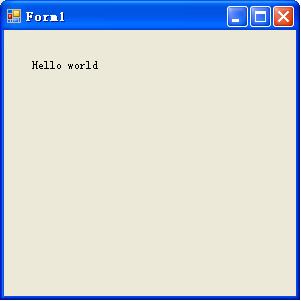
[**WinForm控件开发总结(一)------开篇**](http://www.cnblogs.com/guanjinke/archive/2006/12/04/582084.html)

      我本人不是专业的控件开发人员，只是在平常的工作中，需要自己开发一些控件。在自己开发WinForm控件的时候，没有太多可以借鉴的资料，只能盯着MSDN使劲看，还好总算有些收获。现在我会把这些经验陆陆续续的总结出来，写成一系列方章，希望对看到的朋友有所帮助。今天我来开个头。  
      其实开发WinForm控件并不是很复杂，[.NET](http://www.cnblogs.com/guanjinke/archive/2006/12/04/guanjinke.cnblogs.com)为我们提供了丰富的底层支持。如果你有MFC或者API图形界面的开发经验，那么学会WinForm控件可能只需要很短的时间就够了。  
      自己开发的WinForm控件通常有三种类型：复合控件（Composite Controls），扩展控件（Extended Controls），自定义控件（Custom Controls）。     
      **复合控件**：将现有的各种控件组合起来，形成一个新的控件，将集中控件的功能集中起来。  
     **扩展控件**：在现有控件的控件的基础上派生出一个新的控件，为原有控件增加新的功能或者修改原有控件的控能。  
      **自定义控件**：直接从System.Windows.Forms.Control类派生出来。Control类提供控件所需要的所有基本功能，包括键盘和鼠标的事件处理。自定义控件是最灵活最强大的方法，但是对开发者的要求也比较高，你必须为Control类的OnPaint事件写代码，你也可以重写Control类的WndProc方法，处理更底层的Windows消息，所以你应该了解GDI＋和Windows API。      
      本系列文章主要介绍自定义控件的开发方法。  
      控件（可视化的）的基本特征：  
      1.       可视化。  
      2.       可以与用户进行交互，比如通过键盘和鼠标。  
      3.       暴露出一组属性和方法供开发人员使用。  
      4.       暴露出一组事件供开发人员使用。  
      5.       控件属性的可持久化。  
      6.       可发布和可重用。  
      这些特征是我自己总结出来，不一定准确，或者还有遗漏，但是基本上概括了控件的主要方面。  
      接下来我们做一个简单的控件来增强一下感性认识。首先启动VS2005创建一个ClassLibrary工程，命名为CustomControlSample，VS会自动为我们创建一个solution与这个工程同名，然后删掉自动生成的Class1.cs文件，最后在Solution explorer里右键点击CustomControlSample工程选择Add->Classes…添加一个新类，将文件的名称命名为FirstControl。下边是代码：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Collections.Generic;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Text;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Windows.Forms;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.ComponentModel;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Drawing;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifnamespace CustomControlSample  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    public class FirstControl : Control  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public FirstControl()  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        // ContentAlignment is an enumeration defined in the System.Drawing  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        // namespace that specifies the alignment of content on a drawing   
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        // surface.  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private ContentAlignment alignmentValue = ContentAlignment.MiddleLeft;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        [  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        Category("Alignment"),  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        Description("Specifies the alignment of text.")  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        ]  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public ContentAlignment TextAlignment  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            get  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return alignmentValue;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            set  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                alignmentValue = value;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                // The Invalidate method invokes the OnPaint method described   
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                // in step 3.  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Invalidate();  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        protected override void OnPaint(PaintEventArgs e)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            base.OnPaint(e);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            StringFormat style = new StringFormat();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            style.Alignment = StringAlignment.Near;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            switch (alignmentValue)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                case ContentAlignment.MiddleLeft:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    style.Alignment = StringAlignment.Near;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    break;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                case ContentAlignment.MiddleRight:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    style.Alignment = StringAlignment.Far;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    break;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                case ContentAlignment.MiddleCenter:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    style.Alignment = StringAlignment.Center;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    break;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            // Call the DrawString method of the System.Drawing class to write     
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            // text. Text and ClientRectangle are properties inherited from  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            // Control.  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            e.Graphics.DrawString(  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Text,  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Font,  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                new SolidBrush(ForeColor),  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                ClientRectangle, style);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}

 在上一篇文章里我们创建了一个简单的控件FirstControl，现在我来介绍一下怎么使用和调试自己的控件。我希望将过程写的尽可能的详细，让想学习控件开发的朋友容易上手，高手们见谅。  
      在同一个solution里添加一个Windows Application工程（在Solution Explorer里右键点击CustomControlSample solution选择Add->New Project…），命名为TestControl。VS会为你自动生成一个Form，文件名为Form1.cs。在Solution Explorer里双击Form1.cs文件进入到Form设计界面。现在我们将FirstControl控件添加到工具箱（ToolBox）里，在Toolbox上右键点击，在弹出的菜单中选择Choose Items…，在出现的Choose Toolbox Items对话框中点击Browse…按钮，在Open对话框中选择我们的控件工程生成的dll（我的dll在F:\Programs\C#\CustomControlSample\CustomControlSample\bin\Debug目录下，你可以根据实际情况去找）。完成这一步，在Toolbox就会出现我们设计的控件，图标是一个蓝色的齿轮（默认的都是这个，当然你也可以修改，后边的文章我会介绍），名称是FirstControl。  
      现在我们在Toolbox中选中FirstControl，在form设计器上左键点击，或者按住鼠标拖放。我们制作的控件出现在了Form设计器上，在Form设计器上选中这个控件，然后在属性浏览器中将Text属性设为Hello World，现在我们的控件上的文字变成了Hello World。接下来我们要运行测试的工程，看看实际的效果。在运行之前，将测试工程设为启动工程，具体做法是，在solution explorer中右键点击TestControl工程，选择“Set as Startup Project”。点击工具栏里的运行按钮，或者按键盘的F5功能键。实际效果如下图所示：  
      

[**WinForm控件开发总结(三)------认识WinForm控件常用的Attribute**](http://www.cnblogs.com/guanjinke/archive/2006/12/06/584714.html)

      在前面的文章里我们制作了一个非常简单的控件。现在我们回过头来看看这些代码透露出什么信息。  
      这个类是直接从Control类派生出来的，自定义控件都是直接从Control类派生出来的。这个类定义了一个属性TextAlignment，用来控制文本在控件中显示的位置：

