Chartlet 参考手册

一. 前言

一个属性设置，一个数据源绑定，展现给你一幅漂亮的图表。这就是 Chartlet，2 句代码搞定一幅漂亮的图表，让我们的编程更简洁，让我的图表更漂亮。

Chartlet是一个为ASP.NET设计的免费的图表控件，用来在ASP.NET页面上将数据库的数据动态转换成统计图表显示。如今成熟的.net图表控件有很多，但是大多数都是国外的，而且几乎都是收费的，只适合大型的商业应用，不适合一般的应用开发，而Chartlet正是为了满足大家一般应用而开发的免费的控件，Chartlet提供的图表应该可以满足大家的基本图表要求，而且我们一直在更新它，使它变得更加强大和完善。

二. 快速上手

Chartlet的使用分简单模式和复杂模式两种。简单模式下你只需要设置一个属性，绑定一个数据源就能搞定一个漂亮的图标；复杂模式下你可以选择所有属性，这种方式的灵活性非常高，你可以控制图标显示效果的每一个细节。

如果你的使用要求不是很苛刻，我们建议你使用简单模式，这样不会占据你宝贵的开发时间，让你专注于你的项目开发，而不是专研控件的使用。而且对于一般的开发人员来说选择漂亮的配色，配置图形的显示边框，投影效果，水晶高光效果也不是一个很容易的事情。我们在简单模式下提供的显示效果都是我们精心挑选出来的，应该能满足一般的需求。当然你有时间可以了解一下 Chartlet 的所有属性对你是极其有帮助的，这样会给你有足够的灵活度，在简单模式下无法达到的效果你可以手动选择相应的属性来实现，你可以先选择一个简单模式下的显示效果，然后再手动设置你要修改的属性来进行微调。(简单模式实际上就是把漂亮效果实现的众多的属性设置都打包，你只需要选来用)

一个属性是指 Chartlet.AppearanceStyle，它的取值来自 FanG.Chartlet.AppearanceStyles 枚举类型，你在设置的时候只需要选择一个枚举项就好了。

```
Chartlet1.AppearanceStyle=FanG.Chartlet.AppearanceStyles.Bar_2D_StarryNight_FlatCrystal_Glow_TextureBorder;
```

一个数据源的绑定是指调用Chartlet. BindChartData(SqlDataSource DataSource)方法，将ASP.NET网页上的一个SqlDataSource作为数据源作为参数传入就好了。

上面的属性和方法可以在下文里找到详细的介绍，如果你下载控件是带有演示页面的，那么在 index.aspx 里面有详细的演示，你会看到将数据库中的数据显示成统计图表真的就是 2 句代码搞定的，而且真的也非常漂亮。

