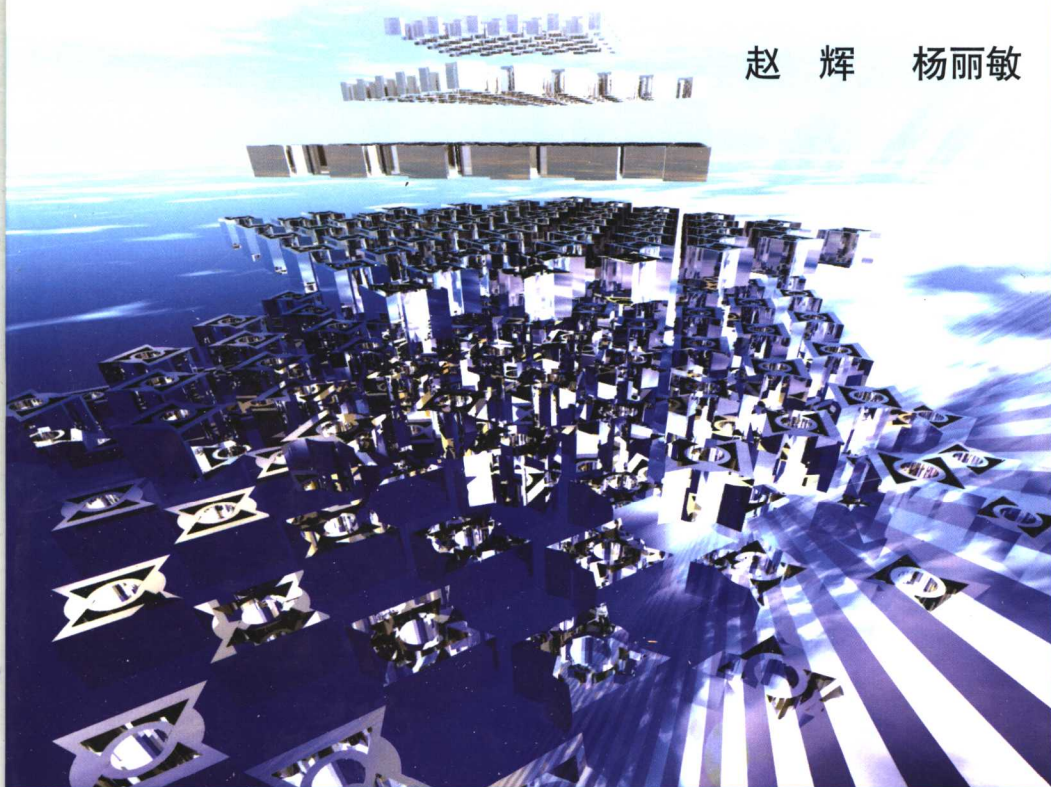


商业开发 代码库系列

ASP.NET

案例开发集锦

赵辉 杨丽敏 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

商业开发代码库系列

ASP.NET案例开发集锦

赵 辉 杨丽敏 编著

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

随着网络技术的迅猛发展,各种网站开发语言不断出现。而微软的.NET技术无疑代表了未来Internet技术的方向,ASP.NET技术就是由微软的.NET技术细化而来的,ASP.NET完全基于模块与组件,具有更好的可扩展性与可定制性,数据处理方面更是引入了许多激动人心的新技术。

本书以案例为主线,将从开发实践中产生的ASP.NET案例贯穿于ASP.NET技术的各个方面:控件应用、数据库访问、应用程序、文件操作、XML文档处理、WebService应用以及综合案例等等。通过案例的详细讲解,使读者快速掌握ASP.NET的开发方法,可以迅速达到实际应用的水平。

本书适合于广大ASP.NET技术的爱好者和专业从事网络应用开发的技术人员,同时也可作为大中专院校学生学习的辅助教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

ASP.NET案例开发集锦/赵辉,杨丽敏编著.一北京:电子工业出版社,2006.1

(商业开发代码库系列)