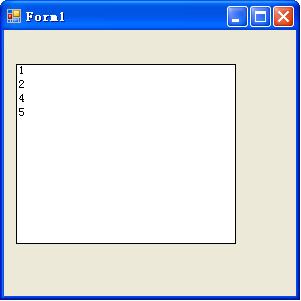
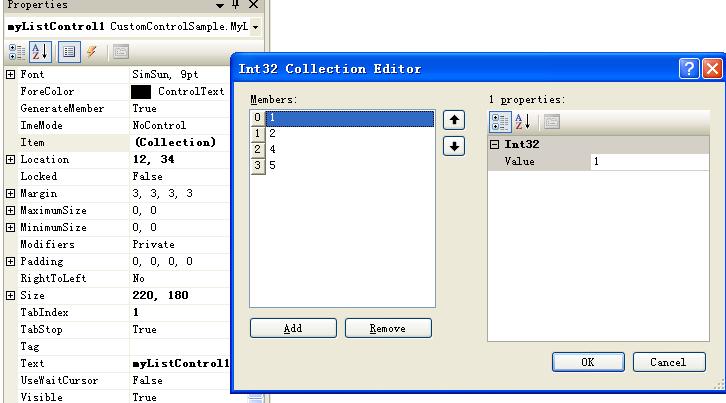
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        [  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        Category("Alignment"),  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        Description("Specifies the alignment of text.")  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        ]  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        public ContentAlignment TextAlignment  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            get  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return alignmentValue;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            set  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                alignmentValue = value;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                // The Invalidate method invokes the OnPaint method described   
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                // in step 3.  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Invalidate();  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

   在这个属性之上有两个Attribute，这两个attribute描述了控件在设计时所表现出来的特征。我们来看看在控件设计中有哪些主要用到的设计时Attribute。   
   **BrowsableAttribute**：描述是否一个属性或事件应该被显示在属性浏览器里。  
   **CategoryAttribute**：描述一个属性或事件的类别，当使用类别的时候，属性浏览器按类别将属性分组。  
   **DescriptionAttribute**：当用户在属性浏览器里选择属性的时候，description里指定的文本会显示在属性浏览器的下边，向用户显示属性的功能。  
   **BindableAttribute**：描述是否一个属性倾向于被绑定。  
   **DefaultPropertyAttribute**：为组件指定一个默认的属性，当用户在Form设计器上选择一个控件的时候，默认属性会在属性浏览器里被选中。     
   **DefaultValueAttribute**：为一个简单类型的属性设置一个默认值。  
   **EditorAttribute**：为属性指定一个特殊的编辑器。  
   **LocalizableAttribute**：指示一个属性是否能被本地化，任何有这个Attribute的属性将会被持久化到资源文件里。     
   **DesignerSerializationVisibilityAttribute**：指示一个属性是否或者如何持久化到代码里。  
   **TypeConverterAttribute**：为属性指定一个类型转换器，类型转换器能将属性的值转化成其它的数据类型。  
   **DefaultEventAttribute**：为组件指定一个默认的事件，当用户在form设计其中选择一个控件的时候，在属性浏览器中这个事件被选中。  
   这些设计时的Attribute时很重要的，如果使用的好，将会对用户的使用带来很大的便利。  
   这一章我主要介绍了设计时的Attribute，接下来的文章我将通过代码来介绍这些Attribute。

[**WinForm控件开发总结(四)-----控件属性的串行化**](http://www.cnblogs.com/guanjinke/archive/2006/12/11/588609.html)

       前一篇文章介绍了常用的设计时Attribute。其中BrowsableAttribute，CategoryAttribute，DescriptionAttribute，DefaultPropertyAttribute，DefaultEventAttribute都是比较简单的，也是可有可无，但是为了提供更好的用户体验这些Attribute最好不要省掉，如果你对这些Attribute还不熟悉，可以参考我前一篇文章的描述或者查看MSDN，这里我就不在赘述了。  
       下来我们主要介绍一下DesignerSerializationVisibilityAttribute和TypeConverterAttribute。  
       DesignerSerializationVisibilityAttribute的功能是指示一个属性是否串行化和如何串行化，它的值是一个枚举，一共有三种类型Content，Hidden，Visible。Content指示代码生成器为对象包含的内容生成代码，而不是为对象本身，Hidden指示代码生成器不为对象生成代码，visible指示代码生成器为对象生成代码。假如你的控件有一个集合属性，又想在设计时自动将集合属性的内容生成代码，那么就使用这个Attribute，并将值设为DesignerSerializationVisibility.Content。  
      TypeConverterAttribute的作用就更大一些，也稍微复杂一些。TypeConverterAttribute主要的目的是为属性指定一个类型转换器，这个转化器可以将属性的值转换城其它的类型。[.NET](http://www.cnblogs.com/guanjinke/admin/guanjinke.cnblogs.com)框架已经为大部分常用的类型都提供了类型转换器，比如Color就有ColorConverter，枚举类型就有EnumConverter，等等，所以一般情况下你没有必要写类型转换器，如果你的属性的特殊的类型或者自定义的类型那么就必须要写了。类型转换器都是从System.ComponentModel.TypeConverter派生出来的，你需要重写其中的一些方法来达到转换的目的，在我们开发的过程中，其实只关心属性的值如何转换成字符串（因为属性的值需要在属性浏览器里显示出来，属性浏览器里显示的都是字符串）和源代码（需要自动为属性的值生成源代码以实现持久化），当然反过来，也要将字符串和源代码转换成属性的值。另外使用TypeConverter也可以实现子属性，让属性的子属性也显示在属性浏览器里，并且可以折叠。  
       接下来我就写一个简单的控件来演示一下这个控件。代码如下：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Collections.Generic;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Text;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Windows.Forms;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Drawing;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.ComponentModel;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Collections;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifnamespace CustomControlSample  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    public class MyListControl:System.Windows.Forms.Control  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private List<Int32> \_list = new List<Int32>();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public MyListControl()  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        [Browsable(true)]  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public List<Int32> Item  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            get  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return \_list;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            set  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                \_list = value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        protected override void OnPaint(PaintEventArgs e)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            base.OnPaint(e);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Graphics g = e.Graphics;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            //绘制控件的边框  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            g.DrawRectangle(Pens.Black,new Rectangle(Point.Empty,new Size(Size.Width-1,Size.Height-1)));  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif     
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            for (Int32 i = 0; i < \_list.Count; i++)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                g.DrawString(\_list[i].ToString(), Font, Brushes.Black,1, i \* FontHeight);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

      我创建了一个简单的List控件，将用户输入的数据显示在控件中，效果图如下：  
     
     在这个控件中，我声明了一个集合属性Item供用户输入要显示的整型数值。我们按照WinForm控件制作教程（二）中的方法将控件加到ToolBox里，然后拖到Form设计器中，然后选中控件，在属性浏览中查看控件的属性，属性中有一个Item的属性，属性右边的值显示为Collection，当你点击这个值的时候，值的右边出现一个小按钮，点击这个小按钮，就会出现弹出一个Collection Editor窗口，你可以在在这个编辑器里添加你想显示的整型值，如图：  
        
