Limnor Studio – References

Part – II

2D Drawings

Contents

1 Drawing Page

This is a Form for making it easier to do 2D drawing. It is derived from Form class in the Microsoft .Net Framework. For reference of the Form class, see http://msdn.microsoft.com/en-us/library/system.windows.forms.form.aspx

1.1 Properties

1.1.1 PageAttributes - PageAttrs

Gets and sets the attributes describing the drawing page

1.1.2 DrawingLayers - DrawingLayerCollection

Gets the drawing layers

1.2 Methods

1.2.1 ShowLayerByName(String name,Boolean show)

Make drawings in a drawing layer visible or hidden

1.2.2 AddDrawing(DrawingItem draw)

Add a new drawing to the drawing layer specified by the drawing's LayerId property. If the LayerId property does not match an existing layer in the page then the drawing is added to the first layer. If draw has a matching DrawingId with an existing drawing item then the existing drawing item will be replaced by draw.

1.2.3 RemoveDrawing(DrawingItem draw)

Remove a drawing

1.2.4 GetLayerById(Guid id) DrawingLayer

Find a drawing layer by its guid

1.2.5 GetLayerByName(String name) DrawingLayer

Find a drawing layer by its name

1.2.6 GetDrawingItemById(Guid id) DrawingItem

Find a drawing by its guid

1.2.7 SaveDrawingsToFile(String filename)

Save the drawings contained in this page to a file. LoadDrawingsFromFile method can be used to load the drawings from the file

1.2.8 LoadDrawingsFromFile(String filename)

Load drawings from the file. Usually the file is generated by SaveDrawingsToFile method

1.2.9 EditDrawings() Boolean

Launch drawing editor to modify the drawings. It returns false if the editing is canceled.

1.2.10 PrintFormSnapshot(String documentName,Boolean preview)

Print this form as it appears on the screen

1.2.11 PrintDrawingLayers(String layerNames,String documentName,Boolean preview)

Print drawing layers. Specify the drawing layers using semi-column seperated layer names

1.2.12 PrintDrawings(String documentName,Boolean preview)

Print all drawings

1.2.13 ShowNextForm() Form

If this form was displayed using ShowPreviousForm from another form then calling this method will display that form.

1.2.14 ShowPreviousForm() Form

If this form was displayed by a Show action from another form then calling this method will display that form.

2 DrawingItem

This is the base class for all other 2D drawing classes. It is an abstract class providing common features for all other drawing classes.

2.1 Properties

2.1.1 Guid DrawingId

Gets a GUID uniquely identifying the drawing.

2.1.2 MouseInDrawing - Boolean

Indicates whether the mouse pointer is within the drawing object

2.1.3 Bounds - Rectangle

Gets and sets the bounds of the drawing

2.1.4 Left - Int32

Gets and sets the left position of the drawing

2.1.5 Top - Int32

Gets and sets the top position of the drawing

2.1.6 Location - Point

Gets and sets the location of the drawing

2.1.7 Center - Point
Center point of this drawing object

2.1.8 Visible - Boolean
Gets and sets a value indicating whether this drawing should be displayed

2.1.9 DrawingLayer - String
Gets the name of the drawing layer

2.1.10 Color - Color
Gets and sets the color of the drawing

2.1.11 Cursor - Cursor
Gets and sets the cursor that appears when the mouse passes over the drawing

2.1.12 ZOrder - Int32
Gets or sets an integer that specifies the order in which a series is rendered from front to back.

2.2 Methods

2.2.1 MoveByStep(Int32 dx,Int32 dy)
Move the drawing object incrementally by given distances along X and Y direction

2.2.2 MoveTo(Point p)
Move the drawing object to a specified point

2.2.3 MoveCenterTo(Point p)
Move the drawing by moving its center to the specified point

2.2.4 ResetGuid

Recreate the Guid for this drawing. After cloning a drawing, call this method of the clone to create a new DrawingId if you want the clone to be a different drawing.

2.2.5 Rotate(Point rotationCenter,Single rotationAngle)

Rotate the drawing by a specified angle around a specified point

2.2.6 DistanceFromPointToLine(Point p,Point linePoint1,Point linePoint2) Double

Calculate the distance from a point to a line

2.2.7 Rotate(Double x,Double y,Double xc,Double yc,Double angle,Double& xo,Double& yo)

Calculate a point of a rotation. (x,y) is the point to be rotated. (xc, yc) is the center of the rotation. angle is the rotation angle. (x0,y0) is the point after rotation.