ISBN 7-121-01826-8

I. A… II. ①赵… ②杨… III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆CIP数据核字(2005)第117610号

责任编辑:徐云鹏 李红玉

印 刷:北京天竺颖华印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

北京市海淀区翠微东里甲2号 邮编:100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:24 字数:610千字

印 次:2006年1月第1次印刷

定 价:37.00元(含光碟1张)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换,若书店售缺,请与本社发行部联系。联系电话:010-68279077。质量投诉请发邮件至zlts@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

前 言

在网络发展日新月异的今天，随着人们不断的创新，原来的那些HTML超文本语言已经不能完全满足我们做网页的需要了。我们需要的是一个动感性、交互性都很强的网站，这样才能和广大的网友和用户进行交流。

而微软的.NET技术目前正是蒸蒸日上，作为全球软件业最大的公司，微软在.NET技术上投入了大量的人力和物力，把公司未来战略重心放在了.NET技术上。而从目前看来，.NET技术无疑代表了未来Internet技术的方向，是当今计算机技术通向信息时代的一个非常重要的里程碑。

ASP.NET又叫ASP+，但其并不仅仅是ASP的简单升级，而是微软推出的新一代Active Server Pages脚本语言。作为微软发展的新型体系结构.NET的一部分，ASP.NET的全新技术架构会让开发基于网络的应用变得更加的简单和方便。ASP.NET利用Common Language Runtime (CLR)和服务框架网络应用程序提供了一个可靠的、自动化的、可扩展的主机环境。ASP.NET也受益于CLR的集成模板，简化了应用程序的配制。另外，它提供简化应用程序开发的服务（如状态管理服务）以及高水平的编程模板（如ASP.NET Web Forms和ASP.NET Web-Service）。

ASP.NET具有良好的可扩展性与可定制性，在数据处理方面引入了更多的新技术，让ASP.NET远远超越了原有的Web开发语言，提供给Web开发人员更好的灵活性，有效缩短了Web应用程序的开发周期。

本书以ASP.NET案例为核心，详细介绍了每个案例涉及到的ASP.NET技术知识以及相关的开发方法和经验，使读者迅速掌握相应的内容，达到实战的水平。同时，本书中的案例均调试运行通过，同时力求注释明晰，方便读者将案例应用到实际的工作中去。

本书分为8章，分别介绍了ASP.NET技术所涉及到的控件应用、数据库访问、应用程序、文件操作、操作XML、WebService应用、高级应用以及综合案例等内容。其中案例的具体内容有可能存在穿插，但案例内容上力求循序渐进的原则。另外，由于ASP.NET的开发语言众多，所以实际案例的开发语言也不尽相同。本书中的案例主要使用VB和C#开发，由此也可认识到ASP.NET的开放性。

本书拥有大量的案例，使读者在实践中学习ASP.NET技术，并可以很快达到一个较高的应用层次。作者长期使用ASP.NET，对该Web开发语言的特点、功能及应用有较深入的理解和体会，因此本书力求能够兼顾各方人士的不同需求。

另外，本书的配套光碟收录了书中各个章节的案例的源代码，读者通过Windows中的“Internet服务管理器”给默认站点添加一个虚拟目录，将案例程序复制到所建的虚拟目录下，就可以运行（对于需要配置的，请参阅本书中相关的案例说明）。

无论是ASP.NET的使用新手，还是已经有相当经验的Web开发高手，在本书中都可以找到自己所需要的内容。随着本书案例的学习，希望读者可以迅速高效地掌握ASP.NET的开发方法和经验。

在本书的编写过程中得到了电子工业出版社及北京美迪亚电子信息有限公司的大力协助，在此表示衷心的感谢。本书由赵辉、杨丽敏编写，赵忠诚、王秀荣、李艳君、赵敏、李军琪、李逸凡、姚忠朋、杨丽明、姚胤含等同志也参与了书稿的编写和整理工作。