     添加完以后，关闭Collection Editor。现在我们看看Form设计器为我们生成了什么代码。对于用户在Form设计器中设计的内容，设计器的代码生成器会将代码生成到窗口类的InitializeComponent（）方法中，对于vs2005来说，这个方法位于\*\*\*.Designer.cs文件中，在我当前的工程中位于Form1.Designer.cs文件中。在solution浏览器中双击打开这个文件，看看Form设计器为我们生成了什么代码：

http://www.cnblogs.com/Images/OutliningIndicators/None.gif            //   
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            // myListControl1  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            //   
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.BackColor = System.Drawing.SystemColors.ActiveCaptionText;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item = ((System.Collections.Generic.List<int>)(resources.GetObject("myListControl1.Item")));  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Location = new System.Drawing.Point(12, 34);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Name = "myListControl1";  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Size = new System.Drawing.Size(220, 180);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.TabIndex = 1;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Text = "myListControl1";  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

      设计器将Item的内容串行化到了资源文件里。现在我们修改控件的代码，让设计器将Item的内容串行化到源代码里。我们为Item属性添加DesignerSerializationVisibilityAttribute，代码片断如下：

http://www.cnblogs.com/Images/OutliningIndicators/None.gif[Browsable(true)]  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        [DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Content)]  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        public List<Int32> Item  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            get  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return \_list;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            set  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                \_list = value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

      编辑完以后，Build控件工程，回到测试工程里，将Item属性里的值，删掉重新添加，添加完以后，我们再来看看设计器生成的代码：

http://www.cnblogs.com/Images/OutliningIndicators/None.gif//   
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            // myListControl1  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            //   
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.BackColor = System.Drawing.SystemColors.ActiveCaptionText;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item.Add(1);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item.Add(2);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item.Add(3);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item.Add(6);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item.Add(8);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item.Add(9);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Location = new System.Drawing.Point(12, 34);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Name = "myListControl1";  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Size = new System.Drawing.Size(220, 180);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.TabIndex = 1;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Text = "myListControl1";  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

      现在设计器将Item的内容串行化到源代码里了。  
      时间有限，今天就写到这里，下一篇文章我来介绍TypeConverterAttribute。

[**WinForm控件开发总结(五)-----为控件的复杂属性提供类型转换器**](http://www.cnblogs.com/guanjinke/archive/2006/12/11/589372.html)

      上一篇文章我已经介绍了TypeConverterAttribute元数据的作用，本文将通过代码向你展示具体的实现。在这个例子中，我要给控件添加一个复杂的属性，这个属性对这个控件没有什么功用，纯粹是为了演示，有些牵强附会了。  
       现在在前一篇文章中的创建的控件代码中添加一个Scope属性：

http://www.cnblogs.com/Images/OutliningIndicators/None.gif        [Browsable(true)]  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        public Scope Scope  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            get  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return \_scope;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            set  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                \_scope = value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

      这个属性的类型是Scope类，代码如下：

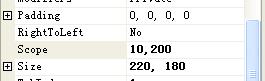
http://www.cnblogs.com/Images/OutliningIndicators/None.gifpublic class Scope  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private Int32 \_min;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private Int32 \_max;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public Scope()  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifpublic Scope(Int32 min, Int32 max)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            \_min = min;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            \_max = max;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        [Browsable(true)]  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public Int32 Min  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            get  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return \_min;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            set  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                \_min = value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        [Browsable(true)]  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public Int32 Max  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            get  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return \_max;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            set  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                \_max = value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif             
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

       添加完属性后，build控件工程，然后在测试的工程里选中添加的控件，然后在属性浏览器里观察它的属性，发现Scope属性是灰的，不能编辑。前一篇文章提到了，在属性浏览器里可以编辑的属性都是有类型转换器的，而[.NET](http://www.cnblogs.com/guanjinke/admin/guanjinke.cnblogs.com)框架为基本的类型和常用的类型都提供了默认的类型转换器。接下来我们为Scope类添加一个类型转换器，以便这个属性能够被编辑，而且也可以在源代码文件里自动生成相应的代码。下面是类型转换器的代码：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifpublic class ScopeConverter : TypeConverter  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override bool CanConvertFrom(ITypeDescriptorContext context, Type sourceType)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (sourceType == typeof(String)) return true;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.CanConvertFrom(context, sourceType);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override bool CanConvertTo(ITypeDescriptorContext context, Type destinationType)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (destinationType == typeof(String)) return true;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (destinationType == typeof(InstanceDescriptor)) return true;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.CanConvertTo(context, destinationType);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override object ConvertTo(ITypeDescriptorContext context, System.Globalization.CultureInfo culture, object value, Type destinationType)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            String result = "";  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (destinationType == typeof(String))  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Scope scope = (Scope)value;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                result = scope.Min.ToString()+"," + scope.Max.ToString();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return result;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (destinationType == typeof(InstanceDescriptor))  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif                ConstructorInfo ci = typeof(Scope).GetConstructor(new Type[] http://www.cnblogs.com/Images/dot.gif{typeof(Int32),typeof(Int32) });  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Scope scope = (Scope)value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif                return new InstanceDescriptor(ci, new object[] http://www.cnblogs.com/Images/dot.gif{ scope.Min,scope.Max });  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.ConvertTo(context, culture, value, destinationType);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override object ConvertFrom(ITypeDescriptorContext context, System.Globalization.CultureInfo culture, object value)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (value is string)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                String[] v = ((String)value).Split(',');  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                if (v.GetLength(0) != 2)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif                http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    throw new ArgumentException("Invalid parameter format");  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif                }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Scope csf = new Scope();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                csf.Min = Convert.ToInt32(v[0]);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                csf.Max = Convert.ToInt32(v[1]);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return csf;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.ConvertFrom(context, culture, value);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif    }  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

      现在我们为类型提供类型转换器，我们在类型前面添加一个TypeConverterAttribute，如下：

http://www.cnblogs.com/Images/OutliningIndicators/None.gif    [TypeConverter(typeof(ScopeConverter))]  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif    public class Scope  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

        添加完以后build工程，然后切换到测试工程，选中控件，在属性浏览器里查看属性，现在的Scope属性可以编辑了，如下图所示：   
           
         我们修改默认的值，然后看看Form设计器为我们生成了什么代码：

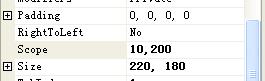
http://www.cnblogs.com/Images/OutliningIndicators/None.gifthis.myListControl1.BackColor = System.Drawing.SystemColors.ActiveCaptionText;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item.Add(1);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item.Add(2);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item.Add(3);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item.Add(6);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item.Add(8);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Item.Add(9);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Location = new System.Drawing.Point(12, 34);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Name = "myListControl1";  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Scope = new CustomControlSample.Scope(10, 200);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.Size = new System.Drawing.Size(220, 180);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif            this.myListControl1.TabIndex = 1;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        this.myListControl1.Text = "myListControl1";  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

