

新时尚Windows8开发（22）：ListBox与ComboBox

作者：有故事的人 来源：范文网 www.wtabcd.cn/fanwen/

本文原地址：<https://www.wtabcd.cn/fanwen/zuowen/01d655825d6023b5ccf152c3c53f3f02.html>

范文网，为你加油喝彩！

这两个家伙，对我们来说，是绝对不陌生的，从winform到wpf，到asp.net，我们都会接触到这两个控件，而且我相信我们也经常使用。

listbox

先说listbox，这个其实很简单，应该说，对于所有的集合控件，都是一样的使用方法，往里面放东西就两种途径：

- 1、数据绑定；
- 2、手动添加项。

而listbox对应的项是listboxitem，说得更明白一些，它就是一个contentcontrol，就像button一样，都有一个content属性，而我们就通过这个属性来设置项里面要显示的东西。对于数据绑定而言，因为是批量化的，所以，这个时候，我们就要用上datatemplate了。

先来看看listbox的数据绑定的例子。

[html]

```
<grid background= " {staticresource applicationpagebackgroundthemebrush} " >
<listbox x:name= " list " width= " 365&#8搞笑小说推荐243;/>
</grid>
```

```
<grid background= " {staticresource applicationpagebackgroundthemebrush} " >
<listbox x:name= " list " width= " 365 " />
</grid>
```

在代码视图中，我们在mainpage类的构造函数中设置数据源，其实就是设置它的itemssource属性。

[csharp]

```
public mainpage()
{
    this.initializecomponent();
}
```

```
string[] items = { “ 夏 ”, “ 商 ”, “ 周 ”, “ 春秋战国 ”, “ 秦 ”, “ 汉 ”, “ 三国魏晋南北朝 ”,  
“ 隋 ”, “ 唐 ” };  
this.list.itemssource = items;  
}
```

```
public mainpage()  
{  
this.initializecomponent();
```

```
string[] items = { “ 夏 ”, “ 商 ”, “ 周 ”, “ 春秋战国 ”, “ 秦 ”, “ 汉 ”, “ 三国魏晋南北朝 ”,  
“ 隋 ”, “ 唐 ” };  
this.list.itemssource = items;  
}
```

然后按下f5，你能看到以下的效果。

可是，有时候，似乎只显示一个文本还不够，有可能我们用的数据源较为复杂。

好的，现在我们来伪造一些测试数据，先定义实体类。

```
[csharp]  
public class employee  
{  
public string emname { get; set; }  
public string emno { get; set; }  
public int emage { get; set; }  
  
public windows.ui.xaml.media.imaging.bitmapimage photo { get; set; }  
}
```

```
public class employee  
{  
public string emname { get; set; }  
public string emno { get; set; }  
public int emage { get; set; }
```

```
public windows.ui.xaml.media.imaging.bitmapimage photo { get; set; }  
}
```