由于本书的编写时间仓促加之编者水平有限，书中错误与不妥之处在所难免，恳请读者批评指正。

编者

目 录

第1章 ASP.NET控件应用	1
案例1 页面数据显示的多样性示例	1
案例2 多按钮事件的处理	8
案例3 网站个人信息注册(数据有效性验证)	12
案例4 网页中一个可伸缩导航面板的自定义控件	17
案例5 自定义页面计算器	22
案例6 自定义文件上传控件	28
本章总结	41
第2章 ASP.NET数据库访问	42
案例1 维护一个学校的学生成绩	42
案例2 学生成绩分页汇总	47
案例3 网站信息检索显示	51
案例4 一个简单的Q&A	67
案例5 用户/角色管理模块	81
本章总结	102
第3章 ASP.NET应用程序	103
案例1 配置文件Global.asax应用示例	103
案例2 网站用户登录时使用Cookie	133
案例3 显示网站在线人数	144
本章总结	155
第4章 ASP.NET的文件操作	156
案例1 操作本地文件示例	156
案例2 读取二进制文件	158
案例3 服务器端文件管理	161
本章总结	181
第5章 ASP.NET操作XML	182
案例1 读取人员花名册	182
案例2 在ASP.NET中执行XSL转换	185

案例3 在XML中进行读取和写入	186
本章总结	193
第6章 ASP.NET的WebService应用	194
案例1 学生考试成绩查询显示	194
案例2 使用参数获得公共新闻网站的新闻列表	199
案例3 向网上学校定制学习计划和内容	201
本章总结	204
第7章 ASP.NET的高级应用	205
案例1 获得服务器的系统信息	205
案例2 网络页面的跟踪	210
案例3 代码与网页框架分离1	213
案例4 代码与网页框架分离2	216
案例5 使用页面数据缓存读取客户信息	229
本章总结	234
第8章 ASP.NET综合案例	235
案例1 一个简单的社区论坛	235
案例2 员工基本信息管理	307
本章总结	377

第1章

ASP.NET控件应用



本章内容:

- ☞ 页面数据显示的多样性示例
- ☞ 多按钮事件的处理
- ☞ 网站个人信息注册 (数据有效性验证)
- ☞ 网页中一个可伸缩导航面板的自定义控件
- ☞ 自定义页面计算器
- ☞ 自定义文件上传控件
- ☞ 本章总结