        关键是这一行this.myListControl1.Scope = new CustomControlSample.Scope(10, 200)，Scope类的类型转换器为属性提供了实例化的代码。

[**WinForm控件开发总结(六)-----控件属性类型转换器代码详解**](http://www.cnblogs.com/guanjinke/archive/2006/12/14/592605.html)

      在上一篇文章，我为控件添加一个一个复杂属性，并且为这个属性的类型的编写了一个类型转换器，现在我们来看看这个类型转换器的代码，并解释一下这些代码的意义。  
       要实现一个类型转换器，我们必须要重写（override）四个方法：  
       **CanConvertFrom（）――**根据类型参数进行测试，判断是否能从这个类型转换成当前类型,在本例中我们只提供转换string和InstanceDescriptor类型的能力。  
       **CanConvertTo（）――**根据类型参数进行测试，判断是否能从当前类型转换成指定的类型。  
       **ConvertTo()――**将参数value的值转换为指定的类型。  
       **ConvertFrom（）――**串换参数value，并返回但书类型的一个对象。

http://www.cnblogs.com/Images/OutliningIndicators/None.gif public override object ConvertTo(ITypeDescriptorContext context, System.Globalization.CultureInfo culture, object value, Type destinationType)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            String result = "";  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (destinationType == typeof(String))  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Scope scope = (Scope)value;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                result = scope.Min.ToString()+"," + scope.Max.ToString();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return result;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (destinationType == typeof(InstanceDescriptor))  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif                ConstructorInfo ci = typeof(Scope).GetConstructor(new Type[] http://www.cnblogs.com/Images/dot.gif{typeof(Int32),typeof(Int32) });  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Scope scope = (Scope)value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif                return new InstanceDescriptor(ci, new object[] http://www.cnblogs.com/Images/dot.gif{ scope.Min,scope.Max });  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.ConvertTo(context, culture, value, destinationType);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

         上面是ConvertTo的实现，如果转换的目标类型是string，我将Scope的两个属性转换成string类型，并且用一个“，”连接起来，这就是我们在属性浏览器里看到的表现形式，如图：  
        
      如果转换的目标类型是实例描述器（InstanceDescriptor，它负责生成实例化的代码），我们需要构造一个实例描述器，构造实例描述器的时候，我们要利用反射机制获得Scope类的构造器信息，并在new的时候传入Scope实例的两个属性值。实例描述器会为我们生成这样的代码：this.myListControl1.Scope = new CustomControlSample.Scope(10, 200)；在最后不要忘记调用 base.ConvertTo(context, culture, value, destinationType)，你不需要处理的转换类型，交给基类去做好了。

http://www.cnblogs.com/Images/OutliningIndicators/None.gifpublic override object ConvertFrom(ITypeDescriptorContext context, System.Globalization.CultureInfo culture, object value)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (value is string)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                String[] v = ((String)value).Split(',');  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                if (v.GetLength(0) != 2)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif                http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    throw new ArgumentException("Invalid parameter format");  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif                }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Scope csf = new Scope();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                csf.Min = Convert.ToInt32(v[0]);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                csf.Max = Convert.ToInt32(v[1]);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return csf;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.ConvertFrom(context, culture, value);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif    }  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

      上面是ConvertFrom的代码，由于系统能够直接将实例描述器转换为Scope类型，所以我们就没有必要再写代码，我们只需要关注如何将String（在属性浏览出现的属性值的表达）类型的值转换为Scope类型。没有很复杂的转换，只是将这个字符串以“，”分拆开，并串换为Int32类型，然后new一个Scope类的实例，将分拆后转换的两个整型值赋给Scope的实例，然后返回实例。在这段代码里，我们要判断一下用户设定的属性值是否有效。比如，如果用户在Scope属性那里输入了“10200”，由于没有输入“，”，我们无法将属性的值分拆为两个字符串，也就无法进行下面的转换，所以，我们要抛出一个异常，通知用户重新输入。

0

0

0

(请您对文章做出评价)

[**WinForm控件开发总结(七)-----为复杂属性的子属性提供编辑功能**](http://www.cnblogs.com/guanjinke/archive/2006/12/15/593784.html)

      前面的几篇文章中，我们给控件添加一个复杂的类型Scope，并且给它的类型提供的一个类型转换器，现在我们可以在属性浏览器中编辑它的值，并且它的值也被串行化的源代码里了。但是你有没有发现，在属性浏览器里编辑这个属性的值还是不太方便。因为属性只是“10，200”这种形式的，所以，你必须按照这种格式来修改，一旦格式错误就会引发异常，比如输入一个“10200”。我们期望这个属性的每一子属性都能够被独立的编辑就好了，这并非不能实现，而且实现还很简单。  
      为了在属性浏览器里能够独立的编辑子属性，我们还要重写两个方法：GetPropertiesSupported（）和GetProperties（）；下面是ScopeConverter的完整代码：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifpublic class ScopeConverter : TypeConverter  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override bool CanConvertFrom(ITypeDescriptorContext context, Type sourceType)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (sourceType == typeof(String)) return true;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.CanConvertFrom(context, sourceType);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override bool CanConvertTo(ITypeDescriptorContext context, Type destinationType)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (destinationType == typeof(String)) return true;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (destinationType == typeof(InstanceDescriptor)) return true;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.CanConvertTo(context, destinationType);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override object ConvertTo(ITypeDescriptorContext context, System.Globalization.CultureInfo culture, object value, Type destinationType)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            String result = "";  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (destinationType == typeof(String))  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Scope scope = (Scope)value;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                result = scope.Min.ToString()+"," + scope.Max.ToString();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return result;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (destinationType == typeof(InstanceDescriptor))  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif                ConstructorInfo ci = typeof(Scope).GetConstructor(new Type[] http://www.cnblogs.com/Images/dot.gif{typeof(Int32),typeof(Int32) });  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Scope scope = (Scope)value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif                return new InstanceDescriptor(ci, new object[] http://www.cnblogs.com/Images/dot.gif{ scope.Min,scope.Max });  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.ConvertTo(context, culture, value, destinationType);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override object ConvertFrom(ITypeDescriptorContext context, System.Globalization.CultureInfo culture, object value)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (value is string)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                String[] v = ((String)value).Split(',');  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                if (v.GetLength(0) != 2)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif                http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    throw new ArgumentException("Invalid parameter format");  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif                }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Scope csf = new Scope();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                csf.Min = Convert.ToInt32(v[0]);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                csf.Max = Convert.ToInt32(v[1]);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return csf;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.ConvertFrom(context, culture, value);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override bool GetPropertiesSupported(ITypeDescriptorContext context)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return true;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override PropertyDescriptorCollection GetProperties(ITypeDescriptorContext context, object value, Attribute[] attributes)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return TypeDescriptor.GetProperties(typeof(Scope), attributes);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