2.2.8 OnDraw(Graphics g)

When override this function by a derived class do actual drawings specific to this object.

2.2.9 SetMarks

When override this function by a derived class, create markers visually design this drawing object.

2.2.10 Randomize(Rectangle bounds)

Randomly change the drawing

2.2.11 Hide

Make the drawing object invisible

2.2.12 Show

Make the drawing object visible

2.2.13 Refresh
Redraw the drawing object

2.2.14 Initialize
Initialize the drawing

2.2.15 ToString() String
Make a string representation

2.3 Events

2.3.1 MouseDown - MouseEventHandler
Occurs when the mouse is pressed down on the drawing

2.3.2 MouseMove - MouseEventHandler
Occurs when the mouse is moving within the drawing

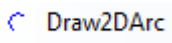
2.3.3 MouseUp - MouseEventHandler
Occurs when the mouse is released

2.3.4 MouseEnter - EventHandler
Occurs when the mouse pointer enters the drawing

2.3.5 MouseLeave - EventHandler
Occurs when the mouse pointer leaves the drawing

2.3.6 SizeChanged - EventHandler
Occurs when the size of the drawing is changed

3 DrawArc



This object represents an Arc.

3.1 Properties

3.1.1 Rectangle - Rectangle

The Rectangle defining the arc

3.1.2 StartAngle - Single

Starting angle defining the arc

3.1.3 SweepAngle - Single

Sweeping angle defining the arc

3.1.4 LineWidth - Single

Line width of the arc

3.1.5 Bounds - Rectangle

Gets and sets the bounds of the drawing

3.1.6 Left - Int32

Gets and sets the left position of the drawing

3.1.7 Top - Int32

Gets and sets the top position of the drawing

3.1.8 Center - Point

Center point of this drawing object

3.2 Methods

3.2.1 MoveByStep(Int32 dx,Int32 dy)

Move the drawing object incrementally by given distances along X and Y direction

3.2.2 MoveTo(Point p)

Move the drawing object to a specified point

3.2.3 SetMarks

When override this function by a derived class, create markers visually design this drawing object.

3.2.4 OnDraw(Graphics g)

When override this function by a derived class do actual drawings specific to this object.

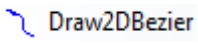
3.2.5 ToString() String

Make a string representation

3.2.6 Randomize(Rectangle bounds)

Randomly change the drawing

4 DrawBezier



This object represents a Bezier curve.