案例1 页面数据显示的多样性示例

案例运行效果与操作

本案例主要演示了ASP.NET服务器控件中, 最常使用到的两种控件: DataList和DataGrid。这两种控件属性各有不同, 从而使用方法上有较大差异。

图1-1是使用了DataList控件进行数据的排列的效果, 而且可以修改数据。用户在单击了“详细”链接后, 可以更新图书的书名和价格。

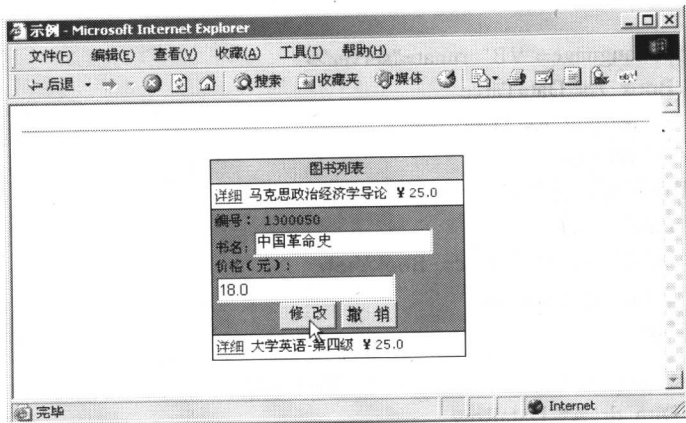


图1-1

图1-2是使用了DataGrid控件排列数据的效果，用户订购的图书在另一张表中显示出来，并且可以退订图书。

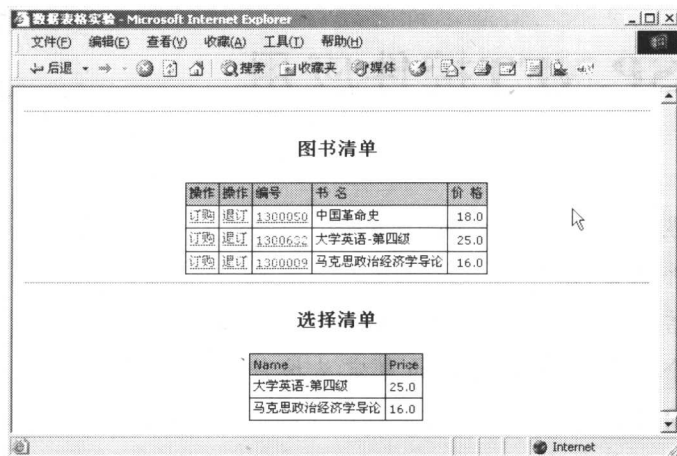


图1-2

制作要点

1. 数据列表显示控件 (DataList) 中多种模板的控制处理。
2. 数据表格显示控件 (DataGrid) 中数据显示的控制处理。

步骤详解

在设置的虚拟目录下，建立aspx文件即可。

程序源代码与解释

1. DataListEdit.aspx页面:

```
<%@ Import Namespace="System.Data" %>
<html>
    <script language="VB" runat="server">
        dim Book As DataTable
        dim BookView As DataView
        dim i As Integer
        '设置数据源，并绑定
        Sub BindList()
            DataList1.DataSource= BookView
            DataList1.DataBind
        End Sub

        Sub Page_Load(s As Object, e As EventArgs)
            Dim dr As DataRow
            '如果没有连接变量session_book，定义数据表Book，并载入实验数据
            if session("session_Book") is Nothing then
                Book = New DataTable()
```

```

Book.Columns.Add(new DataColumn("num", GetType(string)))
Book.Columns.Add(new DataColumn("name", GetType(String)))
Book.Columns.Add(new DataColumn("price", GetType(String)))
session("session_Book") = Book
'载入测试数据
dr = Book.NewRow()
    dr(0)=1300050
    dr(1) = "中国革命史"
    dr(2) = "18.0"
Book.Rows.Add(dr)
dr = Book.NewRow()
    dr(0)=1300632
    dr(1) = "大学英语-第四级"
    dr(2) = "25.0"
Book.Rows.Add(dr)
dr = Book.NewRow()
    dr(0)=1300009
    dr(1) = "马克思政治经济学导论"
    dr(2) = "16.0"
Book.Rows.Add(dr)
' 如果有session_book变量, 直接引用
Else
    Book = session("session_Book")
end if
'产生数据视图, 并按num字段排序
BookView = New DataView(Book)
BookView.Sort="num"
'初次需绑定数据源
if Not IsPostBack then
BindList
End If

End Sub
'编辑处理函数
Sub DataList_EditCommand(sender As Object, e As DataListCommandEventArgs)
    DataList1.EditItemIndex = e.Item.ItemIndex
    BindList
End Sub
'取消处理函数
Sub DataList_CancelCommand(sender As Object, e As DataListCommandEventArgs)
    DataList1.EditItemIndex = -1
    BindList
End Sub
'更新处理函数
Sub DataList_UpdateCommand(sender As Object, e As DataListCommandEventArgs)
    Dim lbl1 As Label = e.Item.FindControl("lblNum")
    Dim txt2 As TextBox = e.Item.FindControl("txtBook")
    Dim txt3 As TextBox = e.Item.FindControl("txtPrice")