      在GetProperties方法里，我用TypeDescriptor获得了Scope类的所有的属性描述器并返回。如果你对TypeDescriptor还不熟悉的话，可以参考MSDN。  
      重写这两个方法并编译以后，在测试工程里查看控件的属性，你可以看到Scope是如下的形式：  
      http://images.cnblogs.com/cnblogs_com/guanjinke/77957/o_MyListControl4.JPG

[**WinForm控件开发总结(八)-----为属性提供弹出式编辑对话框**](http://www.cnblogs.com/guanjinke/archive/2006/12/18/596127.html)

      前几篇文章我们一直在讨论如何更方便的编辑复杂类型的属性，在这个过程中我介绍了类型转换器以及如何制作自己的类型转换器来实现属性值的串行化和实现子属性的编辑。对于Scope这种级别的复杂属性，一个类型转换器就已经足够了，但是对于更为复杂的属性，单单使用类型转换器已经不足以应付了，比如我们常用的Font属性。  
       在这种情况下，我们就需要提供更为复杂的编辑方式，比如属性编辑对话框，你还记得Font对话框吗？现在我们就来看看如何实现更复杂的属性编辑。复杂的属性编辑器分为两种类型，一种是弹出式模态对话框属性编辑器，一种式下拉式属性编辑器。如果你还没有感性的认识的话，可以观察一下TextBox控件的属性，Font属性的编辑器是模态对话框属性编辑器，Dock属性的编辑器是下拉式属性编辑器。  
       接下来我们来制作一个模态对话框编辑器，虽然Scope属性并不复杂，但是为了演示的方便，我们还是用它来做例子。  
       首先我们要做一个用来编辑属性的对话框，在对话框的构造函数里传入要编辑的属性的值。在对话框类里，声明一个Scope类型的私有变量\_scope用以保存传入和编辑后的值。还要增加一个Scope属性，以便外部环境能够获取编辑后的结果。对话框的外观如下：   
        
       在这个对话框里，我们要把OK按钮的DialogResult属性设为OK（当点击OK按钮时，模态对话框关闭，并返回DialogResult.OK），将Cancel按钮的DialogResult属性设为Cancel(当点击OK按钮时，模态对话框关闭，并返回DialogResult.OK)。另外我们要对用户输入的值做验证，以保证Scope的min和max值都是Int32类型。下边是对话框的代码：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Collections.Generic;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.ComponentModel;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Data;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Drawing;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Text;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Windows.Forms;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifnamespace CustomControlSample  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    public partial class ScopeEditorDialog : Form  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private Scope \_scope = null;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public ScopeEditorDialog(Scope scope)    
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            InitializeComponent();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            \_scope = scope;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            textBox1.Text = \_scope.Min.ToString();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            textBox2.Text = \_scope.Max.ToString();  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private void button1\_Click(object sender, EventArgs e)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            \_scope.Min = Convert.ToInt32(textBox1.Text);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            \_scope.Max = Convert.ToInt32(textBox2.Text);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private void textBox1\_Validating(object sender, CancelEventArgs e)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            try  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Int32.Parse(textBox1.Text);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            catch (FormatException)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                e.Cancel = true;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                MessageBox.Show("无效的值", "验证错误", MessageBoxButtons.OK, MessageBoxIcon.Error);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private void textBox2\_Validating(object sender, CancelEventArgs e)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            try  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Int32.Parse(textBox2.Text);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            catch (FormatException)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                e.Cancel = true;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                MessageBox.Show("无效的值", "验证错误", MessageBoxButtons.OK, MessageBoxIcon.Error);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public Scope Scope  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            get  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return \_scope;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            set  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                \_scope = value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

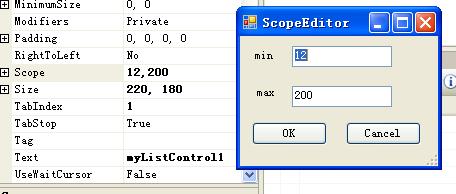
         每一个属性的编辑器都是直接或者间接的派生于UITypeEditor。开发环境从来也不会直接调用我们编写的模态对话框来编辑属性，而是调用UITypeEditor的某些虚方法，所以我们还必须提供一个派生于UITypeEditor的类来与开发环境通信。下边的代码实现了Scope的编辑器：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.ComponentModel;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Drawing.Design;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Windows.Forms.Design;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Windows.Forms;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifnamespace CustomControlSample  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    public class ScopeEditor:UITypeEditor  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override UITypeEditorEditStyle GetEditStyle(ITypeDescriptorContext context)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (context != null && context.Instance != null)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return UITypeEditorEditStyle.Modal;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.GetEditStyle(context);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override object EditValue(ITypeDescriptorContext context, IServiceProvider provider, object value)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            IWindowsFormsEditorService editorService = null;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (context != null && context.Instance != null && provider != null)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                editorService = (IWindowsFormsEditorService)provider.GetService(typeof(IWindowsFormsEditorService));  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                if (editorService != null)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif                http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    MyListControl control = (MyListControl)context.Instance;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    ScopeEditorDialog dlg = new ScopeEditorDialog(control.Scope);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    if (dlg.ShowDialog()== DialogResult.OK)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif                    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                        value = dlg.Scope;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                        return value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif                    }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif                }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

       在这个类里，我们重写了两个方法，一个是GetEditStyle，在这个方法里，我们通知开发环境，属性的编辑器是一个模态对话框。另一个方法是EditValue，这是最核心的方法，在这个方法里，我们通过上下文环境获得了正在编辑的控件的实例，并将实例的Scope属性传递给属性编辑对话框，显示对话框供用户编辑属性的值，用户编辑完属性的值，并关闭对话框，这时，我们从对话框里获取编辑后的结果反会给开发环境。       编写完Editor，我们就要将它应用到MyListControl的Scope属性上,现在的Scope属性定义如下:

http://www.cnblogs.com/Images/OutliningIndicators/None.gif[Browsable(true)]  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        [Editor(typeof(ScopeEditor),typeof(UITypeEditor))]  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        public Scope Scope  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            get  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return \_scope;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            set  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                \_scope = value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

             我们在Scope属性前加上了[Editor(typeof(ScopeEditor),typeof(UITypeEditor))]元数据。Build工程，查看实际的效果。在测试工程的窗体上，选中控件，观察Scope属性，当我们单击Scope属性的值时，在属性值的后边出现了一个按钮，如图：  
      http://images.cnblogs.com/cnblogs_com/guanjinke/77957/o_MyListControl6.JPG