三. Chartlet 参考

下面介绍一下 Chartlet 的属性、方法和一些使用技巧，让大家能够很快地上手。

命名空间：FanG

类：Chartlet

一. 公用枚举

1. ChartTypes 图表类型

Bar 柱状图

Line 折线图

2. LineConnectionTypes 折线图连接点样式

Round 圆形

Square 方形
None 没有任何样式

3. Direction 方向枚举
LeftRight 从左到右
TopBottom 从上到下
RightLeft 从右到左
BottomTop 从下到上

4. ColorStyles 颜色风格
None 不使用任何风格
Breeze 使用 明快 颜色数组
Aurora 使用 绚烂 颜色数组
StarryNight 使用 厚重 颜色数组

5. AppearanceStyles 外观风格（简单模式下你只需要设置这一个属性）
None_None_None_None_None_None,
Bar_2D_Breeze_NoCrystal_NoGlow_NoBorder,
Bar_2D_Breeze_NoCrystal_Glow_NoBorder,
Bar_2D_Breeze_NoCrystal_Glow_WhiteBorder,
Bar_2D_Breeze_FlatCrystal_NoGlow_NoBorder,
Bar_2D_Breeze_FlatCrystal_Glow_NoBorder,
Bar_2D_Breeze_FlatCrystal_Glow_WhiteBorder,
Bar_2D_Breeze_FlatCrystal_Glow_TextureBorder,
Bar_2D_Aurora_NoCrystal_NoGlow_NoBorder,
Bar_2D_Aurora_NoCrystal_Glow_NoBorder,
Bar_2D_Aurora_NoCrystal_Glow_WhiteBorder,
Bar_2D_Aurora_FlatCrystal_NoGlow_NoBorder,
Bar_2D_Aurora_FlatCrystal_Glow_NoBorder,
Bar_2D_Aurora_FlatCrystal_Glow_WhiteBorder,
Bar_2D_Aurora_FlatCrystal_Glow_TextureBorder,
Bar_2D_Aurora_GlassCrystal_NoGlow_NoBorder,
Bar_2D_Aurora_GlassCrystal_Glow_NoBorder,
Bar_2D_Aurora_GlassCrystal_Glow_WhiteBorder,
Bar_2D_StarryNight_FlatCrystal_Glow_NoBorder,
Bar_2D_StarryNight_FlatCrystal_Glow_WhiteBorder,
Bar_2D_StarryNight_FlatCrystal_Glow_TextureBorder,
Bar_2D_StarryNight_GlassCrystal_NoGlow_NoBorder,
Line_2D_StarryNight_ThickRound_NoGlow_NoBorder,
Line_2D_StarryNight_ThickRound_Glow_NoBorder,
Line_2D_StarryNight_ThickSquare_NoGlow_NoBorder,
Line_2D_StarryNight_ThickSquare_Glow_NoBorder,
Line_2D_StarryNight_ThinRound_NoGlow_NoBorder,
Line_2D_StarryNight_ThinRound_Glow_NoBorder,
Line_2D_StarryNight_ThinSquare_NoGlow_NoBorder,
Line_2D_StarryNight_ThinSquare_Glow_NoBorder,
Pie_2D_Breeze_NoCrystal_NoGlow_NoBorder,

Pie_2D_Breeze_NoCrystal_NoGlow_WhiteBorder,
Pie_2D_Breeze_NoCrystal_Glow_WhiteBorder,
Pie_2D_Breeze_FlatCrystal_NoGlow_NoBorder,
Pie_2D_Breeze_FlatCrystal_Glow_WhiteBorder,
Pie_2D_Breeze_GlassCrystal_NoGlow_NoBorder,
Pie_2D_Breeze_GlassCrystal_Glow_WhiteBorder,
Pie_2D_Aurora_NoCrystal_NoGlow_NoBorder,
Pie_2D_Aurora_NoCrystal_NoGlow_WhiteBorder,
Pie_2D_Aurora_NoCrystal_Glow_WhiteBorder,
Pie_2D_Aurora_FlatCrystal_NoGlow_NoBorder,
Pie_2D_Aurora_FlatCrystal_Glow_WhiteBorder,
Pie_2D_Aurora_GlassCrystal_NoGlow_NoBorder,
Pie_2D_Aurora_GlassCrystal_Glow_WhiteBorder,
Pie_2D_StarryNight_NoCrystal_NoGlow_NoBorder,
Pie_2D_StarryNight_NoCrystal_NoGlow_WhiteBorder,
Pie_2D_StarryNight_NoCrystal_Glow_WhiteBorder,
Pie_2D_StarryNight_FlatCrystal_NoGlow_NoBorder,
Pie_2D_StarryNight_FlatCrystal_Glow_WhiteBorder,
Pie_2D_StarryNight_GlassCrystal_NoGlow_NoBorder,
Pie_2D_StarryNight_GlassCrystal_Glow_WhiteBorder,

注：图表类型_维数效果_颜色样式_水晶效果_投影效果_边框效果

二. 属性：

1. ChartTitle　string 类型，代表图表的名称，将显示在图表的上方
2. ChartType　枚举 ChartTypes 类型，代表图表的类型
3. XUnit　string 类型，代表横坐标的单位，将显示在横坐标的右边
4. YUnit　string 类型，代表纵坐标的单位，将显示在纵坐标的上方
5. RootPath　string 类型，设计模式下图片暂存路径，一般不需要设置
6. GroupSize　int 类型，BarChart 的组的大小，LineChart 的条数
7. MaxValueY　double 类型，纵坐标的最大值，用来标准化纵坐标的数值，和统一调整所有图形的高度