```

```

        dim strNum as String
        dim strBook as String
        dim strPrice as String
        strNum=lbl1.text
        strBook=txt2.text
        strPrice=txt3.text
        '用先删除再插入的方式，实现数据的更新操作
        BookView.RowFilter = "num=" & strNum & ""
        If BookView.Count > 0 Then
            BookView.Delete(0)
        End If
        BookView.RowFilter = ""
        dim dr as DataRow=Book.NewRow()
        dr(0) = strNum
        dr(1) = strBook
        dr(2) = strPrice
        Book.Rows.Add(dr)
        DataList1.EditItemIndex = -1
        BindList
    End Sub
</script>
<head>
<title>
示例
</title>
</head>
<body>
<center>
<hr>
<form runat=server>
<font face="Verdana" size="-1">
<asp:DataList id="DataList1" runat="server"
    BorderColor="black"
    BorderWidth="1"
    GridLines="Both"
    CellPadding="3"
    CellSpacing="0"
    Font-Name="Verdana"
    Font-Size="8pt"
    Width="220px"
    HeaderStyle-BackColor="#bbbbdd"
    AlternatingItemStyle-BackColor="Gainsboro"
    EditItemStyle-BackColor="#11aacc"
    OnEditCommand="DataList_EditCommand"
    OnUpdateCommand="DataList_UpdateCommand"
    OnCancelCommand="DataList_CancelCommand"
>
    <HeaderTemplate>

```

```

<center><h>图书列表</h></center>
</HeaderTemplate>
<ItemTemplate>
    <asp:LinkButton id="button1" runat="server" Text="详细"
CommandName="edit" />
    <%# Container.DataItem("name") %>
    ¥<%# Container.DataItem("price") %>
</ItemTemplate>
<EditItemTemplate>
    编号:
    <asp:Label id="lblNum" runat="server" Text='<%# Container.DataItem
("num") %>' /><br>
    书名:
    <asp:TextBox id="txtBook" runat="server" Text='<%#
Container.DataItem("name") %>' /><br>
    价格(元):
    <asp:TextBox id="txtPrice" runat="server" Text='<%# DataBinder.Eval
(Container.DataItem, "price") %>' />
    <br>
<center>
    <asp:Button id="button2" runat="server" Text="修 改"
CommandName="update" />
    <asp:Button id="button3" runat="server" Text="撤 销"
CommandName="cancel" />
</center>
</EditItemTemplate>
</asp:DataList>
</font>
</form>
</center>
</body>
</html>

```

2. DataGrid.aspx页面:

```

<%@ Import Namespace="System.Data" %>
<%@ Page Language="VB" Debug="true" %>
<html>
<script language="VB" runat="server">
    dim Order as DataTable
    dim OrderView as DataView
    '对数据表格1创建数据表,并返回数据视图
    Function LoadData() As ICollection
        Dim dt As DataTable
        Dim dr As DataRow
        Dim i As Integer
        '创建数据表
        dt = New DataTable

```

```

dt.Columns.Add(new DataColumn("Num", GetType(String)))
dt.Columns.Add(new DataColumn("Name", GetType(String)))
dt.Columns.Add(new DataColumn("Price", GetType(String)))
'载入测试数据
    dr = dt.NewRow()
        dr(0)=1300050
        dr(1) = "中国革命史"
        dr(2) = "18.0"
    dt.Rows.Add(dr)
    dr = dt.NewRow()
        dr(0)=1300632
        dr(1) = "大学英语-第四级"
        dr(2) = "25.0"
    dt.Rows.Add(dr)
    dr = dt.NewRow()
        dr(0)=1300009
        dr(1) = "马克思政治经济学导论"
        dr(2) = "16.0"
    dt.Rows.Add(dr)
LoadData = New DataView(dt)
End Function
'页面初始化, 分别对DataGrid1和DataGrid2绑定数据源
Sub Page_Load(sender As Object, e As EventArgs)

    If Session("session_order") is Nothing Then
        Order = New DataTable()
        Order.Columns.Add(new DataColumn("Name", GetType(string)))
        Order.Columns.Add(new DataColumn("Price", GetType(string)))
        Session("session_order") = Order
    Else
        Order = Session("session_order")
    End If
    OrderView = New DataView(Order)
    DataGrid2.DataSource = OrderView
    DataGrid2.DataBind
    If Not IsPostBack Then
DataGrid1.DataSource = LoadData()
        DataGrid1.DataBind
    End If
End Sub
'对ButtonColumns的处理函数集合
Sub Grid_Command(sender As Object, e As DataGridCommandEventArgs)
    Dim dr As DataRow = order.NewRow()
    Dim Cell1 As TableCell = e.Item.Cells(3)
    Dim Cell2 As TableCell = e.Item.Cells(4)
    Dim name As String = Cell1.Text
    Dim price As String = Cell2.Text
    If e.CommandSource.CommandName = "Add" Then