4.1 Properties

4.1.1 Bounds - Rectangle

Gets and sets the bounds of the drawing

4.1.2 Center - Point
Center point of this drawing object

4.1.3 ControlPoint1 - Point
The first control point

4.1.4 ControlPoint2 - Point
The second control point

4.1.5 EndPoint - Point
End point

4.1.6 Left - Int32
Gets and sets the left position of the drawing

4.1.7 LineWidth - Single
Curve line width

4.1.8 StartPoint - Point
Start point

4.1.9 Top - Int32
Gets and sets the top position of the drawing

4.2 Methods

4.2.1 MoveByStep(Int32 dx,Int32 dy)
Move the drawing object incrementally by given distances along X and Y direction

4.2.2 MoveTo(Point p)
Move the drawing object to a specified point

4.2.3 OnDraw(Graphics g)
When override this function by a derived class do actual drawings specific to this object.

4.2.4 Randomize(Rectangle bounds)
Randomly change the drawing

4.2.5 SetMarks
When override this function by a derived class, create markers visually design this drawing object.

4.2.6 ToString() String
Make a string representation

5 DrawCircle
Draw2DCircle This object represents a Circle.

5.1 Properties

5.1.1 Bounds - Rectangle
Gets and sets the bounds of the drawing

5.1.2 Center - Point
Center of the circle

5.1.3 CircleCenter - Point
Center of the circle

5.1.4 Fill - Boolean
True:fill the circle with the FillColor

5.1.5 FillColor - Color
The color to fill the circle if Fill is True

5.1.6 Left - Int32
Gets and sets the left position of the drawing

5.1.7 LineWidth - Single
Width of the circle line

5.1.8 Radius - Int32
Radius of the circle

5.1.9 Top - Int32
Gets and sets the top position of the drawing

5.2 Methods

5.2.1 MoveByStep(Int32 dx,Int32 dy)
Move the drawing object incrementally by given distances along X and Y direction

5.2.2 MoveCenterTo(Point p)
Move the drawing by moving its center to the sepcified point

5.2.3 MoveTo(Point p)
Move the drawing object to a specified point

5.2.4 OnDraw(Graphics g)

When override this function by a derived class do actual drawings specific to this object.

5.2.5 Randomize(Rectangle bounds)

Randomly change the drawing

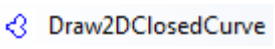
5.2.6 SetMarks

When override this function by a derived class, create markers visually design this drawing object.

5.2.7 ToString() String

Make a string representation

6 DrawClosedCurve



This object represents a Closed Curve.

6.1 Properties

6.1.1 Bounds - Rectangle

Gets and sets the bounds of the drawing

6.1.2 Center - Point

Center point of this drawing object

6.1.3 Fill - Boolean

If it is True then the closed curve is filled with a color indicated by the FillColor property

6.1.4 FillColor - Color

The color to fill the closed curve if Fill is True

6.1.5 Left - Int32
Gets and sets the left position of the drawing

6.1.6 LineWidth - Single
Width of the curve line

6.1.7 PointCount - Int32
Gets the number of the points defining the curve

6.1.8 Points – List<Point>
Points defining the closed curve

6.1.9 Tension - Single
Value greater than or equal to 0.0 that specifies the tension of the curve.

6.1.10 Top - Int32
Gets and sets the top position of the drawing

6.2 Methods

6.2.1 GetPoint(Int32 index) Point
Returns the point specified by the index. If the index is invalid then it returns (0,0).

6.2.2 Initialize
Initialize the drawing

6.2.3 MoveByStep(Int32 dx,Int32 dy)
Move the drawing object incrementally by given distances along X and Y direction

6.2.4 MoveCenterTo(Point p)
Move the drawing by moving its center to the specified point

6.2.5 MoveTo(Point p)
Move the drawing object to a specified point

6.2.6 OnDraw(Graphics g)
When override this function by a derived class do actual drawings specific to this object.

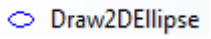
6.2.7 Randomize(Rectangle bounds)
Randomly change the drawing

6.2.8 SetMarks
When override this function by a derived class, create markers visually design this drawing object.

6.2.9 StringExp() String
Returns a string describing this drawing

6.2.10 ToString() String
Make a string representation

7 DrawEllipse



This object represents an Ellipse.

7.1 Properties

7.1.1 Bounds - Rectangle
Gets and sets the bounds of the drawing

7.1.2 Rectangle - Rectangle

Gets and sets the rectangle defining the Ellipse

7.1.3 LineWidth - Single

Gets and sets line width

7.1.4 Fill - Boolean

Gets and sets a Boolean indicating whether the background of the object is filled with a color indicated by the FillColor property

7.1.5 FillColor - Color

Gets and sets the color to fill the background of the object

7.1.6 RotateAngle - Double

Gets and sets the angle to rotate the object

7.1.7 Left - Int32

Gets and sets the left position of the drawing

7.1.8 Top - Int32

Gets and sets the top position of the drawing

7.1.9 Center - Point

Gets and sets the center position of the drawing

7.1.10 Width - Int32

Gets and sets the width of the drawing

7.1.11 Height - Int32
Gets and sets the height of the drawing

7.1.12 Right - Int32
Gets the right position of the drawing

7.1.13 Bottom - Int32
Gets the bottom position of the drawing

7.2 Methods

7.2.1 SetBoundsWidth(Int32 w)
Set the width of this drawing

7.2.2 SetBoundsHeight(Int32 h)
Set the height of this drawing

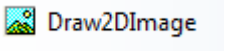
7.2.3 MoveByStep(Int32 dx,Int32 dy)
Move the drawing by step dx and dy

7.2.4 MoveTo(Point p)
Move this drawing to the point p

7.2.5 ToString() String
Make a string representation

7.2.6 Randomize(Rectangle bounds)
Randomly change the drawing

8 DrawImage



This object represents an Image.