<下面两个属性仅用于柱状图>

8. RoundRectangle　bool 类型，是否使柱状图使用圆角矩形
9. RoundRadius　int 类型，圆角半径
10. LineWidth　int 类型，折线图的线宽，其他图形的边框宽度
11. LineTextureEnable　bool 类型，是否使用纹理边框
12. LineTextureStyle　HatchStyle 类型，纹理风格

<下面三个属性仅用于折线图>

13. LineConnectionType　LineConnectionTypes 类型，折线连接点的类型
14. LineConnectionRadius　int 类型，折线链接点的半径或者宽度
15. FillColor1　Color 类型，填充颜色 1
16. FillColor2　Color 类型，填充颜色 2
17. FillColor3　Color 类型，填充颜色 3
18. StrokeColor1　Color 类型，边框颜色 1
19. StrokeColor2　Color 类型，边框颜色 2

20. StrokeColor3　Color 类型，边框颜色 3
21. FillColorStyle　ColorStyles 类型，填充颜色使用哪一组颜色
22. StrokeColorStyle　ColorStyles 类型，边框(线)颜色使用哪一组颜色
23. FillShiftStep　int 类型，颜色数组循环偏移量，从颜色数组中选择不同的颜色
24. StrokeShiftStep　int 类型，颜色数组循环偏移量，从颜色数组中选择不同的颜色

<下面是投影的相关属性> **Shadow**

25. Enable　bool 类型，是否使用投影
26. Radius　int 类型，阴影半径
27. Distance　intl 类型，阴影偏移距离
28. Angle　float 类型，阴影偏移角度
29. Alpha　byte 类型，投影颜色的透明度 0 ~ 255
30. Color　Color 类型，阴影的颜色
31. Hollow　bool 类型，是否使用空心投影(只对图形边框进行投影)

注：在页面(xx.aspx)上使用上面的属性时写法如：Shadow-Enable="true"
在代码(xx.cs)里使用上面的属性时写法如：Shadow.Enable=true;
似乎在 Visual web developer 的设计视图中，在属性窗口中修改上述属性不起作用

<下面是水晶效果的相关属性> **Crystal**

32. Enable　bool 类型，是否使用水晶效果
33. CoverFull　bool 类型，　水晶效果覆盖图形的全部还是一般
34. Contraction　int 类型，　水晶效果收缩像素
35. Direction　Direction 枚举，　水晶效果放射的方向

注：Crystal 属性的使用方法和 Shadow 一样

三. 方法

1. BindChartData(SqlDataSource DataSource)

此方法将 SqlDataSouce 中的数据绑定到 Chartlet 控件，需要在页面上放置一个 SqlDataSource 控件，设置好数据源，然后在后台代码中(xx.cs)中调用一下这个方法，就可将数据库中的数据绑定到 Chartlet 了。

