Note: Put this all on ONE line.



Graphics Exercises - Part I

_____ _____

Name Date

RGB stand for _____, _____, _____

The individual values for the RGB() function can range from 0 to _____

List the RGB color codes for the following …

White	RGB (_____, _____, _____)
Black	RGB (_____, _____, _____)
Red	RGB (_____, _____, _____)
Green	RGB (_____, _____, _____)
Blue	RGB (_____, _____, _____)
Yellow	RGB (_____, _____, _____)
Magenta	RGB (_____, _____, _____)
Cyan	RGB (_____, _____, _____)

In a PictureBox control, the coordinates (0, 0) are found in the…

_____ bottom, left-hand corner of the screen

_____ bottom, right-hand corner of the screen

_____ top, left-hand corner of the screen

_____ top, right-hand corner of the screen

_____ center of the screen

Graphics Exercises - Part I

Name _____ Date _____

Assuming you have a PictureBox control (named PictureBox1), write the code that will …

a. Clears the contents of the PictureBox

b. Places a red dot at (100, 100)

c. Draws a black line from (100, 100) to (200, 200)

d. Draws a blue box (outline only) with top left-hand corner at (50, 100) and bottom right-hand corner at (200, 250).

e. Draws a blue box (filled in) with top left-hand corner at (50, 100) and bottom right-hand corner at (200, 250).

f. Draws a blue circle with left, top corner of (50, 100) and a radius of 75

g. Draws a blue circle with center of (200, 200) and a radius of 50

h. Draws a black line from the top left-hand corner to the bottom right-hand corner.
Your code should work regardless of the size of the PictureBox.
Hint: Use PictureBox1.Width and PictureBox1.Height