

深入体验 A SP.N et

焦荐沧 龙晓敏 主编

西安电子科技大学出版社

2001

内 容 简 介

本书以通俗、简明的语言着重介绍了基于交互式网站开发的 ASP.Net 技术。本书结合大量的开发实例，剖析了如何基于 ASP.Net 平台构建高速、高效的电子商务平台，并将 ASP.Net 与传统的 ASP 技术作了比较，充分展示了 ASP.Net 的强大功能和用 ASP.Net 技术开发互动性商务网站的优越性。

本书侧重于介绍构成网页的基本元素——控件，以控件的介绍为主线并搭配丰富的实例进行讲解。本书适合于 ASP.Net 初学者，同时还可以作为广大工程技术人员、管理开发人员学习 ASP.Net 的参考指南。

图书在版编目(CIP) 数据

深入体验 ASP.Net / 焦荐沧, 龙晓敏主编.

—西安: 西安电子科技大学出版社, 2001. 10

ISBN 7-5606-1063-3

. 深... . 焦... 龙... . 主页制作-应用软件, ASP.Net . TP393.092

中国版本图书馆 CIP 数据核字(2001)第 063102 号

责任编辑 臧延新 龙 晖

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)8227828 邮编 710071

http: //www.xduph.com E-mail: xdupfb@pub.xaonline.com

经 销 新华书店

印 刷 西安文化彩印厂

版 次 2001 年 10 月第 1 版 2001 年 10 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 24.75

字 数 593 千字

印 数 1~4 000 册

定 价 34.00 元

ISBN 7-5606-1063-3/TP · 0530

如有印装问题可调换

本书封面贴有西安电子科技大学出版社的激光防伪标志, 无标志者不得销售。

前 言

ASP(Active Server Pages) 是微软推出的开发 Web 动态网站的重要、快速、有效的工具。在网络编程中, 它的出现早于 JSP、PHP。随着 Microsoft®.Net 技术的成熟, Microsoft.Net 基于 XML 的松散型编程模式引入了一个新概念, 创造基于 XML 的网络服务。Microsoft.Net 编程模式提供了一个内在机制, 可使所有网站和服务商结成同盟, 协调工作。ASP.Net 作为下一代交互式的商务 Web 开发平台, 具有得天独厚的优势。

为了体现 ASP.Net 支持多种语言, 本书中的一些实例分别使用 VB、C#、JavaScript 三种语言进行编写, 以满足具有不同编程习惯的用户之要求, 使其可以在参考的过程中各取所需。尤其是在访问 Web 数据库方面, ASP.Net 提供了强大的数据库访问功能。例如分页浏览 Web 数据库中的数据, 只需要一个控件就可以实现 ASP 近 100 行代码所能实现的分页浏览功能。作为微软面向 21 世纪的战略产品, ASP.Net 带来的不仅仅是上面所讲的功能, 更多的是希望读者在学习运用过程中去细细体会。

本书共分为七章, 前面三章主要介绍 ASP.Net 的背景, 以及 .Net Framework 的安装及运行, 同时写出了第一个基于 ASP.Net 的程序。第 4 章主要介绍 ASP.Net 内置服务器的 HTML 控件, 给出了使用语法, 同时列举了诸多实例, 供参考借鉴。第 5 章是本书的重点, 主要介绍 ASP.Net 内置服务器的 Web 控件。由于基于 Web 控件的编程, 使得 ASP.Net 在网页编写程序方面有重大突破。同时列出了丰富的实例, 并加以注释, 配合基于控件的语法介绍, 易于使用户理解和学习。尤其是对于涉及到访问数据库方面的 Web 控件, 由于其功能强大, 使开发过程方便快捷, 因此在本书中作了特别详细的讲解。第 6 章主要对数据库的存取技术作了介绍, 并且介绍了在 ASP.Net 中访问 Web 数据库的初步知识, 同时还介绍了目前最流行的 XML 语言, 以及如何访问 XML 格式的数据库。第 7 章是综合实例, 是基于控件的综合应用, 同时也介绍了如何排除网页程序中的错误, 以及利用 Trace 功能检测网页程序的运行状态。当然, ASP.Net 不仅仅局限于本书中讲解的内容, 希望读者在实际运用过程中, 真正理解 ASP.Net 的内涵。

另外, 本书中采用了大量的语法说明和语法, 其中有部分斜体的英文, 它表示这部分内容可由用户任意录入。

本书由焦荐沧和龙晓敏组织编写, 参加本书编写工作的还有陈晓玲、彭涛、陈晓琴、陈奕廷、焦英芹、焦国良、毛翔、王坤、张强、代亚勇、赵莉丽、邓

勋、李婕、黄新民、刘旭升、魏军、张建军、于建忠、程艳军、曹发辉、许海亮、任贺靖、陈世超、李红月、高静娟、王宁、闫桂玲、张爱玲、许增伟、陈云、丁文兵、伍华聪、梁亚新、李振捷、裴威里、郑璟、吴长河、龙晓东、李俊锋等。本书的编写还得到了很多朋友的帮助，在此一并表示感谢。