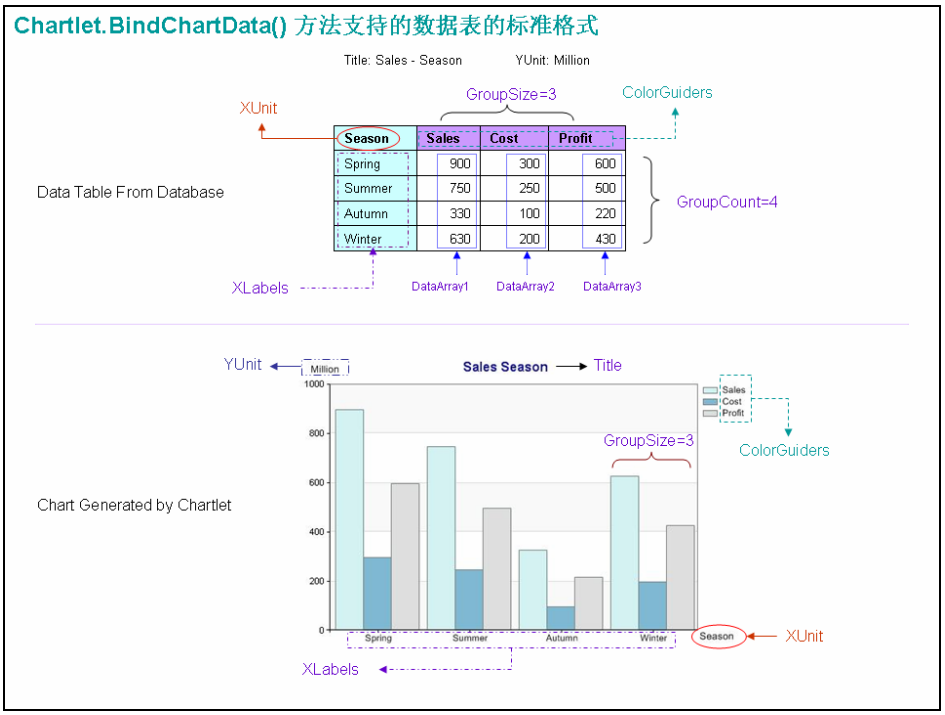
注：需要将数据库的中数据在 SQL 里组织成标准的数据表格式，格式见附图

2. InitializeData(ArrayList[ ] ChartData, ArrayList XLabel, ArrayList ColorGuider)

此方法用 ArrayList 绑定数据，需要你事先在代码里把数据库里查询来的数据组织成 ArrayList，再把 ArrayList 当作参数传入 InitializeData()方法绑定到 Chartlet 控件，这个方法比较繁琐，但是却很灵活，当你的数据库结构比较复杂，无法组织成标准的数据表结构时，就可以使用这个方法

ArrayList[ ] ChartData　ArrayList 数据
ArrayList XLabel　横坐标标识数组
ArrayList ColorGuider　图例说明

