Dynamic Charts

Source http://www.dotnetspider.com/kb/Article1622.aspx

Introduction

Charts and/or graphs can be very useful in both Windows-based and Web-based applications for pictorially displaying information and statistics. There are tools such as Crystal Reports that can be used to generate reports. There also area number of charting tools such as Chart FX, Dundas Chart, and TeeChart that will allow you to create robust charts and graphs for display within your applications. However, it isn't always feasible to purchase licenses for these tools, especially if the application is relatively simple or is constrained by a low budget. Lucky for those with a shoe string budget, the Microsoft .NET Framework includes the classes in the System.Drawing namespace.

How to Dynamically Create a Chart or Graph?

Creating a chart or graph using the Microsoft .NET Framework is easier than one might expect. The degree of difficulty of the chart is based on the complexity of the information you are trying to display and the shape you are trying to use to display it. The basic steps of creating a chart or graph are as follows:

1. Create a new image in the desired format.
2. Draw the desired shape using the drawing classes in the System.Drawing namespace.
3. Save the image to a stream (memory, file, and so forth) so that it can be used.

Creating a Pie Chart Sample Code

The following sample class demonstrates how to create a dynamic image that contains a pie chart. It takes the background color, width, height, and an array of decimal values as the input. The PieChart object sums up the values and draws the pie chart for each value as a percent of the total. Each section of the chart will be drawn in one of the ten colors that have been set up.

```
using System;
using System.Drawing;
using System.Drawing.Imaging;

namespace DotNetSpider.SampleCode
{
 ///
 /// Draw a pie chart using the given information.
 ///
 public class PieChart
 {
   public Bitmap Draw(Color bgColor, int width, int height,
          decimal[] vals)
 {
   // Create a new image and erase the background
   Bitmap bitmap = new Bitmap(width,height,
                              PixelFormat.Format32bppArgb);
   Graphics graphics = Graphics.FromImage(bitmap);
   SolidBrush brush = new SolidBrush(bgColor);
   graphics.FillRectangle(brush, 0, 0, width, height);
   brush.Dispose();

   // Create brushes for coloring the pie chart
   SolidBrush[] brushes = new SolidBrush[10];
   brushes[0] = new SolidBrush(Color.Yellow);
   brushes[1] = new SolidBrush(Color.Green);
   brushes[2] = new SolidBrush(Color.Blue);
   brushes[3] = new SolidBrush(Color.Cyan);
   brushes[4] = new SolidBrush(Color.Magenta);
   brushes[5] = new SolidBrush(Color.Red);
   brushes[6] = new SolidBrush(Color.Black);
   brushes[7] = new SolidBrush(Color.Gray);
   brushes[8] = new SolidBrush(Color.Maroon);
   brushes[9] = new SolidBrush(Color.LightBlue);

   // Sum the inputs to get the total
   decimal total = 0.0m;
   foreach( decimal val in vals )
     total += val;

   // Draw the pie chart
   float start = 0.0f;
   float end = 0.0f;
   decimal current = 0.0m;
   for( int i = 0; i < vals.Length; i++ )
   {
     current += vals[i];
     start = end;
     end = (float) (current / total) * 360.0f;
     graphics.FillPie(brushes[i % 10], 0.0f, 0.0f, width,
                      height, start, end - start);
   }

   // Clean up the brush resources
   foreach( SolidBrush cleanBrush in brushes )
     cleanBrush.Dispose();

   return bitmap;
   }
 }
}
```

***~~~ End of Article ~~~***