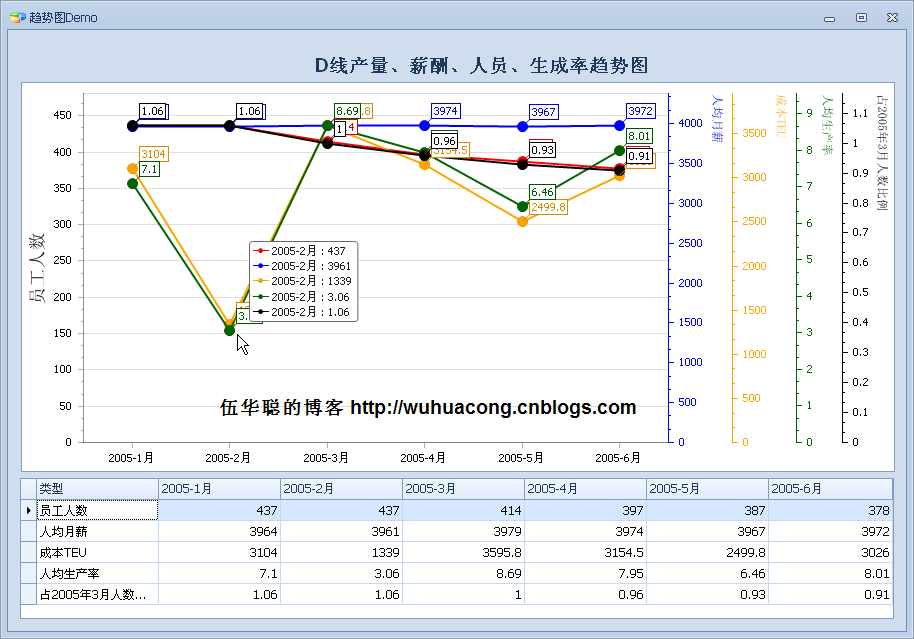
[**DevExpress控件使用之多重坐标图形的绘制**](http://www.cnblogs.com/wuhuacong/archive/2013/05/13/3076142.html)

有时候，基于对一些年份、月份的统计，需要集成多个数值指标进行分析，因此就需要把多种数据放到一个图形里面展现，也成为多重坐标轴，多重坐标轴可以是多个X轴，也可以是Y轴，它们的处理方式类似。本文通过一个例子对这个方面进行介绍，希望给大家有一个很好的参考。

首先我们先来看一个图形例子，我们可以从里面图形的右边看到有多个Y轴，一个Y轴代表一个指标分析，X轴为月份。



上图是采用了DevExpress的ChartControl图表控件来实现的，这个控件提供了SecondaryAxisY对象来处理多重坐标的问题。

**1、准备数据并绑定**

首先，，拖动ChartControl控件到Form界面上，然后设计好布局。

下面为了测试准备了几项数据，绑定在下面列表GridControl对象里面，然后把数据绑定到图表对象里面，如下代码。具体处理的时候，我们从数据库获取对应指标的数据即可实现动态绑定。

[复制代码](javascript:void(0);)

/// <summary>

/// 准备数据内容

/// </summary>

/// <returns></returns>

private DataTable CreateData()

{

DataTable dt = new DataTable();

dt.Columns.Add(new DataColumn("类型"));

dt.Columns.Add(new DataColumn("2005-1月", typeof(decimal)));

dt.Columns.Add(new DataColumn("2005-2月", typeof(decimal)));

dt.Columns.Add(new DataColumn("2005-3月", typeof(decimal)));

dt.Columns.Add(new DataColumn("2005-4月", typeof(decimal)));

dt.Columns.Add(new DataColumn("2005-5月", typeof(decimal)));

dt.Columns.Add(new DataColumn("2005-6月", typeof(decimal)));

dt.Rows.Add(new object[] { "员工人数", 437, 437, 414, 397, 387, 378 });

dt.Rows.Add(new object[] { "人均月薪", 3964, 3961, 3979, 3974, 3967, 3972 });

dt.Rows.Add(new object[] { "成本TEU", 3104, 1339, 3595.8, 3154.5, 2499.8, 3026 });

dt.Rows.Add(new object[] { "人均生产率", 7.1, 3.06, 8.69, 7.95, 6.46, 8.01 });

dt.Rows.Add(new object[] { "占2005年3月人数比例", 1.06, 1.06, 1, 0.96, 0.93, 0.91 });

return dt;

}

private void Form1\_Load(object sender, EventArgs e)

{

DataTable dt = CreateData();

this.gridControl1.DataSource = dt;

CreateChart(dt);

}

[复制代码](javascript:void(0);)

**2、创建图表图形**

[复制代码](javascript:void(0);)

private void CreateChart(DataTable dt)