8.1 Properties

8.1.1 AllowPanningByMouse - Boolean

When the SizeMode is Normal, the Angle is 0, and the image is larger than the size of this component, setting this property to true will allow the user to pan the image by mouse

8.1.2 Angle - Double

The angle to rotate the image

8.1.3 Bounds - Rectangle

Gets and sets the bounds of the drawing

8.1.4 Center - Point

Center point of this drawing object

8.1.5 Filename - String

Full path to the image file

8.1.6 Image - Image

Image to be displayed

8.1.7 ImageStartPoint - Point

When the SizeMode is Normal, the Angle is 0, and the image is larger than the size of this component, this property indicate the point to start drawing the image.

8.1.8 Left - Int32

Gets and sets the left position of the drawing

8.1.9 Rectangle - Rectangle

Rectangle containing the image

8.1.10 SizeMode - PictureBoxSizeMode

This property defines how to resize the picture.

```
// Summary:
//     The image is placed in the upper-left corner of the System.Windows.Forms.PictureBox.
//     The image is clipped if it is larger than the System.Windows.Forms.PictureBox
//     it is contained in.
Normal = 0,
//
// Summary:
//     The image within the System.Windows.Forms.PictureBox is stretched or shrunk
//     to fit the size of the System.Windows.Forms.PictureBox.
StretchImage = 1,
//
// Summary:
//     The System.Windows.Forms.PictureBox is sized equal to the size of the image
//     that it contains.
AutoSize = 2,
//
// Summary:
//     The image is displayed in the center if the System.Windows.Forms.PictureBox
//     is larger than the image. If the image is larger than the System.Windows.Forms.PictureBox,
//     the picture is placed in the center of the System.Windows.Forms.PictureBox
//     and the outside edges are clipped.
CenterImage = 3,
//
// Summary:
//     The size of the image is increased or decreased maintaining the size ratio.
Zoom = 4,
```

8.1.11 Top - Int32

Gets and sets the top position of the drawing

8.2 Methods

8.2.1 MoveByStep(Int32 dx,Int32 dy)

Move the drawing object incrementally by given distances along X and Y direction

8.2.2 MoveCenterTo(Point p)

Move the drawing by moving its center to the sepcified point

8.2.3 MoveTo(Point p)

Move the drawing object to a specified point

8.2.4 OnDraw(Graphics g)

When override this function by a derived class do actual drawings specific to this object.

8.2.5 PanImage(Point newStartPoint,Int32 speed)

Pan the image by moving the image to the new starting point specified by parameter newStartPoint. Parameter speed is a value greater than 0 indicating how many pixels it pans each time. The larger the value is the faster of the panning.

8.2.6 Randomize(Rectangle bounds)

Randomly change the drawing

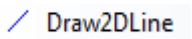
8.2.7 SetMarks

When override this function by a derived class, create markers visually design this drawing object.

8.2.8 ToString() String

Make a string representation

9 DrawLine



This object represents a Line.

9.1 Properties

9.1.1 Bounds - Rectangle
Gets and sets the bounds of the drawing

9.1.2 Center - Point
Center point of this drawing object

9.1.3 Left - Int32
Gets and sets the left position of the drawing

9.1.4 LineWidth - Single
The width of the line

9.1.5 Point1 - Point
The first point of the line

9.1.6 Point2 - Point
The second point of the line

9.1.7 Top - Int32
Gets and sets the top position of the drawing

9.2 Methods

9.2.1 Initialize
Initialize the drawing

9.2.2 MoveByStep(Int32 dx,Int32 dy)
Move the drawing object incrementally by given distances along X and Y direction

9.2.3 MoveCenterTo(Point p)

Move the drawing by moving its center to the sepcified point

9.2.4 MoveTo(Point p)

Move the drawing object to a specified point

9.2.5 OnDraw(Graphics g)

When override this function by a derived class do actual drawings specific to this object.

9.2.6 Randomize(Rectangle bounds)

Randomly change the drawing

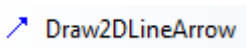
9.2.7 SetMarks

When override this function by a derived class, create markers visually design this drawing object.