如果读者需要本书实例的源代码或配套光盘，请与我们联系。我们的电子邮件地址是kj510@263.net。

由于时间仓促，加之我们水平有限，书中错误和缺点在所难免，恳请广大读者批评指正。

编 者

2001 年 7 月

目 录

第 1 章 ASP.Net 简介	1	3.2.3 编写可以显示结果的 Web Form.....	32
1.1 Microsoft.Net 背景.....	1	3.3 使用服务器控件编写 Web Form 页面.....	38
1.2 ASP.Net 与 JSP、PHP 的比较	2	3.3.1 使用服务器控件.....	38
1.2.1 ASP.Net 的特点.....	2	3.3.2 使用<asp:adrotator>控件举例.....	41
1.2.2 与 JSP 的比较.....	5	3.4 本章小结.....	44
1.2.3 与 PHP 的比较.....	5		
1.3 从 ASP 到 ASP.Net	6	第 4 章 ASP.Net 的内置服务器控件	
1.3.1 ASP 综述	6	HTML 控件	45
1.3.2 从 ASP 到 ASP.Net 编写第一个程序	7	4.1 HTML 控件简介	45
1.4 ASP.Net 的新特点	10	4.1.1 Attributes 属性.....	46
1.4.1 Web Forms	10	4.1.2 Disable 属性	47
1.4.2 ASP.Net 控件	12	4.1.3 InnerHtml 及 InnerText 属性.....	47
1.5 本章小结	17	4.1.4 Style 属性.....	49
第 2 章 .Net Framework 的安装及运行	18	4.2 HtmlAnchor 控件	50
2.1 .Net Framework 的安装准备	18	4.3 HtmlButton 控件	60
2.1.1 运行机制不同	18	4.4 HtmlForm 控件	65
2.1.2 要求的配置不同	18	4.5 HtmlGenericControl 控件	70
2.2 .Net Framework 的安装过程.....	19	4.6 HtmlImage 控件	73
2.2.1 Internet Explorer 5.5 的安装	19	4.7 HtmlInputButton 控件	77
2.2.2 .Net Framework 的安装.....	20	4.7.1 HtmlInputButton 控件举例一	78
2.3 进入 ASP.Net.....	22	4.7.2 HtmlInputButton 控件举例二	83
2.3.1 安装步骤.....	22	4.8 HtmlInputCheckBox 控件.....	85
2.3.2 安装时的常见问题.....	23	4.9 HtmlInputFile 控件.....	90
2.4 本章小结	24	4.10 HtmlInputHidden 控件.....	95
第 3 章 ASP.Net 的 ASP Web Form		4.11 HtmlInputImage 控件	100
快速入门	25	4.12 HtmlInputRadioButton 控件	105
3.1 什么是 ASP.Net 的 Web Forms	25	4.13 HtmlInputText(Password、Text)	
3.2 编写第一个 Web Form 页面	26	控件	110
3.2.1 编写第一个 Web Form 页面.....	26	4.14 HtmlSelect 控件	115
3.2.2 使用 ASP 的<% %>块	27	4.15 HtmlTable、HtmlTableCell 和	
		HtmlTableRow 控件	125
		4.16 HtmlTextArea 控件.....	132
		4.17 本章小结.....	136

第 5 章 ASP.Net 的内置服务器控件

——Web 控件	137
5.1 Web 控件简介	137
5.2 AdRotator 控件	138
5.3 Button Web 控件	141
5.4 Calenda Web 控件	150
5.5 CheckBox Web 控件	170
5.6 CheckBoxList Web 控件	173
5.7 CompareValidator Web 控件	179
5.8 CustomValidator Web 控件	185
5.9 DataGrid Web 控件	189
5.9.1 使用 DataGrid Web 控件浏览数据	192
5.9.2 使用 DataGrid Web 控件浏览数据 更改表头	195
5.9.3 使用 DataGrid Web 在数据中 加入链接	199
5.9.4 使用 DataGrid Web 控件用 Session 临时存储数据	205
5.9.5 给 DataGrid Web 控件中加入模板	212
5.9.6 使用 DataGrid Web 控件编辑数据	219
5.9.7 使用 DataGrid Web 控件隐藏 一系列数据	226
5.9.8 使用 DataGrid Web 控件设置 数据排序	232
5.9.9 使用 DataGrid Web 控件自定义 排序	235
5.9.10 使用 DataGrid Web 控件分页 显示数据	242
5.9.11 使用 DataGrid Web 控件详细地 数据分页	247
5.10 DataList Web 控件	256
5.11 DropDownList Web 控件	270
5.12 Hyperlink Web 控件	273
5.13 Image Web 控件	277
5.14 ImageButton Web 控件	280
5.15 Label Web 控件	290
5.16 LinkButton Web 控件	292
5.17 ListBox Web 控件	294

5.18 Panel Web 控件	300
5.19 RadioButton Web 控件	305
5.20 RadioButtonList Web 控件	308
5.21 RangeValidator Web 控件	313
5.22 RegularExpressionValidator Web 控件	318
5.23 Repeater Web 控件	323
5.24 Table、TableRow 和 TableCell 控件	327
5.25 TextBox Web 控件	329
5.26 ValidationSummary Web 控件	330
5.27 本章小结	334

第 6 章 ASP.Net 数据库存取技术

6.1 ADO+ 对象	335
6.1.1 在 ASP.Net 中访问数据库	335
6.1.2 DataSetCommand 对象与 Command 对象	336
6.2 DataReader 对象	337
6.3 DataSet 对象	340
6.4 DataView 对象	346
6.5 Command 对象	349
6.6 XML 与 DataSet 对象	355
6.6.1 什么是 XML	356
6.6.2 XML 格式的数据库文件	362
6.7 本章小结	366

第 7 章 实例介绍

7.1 留言簿	367
7.2 客户注册系统	371
7.3 文件上传	377
7.4 发送 E-mail	379
7.5 差错及排除错误	382
7.5.1 设定 config.web 区段	382
7.5.2 使用 Trace 追踪功能	383
7.5.3 查错工具(Debugger)	386
7.5.4 ASP.Net 中的 Session 对象	389
7.5.5 提高执行效率的 Cache	389
7.6 本章小结	390

第 1 章

ASP.Net 简介

主要内容：

- | Microsoft.Net 背景
- | 与 JSP、PHP 的比较
- | 从 ASP 到 ASP.Net

本章将用最简洁的说明来介绍 ASP.Net 出现的背景、与 PHP、JSP 的比较以及从 ASP 到 ASP.Net 的升级，并使用 ASP.Net 编写第一个网页。

