.Net notes

Saturday, April 4, 2015 9:37 AM

Bitmap Class - System.Drawing

Bitmap (string filename) - filename: Bitmap file name and path
fileNotFoundException - specified File Not Found

Color GetPixel (int X, int Y)
X - X-coord of pixel to get
Y - Y-coord of pixel to get
returns Color struct of specified Pixel
ArgumentOutOfRangeException - X < 0 or = Width or Y < 0 or = Height
Exception - operation Failed.

Color Struct : System.Drawing
Color FromARGB (int Alpha, int Red, int Green, int Blue)
alpha - 0-255
Red - 0-255
Green - 0-255
Blue - 0-255
Returns Color Struct
ArgumentException - Alpha, Red, green or blue < 0 or > 255

RectangleF GetBounds (ref GraphicsUnit PageUnit)
PageUnit - One of GraphicsUnit values indicating unit of measurement for bounding rectangle.
returns RectangleF represents Bounds of image in specified unit.

GraphicsUnit - Display
Document
Inch
Millimeter
Pixel
Point
World

RectangleF Structure (System.Drawing)
Stores set of 4 floats represent location and size of a rectangle.

Drag and Drop

Not Enabled by Default.
Set: AllowDrop = true

Typically involves 2 parties:
- Drag source
- Drop target

Can be same App or different.
Actions Are App Specific, determined by context.
In WPF, any UIElement or ContentElement can participate
Events and Methods - DragDrop class System.Windows.DragDrop
Analogous to copy-paste.

Data Transfer
requires:
- Source object (Provides Data)
- Way to temp. store transferred Data
- Target object (receives Data)

Drag Source initiates D/D calling static DragDrop.DoDragDrop method - pass data.
Drop Target responsible for extracting data.

Drag Drop Effects
WPF Defines DragDropEffects enum.
All - Data is copied, removed from source, and scrolled in Target.
Copy - Data is copied to Target
Link - Data from Source is Linked
Move - Data from source is Moved To Target
None - Target does not accept Data
Scroll - scrolling About to start or is currently occuring in Target

Events
DragEnter
DragLeave
DragOver
Drop - occurs when object Dropped on Target

Implementing

Identify Element that will be source
Create Event handler on Source to initiate Drag/Drop ops.
Call DoDragDrop in source event handler
Identify element that will be Target
Set AllowDrop = "True" on Target Element
Create Drop event handler to process Data
Extract Data from DragEventArgs by Using GetDataPresent and Get Data
Use Data to perform Desired effect.

```
Private foo_Drag(Object sender, DragEventArgs e)
{
   if(e.Data.GetDataPresent(DataFormats.FileDrop))
   {
      String[] fileList = (String[]) e.Data.GetData(DataFormats.FileDrop);
   }
}
```