9.2.8 ToString() String

Make a string representation

10 DrawLineArrow



This object represents a Line with an arrow head. It is derived from DrawLine.

10.1 Properties

10.2 Methods

10.2.1 DrawArrowHead(Graphics g,Point p1,Point p2,Single lineWidth,SolidBrush sb,Pen pen)

Draw an arrow head for a line from point p1 to p2.

10.2.2 OnDraw(Graphics g)
When override this function by a derived class do actual drawings specific to this object.

10.2.3 ToString() String
Make a string representation

11 DrawLineArrow2
Draw2DLineArrow2

This object represents a Line with arrow heads on both ends.

It is derived from DrawLineArrow

11.1 Methods

11.1.1 OnDraw(Graphics g)
When override this function by a derived class do actual drawings specific to this object.

11.1.2 ToString() String
Make a string representation

12 DrawPolygon
Draw2DPolygon

This drawing object represents a Polygon.

It is derived from DrawClosedCurve

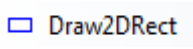
12.1.1 Methods

12.1.2 OnDraw(Graphics g)
When override this function by a derived class do actual drawings specific to this object.

12.1.3 ToString() String

Make a string representation

13 DrawRect



This object represents a Rectangle.

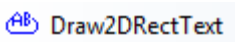
It is derived from DrawEllipse

13.1 Methods

13.1.1 ToString() String

Make a string representation

14 DrawRectText



This object represents a Rectangle with rounded corners and a text on top.

It is derived from DrawRoundRectangle

14.1 Properties

14.1.1 TextBackColor - Color

It indicates the Background color of the text when the UseFormBackgroundColor property is false

14.1.2 TextBoxCornerRadius - Single

The radius for each corner of the text box

14.1.3 TextBoxSize - Size
The rectangle size for displaying the text

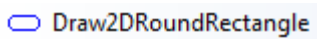
14.1.4 TextFont - Font
Font of the text

14.1.5 TextString - String
The text to be displayed

14.1.6 UseFormBackgroundColor - Boolean
If this property is true then the TextBackColor property is ignored and the form's background color is used as the text background color

14.2 Methods

14.2.1 OnDraw(Graphics g)
When override this function by a derived class do actual drawings specific to this object.

15 DrawRoundRectangle


This object represents a Rectangle with rounded corners.

It is derived from DrawingItem

15.1 Properties

15.1.1 Bottom - Int32
Gets the bottom edge of the drawing.

15.1.2 Bounds - Rectangle
Gets and sets the bounds of the drawing

15.1.3 Center - Point
Center point of this drawing object

15.1.4 CornerRadius - Single
The radius for each corner

15.1.5 Fill - Boolean
If it is True then the background of the object is filled with a color indicated by the FillColor property

15.1.6 FillColor - Color
The color to fill the background of the object

15.1.7 Height - Int32
Gets the height of the drawing.

15.1.8 Left - Int32
Gets and sets the left position of the drawing

15.1.9 LineWidth - Single
Line width

15.1.10 Rectangle - Rectangle
The rectangle defining the object

15.1.11 Right - Int32
Gets the right edge of the drawing.

15.1.12 RotateAngle - Double
The angle to rotate the object

15.1.13 Top - Int32

Gets and sets the top position of the drawing

15.1.14 Width - Int32

Gets the width of the drawing.

15.2 Methods

15.2.1 MoveByStep(Int32 dx,Int32 dy)

Move the drawing object incrementally by given distances along X and Y direction

15.2.2 MoveTo(Point p)

Move the drawing object to a specified point

15.2.3 OnDraw(Graphics g)

When override this function by a derived class do actual drawings specific to this object.

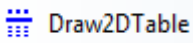
15.2.4 Randomize(Rectangle bounds)

Randomly change the drawing

15.2.5 SetMarks

When override this function by a derived class, create markers visually design this drawing object.

16 DrawTable



This component displays a table of data, usually used in a report.

It is derived from DrawRect

16.1 Properties

16.1.1 ColumnProperties - ColumnAttributesCollection

The attributes for each column

16.1.2 CurrentCellColumn - Int32

This is the column number of the cell under mouse point.

16.1.3 CurrentCellRow - Int32

This is the row number of the cell under mouse point.

