# Drawing Rubber-Band Boxes

I am sure that you have seen the rubber-band boxes, sometimes called selection boxes, that some programs use to select areas on the screen. Point with the mouse, hold down the mouse button, and move the mouse, and a rectangular box is drawn between the original and the current mouse positions. The term "rubber-band" comes from the way the box grows and shrinks as the mouse is moved. When you release the mouse button the box remains on-screen. This tip shows you how to create rubber-band boxes in Visual Basic.

Drawing a box on a form or a Picture Box control is easily done with the Line method. The problem is how to make the previous box disappear when the mouse is moved and a new box is drawn. The answer to this lies in the DrawMode property. The default setting draws a solid line, but this won't work for our purposes. Rather we will use the setting vbInvert which means that drawing operations such as the Line method draw using the inverse of the color that is already there on the form or Picture Box control. Thus, if the background is white, vbInvert draws a black line, and if the background is black a white line is drawn. Colors have their inverses too. This solves two problems:

The box will always be visible against the background.

If we draw the box a second time the colors will be re-inverted back to their original values and the box will disappear.

The approach, then, is as follows:

When the user presses the mouse button, record the current mouse coordinates. This will be the fixed corner of the box. At the same time, set a "drawing" flag to True.

When the mouse is moved, erase the previous box by drawing it again. This step is not required the first time a box is drawn but it required all subsequent times.

Also when the mouse is moved, draw a box between the starting coordinates and the new mouse coordinates.

When the mouse button is released, set the drawing flag to False.

This is demonstrated in the following code snippet. To try this out, create a Standard EXE project and place a Picture Box control on the form. Load the Picture Box with an image of your choice, and set its DrawMode property to 6 - Invert and its AutoRedraw property to True. Next, put the following variable declarations at the module-level in the form:

```
   ' True if a box is being drawn.
    Dim drawing As Boolean

    ' True if the box has just started.
    Dim first As Boolean

    ' Previous box's mouse coordinates.
    Dim oldX As Single, oldY As Single

    ' The box's starting coordinates.
    Dim startX As Single, startY As Single

Finally, put the code shown here in the mouse-related event procedures for the Picture Box
control:

    Private Sub Picture1_MouseDown(Button As Integer, _
      Shift As Integer, X As Single, Y As Single)

        drawing = True
        first = True
        startX = X
        startY = Y

    End Sub

    Private Sub Picture1_MouseMove(Button As Integer, _
      Shift As Integer, X As Single, Y As Single)

        If Not drawing Then Exit Sub
        If Not first Then
      Picture1.Line (startX, startY)-(oldX, oldY), , B
        Else
            first = False
        End If

  Picture1.Line (startX, startY)-(X, Y), , B
        oldX = X
        oldY = Y

    End Sub

    Private Sub Picture1_MouseUp(Button As Integer, _
      Shift As Integer, X As Single, Y As Single)

        drawing = False

    End Sub
```

When you run the program you will be able to draw rubber band boxes on the picture. Each box you draw remains in place, and you can draw essentially as many as you like. If you want an existing box to vanish when a new one is started, put code in the MouseUp event procedure to save the coordinates of the existing box. Then, in MouseDown, erase any existing box by drawing over it. You would want to save the coordinates of the box in any case so your program can make use of them, for example to perform processing on the selected region of the image.