      当我们点击这个按钮后，弹出了属性编辑的对话框，如图：  
        
      我们在对话框里编辑属性的值，并点击OK关闭对话框，现在Scope属性值已经被修改了。

[**WinForm控件开发总结(九)-----为属性提下拉式属性编辑器**](http://www.cnblogs.com/guanjinke/archive/2006/12/19/597260.html)

      在上一篇文章，我介绍了如何编写模态对话框属性编辑器，这篇文章我将介绍如何编写下拉式属性编辑器。下拉式（DropDown）属性编辑器和模态对话框属性编辑器的不同之处就是，当你点击属性值修改的时候，模态对话框编辑器是弹出一个模态对话框，而下拉式属性编辑器却是在紧贴着属性值的地方显示一个下拉的控件。不知道大家注意到了没有，这里我说的是显示一个下拉的控件，而这个控件也是需要你去开发的，接下来我还是以Scope属性为例，介绍一下具体的实现。  
      首先我们要创建一个用于编辑属性的控件，在本系列文章的开始，我们介绍了自定义控件有三种类型：复合控件，扩展控件，自定义控件。在本例中我们制作一个复合控件（Compsite control），复合控件的开发比较简单，不在本系列文章的讲解范围，我简单做个介绍，在Solution 浏览器里右键点击CustomControlSample工程选择Add->User Control…,输入文件名ScopeEditorControl.cs。我们做的这个复合控件上一篇文章介绍的模态对话框所包含子控件基本一样，除了用于确认和取消的按钮，如下图：  
        
      由于我们取消了用于确认和取消的按钮，并且是一个下拉的编辑器控件，在出现下面三种情况的时候下拉的编辑器控件会关闭：用户敲了回车，用户敲了ESC键，用户点击了编辑器以外的地方。当下拉编辑器控件关闭的时候我们就需要更新属性的值。下边是这个控件的代码：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Collections.Generic;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.ComponentModel;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Drawing;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Data;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Text;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Windows.Forms;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifnamespace CustomControlSample  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    public partial class ScopeEditorControl : UserControl  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private Scope \_oldScope;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private Scope \_newScope;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private Boolean canceling;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif          
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public ScopeEditorControl(Scope scope)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            \_oldScope = scope;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            \_newScope = scope;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            InitializeComponent();  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public Scope Scope  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            get  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return \_newScope;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private void textBox1\_Validating(object sender, CancelEventArgs e)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            try  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Int32.Parse(textBox1.Text);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            catch (FormatException)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                e.Cancel = true;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                MessageBox.Show("无效的值", "验证错误", MessageBoxButtons.OK, MessageBoxIcon.Error);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private void textBox2\_Validating(object sender, CancelEventArgs e)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            try  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Int32.Parse(textBox2.Text);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            catch (FormatException)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                e.Cancel = true;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                MessageBox.Show("无效的值", "验证错误", MessageBoxButtons.OK, MessageBoxIcon.Error);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        protected override bool ProcessDialogKey(Keys keyData)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (keyData == Keys.Escape)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                \_oldScope = \_newScope;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                canceling = true;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.ProcessDialogKey(keyData);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private void ScopeEditorControl\_Leave(object sender, EventArgs e)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (!canceling)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                \_newScope.Max = Convert.ToInt32(textBox1.Text);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                \_newScope.Min = Convert.ToInt32(textBox2.Text);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private void ScopeEditorControl\_Load(object sender, EventArgs e)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            textBox1.Text = \_oldScope.Max.ToString();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            textBox2.Text = \_oldScope.Min.ToString();  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

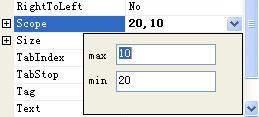
         和模态对话框编辑器一样，开发环境并不会直接调用我们的编辑器控件，而是用过UITypeEditor类的派生来实现编辑器的调用，所以我们必须实现一个下拉式编辑器。代码如下：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.ComponentModel;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Drawing.Design;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Windows.Forms.Design;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Windows.Forms;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifnamespace CustomControlSample  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    public class ScopeDropDownEditor : UITypeEditor  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override UITypeEditorEditStyle GetEditStyle(ITypeDescriptorContext context)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (context != null && context.Instance != null)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return UITypeEditorEditStyle.DropDown;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.GetEditStyle(context);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public override object EditValue(ITypeDescriptorContext context, IServiceProvider provider, object value)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            IWindowsFormsEditorService editorService = null;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if (context != null && context.Instance != null && provider != null)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                editorService = (IWindowsFormsEditorService)provider.GetService(typeof(IWindowsFormsEditorService));  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                if (editorService != null)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif                http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    MyListControl control = (MyListControl)context.Instance;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    ScopeEditorControl editorControl = new ScopeEditorControl(control.Scope);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    editorService.DropDownControl(editorControl);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    value = editorControl.Scope;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    return value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif                }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

      看过上一篇文章的朋友应该对这段代码很熟悉，是的，这两个编辑器的代码只有几行不同之处，在GetEditStyle方法中，我们返回的是UITypeEditorEditStyle.DropDown，而不是UITypeEditorEditStyle.Modal，表明我们的编辑器是一个下拉式的编辑器。在EditValue中的不同之处是，我们使用DropDownControl方法来显示编辑器。编辑器制作完毕，我们把Scope以前的编辑器替换成下拉式编辑器，如下：

http://www.cnblogs.com/Images/OutliningIndicators/None.gif[Browsable(true)]  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        [Editor(typeof(ScopeDropDownEditor), typeof(UITypeEditor))]  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif        public Scope Scope  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            get  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return \_scope;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            set  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                \_scope = value;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

      现在build CustomControlSample工程，然后切换到测试工程查看Scope属性。当我们点击属性的值，在属性值的后边出现了一个按钮：  
      http://images.cnblogs.com/cnblogs_com/guanjinke/77957/o_MyListControl8.JPG

   当点击这个按钮的时候，下拉的属性编辑器出现了：  
   

   好了，属性的编辑到这里就讲完了。

[**WinForm控件开发总结(十)-----为属性设置默认值**](http://www.cnblogs.com/guanjinke/archive/2006/12/24/602451.html)