具体使用见 Test.aspx 实例



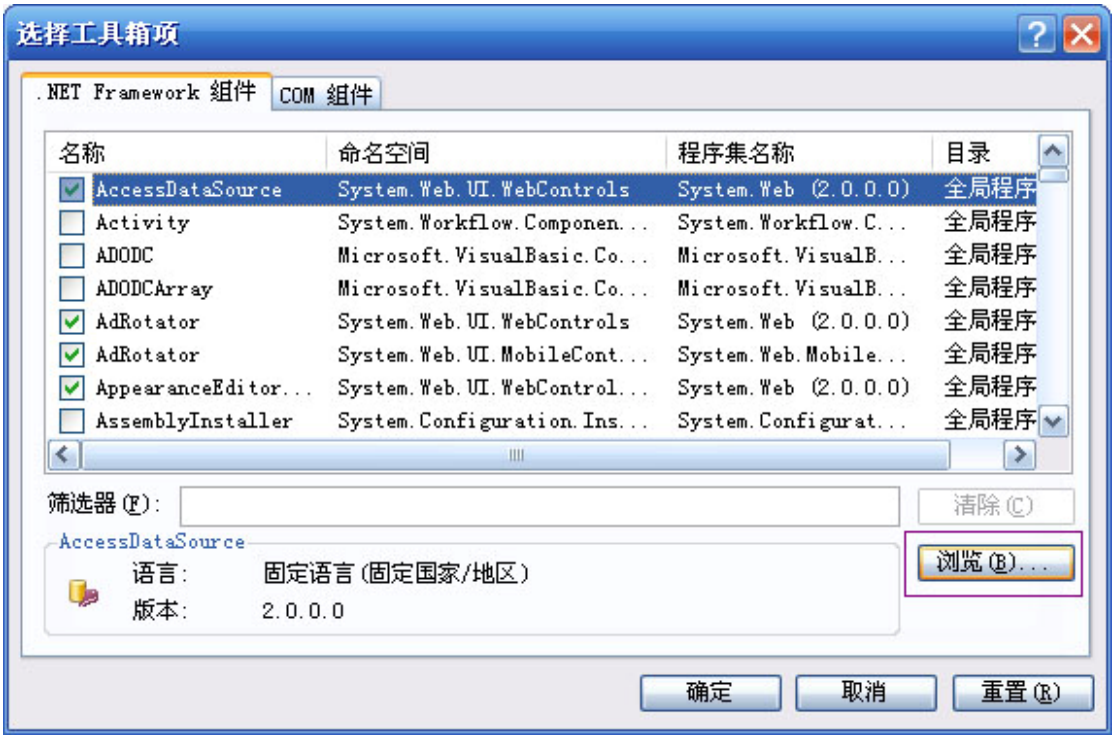
Chartlet.BindChartData()标准数据表格式

四．使用帮助

一．将 Chartlet 控件加入 Visual Web Developer(Visual Studio)工具箱。

步骤：在工具箱空白位置右击 -> 选择项 -> 浏览　找到你下载的 Chartlet.dll，然后确定

添加工具箱



添加工具箱

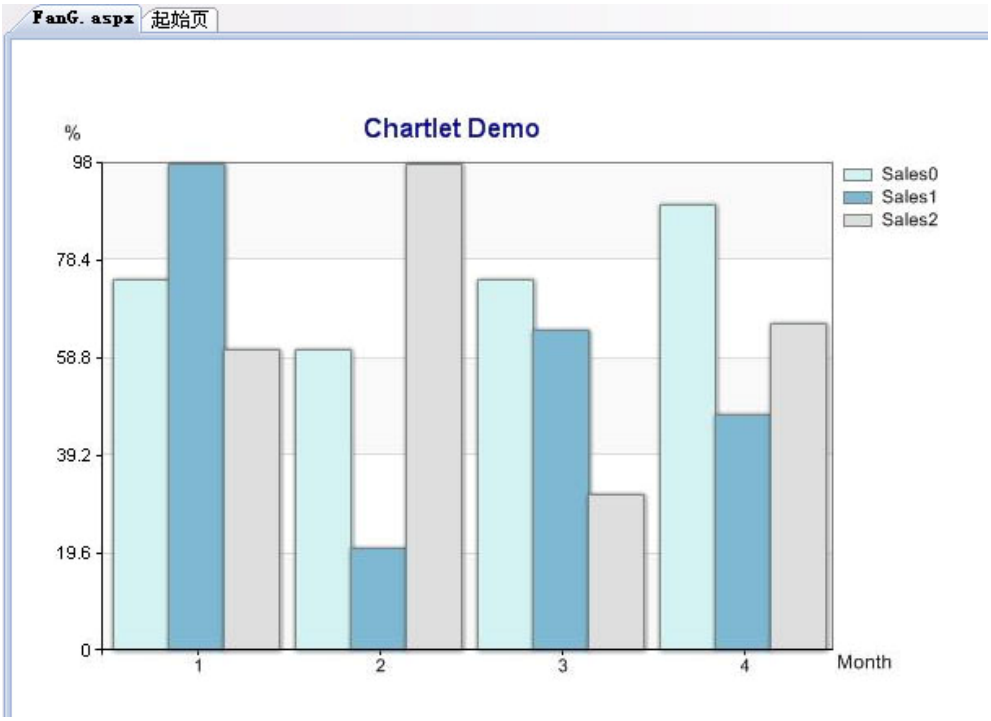
成功添加到工具箱

（如图所示，依照上面三个步骤就可以将 Chartlet 控件加入到工具箱了，Chartlet 图标会显示在工具箱内，当然你下载的 FanG.ico 如果被删除了，就无法显示图标了，会变成默认的齿轮图标）

二．在 ASP.NET 页面上使用 Chartlet 控件

步骤：

1. 直接从工具箱中将控件拖到你的 ASP.NET 页面上，页面上就会显示默认的图表



将 Chartlet 控件拖到页面上

（注：在 Visual Studio 设计视图里显示的数据是 Chartlet 里的默认数据，只提供设计时视觉参考，让你在设置了控件属性后能有一个感官认识，运行时数据是你下面要在后台代码里从数据库加载的数据）