接着弄好xaml。

```
[html]  
<listbox x:name= " listemp " selectionmode= " single " >  
<listbox.itemtemplate>  
<datatemplate>
```

```
<grid>
<grid.columndefinitions>
<columndefinition width= " auto " />
<columndefinition/>
</grid.columndefinitions>
<grid margin= " 10 " grid.column= " 1 " >
<grid.rowdefinitions>
<rowdefinition height= " auto " />
<rowdefinition height= " auto " />
<rowdefinition height= " auto " />
</grid.rowdefinitions>
<grid.columndefinitions>
<columndefinition width= " auto " />
<columndefinition/>
</grid.columndefinitions>
<textblock grid.row= " 0 " grid.columnspan= " 2 " fontweight= " bold " fontsize= " 24 "
margin= " 3,2,3,2 " text= " {binding emname} " />
<textblock text= " 工号： " grid.row= " 1 " grid.column= " 0 " />
<textblock grid.row= " 1 " grid.column= " 1 " text= " {binding emno} " />
<textblock grid.row= " 2 " grid.column= " 0 " text= " 年龄： " />
<textblock grid.row= " 2 " grid.column= " 1 " text= " {binding emage} " />
</grid>
<image grid.column= " 0 " margin= " 4 " source= " {binding photo} " stretch= " uniform " />
</grid>
</datatemplate>
</listbox.itemtemplate>
</listbox>
```

```
<listbox x:name= " listemp " selectionmode= " single " >
<listbox.itemtemplate>
<datatemplate>
<grid>
<grid.columndefinitions>
<columndefinition width= " auto " />
<columndefinition/>
</grid.columndefinitions>
<grid margin= " 10 " grid.column= " 1 " >
<grid.rowdefinitions>
<rowdefinition height= " auto " />
<rowdefinition height= " auto " />
<rowdefinition height= " auto " />
</grid.rowdefinitions>
<grid.columndefinitions>
<columndefinition width= " auto " />
<columndefinition/>
</grid.columndefinitions>
```

```
<textblock grid.row= " 0 " grid.columnspan= " 2 " fontweight= " bold " fontsize= " 24 "
margin= " 3,2,3,2 " text= " {binding emname} " />
<textblock text= " 工号： " grid.row= " 1 " grid.column= " 0 " />
<textblock grid.row= " 1 " grid.column= " 1 " text= " {binding emno} " />
<textblock grid.row= " 2 " grid.column= " 0 " text= " 年龄： " />
<textblock grid.row= " 2 " grid.column= " 1 " text= " {binding emage} " />
</grid>
<image grid.column= " 0 " margin= " 4 " source= " {binding photo} " stretch= " uniform " />
</grid>
</datatemplate>
</listbox.itemtemplate>
</listbox>
在代码视图中进行绑定。
```

```
[csharp]
list<employee> emplist = new list<employee>();
employee e1 = new employee
{
    emname = " 胡扯 ",
    emno = " hc-22556854 ",
    emage = 38,
    photo = new windows.ui.xaml.media.imaging.bitmapimage(new uri( " ms-
appx:///assets/1.jpg " ))
};
emplist.add(e1);

employee e2 = new employee
{
    emname = " 张大腿 ",
    emno = " hc-62254585 ",
    emage = 41,
    photo = new windows.ui.xaml.media.imaging.bitmapimage(new uri( " ms-appx:///assets/1.jpg " ))
};
emplist.add(e2);

employee e3 = new employee
{
    emname = " 草先生 ",
    emno = " hc-2000355462 ",
    emage = 41,
    photo = new windows.ui.xaml.media.imaging.bitmapimage(new uri( " ms-appx:///assets/1.jpg " ))
};
emplist.add(e3);

this.listemp.itemssource = emplist;
```

```
list<employee> emplist = new list<employee>();  
employee e1 = new employee  
{  
    emname = “ 胡扯 ” ,  
    emno = “ hc-22556854 ” ,  
    emage = 38,  
    photo = new windows.ui.xaml.media.imaging.bitmapimage(new uri( “ ms-appx:///assets/1.jpg ” ))  
};  
emplist.add(e1);
```

```
employee e2 = new employee  
{  
    emname = “ 张大腿 ” ,  
    emno = “ hc-62254585 ” ,  
    emage = 41,  
    photo = new windows.ui.xaml.media.imaging.bitmapimage(new uri( “ ms-appx:///assets/1.jpg ” ))  
};  
emplist.add(e2);
```

```
employee e3 = new employee  
{  
    emname = “ 草先生 ” ,  
    emno = “ hc-2000355462 ” ,  
    emage = 41,  
    photo = new windows.ui.xaml.media.imaging.bitmapimage(new uri( “ ms-appx:///assets/1.jpg ” ))  
};  
emplist.add(e3);
```

```
this.listemp.itemssource = emplist;
```

这里我们就用到了数据模板了，因为我们每一项要显示的内容有员工头像，姓名，工号，年龄多个字段，明显仅依靠一个字符串是做不到的，因此，必要时，我们出动数据模板。

可能有人会想了，这listbox默认是纵向排列的，有没有办法让它的项水平排列呢？当然有，items panel属性正是让我们来决定它的项列表如何排列的，它可以设置一个面板，如

```
[html]  
<listbox.itemspanel>  
<itemspaneltemplate>  
<stackpanel orientation= ” horizontal ” />  
</itemspaneltemplate>  
</listbox.itemspanel>
```

```
<listbox.itemspanel>  
<itemspaneltemplate>
```

```
<stackpanel orientation= " horizontal " />
</itemspaneltemplate>
</listbox.itemspanel>
```

再运行一下，你就能看到它们水平排列了。

怎么样，这三位哥们儿挺帅的吧。

接下来是手动添加项，这个就更简单了，看看下面的xaml你就懂了。

```
[html] view plaincopyprint?
<listbox>
<listboxitem>项目一</listboxitem>
<listboxitem>
<listboxitem.content>
<rectangle width= " 480 " height= " 80 " fill= " orange " />
</listboxitem.content>
</listboxitem>
<listboxitem>
<listboxitem.content>
<grid>
<grid.columndefinitions>
<columndefinition width= " auto " />
<columndefinition/>
</grid.columndefinitions>
<textblock grid.column= " 0 " text= " 请输入一个数字： " fontsize= " 25 " />
<textbox grid.column= " 1 " width= " 260 " />
</grid>
</listboxitem.content>
</listboxitem>
</listbox>

<listbox>
<listboxitem>项目一</listboxitem>
<listboxitem>
<listboxitem.content>
<rectangle width= " 480 " height= " 80 " fill= " orange " />
</listboxitem.content>
</listboxitem>
<listboxitem>
<listboxitem.content>
<grid>
<grid.columndefinitions>
<columndefinition width= " auto " />
<columndefinition/>
</grid.columndefinitions>
<textblock grid.column= " 0 " text= " 请输入一个数字： " fontsize= " 25 " />
<textbox grid.column= " 1 " width= " 260 " />
</grid>
</listboxitem.content>
</listboxitem>
</listbox>
```

