Drag and Drop

Many important applications involve **drag and drop** operations. Create an application and place a picturebox control on the form. Add a picture to the control. Run the application - you will notice that you will not be able to move the picture. You must write code that initiates a drag-and-drop operation in the source control, and you must also write code in the target control. Let’s create a simple application with only two picture boxes as follows:

Load a picture to PictureBox1. In the form_load event, write the following code:

```vbnet
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Me.PictureBox1.AllowDrop = True
    Me.PictureBox2.AllowDrop = True
End Sub
```

The above is to allow both picture boxes as the target for drag-and-drop event. By default, all control’s AllowDrop property is False.

The following subroutine handles the MouseDown event for both PictureBox controls. This event fires when the mouse is in the control’s bounds and mouse button is clicked.

```vbnet
Private Sub PictureBox_MouseDown(sender As Object, e As MouseEventArgs)
    Handles PictureBox1.MouseDown, PictureBox2.MouseDown
    Dim pic As PictureBox = sender
    If (pic.Image Is Nothing) = False Then
        DoDragDrop(pic.Image, DragDropEffects.Move)
    End If
End Sub
```

The If statement: **If (pic.Image Is Nothing) = False** checks to see if the current picture has image in it, if Yes, then perform drag-and-drop, otherwise do nothing. The DoDragDrop method actually initiates the drag-and-drop operation. We then need to write code for both the DragEnter and the DragDrop events of the target control. In the following DragEnter event, we assign values (Copy, Move, or None) to the Effect property.

```vbnet
Private Sub PictureBox_DragEnter(sender As Object, e As DragEventArgs) Handles PictureBox1.DragEnter, PictureBox2.DragEnter
    e.Effect = DragDropEffects.Move
End Sub
```
In the DragDrop event, we write the code to manage the actual data processing:

```vbnet
Private Sub PictureBox_DragDrop(sender As Object, e As DragEventArgs) Handles PictureBox1.DragDrop, PictureBox2.DragDrop
    sender.Image = e.Data.GetData(DataFormats.Bitmap)
    If sender.Name = "PictureBox1" Then
        Me.PictureBox2.Image = Nothing
    Else
        Me.PictureBox1.Image = Nothing
    End If
End Sub
```

To summarize the four steps in drag and drop code:
1. Assign True value to both the initiator’s and target’s AllowDrop property;
2. Go to the MouseDown event of the initiator, and specify
   `DoDragDrop(what-to-drag, DragDropEffects.Move/Copy...)`
3. Go to the DragEnter event of the target, and specify
   `e.Effect = DragDropEffects.Move/Copy...`
4. Go to the DragDrop event of the target to finish the code by using the
   `e.Data.GetData(DataFormats.Text/Bitmap...)`

Now let’s practice the drag-and-drop operation using other controls. Let’s design an interface as follows. The system has two listboxes, the first listbox lists all available menu items for a fast food restaurant that a customer can order from. The second listbox shows the current orders. To order an item from the menu, a customer drags the intended item from the menu listbox to the current order listbox. Multiple orders of the same item can be made. To delete an ordered item, the customer only has to drag an already-ordered item to the trash bin. A label at the bottom of the screen also records the current order’s total cost.

Write appropriate code to complete the application.