{

#region Series

//创建几个图形的对象

Series series1 = CreateSeries("员工人数", ViewType.Line, dt, 0);

Series series2 = CreateSeries("人均月薪", ViewType.Line, dt, 1);

Series series3 = CreateSeries("成本TEU", ViewType.Line, dt, 2);

Series series4 = CreateSeries("人均生产率", ViewType.Line, dt, 3);

Series series5 = CreateSeries("占2005年3月人数比例", ViewType.Line, dt, 4);

#endregion

List<Series> list = new List<Series>() { series1, series2, series3, series4, series5 };

chartControl1.Series.AddRange(list.ToArray());

chartControl1.Legend.Visible = false;

chartControl1.SeriesTemplate.LabelsVisibility = DefaultBoolean.True;

for (int i = 0; i < list.Count; i++)

{

list[i].View.Color = colorList[i];

CreateAxisY(list[i]);

}

}

[复制代码](javascript:void(0);)

为了简化代码，并方便处理，上面代码中提取了两个函数进行了独立处理。

CreateSeries用于创建一个典型的图形，如一条曲线。CreateAxisY用来创建一个多重坐标轴。

CreateSeries用于创建一个典型的图形的源码如下所示。值得注意的是series.ArgumentScaleType = **ScaleType.Qualitative**;这句代码必须设置，否则默认会把“2005年1月”内容转换为日期类型，显示不恰当的内容。

[复制代码](javascript:void(0);)

/// <summary>

/// 根据数据创建一个图形展现

/// </summary>

/// <param name="caption">图形标题</param>

/// <param name="viewType">图形类型</param>

/// <param name="dt">数据DataTable</param>

/// <param name="rowIndex">图形数据的行序号</param>

/// <returns></returns>

private Series CreateSeries(string caption, ViewType viewType, DataTable dt, int rowIndex)

{

Series series = new Series(caption, viewType);

for (int i = 1; i < dt.Columns.Count; i++)

{

string argument = dt.Columns[i].ColumnName;//参数名称

decimal value = (decimal)dt.Rows[rowIndex][i];//参数值

series.Points.Add(new SeriesPoint(argument, value));

}

//必须设置ArgumentScaleType的类型，否则显示会转换为日期格式，导致不是希望的格式显示

//也就是说，显示字符串的参数，必须设置类型为ScaleType.Qualitative

series.ArgumentScaleType = ScaleType.Qualitative;

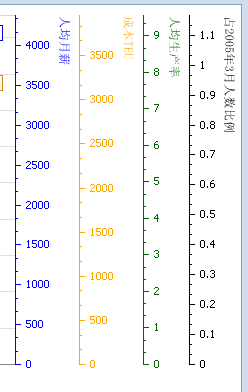
series.LabelsVisibility = DevExpress.Utils.DefaultBoolean.True;//显示标注标签

return series;

}

[复制代码](javascript:void(0);)

CreateAxisY用来创建一个多重坐标轴的代码如下所示，注意这里多重坐标，使用了和Series一直的**View.Color**颜色，这样方便区分。



[复制代码](javascript:void(0);)

/// <summary>

/// 创建图表的第二坐标系

/// </summary>

/// <param name="series">Series对象</param>

/// <returns></returns>

private SecondaryAxisY CreateAxisY(Series series)

{

SecondaryAxisY myAxis = new SecondaryAxisY(series.Name);

((XYDiagram)chartControl1.Diagram).SecondaryAxesY.Add(myAxis);

((LineSeriesView)series.View).AxisY = myAxis;

myAxis.Title.Text = series.Name;

myAxis.Title.Alignment = StringAlignment.Far; //顶部对齐

myAxis.Title.Visible = true; //显示标题

myAxis.Title.Font = new Font("宋体", 9.0f);

Color color = series.View.Color;//设置坐标的颜色和图表线条颜色一致

myAxis.Title.TextColor = color;

myAxis.Label.TextColor = color;

myAxis.Color = color;

return myAxis;

}

[复制代码](javascript:void(0);)

**3、Web界面的展现和代码处理**

本来以为在Web上，使用DevExpress控件实现上图的图表很麻烦，没想到它们的对象关系及属性完全一样，复制代码就基本解决问题了，界面代码变化一点点而已。

[复制代码](javascript:void(0);)

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="WebLineStatisticChart.Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<p><h3 align="center" style="text-align: left">D线产量、薪酬、人员、生成率趋势图</h3></p>

<dx:WebChartControl ID="chartControl1" runat="server" Height="452px"

Width="1013px">

</dx:WebChartControl>

<div>

</div>

<br />

<dx:ASPxGridView ID="ASPxGridView1" runat="server" Theme="Aqua">

<SettingsPager Visible="False">

</SettingsPager>

</dx:ASPxGridView>

</form>

</body>

</html>

[复制代码](javascript:void(0);)

后台代码，除了绑定数据的代码略有不同外，其他完全一致。

[复制代码](javascript:void(0);)

public partial class Default : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (!this.IsPostBack)

{

//D线产量、薪酬、人员、生成率趋势图

DataTable dt = CreateData();

//绑定数据源到表格里面

this.ASPxGridView1.DataSource = dt;

this.ASPxGridView1.DataBind();

CreateChart(dt);

}

}

.............