1.1 Microsoft.Net 背景

华盛顿州 Redmond 2000 年 6 月 22 日消息：微软公司今天在论坛 2000 上发布了名为 Microsoft.Net 的新一代平台。有了它，以新网络工具和可编程网络服务的有力组合为特征的第三代因特网将可以让所有的开发商、公司用户和消费者都得到更多的益处。基于诸如 XML 和 SOAP 等因特网标准的 Microsoft.Net 融合了当今最优秀的计算和通信方案，能够促进因特网从基于 HTML 协议的单纯陈述飞跃到基于 XML 标准的可编程信息。通过让消费者随心所欲地对网络进行剪裁，Microsoft.Net 服务程序、产品和工具将使公司得以高效、安全地与消费者和商业伙伴进行沟通。这为消费者提供了一种完整、安全和随意的网络体验。

Microsoft.Net 的出现意味着信息工作者可以只用一种简单的界面就可以编写、浏览、编辑和分享信息，而且还得到了功能强大的信息管理工具，无论身处何地、通过何种设备，都可以随意使用。同时，因为其文件以 XML 格式存在，所以整个公司以至于所有因特网用户都可以方便地查找和使用其中的信息。任何规模的公司都可以使用相同的工具与供应商、商业伙伴和客户高效地沟通和分享信息，并且能够创造出更多的可以与原有系统、客户端程序和其他因特网资源协调一致的分布式服务程序。所有这些，都可以帮助用户抓住新的商机并更有效地管理公司。

ASP 现在被从头到脚地重建了，结果就是 ASP.Net。它并不只是 ASP 4.0，而是一个用于 Web 开发的全新框架，其中包含了许多新的特性。ASP.Net 提供了更易于编写、结构更清晰的代码，这些代码能很容易地进行再利用和共享；ASP.Net 使用编译后的语言，从而提

高了性能，增大了伸缩性；ASP.NET 使用 Web 表单，使开发更直观。利用面向对象技术可促进组件的再利用。另外，ASP.NET 中还包括页面事件、Web 控件、缓冲技术以及服务器控件和对数据捆绑的改进。供 ASP.NET 使用的库以及在 Microsoft.NET 框架中允许通过 Web 使用客户商用函数，为程序员提供了更多新的开发机会。

1.2 ASP.NET 与 JSP、PHP 的比较

互联网在其短暂的历史中，已经从一个大部分显示静态信息的网络演化到对股票进行交易和进行购物操作的一个基础设施。在各种各样的应用程序中，对于可能使用的基于 Web 的客户端，看上去并没有任何限制。

基于浏览器客户端的应用程序与传统的基于客户机/服务器的应用程序相比较，具有几个优点，包括几乎没有限制的客户端访问和极其简化的应用程序部署和管理（要更新一个应用程序，管理人员只需要更改一个基于服务器的程序，而不是成千上万的安装在客户端的应用程序）。这样，软件工业正迅速地向建造基于浏览器客户端的多层次应用程序迈进。这些快速增长的精巧的基于 Web 的应用程序要求开发技术上的改进。静态 HTML 对于显示相对静态的内容是不错的选择；新的挑战在于创建交互的基于 Web 的应用程序，在这些程序中，页面的内容是基于用户的请求或者系统的状态，而不是预先定义的文字。对于这个问题的一个早期解决方案是使用 CGI-BIN 接口；开发人员编写与接口相关的单独的程序，以及基于 Web 的应用程序，后者通过 Web 服务器来调用前者。这个方案有着严重的扩展性问题——每个新的 CGI 要求在服务器上新增一个进程。如果多个用户并发地访问该程序，则这些进程将消耗该 Web 服务器所有的可用资源，并且系统性能会降低到极其低下的地步。

1.2.1 ASP.NET 的特点

现有的 ASP 应用程序，不管将它们写得怎么样，其中总是充满了长长短短的代码。而 ASP.NET 的代码不仅更容易编写，而且更加简洁，比 ASP 代码更易于阅读。同时，ASP.NET 代码的结构方式提高了其再利用性和共享性。

ASP.NET 改进了配置、伸缩性、安全性和可靠性。对于简单的 ASP 应用程序，配置其实并不算什么问题，但是当移植到一个利用组件的 N 层结构中时就会遇到问题。当对这些应用程序进行配置和维护时，DLL 陷阱问题（例如组件注册、版本、锁定的 DLL 等）就会出现。ASP.NET 中则取消了组件注册及 DLL 锁定，全面使用了 XML 配置文件，从而解决了这个问题。这样只需要执行复制文件的工作就能配置一个 Web 应用程序。

浏览器兼容性问题常常困扰着 ASP 开发人员。他们为了解决此类问题，要么为低一级的浏览器编写代码，比如采用 HTML3.2，要么就要限制页面的浏览范围。无线应用程序协议 WAP 设备的引入加剧了兼容性问题。但是，ASP.NET 却较好地解决了这一问题，这一过程可通过 Web 表单了解到。

ASP.NET 造就了一类新的 Web 应用程序。目前的 Web 应用程序通常都是同一种模式：一套线性应用程序，然后将逻辑植入其中之一。ASP.NET 允许开发人员打破这种单一模式，创造出动态感和伸缩性更强的应用程序，这种应用程序能够更好地满足公司的商业需求，

并提供一个更加丰富的开发环境。

这时读者可能会想，虽然这些目标都很高尚，但是它们会使编写应用程序变得困难。然而恰恰相反，用 ASP.Net 进行开发要简单得多！

现有的 ASP 版本是基于脚本语言的，如 VBScript 和 JavaScript。尽管脚本语言有两大缺陷：属解释型，分类不严格，但 ASP.Net 没有完全抛弃脚本语言的概念，而引入了对全编译语言的支持，使用户能够用 Visual Basic 编写自己的服务器端代码，例如：

```
<script language="vb"runat="server">
```