```

```

        dr(0) = name
        dr(1) = price
        order.Rows.Add(dr)
    Else
        OrderView.RowFilter = "name=" & name & ""
        If OrderView.Count > 0 Then
            OrderView.Delete(0)
        End If
        OrderView.RowFilter = ""
    End If
    DataGrid2.DataBind()
End Sub
</script>
<head>
<title>
数据表格实验
</title>
</head>
<body>
<center>
<hr>
<form runat=server>
<h3><b>图书清单</b></h3>
<ASP:DataGrid id="DataGrid1" runat="server"
    BorderColor="black"
    BorderWidth="1"
    GridLines="Both"
    CellPadding="3"
    CellSpacing="0"
    Font-Name="Verdana"
    Font-Size="8pt"
    HeaderStyle-BackColor="#aaaadd"
    AutoGenerateColumns="false"
    OnItemCommand="Grid_Command">
    <Columns>
        <asp:ButtonColumn HeaderText="操作" Text="订购" CommandName="Add" />
        <asp:ButtonColumn HeaderText="操作" Text="退订" CommandName="Remove" />
        <asp:HyperLinkColumn
            HeaderText="编号"
            DataNavigateUrlField="Num"
            DataNavigateUrlFormatString="DataGridResult.aspx?id={0}"
            DataTextField="Num"
            Target="_new"
        />
        <asp:BoundColumn HeaderText="书 名" DataField="Name" />
        <asp:BoundColumn HeaderText="价 格" DataField="Price" DataFormatString
           ="{0:c}" ItemStyle-HorizontalAlign="right" />
    </Columns>
</ASP:DataGrid>
</form>

```

```

</Columns>
</asp:DataGrid>
<hr>
<h3><b>选择清单</b></h3>
<ASP:DataGrid id="DataGrid2" runat="server"
    BorderColor="black"
    BorderWidth="1"
    CellPadding="3"
    Font-Name="Verdana"
    Font-Size="8pt"
    HeaderStyle-BackColor="#aaaadd"
/>
</form>
</center>
</body>
</html>

```

3. DataGridResult.aspx 页面:

```

<html>
<head>
<title>
数据表格链接测试实验
</title>
<script language="VB" runat="server">
    Dim num As String
    Sub Page_Load(sender As Object, e As EventArgs)
        num=Request.QueryString("id")
    End Sub
</script>
</head>
<body bgcolor=#ccccff>
<center>
<hr>
<p></p>
<h4>您选择的是 第<u> <%= num %></u>本图书</h4>
</body>
</html>

```

案例2 多按钮事件的处理

案例运行效果与操作

本案例通过一种在实践中经常使用的多种选择的模式，展示了ASP.NET在多个按钮事件处理上的优势。如果使用常用的JavaScript代码处理该页面，则需要大量的代码，而且调