16.1.4 CurrentCellValue - Object

This is the cell value of the cell under mouse point.

16.1.5 DatabaseConnection - ConnectionItem

Connection to the database

16.1.6 HasMorePage - Boolean

Gets a Boolean value indicating whether more report pages are available

16.1.7 IsConnectionReady - Boolean

Accessing this property will test connect to the database. If the connection is made then this property is True; otherwise it is False.

16.1.8 PageNumber - Int32

Gets or sets an integer indicating the current page index

16.1.9 QueryOnStart - Boolean

Indicates whether to make database query when this object is created.

16.1.10 SQL - SQLStatement

SQL statement for querying database

16.1.11 StringCount - Int32

Gets the number of the columns

16.1.12 TotalPage - Int32

The number of pages

16.2 Methods

16.2.1 GetColumnSum(String fieldName) Double

Get the sum of all rows for each column.

16.2.2 GetColumnSumOnCurrentPage(String fieldName) Double

Get the sum of all rows on the current page for the specified column.

16.2.3 HitTestCell(Control owner,Int32 x,Int32 y) Boolean

Returns True if (x,y) is within a cell. If it returns True then the cell is represented by property CurrentCellValue

16.2.4 MoveFirst() Boolean

Show the first page

16.2.5 MoveNext() Boolean

Show next report page

16.2.6 MovePrevious() Boolean

Show previous report page

16.2.7 PrintAllPages(String documentName)

Print all report pages from page 1 to the last page

16.2.8 Query

Compose the query commands and query the database for data

16.2.9 QueryWithParameterValues(Object[] values)

Execute database query with the parameter values

16.2.10 StringItem(Int32 i) String

Returns the column name by index. It returns empty if the index is not valid.

16.2.11 ToString() String

Make a string representation

16.3 Events

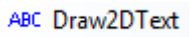
16.3.1 CellClick - EventHandler

Occurs when clicking on a cell. The following properties provide information about the cell: CurrentCellColumn, CurrentCellRow and CurrentCellValue.

16.3.2 CellEnter - EventHandler

Occurs when mouse rnters a cell. The following properties provide information about the cell: CurrentCellColumn, CurrentCellRow and CurrentCellValue.

17 DrawText



This object represents a Text.

It is derived from DrawingItem

17.1 Properties

17.1.1 Bounds - Rectangle
Gets and sets the bounds of the drawing

17.1.2 Center - Point
Center point of this drawing object

17.1.3 Left - Int32
Gets and sets the left position of the drawing

17.1.4 Location - Point
The location of the text

17.1.5 TextAngle - Double
The angle to rotate the text

17.1.6 TextContent - String
The text to be displayed

17.1.7 TextFont - Font
Font of the text

17.1.8 TextSize - Size
The width and height of the text

17.1.9 Top - Int32
Gets and sets the top position of the drawing

17.2 Methods

17.2.1 MoveByStep(Int32 dx,Int32 dy)

Move the drawing object incrementally by given distances along X and Y direction

17.2.2 MoveCenterTo(Point p)

Move the drawing by moving its center to the sepcified point

17.2.3 MoveTo(Point p)

Move the drawing object to a specified point

17.2.4 OnDraw(Graphics g)

When override this function by a derived class do actual drawings specific to this object.

17.2.5 Randomize(Rectangle bounds)

Randomly change the drawing

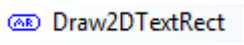
17.2.6 SetMarks

When override this function by a derived class, create markers visually design this drawing object.

17.2.7 ToString() String

Make a string representation

18 DrawTextRect



This object represents a Rectangle with rounded corners and a text in the center.

It is derived from DrawRoundRectangle

18.1 Properties

18.1.1 HideRectangle - Boolean
Indicates if the rectangle is hidden

18.1.2 TextAlign - ContentAlignment
Text alignment

18.1.3 TextColor - Color
Color of the text

18.1.4 TextFont - Font
Font of the text

18.1.5 TextString - String
The text to be displayed

18.1.6 WordWrap - Boolean
Indicates if lines are automatically word-wrapped for multiline text

18.2 Methods

18.2.1 OnDraw(Graphics g)
When override this function by a derived class do actual drawings specific to this object.