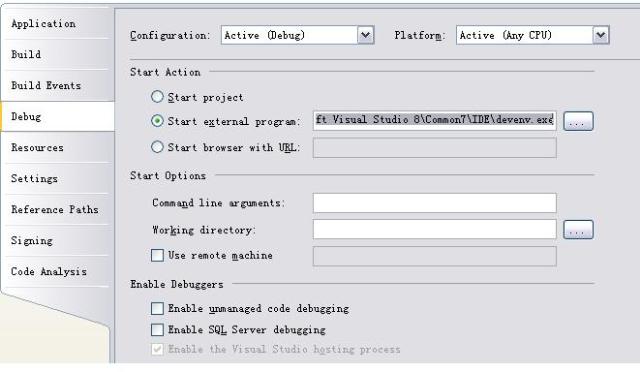
      本系列的前面几篇文章讲解了如何来定义属性以及更有效的编辑属性,接下来我要讲一下控件属性的默认值。如果我们希望自己开发的控件更易于被其它开发者使用，那么提供默认值是非常值得的。  
      如果你为属性设定了默认值，那么当开发者修改了属性的值，这个值在Property Explorer中将会以粗体显示。VS为属性提供一个上下文菜单，允许程序员使用控件把值重置为默认值。当VS进行控件的串行化时，他会判断那些值不是默认值，只有不是默认值的属性才会被串行化，所以为属性提供默认值时可以大大减少串行化的属性数目，提高效率。  
      那么VS怎么知道我们的属性值不是默认值了呢？我们需要一种机制来通知VS默认值。实现这种机制有两种方法：  
      对于简单类型的属性，比如Int32，Boolean等等这些Primitive类型，你可以在属性的声明前设置一个DefaultValueAttribute，在Attribute的构造函数里传入默认值。  
      对于复杂的类型，比如Font，Color，你不能够直接将这些类型的值传递给Attibute的构造函数。相反你应该提供Reset<PropertyName> 和ShouldSerialize<PropertyName>方法，比如ResetBackgroundColor(),ShouldSerializeBackgroundColor()。VS能够根据方法的名称来识别这种方法，比如Reset<PropertyName>方法把重置为默认值，ShouldSerialize<PropertyName>方法检查属性是否是默认值。过去我们把它称之为魔术命名法，应该说是一种不好的编程习惯，可是现在微软依然使用这种机制。我还是以前面几篇文章使用的例子代码。

http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Collections.Generic;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Text;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Windows.Forms;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.ComponentModel;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Drawing;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifnamespace CustomControlSample  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    public class FirstControl : Control  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifprivate String \_displayText=”Hello World!”;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifprivate Color \_textColor=Color.Red;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  public FirstControl()  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        // ContentAlignment is an enumeration defined in the System.Drawing  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        // namespace that specifies the alignment of content on a drawing   
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        // surface.  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private ContentAlignment alignmentValue = ContentAlignment.MiddleLeft;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        [  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        Category("Alignment"),  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        Description("Specifies the alignment of text.")  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        ]  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public ContentAlignment TextAlignment  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            get  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return alignmentValue;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            set  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                alignmentValue = value;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                // The Invalidate method invokes the OnPaint method described   
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                // in step 3.  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Invalidate();  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif [Browsable(true)]  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif [DefaultValue(“Hello World”)]  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif public String DisplayText  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifget  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifreturn \_displayText;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifset  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif     \_displayText =value;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    Invalidate();  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif[Browsable(true)]  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifpublic Color TextColor  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifget  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    return \_textColor;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifset  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    \_textColor=value;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifInvalidate();  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifpublic void ResetTextColor()  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    TextColor=Color.Red;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifpublic bool ShouldSerializeTextColor()  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifreturn TextColor!=Color.Red;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gifprotected override void OnPaint(PaintEventArgs e)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            base.OnPaint(e);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            StringFormat style = new StringFormat();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            style.Alignment = StringAlignment.Near;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            switch (alignmentValue)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                case ContentAlignment.MiddleLeft:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    style.Alignment = StringAlignment.Near;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    break;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                case ContentAlignment.MiddleRight:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    style.Alignment = StringAlignment.Far;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    break;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                case ContentAlignment.MiddleCenter:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    style.Alignment = StringAlignment.Center;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    break;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            // Call the DrawString method of the System.Drawing class to write     
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            // text. Text and ClientRectangle are properties inherited from  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            // Control.  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            e.Graphics.DrawString(  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                DisplayText,  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                Font,  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                new SolidBrush(TextColor),  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                ClientRectangle, style);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

      在上面的代码中，我增加了两个属性，一个是DisplayText，这是一个简单属性，我们只需要在它的声明前添加一个DefaultValue Attribute就可以了。另外一个是TextColor属性，这个复杂类型的属性，所以我们提供了ResetTextColor和ShouldSerializeTextColor来实现默认值。  
      默认值的实现就讲完了，但是有一点不要忽视了，你设定了默认值，就应该相应的初始化这些属性，比如我们例子中的代码：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifprivate String \_displayText=”Hello World!”;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifprivate Color \_textColor=Color.Red;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

[**WinForm控件开发总结(十一)-----调试控件的设计时行为**](http://www.cnblogs.com/guanjinke/archive/2007/01/23/628311.html)

      前面的一些文章绝大部分都是要讲控件的设计时的行为，既然涉及到这么多的设计时行为的代码编写，那么就有必要就一下如何来调试控件的设计行为。  
       调试控件的设计时行为和调试DLL的方式非常的相似，因为DLL是不能够单独运行的，而一般的控件也会在一个DLL里。当然如果你不考虑类的可复用性而把控件写在一个Windows Application里面也无可厚非，这样调试倒也变的简单了。但是我们还是要考虑更通常的情况。一般来说，我们调试DLL时，都是创建一个可独立运行的应用程序，在这个应用程序里引用你希望调试的DLL工程，在DLL工程的代码里设置断点，然后调试。所以，调试这一类东西，首要的问题就是找到一个调用它的宿主。调试控件的设计时行为什么样的宿主最好呢，当然是Visual studio了，visual studio里提供了非常全面的设计时支持。下来我就来演示一下具体的做法。  
       首先将你要测试的控件所在的工程设为启动工程。在Solution Explorer里右键点击控件所在的工程，在菜单里选择属性（Properties）进入工程属性设置界面，点击“Debug”页面，将Start Action 选为“Start External Program”，接下来点击后边的选择按钮选中你的Visual Studio的可执行程序，我的Visual Studio程序位于“D:\Program Files\Microsoft Visual Studio 8\Common7\IDE\devenv.exe”，你可以根据自己的情况选择。如下图：  
        
       在设置完以后工程属性以后，在需要调试的地方设置断点，然后点击F5或者点击工具栏的运行按钮。当点击以后，visual studio会运行起来，在运行起来的Visual studio里面打开一个应用你这个Assembly的工程，在这个工程里切换到Form设计器界面，选中你的控件，然后编辑你所要调设的功能，比如，你要调试一个控件的属性的Editor，你在这个editor类里设置断点，接着在属性浏览器里编辑这个属性，程序就会停在你设置的断点。

       今天我也把写前面的文章的时候用到的源码附上，方便朋友们使用。

[**WinForm控件开发总结(十二)-----让控件处理导航键**](http://www.cnblogs.com/guanjinke/archive/2007/05/11/743464.html)