试起来比较繁琐。而使用ASP.NET的按钮事件处理机制，则非常明晰而且快捷地解决了此类问题。

用户可以将左侧的选项转到右侧，可以单选，也可以批量处理，如图1-3所示。同时，如图1-4所示，可以通过“提交”按钮，显示所有右侧列表中的项目。

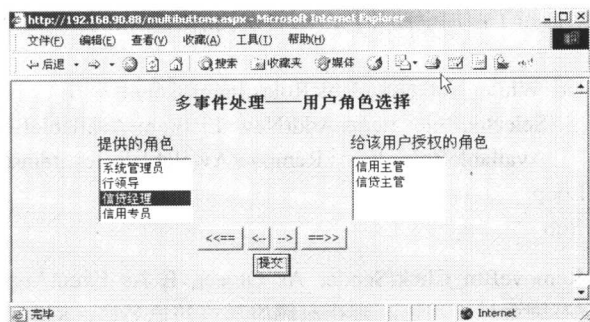


图1-3

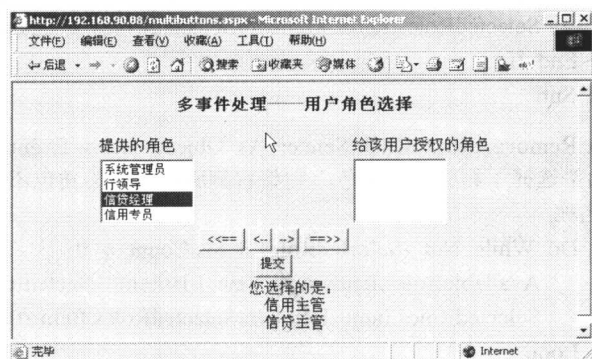


图1-4

制作要点

1. 在一个页面中，处理多个事件的方法。
2. 理解ASP.NET中事件处理的机制。

步骤详解

在设置的虚拟目录下，建立该aspx文件。

程序源代码与解释

multibuttons.aspx页面：

```
<html>
<script language="VB" runat="server">
    Sub AddBtn_Click(Sender As Object, E As EventArgs)
        ' 如果在左侧选择了角色，则在右侧添加该角色名称，同时在左侧删除该角色名称
        If Not (AvailableRoles.SelectedIndex = -1)
```

```

SelectedRoles.Items.Add(New ListItem(AvailableRoles.SelectedItem.Value))
    AvailableRoles.Items.Remove(AvailableRoles.SelectedItem.Value)
End If
End Sub

Sub AddAllBtn_Click(Sender As Object, E As EventArgs)
    ' 如果选择了添加全部角色，则将左侧所有剩余的角色添加到右侧，同时在左侧
    删除该角色名称
    Do While Not (AvailableRoles.Items.Count = 0)
        SelectedRoles.Items.Add(New ListItem(AvailableRoles.Items(0).Value))
        AvailableRoles.Items.Remove(AvailableRoles.Items(0).Value)
    Loop
End Sub

Sub RemoveBtn_Click(Sender As Object, E As EventArgs)
    ' 如果在右侧选择了角色，则在左侧添加该角色名称，同时在右侧删除该角色名称
    If Not (SelectedRoles.SelectedIndex = -1)
        AvailableRoles.Items.Add(New ListItem(SelectedRoles.SelectedItem.Value))
        SelectedRoles.Items.Remove(SelectedRoles.SelectedItem.Value)
    End If
End Sub

Sub RemoveAllBtn_Click(Sender As Object, E As EventArgs)
    ' 如果选择了删除全部角色，则将右侧所有剩余的角色添加到左侧，同时在右侧
    删除该角色名称
    Do While Not (SelectedRoles.Items.Count = 0)
        AvailableRoles.Items.Add(New ListItem(SelectedRoles.Items(0).Value))
        SelectedRoles.Items.Remove(SelectedRoles.Items(0).Value)
    Loop
End Sub

Sub result(Sender As Object,E As EventArgs)
    dim tmpStr as String
    tmpStr="<br>"
    Do While Not (SelectedRoles.Items.Count = 0)
        ' 将右侧所有的角色名称添加到字符串中
        tmpStr=tmpStr & SelectedRoles.items(0).value & "<br>"
        SelectedRoles.items.remove(SelectedRoles.items(0).value)
    Loop
    ' 将所有选择的角色名称在页面中显示
    SelectedItems.Text="您选择的是: "&tmpStr
End Sub
</script>

<body bgcolor="#ccccff">
<center>
    <h3><font face="Verdana">多事件处理——用户角色选择</font></h3>
</center>
<center>

```