运行后就是这注意力训练样：

combobox

其实与listbox是一样的，只不过它是一个下拉列表框罢了。其项列表对应着是comboboxitem类。

考虑下面的例子，我们在combobox里面放些东东，然后，当我们选择了特定项后，textblock中文本的颜色随之改变。

1、先布局ui。

[html]

```
<grid background= " {staticresource applicationpagebackgroundthemebrush} " >
<stackpanel>
<combobox x:name= " cb " width= " 275 " horizontalalignment= " left " >
<combobox.itemtemplate>
<datatemplate>
<grid>
<grid.columndefinitions>
<columndefinition width= " auto " />
<columndefinition width= " * " />
</grid.columndefinitions>
<rectangle grid.column= " 0 " width= " 25 " height= " 25 " margin= " 5 " fill= " {binding path=brush} " />
<textblock grid.column= " 1 " text= " {binding path=colordname} " verticalalignment= " center " margin= " 3,2,2,2 " />
</grid>
</datatemplate>
</combobox.itemtemplate>
</combobox>
<textblock margin= " 5,20,0,0 " fontsize= " 60 " fontweight= " black " text= " 文本颜色 " x:name= " tb " />
</stackpanel>
</grid>

<grid background= " {staticresource applicationpagebackgroundthemebrush} " >
<stackpanel>
<combobox x:name= " cb " width= " 275 " horizontalalignment= " left " >
<combobox.itemtemplate>
<datatemplate>
<grid>
<grid.columndefinitions>
<columndefinition width= " auto " />
<columndefinition width= " * " />
</grid.columndefinitions>
<rectangle grid.column= " 0 " width= " 25 " height= " 25 " margin= " 5 " fill= " {binding path=brush} " />
<textblock grid.column= " 1 " text= " {binding path=colordname} " verticalalignment= " center " margin= " 3,2,2,2 " />
</grid>
</datatemplate>
</combobox.itemtemplate>
</combobox>
<textblock margin= " 5,20,0,0 " fontsize= " 60 " fontweight= " black " text= " 文本颜色 " x:name= " tb " />
</stackpanel>
</grid>
```

接下来嘛，你猜到了，肯定是写处理代码了。

在mainpage类的构造函数中，我们为combobox添加4个可选项。并且处理它的selectionchanged事件，对于绑定，我们这次不必定义一个类这么麻烦，别忘了，.net里面有一个很灵活的类型——dynamic，再配全new关键字，我们就可以动态定义一个对象出来了。

```
[csharp]
public mainpage()
{
    this.initializecomponent();
    list<dynamic> colorlist = new list<dynamic>();
    colorlist.add(new {
        brush = new solidcolorbrush(colors.yellow),
        colorname = “ 黄色 ”
    });
    colorlist.add(new
    {
        brush = new solidcolorbrush(colors以e结尾的动词.blue),
        colorname = “ 蓝色 ”
    });
    colorlist.add(new
    {
        brush = new solidcolorbrush(colors.gray),
        colorname = “ 灰色 ”
    });
    colorlist.add(new
    {
        brush = new solidcolorbrush(colors.pink),
        colorname = “ 粉红色 ”
    });
    this.cb.itemssource = colorlist;

    // 绑定事件，对combobox的选择作出响应
    cb.selectionchanged += (a, args) =>
    {
        if (args.addeditems.count <= 0)
        {
            return;
        }
        dynamic item = (dynamic)args.addeditems[0];
        if (item != null)
        {
            this.tb.foreground = item.brush;
        }
    };

    if (cb.items.count > 0)
    {
        cb.selectedindex = 0;
    }
}

public mainpage()
{
    this.initializecomponent();
```

```
list<dynamic> colorlist = new list<dynamic>();
colorlist.add(new {
brush = new solidcolorbrush(colors.yellow),
colorname = “ 黄色 ”
});
colorlist.add(new
{
brush = new solidcolorbrush(colors.blue),
colorname = “ 蓝色 ”
});
colorlist.add(new
{
brush = new solidcolorbrush(colors.gray),
colorname = “ 灰色 ”
});
colorlist.add(new
{
brush = new solidcolorbrush(colors.pink),
colorname = “ 粉红色 ”
});
this.cb.itemssource = colorlist;

// 绑定事件，对combobox的选择作出响应
cb.selectionchanged += (a, args) =>
{
if (args.addeditems.count <= 0)
{
return;
}
dynamic item = (dynamic)args.addeditems[0];
if (item != null)
{
this.tb.foreground = item.brush;
}
};

if (cb.items.count > 0)
{
cb.selectedindex = 0;
}
}
```

更多 作文 请访问 https://www.wtabcd.cn/fanwen/list/92_0.html

文章生成doc功能，由[范文网](#)开发