Visual Basic 的一大优势是其对严格分类变量的支持，因此以下的代码在 ASP.Net 中是成立的：

```
Dim FirstName As String
```

除了 Visual Basic 和 C++，还可以使用最新的 Microsoft 语言 C#来编写服务器端代码。C#将 C++中不实用的部分去掉了，使之更容易理解。用 ASP.Net 编译动作，可使动作在页面第一次装载时发生。即便代码是脚本语言，它也会在执行前被编译，因此 JavaScript 代码页面的性能也会得到提高。实际上，这是 .Net 框架的一个基本的新特征。以前版本的语言编译器对数据类型和对象是区别对待的，这就意味着惟一可以进行跨语言开发的就是创建 COM 对象。Microsoft .Net 框架中的通用语言运行时间环境 CLR 允许对已经用通用运行时间支持编译过的任何代码进行紧密的交互作用，这就是新的 Visual Basic 和 C#编译器将要做的：创建能在运行时间中被管理的代码。

这样做的最大优点是使得真正的跨语言开发成为可能。有了通用运行时间环境 CLR，可以用 C#创建对象，并在 Visual Basic 中通过继承功能对它们进行扩展。现在的 Visual Basic .Net 支持继承功能，这样就可以在 C#中编写组件，然后在 Visual Basic 中将它们设置成子集。虽然 Visual Basic、C#和脚本语言是在开始的框架结构中被支持的语言，但是稍后其他语言，如 Smalltalk、Eiffel、Pascal 等也将会被支持。Microsoft .Net 框架的一个优势就是扩展非常容易，因此使用一种新的语言仅仅需要为运行时间环境兼容的输出提供编译器支持。

某些 Web 服务器供应商已经尝试通过为服务器提供“插件”和 API 来简化 Web 应用程序的开发。这些解决方案是与特定的 Web 服务器相关的，不能解决跨多个供应商的问题。例如，微软的 Active Server Pages™(ASP) 技术使得在 Web 页面上创建动态内容更加容易，虽然只能工作在微软的 IIS 和 Personal Web Server 上，但是随着 Microsoft .Net 的推出，其功能变得更加强大，并且一直都会领导 Web 程序开发的潮流。当然，还存在其他的解决方案，但是都不能使一个普通的页面设计者能够轻易掌握。例如，像 Java Servlet 以及 PHP 等技术，就可以使编写交互的应用程序的服务器端的代码变得容易。

提示：ASP.Net 提供了 Web Form、Web Control、Web Service 等等面向 Web 应用的改进；Java 则以其出色的跨平台特性，良好的面向对象属性获得了今天 Web 开发的主流地位。

ASP.Net 是 Microsoft .Net 的一部分，作为战略产品，ASP.Net 设计了这样一些策略：易于写出结构清晰的代码，代码易于重用和共享，可用编译类语言编写等等。目的是让程序员更容易地开发出 Web 应用，满足计算向 Web 转移的战略方向的需要。

ASP.Net Beta1 已经于 2000 年 7 月开始测试，下面查看一下它的一些特性。支持编译类语言及 RunTime 支持，如 VB、VC++、C#、`<script language="vb"runat="server">`比 Script

语言、编译类语言运行速度快，更适合编写大型应用。Microsoft.Net 支持多语言混合编程。Page 在首次装入时编译，可以使代码在 RunTime 中处理，这一点像 VB6，代码运行在 virtual 环境中，类似 Java 的 Virtual Machine。

1. Web Form

ASP.Net Web Form 提供了像 VB Form 一样的编程习惯，可以触发 Server 端的 Events。Web Form 中可以布置 Web Control。

2. Page Events

ASP.Net 的设计是 Object Oriented。Page object 是最顶端的 object，Web control、application 和自定义的 Page 都从 Page object 中继承下来。Page object 有 OnLoad 等事件。

3. Web Controls

ASP.Net 将提供 5 类 Web Controls，有与 HTML 中的 Form 元素对应的 Control，比如 Button、Text List 类 Control，比如 Repeater、DataList、DataGrid。还有 Rich Control、Validation Control、Mobile Control，它们为 WAP 设备提供 WML。

4. 开发 Web Services

Web Services 允许系统向 Web 输出函数，输出方式是 URIs，调用之后返回 XML。比如：

```
<% @ WebService Language =" c#" %>
using system.web.services;
public class Shipping{
    <webmethod>
    public string OrderStatus ( string OrderNumber ){
        //代码
    }
}
```

调用协议可以用 HTTP - GET、HTTP - POST、HTTP - SOAP。Web Service 为 Web 系统集成，为基于 Web 的分布式计算提供了良好的框架。

5. Caching 改进

ASP.Net 提供 Cache API，程序员可以控制 Cache 的使用。在某些情况下，可以编写性能更好的应用程序。

6. Debugging 和 Tracing

ASP.Net 内建的 Debugger 支持多种语言的混合编程，可以像在 VB 中一样调试程序。ASP.Net 增加了 Trace 和 Profiler 功能。

1) 分离代码与 HTML 模板

代码可以放在 Class 模块中。分离的好处，既便于程序员与页面制作人员独立工作，也便于代码共享。

2) 库

对于模块化的支持,是编程工具用来编写大型应用程序的必备条件。ASP.Net 的模块化支持非常好。它提供了常用库,如 SMTP SendMail;提供了结构化的代码组织方式,如 `< % @ Import- Namespace ="system.asp.util"% >`,其引用了 util 包。

