

The Power of Graphics: Creating Interesting and Functional Compound Shapes

by Chris Freeman, Sr. Software Developer

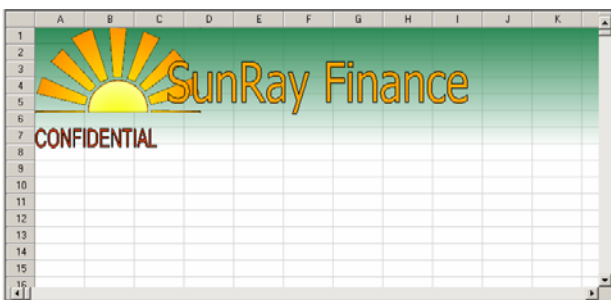
Introduction

FarPoint Spread for Windows Forms version 2.0 has a new shapes layer to allow you to add drawings and graphics that float on a separate layer on the spreadsheet. This article explains a particularly useful technique of creating shapes within shapes to customize the appearance of the Spread component.

Background

Each sheet can have its own shape layer. For instance, in a spreadsheet you could create a shape like a “star” or other graphic that could highlight data or point the user to some aspect of working with the sheet. You could then proceed to customize aspects of that star or graphic from size and background color to rotation angle or gradient. The shapes are available in code (each shape being a separate class in the DrawingSpace namespace) or from the Drawing toolbar in the Spread Designer. One of the inherent capabilities of all shape classes is the ability to have other shape objects embedded in them. This is a great way of making *compound shapes*.

In particular, this article explains how to create a compound shape that serves as a watermark, which can identify information or corporate identity subtly (without obstructing or obscuring the spreadsheet) as shown here.



Example of Compound Shape

To create this compound shape, we define a watermark shape class and set the properties to customize this appearance. In this custom watermark shape class, we have a background gradient that fades from a specified color to transparency; we have an embedded shape that contains an image of the company logo; there is a separate embedded text shape for the company name; and there is a text shape for additional information such as Web site address or company motto or security classification.

Deriving the Custom Class

We start by deriving a class from the standard RectangleShape class. This gives us a rectangular palette in which to place our embedded shapes. You could just as easily use other shapes such as ellipses, or polygons, but your embedded shapes may be “clipped” (because embedded shapes do not exceed the boundaries of the parent shape). Here is the code:

VB:

```
Public Class CompanyWatermark
    Inherits FarPoint.Win.Spread.DrawingSpace.
    RectangleShape
```

C#:

```
public class CompanyWatermark :
    FarPoint.Win.Spread.DrawingSpace.
    RectangleShape
```

Setting Properties

We then create properties for accessing the internal shapes. One of the benefits of creating a compound shape is that we can hide many of the unnecessary properties of the embedded shapes. We can expose, through custom properties, only the pertinent information. The first property is the “CompanyColor” property. This is the main color in the background gradient, which gradually fades

to transparency in the main shape. It is also alpha-blended so as not to fully obscure the sheet below. With the gradient class, other properties such as Style, which determines the gradient direction and type of gradient, can also be set. For this example, we use a Style of GradientStyle.TopDown.

The next property is an image property called “CompanyLogo”. This property sets the graphic to be displayed in the embedded logo shape. For this example we will lock the logo shape to the upper left corner of the watermark. You could very simply extend this with properties to allow alignment of the logo.

Embedded Text Shapes

The most important part of our watermark is the company name. We have created an embedded TextShape object to represent the company name. This has been exposed on the watermark shape as the “CompanyName” property, which is a string. There are many customizable features of the TextShape class that can be exposed such as Font or ForeColor, though we do not expose them here. Also, for simplicity of this sample, the company name shape is locked to be centered in the watermark shape.

And, finally, we have a secondary TextShape object that holds some additional text information in the watermark. This could be a company motto, slogan, Web site address or even words like “CONFIDENTIAL” or “Copyright 2005”. We anchor this text to the bottom left corner of the watermark shape for this example. Many additional properties can be set on this embedded shape. We keep this example simple enough to expose only the ones of interest. The text for this shape is exposed as the “CompanyText” property.

Results

That’s about it. In the figure above, you see the full implementation of the CompanyWatermark class with the other functions that were described. For the pictured watermark, we chose “System.Drawing.Colors.SeaGreen” as the “CompanyColor”, a custom bitmap as the “CompanyLogo”, “SunRay Finance” as the

“CompanyName”, and “CONFIDENTIAL” as the “CompanyText”.

Additional Comments

Here are some additional comments about the code in the online example. Since Spread can scroll the viewport panes, we override the TopChange and LeftChange events of Spread to move the watermark with the current sheet so that it always appears across the top and is as wide as the spreadsheet control. We also override the Top and Left properties of the watermark class to move the embedded objects to always be visible within the watermark.

One of the features of the TextShape class is to always stretch the text to the dimensions of the shape using the Font associated with it. We have changed this behavior in the example with the MeasureText method. We measure the text string using the control’s Graphics object and current Font and then set the TextShape dimensions accordingly. This guarantees that the shape keeps the size locked to the specified font and font size.

This class provides an excellent start for creating your own compound shapes in Spread. Now, you can let your imagination shape your future applications and use FarPoint Spread to make it a reality.

Conclusion

You can copy the sample code provided online for this article and experiment with it. The complete source for this class in Visual Basic .NET and C# is available at [http://www.fpoint.com/\[path here\]/compound_shape](http://www.fpoint.com/[path here]/compound_shape).

While not everyone uses shapes in their application, the object-oriented development of shapes in Spread allow you the most flexibility when you need shapes. For more information, refer to product documentation or contact FarPoint. Find us online at www.fpoint.com.