[复制代码](javascript:void(0);)

Web上图表的效果如下所示，鼠标放到图形节点上，还有动态提示，很友好。



**4、 图表的打印**

ChartControl提供了很好的打印功能，使用代码很简单。

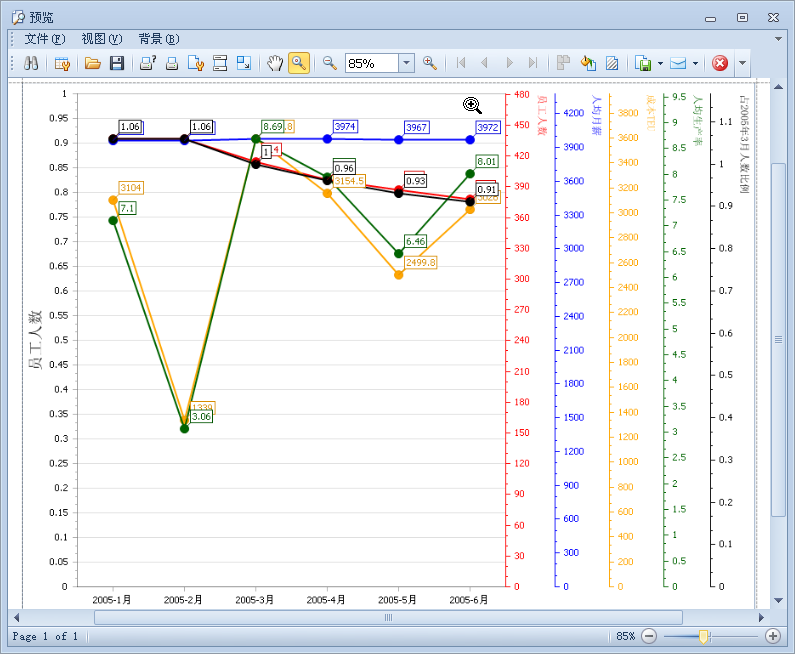
private void btnPrint\_Click(object sender, EventArgs e)

{

this.chartControl1.ShowPrintPreview(DevExpress.XtraCharts.Printing.PrintSizeMode.Zoom);

}

得到的效果如下所示，基本上能够满足要求了，当然复杂的定制打印需要另外的处理代码了。



 打印也可以使用下面的代码，效果差不多，但是定制性比较强一点。

[复制代码](javascript:void(0);)

private void btnPrint\_Click(object sender, EventArgs e)

{

//this.chartControl1.ShowPrintPreview(DevExpress.XtraCharts.Printing.PrintSizeMode.Zoom);

DevExpress.XtraPrintingLinks.CompositeLink compositeLink = new DevExpress.XtraPrintingLinks.CompositeLink();

DevExpress.XtraPrinting.PrintingSystem ps = new DevExpress.XtraPrinting.PrintingSystem();

compositeLink.PrintingSystem = ps;

compositeLink.Landscape = true;

compositeLink.PaperKind = System.Drawing.Printing.PaperKind.A4;

DevExpress.XtraPrinting.PrintableComponentLink link = new DevExpress.XtraPrinting.PrintableComponentLink(ps);

ps.PageSettings.Landscape = true;

link.Component = this.chartControl1;

compositeLink.Links.Add(link);

link.CreateDocument(); //建立文档

ps.PreviewFormEx.Show();//进行预览

}

[复制代码](javascript:void(0);)

http://www.cnblogs.com/Images/OutliningIndicators/None.gif主要研究技术：代码生成工具、会员管理系统、客户关系管理软件、病人资料管理软件、Visio二次开发、酒店管理系统、仓库管理系统等共享软件开发  
 专注于[Winform开发框架/混合式开发框架](http://www.iqidi.com/Framework/index.html)、[Web开发框架](http://www.iqidi.com/Framework/WebIntroduce.htm)、[Bootstrap开发框架](http://www.iqidi.com/Framework/BootstrapIndex.html)、[微信门户开发框架的研究及应用](http://www.iqidi.com/Framework/WeixinIndex.htm)。  
  转载请注明出处：  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif撰写人：伍华聪  [http://www.iqidi.com](http://www.iqidi.com/)