3) 配置和实施

ASP.Net 的配置信息放在 XML 文件中,而不是在注册表中。实施 ASP.Net 应用时,只需要把程序拷贝到一个目录中,而不需要注册任何对象,因为注册有时需要原有的服务 STOP 等一系列问题。

ASP.Net 支持编译过的程序的运行,可以保护技术秘密。

纵观以上特性的介绍,ASP.Net 改进了很多 ASP 的问题,对于熟悉 VB、VC 或其他传统开发平台的程序员,ASP.Net 提供了一个好的开发 Web 应用的平台,可以说,ASP 是基于 Script 的,而 ASP.Net 基于完整的 OO 程序设计思想。ASP.Net 对于 Web Service 的支持,使得 Web 应用的集成非常方便。

1.2.2 与 JSP 的比较

JSP 是 Sun 公司推出的新一代站点开发语言,Sun 公司借助自己在 Java 上的不凡造诣,除 Java 应用程序和 Java Applet 之外,又有新的硕果,它就是 JSR(Java Server Pages)。JSP 可以在 Servlet 和 JavaBeans 的支持下,完成功能强大的站点程序。

JavaServer Pages™(JSP) 技术为创建显示动态生成内容的 Web 页面提供了一个简捷而快速的方法。JSP 技术的设计目的是使构造基于 Web 的应用程序更加容易和快捷,而这些应用程序能够与各种 Web 服务器、应用服务器、浏览器和开发工具共同工作。

在开发 JSP 规范的过程中,太阳微系统公司(Sun Microsystems Inc.)与许多主要的 Web 服务器、应用服务器和开发工具供应商,以及各种各样富有经验的开发团体进行合作,其结果是找到一种为应用和页面开发人员在可移植性和易用性两者间取得折中的开发方法。

由于 JSP 页面的内置脚本语言是基于 Java 编程语言的,而且所有的 JSP 页面都被编译成为 Java Servlet,JSP 页面就具有 Java 技术的所有好处,包括健壮的存储管理和安全性。作为 Java 平台的一部分,JSP 拥有 Java 编程语言“一次编写,各处运行”的特点。随着越来越多的供应商将 JSP 支持添加到他们的产品中,用户可以使用自己所选择的服务器和工具,供应商更改工具或服务器并不影响用户当前的应用。当与 Java 2 平台的企业版(J2EE)和 Enterprise JavaBeans 技术整合时,JSP 页面将提供企业级的扩展性和性能,这对在虚拟企业中部署基于 Web 的应用是必需的。

1.2.3 与 PHP 的比较

PHP 是一种跨平台的服务器端的嵌入式脚本语言。它大量地借用 C、Java 和 Perl 语言的语法,并耦合 PHP 自己的特性,使 Web 开发者能够快速写出动态生成页面。它支持目前的绝大多数数据库。还有一点,PHP 是完全免费的,可以从 PHP 官方站点(<http://www.php.Net>)上自由下载,而且可以不受限制地获得源代码,甚至可以加进自己需要的特色。

PHP 可以编译成具有与许多数据库相连接的函数。PHP 与 MySQL 是当今的绝佳组合，还可以自己编写外围的函数去间接存取数据库。通过这样的途径更换使用的数据库时，可以轻松地更改编码以适应这样的变化。PHPLib 就是最常用的可以提供一般事务需要的一系列基库。但 PHP 提供的数据库接口支持彼此不统一，比如 Oracle、MySQL 和 Sybase 的接口，彼此都不一样。

PHP 提供了类和对象。基于 Web 的编程工作非常需要面向对象编程能力。PHP 支持构造器、提取类等。

综上所述，ASP.Net 中提到的 Web Service 集成的方式，目前微软力推 SOAP、UDDI。从以上关于业界支持、开发模型的观察中，可以得出 Java 目前更适合于广泛的 Web 计算。同时，人们期待 ASP.Net 的到来，期待更好的 Web 计算模型的到来。

1.3 从 ASP 到 ASP.Net

本小节主要讲解 ASP 的发展情况以及 .Net 出现前后的对比，并且介绍了 ASP.Net 的一些新的功能。

1.3.1 ASP 综述

ASP 的发展依赖于 IIS 的发展，从 ASP 1.0 到 ASP 3.0，微软没有用太多的心思改良 ASP，所以尽管版本升级了，但是 ASP 的功能还是没有显著的改变，如表 1-1 所示。

表 1-1 IIS 版本与它所支持的 ASP 版本的对照表

IIS 版本	所支持的 ASP
IIS 3.0	ASP 1.0
IIS 4.0	ASP 2.0
IIS 5.0	ASP 3.0

随着 IIS 5.0 的推出，微软首先开发出 ASP+，然后又更名为 ASP.Net，并对 ASP，ASP.Net 进行了改良。ASP.Net 新增加了哪些功能呢？

在 ASP 中使用功能比较小，但容易编译（Interpret）的 VBScript。而在 ASP.Net 中使用功能强大的 VB、C# 和 JavaScript，以及所有可以编译成 MSIL 的程序语言。下面看一下 ASP 与 ASP.Net 的执行过程。在 ASP 中执行过程如图 1-1 所示。



图 1-1 ASP 中的执行过程

在 ASP.Net 中执行过程如图 1-2 所示。

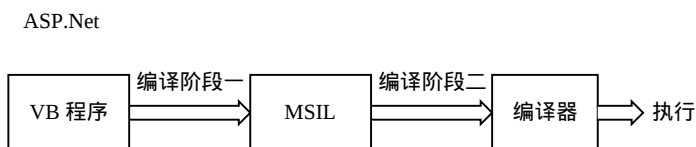


图 1-2 ASP.Net 中的执行过程

ASP.Net 阶段一的执行时间加上阶段二的执行时间大于 ASP 的执行时间，那么是不是 ASP.Net 的速度就比较慢呢？网页如果第二次被浏览，情况会是怎样的呢？