    最近真的真的太忙了，以至于一个多月都没哟更新我的blog。昨天晚上，一个网上的朋友看了我的ToolBox的文章，问我一个问题，他说如何让ToolBox控件也能响应键盘操作，也就是用Up，down按键来选择工具箱控件里的Item，他添加了键盘事件，但是不起作用。一开始做这个控件的时候也只是演示一下控件的制作过程，只用了很短的时间做了一个，只考虑了用鼠标选取，没有考虑键盘操作，我想要添加键盘操作无非重载KeyDown事件，针对Up，Down做一些响应就可以了。可是添加了重载了OnKeyDown事件后，结果和那位朋友所说的一样，没有任何作用，我设了断点，调试了一下，发现KeyDown根本捕获不到Up，Down按键的点击，是什么原因呢，是不是忘记设控件的风格以便让它能够获得焦点？于是，我使用了语句：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifSetStyle(ControlStyles.Selectable, true);

依然没有效果，当我们在控件上按下Down键的时候，另一个控件获得了焦点。这时Up，Down按钮只是起到了导航的作用就像Tab键一样。  
      接下来，我在测试工程的窗体上放置了一个ListBox控件做一个对比，其实ToolBox和ListBox在界面表现上有相似之处，就是都有子Item，并且在ListBox上点击Down是起作用的，ListBox并没有失去焦点，这说明这时Up，Down按键没有成为导航键。我想Windows一定是对默认的导航键Up，Down,Left,Right有默认的处理，除非你希望你的控件希望自己处理这些键。用反汇编工具看了一下ListBoxControl控件的源代码，发现一个有趣的函数：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifprotected override bool IsInputKey(Keys keyData)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    if ((keyData & Keys.Alt) == Keys.Alt)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        return false;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    switch ((keyData & Keys.KeyCode))  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        case Keys.Prior:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        case Keys.Next:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        case Keys.End:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        case Keys.Home:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return true;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    return base.IsInputKey(keyData);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

在这里面，ListBoxControl允许Prior，Next，End，Home成为有效的输入键，接着一路跟下去，看看WinForm控件的基类Control的这个函数是如何处理的：

http://www.cnblogs.com/Images/OutliningIndicators/None.gif[UIPermission(SecurityAction.InheritanceDemand, Window=UIPermissionWindow.AllWindows)]  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifprotected virtual bool IsInputKey(Keys keyData)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    if ((keyData & Keys.Alt) != Keys.Alt)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        int num = 4;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        switch ((keyData & Keys.KeyCode))  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            case Keys.Left:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            case Keys.Up:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            case Keys.Right:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            case Keys.Down:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                num = 5;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                break;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            case Keys.Tab:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                num = 6;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                break;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        if (this.IsHandleCreated)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return ((((int) this.SendMessage(0x87, 0, 0)) & num) != 0);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif    return false;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

      注意这一行return (((([int](http://www.aisto.com/roeder/dotnet/Default.aspx?Target=code://mscorlib:2.0.0.0:b77a5c561934e089/System.Int32)) this.[SendMessage](http://www.aisto.com/roeder/dotnet/Default.aspx?Target=code://System.Windows.Forms:2.0.0.0:b77a5c561934e089/System.Windows.Forms.Control/SendMessage(Int32,Int32,Int32):IntPtr)(0x87, 0, 0)) & num) != 0);0x87是什么windows消息呢，打开WinUser.h文件，发现是WM\_GETDLGCODE,在MSDN中的描述是这样的：  
      The **WM\_GETDLGCODE** message is sent to the window procedure associated with a control. By default, the system handles all keyboard input to the control; the system interprets certain types of keyboard input as dialog box navigation keys. To override this default behavior, the control can respond to the **WM\_GETDLGCODE** message to indicate the types of input it wants to process itself.  
      也就是说windows用这个消息来判断哪些类型的输入交给控件本身来处理。然后，我注意到，对于方向导航键，函数都给于一个值5与this.[SendMessage](http://www.aisto.com/roeder/dotnet/Default.aspx?Target=code://System.Windows.Forms:2.0.0.0:b77a5c561934e089/System.Windows.Forms.Control/SendMessage(Int32,Int32,Int32):IntPtr)(0x87, 0, 0))的返回值进行与操作，那么this.[SendMessage](http://www.aisto.com/roeder/dotnet/Default.aspx?Target=code://System.Windows.Forms:2.0.0.0:b77a5c561934e089/System.Windows.Forms.Control/SendMessage(Int32,Int32,Int32):IntPtr)(0x87, 0, 0))的返回值都可能是什么值呢,WinUser.h中是这样声明的：

http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif/\*\*//\*  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif \* Dialog Codes  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif \*/  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif#define DLGC\_WANTARROWS     0x0001      /\* Control wants arrow keys         \*/  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif#define DLGC\_WANTTAB        0x0002      /\* Control wants tab keys           \*/  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif#define DLGC\_WANTALLKEYS    0x0004      /\* Control wants all keys           \*/  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif#define DLGC\_WANTMESSAGE    0x0004      /\* Pass message to control          \*/  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif#define DLGC\_HASSETSEL      0x0008      /\* Understands EM\_SETSEL message    \*/  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif#define DLGC\_DEFPUSHBUTTON  0x0010      /\* Default pushbutton               \*/  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif#define DLGC\_UNDEFPUSHBUTTON 0x0020     /\* Non-default pushbutton           \*/  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif#define DLGC\_RADIOBUTTON    0x0040      /\* Radio button                     \*/  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif#define DLGC\_WANTCHARS      0x0080      /\* Want WM\_CHAR messages            \*/  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif#define DLGC\_STATIC         0x0100      /\* Static item: don't include       \*/  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif#define DLGC\_BUTTON         0x2000      /\* Button item: can be checked      \*/

      5最贴切的表达就是DLGC\_WANTMESSAGE | DLGC\_WANTARROWS，也就是将方向键发送给控件处理，对于6呢，也就是DLGC\_WANTMESSAGE| DLGC\_WANTTAB，将Tab键发送给控件处理。  
       从这段代码里和控件实际的行为我们可以得出一个结论，那就是，控件本身是不处理方向键和Tab键的，因为他们有默认的行为，也就是支持焦点在窗体的控件之间转换。如果你想要处理这些导航键，那么结论很简单，就是重载IsInputKey方法，它是一个保护类型的虚方法。         
      在ToolBox控件的代码里重载IsinputKey方法：

http://www.cnblogs.com/Images/OutliningIndicators/None.gif        protected override bool IsInputKey(Keys keyData)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedBlock.gif        http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            if ((keyData & Keys.Alt) == Keys.Alt)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                return false;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            switch ((keyData & Keys.KeyCode))  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cnblogs.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cnblogs.com/Images/dot.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                case Keys.Up:  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                case Keys.Down:                  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                    return true;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return base.IsInputKey(keyData);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif

       当用户点击的键是Up，Down的时候，返回true，这时我们的OnKeyDown方法里就可以捕获到Up，Down的点击事件了。