

October, 2004

Advisor Answers

Using polygons on forms

Visual FoxPro 9

Q: I know how to use the Shape control to make rectangles, rounded rectangles, circles and ellipses. Is there a way to put other shapes, like stars and hexagons, on a form?

A: Prior to VFP 9, doing this was tricky, at best. You could create the outline of a shape using a number of Line controls, but you didn't have much control over the result. VFP 9 adds a new property, PolyPoints, that makes creating shapes other than rectangles and circles much easier, though you still need to know some geometry.

The new PolyPoints property accepts the name of an array. The array contains the points that determine the shape to be displayed. Although you can use an array of any dimensions, it's easiest to use a two-column array; in that case, the first column provides the x-position for each point, while the second column supplies the y-position.

You can use any array that's in scope when you run the form, but your best bet is to add the array as a property of the control or the form. To make things easy, subclass the Shape class and add the array property. Then base all your unusual shapes on this subclass. This month's Professional Resource CD contains a class library called Shapes.VCX; it includes shpPolygon, which has a custom aPoints property (initially defined as an array of one element). The PolyPoints property is set to This.aPoints.

A shape with PolyPoints specified can be displayed at design-time. However, in the beta version I'm working with, that works only if the array is in scope. Worse, once you set PolyPoints to an array that's not available, the outline of the shape disappears except when the shape is selected. To find it, use the object dropdown in the Property Sheet.

You specify the points as percentages of the shape's designated Height and Width. (The Height and Width of shpPolygon have both been set to 100.) That is, the values of the points should be between 0 and 100. Figure 1 shows a form (UseShapes.SCX on this month's PRD) containing two shapes, one set up as a 6-pointed star, the other as a

hexagon. To specify the star requires 12 points, so the Init method for the star contains this code:

```
DIMENSION This.aPoints[12,2]

This.aPoints[1,1] = 50
This.aPoints[1,2] = 0
This.aPoints[2,1] = 66
This.aPoints[2,2] = 25
This.aPoints[3,1] = 100
This.aPoints[3,2] = 25
This.aPoints[4,1] = 84
This.aPoints[4,2] = 50
This.aPoints[5,1] = 100
This.aPoints[5,2] = 75
This.aPoints[6,1] = 66
This.aPoints[6,2] = 75
This.aPoints[7,1] = 50
This.aPoints[7,2] = 100
This.aPoints[8,1] = 33
This.aPoints[8,2] = 75
This.aPoints[9,1] = 0
This.aPoints[9,2] = 75
This.aPoints[10,1] = 16
This.aPoints[10,2] = 50
This.aPoints[11,1] = 0
This.aPoints[11,2] = 25
This.aPoints[12,1] = 33
This.aPoints[12,2] = 25
```

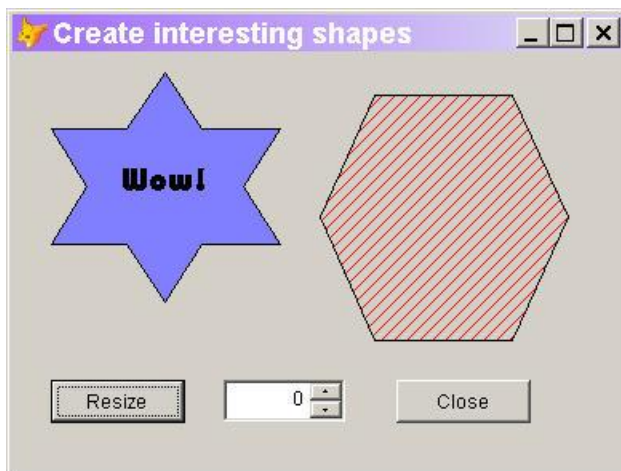


Figure 1. Polygons without pain—The new PolyPoints property of Shapes lets you create all kinds of polygons.

In fact, the star on the form is based on the `shpStar` class in the Shapes class library, while the hexagon is based on `shpHexagon`, and the code shown above is in the `Init` method of `shpStar`.

Because the points are specified as percentages, changing the size of the Shape object changes the shape. The UseShapes form includes a Resize button that has this code in its Click method:

```
ThisForm.shpStar.Height = ThisForm.shpStar.Height/2  
ThisForm.shpHexagon.Width = ThisForm.shpHexagon.Width + 10  
ThisForm.lblWow.Refresh()
```

Figure 2 shows the result the first time you click the Resize button. (The label contains code that keeps it centered on the star.)

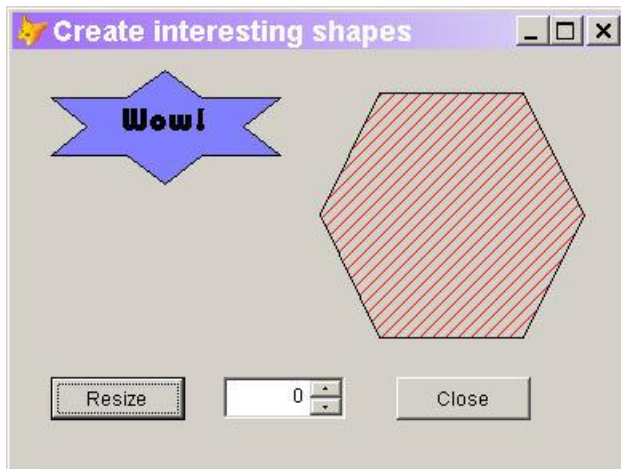


Figure 2. Relative size—The points you specify for a polygon are percentages of the total space occupied by the Shape control. When you resize the Shape, the polygon's appearance changes.

Shapes (and labels) have another new property, Rotation, that lets you change their orientation. Rotation accepts a value between 0 and 360. Rotation goes counterclockwise. The spinner on the UseShapes form controls the rotation of all three objects, the star, the label and the hexagon. Figure 3 shows 45-degree rotation.

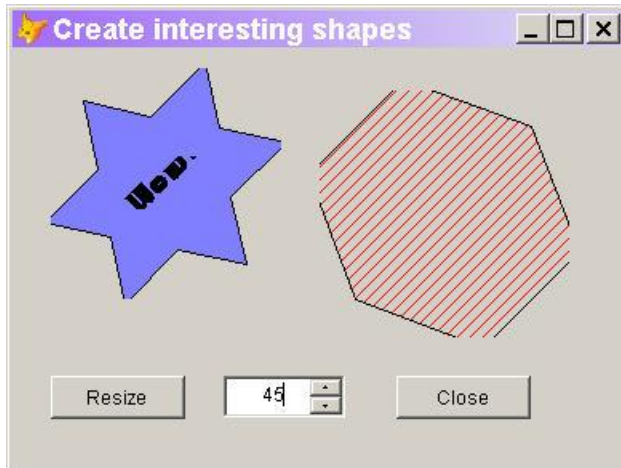


Figure 3. Rotation objects—The new Rotation property lets you change the orientation of Shapes, Lines, and Labels.

The Line object also has PolyPoints and Rotation properties in VFP 9, allowing you to create lines other than straight lines. In addition, the LineSlant property has been enhanced. Specify "S" to signify that the specified points should be connected with a curve (a Bézier curve) rather than a series of straight lines.

The ability to create shapes other than rectangles and circles, as well as to rotate the objects we create, gives us lots of new possibilities in form design.

–Tamar