如图 1-3 所示，从上面的例子中可以看出，ASP.Net 网页第一次被浏览时经过两个阶段的编译，所以速度比较慢。但是第一次浏览以后，MSIL 会被存储下来，以后如果再浏览，那么速度就会加快很多。

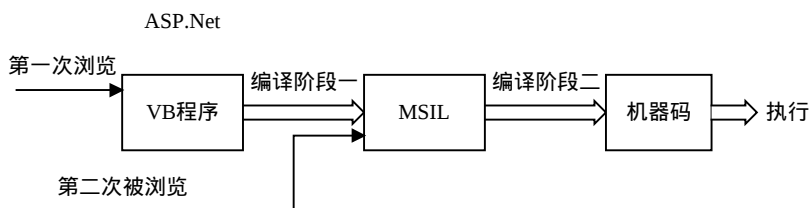


图 1-3 两次 ASP.Net 中执行过程的比较

.Net Framework 是微软最新发布的网页程序开发平台。在 .Net Framework 发布之前，程序是以 WIN32 API 及 COM 对象为接口的，ASP 网页则以 ASP.DLL 所提供的对象为接口。而 .Net Framework 发布之后，不管是程序还是网页，都以 .Net Framework 对象类别库为接口，原理如图 1-4 所示。

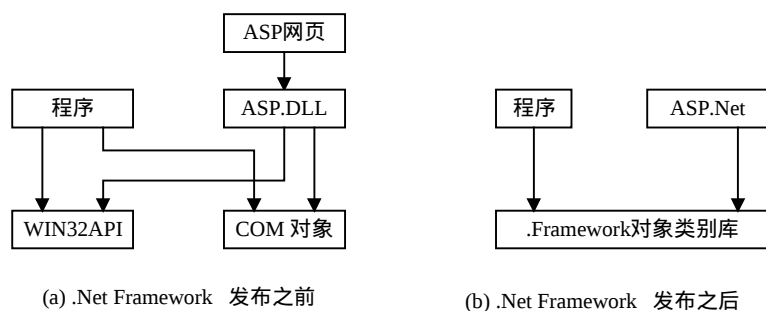


图 1-4 .Net Framework 发布之前与发布之后的对比

.Net Framework 所提供的对象类别库多达数百种，对制作网页来说是非常方便的。

1.3.2 从 ASP 到 ASP.Net 编写第一个程序

下面就通过程序来简单了解一下 ASP 与 ASP.Net 的区别。

在 ASP 中输出：

ASP：好！
ASP：不好！
在 ASPX 中输出：
ASPX：好！
ASPX：会变得更好！
下面的例子对于熟悉 ASP 的人来说是比较简单的。

例：test01.asp

```
<html><head>  
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">  
</head>  
<%Language="Vbscript"  
Function doIt()  
response.write ("asp:好！")  
End Function  
Function doIt2()  
response.write ("asp:不好！")  
End Function  
'调用函数 doIt()  
doIt()  
'显示回车  
response.write("<br>")  
'调用函数 doIt2()  
doIt2()  
%></html>
```

运行程序，结果如图 1 - 5 所示。

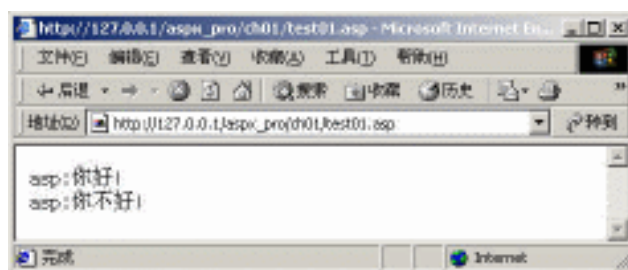


图 1 - 5 test01.asp 运行结果

而在 ASP.Net 中，则写成如下源代码：

例：test01.aspx

```
<%--以下为显示中文--%>  
<%@ CODEPAGE = "936" %>  
<html><head>
```

```
<%--显示中文--%>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<%--在浏览此页面的时候调用自己编写的函数来显示结果--%>
<SCRIPT LANGUAGE="VB" RUNAT="server">
Sub Page_Load(Sender as Object , E as EventArgs)
thestr1.text=doIt()
thestr2.text=doIt2()
End sub
Function doIt()
response.write("ASPX：好！<br>")
End Function
Function doIt2()
response.write("ASPX：会变得更好！")
End Function
</SCRIPT>
<%--以上是人们自己编写的函数--%></head>
<body>
<%--输出变量到浏览器--%>
<asp:label id="thestr1" runat="server"/>
<asp:label id="thestr2" runat="server"/>
</body></html>
```

运行结果如图 1 - 6 所示。

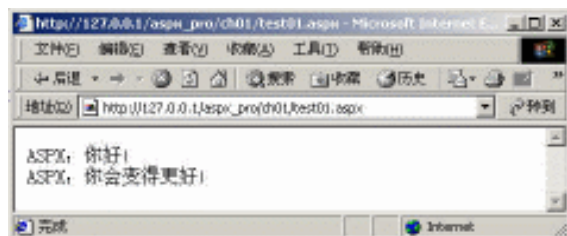


图 1 - 6 程序运行结果

从两者的区别中可以看出，在 ASP 中，所有需要服务器执行的语句都写在<%与%>之间，而在 ASP.Net 中则进行如下操作：

```
<SCRIPT ... RUNAT="server">
...
</SCRIPT>
```

形成的标记中，两者所使用的语言不同。在 ASP 中使用的是 VBScript 语言，而在 ASP.Net 中使用的是功能强大的 VB 语言，还可以使用 C# 和 JavaScript 语言，只有 JavaScript 保持不变。此外凡是可编译成 MSIL(Microsoft Intermediate Language) 的程序语言，都可以成为制作 ASP.Net 网页的程序语言。