此时你转入当前页面（这里是 FanG.aspx）的代码视图，会看到页面上多了两行代码。

```
<%@ Register assembly="Chartlet" namespace="FanG" tagprefix="cc1" %>
...
  <cc1:Chartlet ID="Chartlet1" runat="server" />
```

这两行代码都是 Visual Studio 自动帮你生成的，第一句是在页面上注册 Chartlet 控件，第二句是在页面上使用 Chartlet 控件，如果你没有将 Chartlet.dll 复制到你的网站的 Bin 目录中，它会提示你拷贝，确定拷贝就可以了。

2. 设置 Chartlet 控件的显示属性

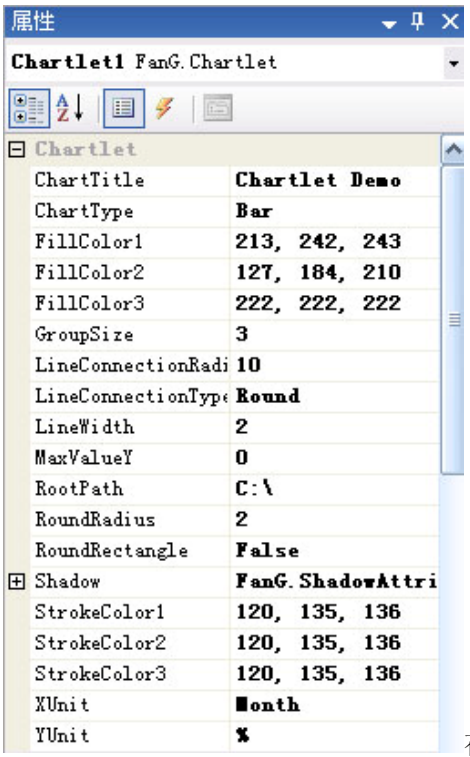
你可以在属性窗口中设置，也可以在代码视图里设置，还可以在后台代码里动态设置

代码视图（FanG.aspx）中设置

```
<cc1:Chartlet ID="Chartlet1" runat="server" ChartTitle="Jesy Sales" Shadow-Enable="true" />
```

后台代码(FanG.aspx.cs)中设置(这里设置的属性只有在页面运行时(浏览器中)才能起作用)

```
protected void Page_Load(object sender, EventArgs e){
        Chartlet1.MaxValueY = 1000;
        Chartlet1.Shadow.Radius = 5;
    }
```



在属性窗口中设置

（注：在页面上（FanG.aspx）设置的属性(包括代码中和属性窗口中)都会在设计视图中实时显示设置效果，其中在属性窗口中设置 Shadow 的各个属性时似乎不起作用，如果要设置请转入代码视图）

<下面就是从数据库中加载数据的工作了>

3. 在页面上放置 SqlDataSource

在页面上（FanG.aspx）放置一个 SqlDataSource，并设置好数据源，连接数据库，选择相应的表和需要的字段（数据格式一定要组织称标准的格式，参看第一节的附图）

4. 在代码(FanG.aspx.cs)里将 SqlDataSource 作为参数传给 Chartlet.BindChartData()方法

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$
ConnectionStrings:FanG Sample %>"
            ProviderName="<%$ ConnectionStrings:FanG Sample.ProviderName %>"
SelectCommand="Select Season,Profit from Sales where User='Jesy'">
        </asp:SqlDataSource>

protected void Page_Load(object sender, EventArgs e)
    {
        Chartlet1.MaxValueY = 1000;
        Chartlet1.Shadow.Radius = 5;
        Chartlet1.BindChartData(SqlDataSource1);
    }
```

(好了，在浏览器中查看你的网页吧，你就可以看到漂亮的图表了，我们也把这个示例(FanG.aspx)放在了下载包中，如果你使用时遇到问题可以作为参考，当然你要修改 Web.config 中的连接字符串)

下面把刚才那个页面 FanG.aspx 运行结果贴出来

FanG.aspx 运行截图

五．致谢

最后，非常感谢大家使用 Chartlet 控件，希望我们的控件能给你的开发带来帮助。
如果有任何问题请不要忘了联系我们，我们将非常感激。