1.4 ASP.Net 的新特点

ASP.Net Web Forms 就是 Web 页面，同现在用 ASP 编写代码所做的工作是一样的。但是还不仅如此，ASP.Net Web Forms 是在一个面向对象的编程模型上设计的，这样代码就能够再利用，并且使应用程序代码与页面内容分隔开。在 VB 中，将一个控件拖曳到一个表单上，然后执行其下的事件程序。在传统的 ASP 中，这是不可能的，因为在用户界面控件和它们的服务器端代码之间没有链接存在。ASP.Net 的出现使得上述所讲的都成为可能。

1.4.1 Web Forms

简单的 ASP.Net Web Form 例程如 test1_01.aspx，其源代码如下：

```
<html>
<head>
    <script language="VB" runat="server">
        按下按钮所调用的程序 Button1_Click
        Sub Button1_Click(sender As Object, e As EventArgs)
            Label1.Text="您刚才点击了按钮"
        End Sub
    </script>
<title>Button 举例一</title>
</head>
<body>
<h3><font face="Verdana">Button 举例一</font>
</h3>
<form runat=server>
<!--布置按钮控件-->
    <asp:Button id=Button1 Text="点击我一下。" onclick="Button1_Click"
runat="server"
/>
    <!--显示运行结果提示信息-->
<asp:Label
id=Label1
runat=server />
</form>
</body>
</html>
```

运行结果如图 1 - 7 所示。

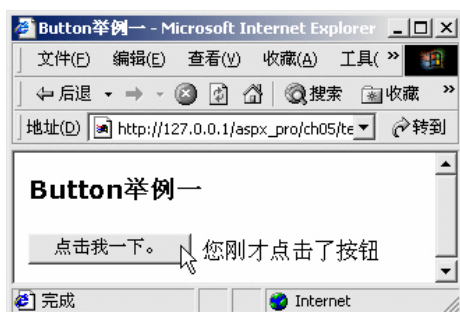


图 1 - 7 Button Web 控件举例运行结果

在上例中有两点值得注意。第一是在表单上 `runat="server"` 属性和 `asp:textbox` 控件的使用，这将告诉 ASP.Net 服务器和客户都可以使用这些控件，如此使用的控件叫做服务器控件。asp: 用作控件名的前缀，它本身就识别了控件是从哪里来的（稍后将讨论这一点）。第二个需要注意的细节是 `OnClick` 事件。开发 DHTML 代码时，经常要用到 `OnClick` 事件，这是为了在浏览器中激活一个事件。因为为控件设置了 `runat="server"` 属性，所以事件只在服务器上被激活。要对这个例子进行扩展，可以去掉 `Response.Write`，然后用一个基于服务器的控件来代替它。

`test1_01.aspx` 代码运行相当好，很像一个传统的客户或 Visual Basic 表单，代码也直观多了。可以用服务器控件来连接事件过程和基于服务器的代码。这些基于服务器的控件向浏览器发送纯 HTML 内容，其中不再含有客户端脚本。实际上，这其中有一个重要的设计，目的就是要坚持使用固有的 HTML 3.2 元素代码，以提供最大可能的浏览器兼容性。

生成的代码是符合 HTML 3.2 标准的，执行一个标准投递，将用户输入发送回同一个文件。这里没有对服务器状态进行维护，也没有客户端脚本对状态进行维护。隐含的域执行对控件状态的维护，这意味着在页面的“提交与返回”之间，控件能自动恢复状态，而不需要任何编程干预。虽然 ASP.Net Web 控件的默认输出是符合 HTML 3.2 标准的内容，但是对于更高级的浏览器，如 Microsoft Internet Explorer 5.0，也可以输出为 DHTML 格式。这样就能只使用一套服务器控件来编写页面，允许控件根据浏览器决定发送哪种类型的输出，从而允许将发送 DHTML 的控件和客户端脚本用到最新版本的 Internet Explorer 中，将纯 HTML 3.2 内容发送到其它浏览器中。

上面提到 ASP.Net 已经被重写了，但是并没有指出它是按照面向对象的思路重建的。在对象树的顶部是 `Page` 对象，即页面对象，ASP.Net 的每个控件、应用程序和页面都是从这个对象中继承来的，也就是说每个页面都是页面对象的一个示例。页面的 `Load`（装载）事件是一个非常重要的事件，如下面的代码所示：

```
<html>
<script language="VB" runat="server">
Sub Page_Load(Source As Object, E As EventArgs)
'页面被浏览是调用的事件程序
End Sub
Sub SubmitButton_Click(Source As Object, E As EventArgs)
```

```

按下按钮所调用的事件程序
End Sub

Sub Page_Unload(Source As Object, E As EventArgs)
'页面卸载是调用的事件程序
End Sub

</script>

<form runat="server">

<asp:Button text="Enter"
OnClick="SubmitButton_Click" runat="server" />

<asp:label id="YouEntered"/>

</form>

</html>

```

在这里看到了以前在 Visual Basic 中同样的 Load/Unload (装载/卸载) 过程。当页面被装载时, Load 事件被激活, 这时所有基于服务器的控件都可用。在与用户的交互作用过程中会生成其它事件。最后, 当页面被卸载时激活 Unload 事件。

1.4.2 ASP.Net 控件

<asp:TextBox>控件代表了一套需要熟练运用的全新控件。这并不难学, 因为在 HTML 中都有对应者。例如, 对于一个很简单的文本框, 在 HTML 中为:

```
<input type="text" value="Your Name"></input>
```

而相应的 Web 控件为:

```
<asp:TextBox Text="Your Name" runat="server" />
```

在这两者中, 能注意到 Web 控件通过代码 asp: 名称空间来识别, 并且同在 XML 中一样, 使用斜线来结束元素。不是必须使用 XML 格式, 也可以使用 HTML 格式, 用一个分号来结束标记。但是会发现, XML 格式被用在许多代码举例中, 并且代码字数也较简洁。名称空间必须要使用, 它负责识别文本框 TextBox 控件来自哪里。所有的标准 Web 控件都是 ASP 名称空间的一部分。当编写自己的控件时, 这一点变得很重要。

TextBox 控件相对于标准的输入框好像没有多少优势, 但是应该考虑下面的三个输入控件:

```

<input type="text" ...>
<input type="password" ...>
<textarea rows="5" ...>

```

它们都是用于 HTML 输入的, 却并没有一致性。但是如果使用下面的, 是不是更简单一些呢?

```

<asp:TextBox runat="server" ...>
<asp:TextBox TextBoxMode="Password" ...>
<asp:TextBox Rows="5" ...>

```

可以看到, 一个简单控件包含 HTML 中的三个控件的功能, 而且更易记忆、更易编码。

ASP.Net 携带了以下五大类 Web 控件：

- 与 HTML 相对应的固有控件。
- 在页面上提供数据流的列表控件。
- 提供更丰富的 UI(用户界面) 内容和功能的 Rich 控件。
- 完成各种表单验证的确认控件。
- 为 WAP 设备封装 WML 的移动控件。

固有的服务器控件与 HTML 控件相同，但是更加合理了，可提供更为一致的用途。这些控件包括链接按钮(LinkButton)、图像按钮(ImageButton)、超级链接(HyperLink)、文本框(TextBox)、复选框(CheckBox)、单选框(RadioButton)、下拉列表(DropDownList)、列表框(ListBox)、图像(Image)、标签(Label)、面板(Panel)、表格(Table)、表格行(TableRow)、表格单元(TableCell)等。

列表控件包括转发器(Repeater)、数据列表(DataList)和数据网格(DataGrid)，还包括单选框列表(RadioButtonList)和复选框列表(CheckBoxList)，这样就使创建单选框和复选框的列表变得简单。

Rich 控件包括日历(Calendar)和广告旋转(AdRotator)。Calendar 控件为低级的浏览器输出纯 HTML，为高级浏览器输出 DHTML(如 Internet Explorer 5.0)。AdRotator 输出图像，它有内置的旋转代码。

确认控件包括请求域确认(RequiredFieldValidator)、比较确认(Compare Validator)、范围确认(RangeValidator)、规范表达确认(RegularExpressionValidator)、顾客确认(CustomValidator)和确认摘要(ValidationSummary)。这些控件为开发人员在表单处理中建立确认提供了简单的途径。

有关移动控件的信息现在还没有发布，但是可以肯定它们有助于构造激活 WAP 的 Web 站点。

并不一定只局限于使用这些系统提供的控件，自己编写控件也相当简单。例如，如果想要一个控件来封装两个文本框(可能是名和姓的输入域)，就可以这样来编写代码：

```
<asp:Panel runat="server">  
  <asp:Textbox id="txtFirstName" text="First Name" runat="server" />  
  <asp:Textbox id="txtLastName" text="Last Name" runat="server" />  
</asp:Panel>
```

可以把这个代码保存在文件 Name.asp(注意这个新扩展名) 中，并将其作为一个 Web 表单控件来对待。接着，就可以向 Web 表单中增加以下内容：

```
<%@ Register TagName="NameControl" TagPrefix="Foo" Src="Name.asc" %>  
<form>  
  <foo:NameControl runat="Server"/>  
</form>
```

这样就可以很容易地创建可以再利用的控件了，还可以直接在 Visual Basic 或 C# 中创建控件，允许它们成为其它控件的子类，并且呈现任何它们所要求的输出。因为控件是用名称空间来识别的，因此在控件之间应该是没有冲突的。实际上，控件甚至可以使用同一个名字，只要它们在不同的名称空间中就行。这使 ASP.Net 非常具有扩展性，并且使编程环

境变得越来越丰富。实际上有一个很大的提供丰富控件的第三方市场。

新的 Web 控件之一是数据网格(DataGrid), 这是用于显示成套数据的内置支持控件。为了从 SQL 生成的数据中产生一个 HTML 表格, 只需要创建 ADO+对象, 并执行指令来获取数据以作为网格的数据源, 比如下面的 test1_02.aspx 的源代码:

```
<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
'建立临时选定人才表格
Dim Cart As DataTable
Dim CartView As DataView
'创建数据视图 CreateDataView
Function CreateDataView(DB as string,table as string) As ICollection
    dim connstr as string
    connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
    dim cmd as ADODatasetCommand
    cmd=new ADODatasetCommand("Select id,name from "& table,Connstr )
    dim ds as DataSet = new DataSet()
    cmd.FillDataSet(ds,table)
    CreateDataView= new DataView(ds.Tables(table))
End Function
'当页面被浏览时所调用的程序 Page_Load
Sub Page_Load(sender As Object, e As EventArgs)
    If Session("DG5VB_ShoppingCart") = Nothing Then
        Cart = New DataTable()
        Cart.Columns.Add(new DataColumn("序号", GetType(Integer)))
        Cart.Columns.Add(new DataColumn("姓名", GetType(string)))
        Session("DG5VB_ShoppingCart") = Cart
    Else
        Cart = Session("DG5VB_ShoppingCart")
    End If
    CartView = New DataView(Cart)
    ShoppingCart.DataSource = CartView
    CartView.Sort=id
    ShoppingCart.DataBind
    MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
```

```

        MyDataGrid.DataBind
    End Sub

    Sub Grid_CartCommand(sender As Object, e As DataGridCommandEventArgs)
        Dim dr As DataRow = Cart.NewRow()
        ' e.Item 是要选定的行
        Dim itemCell As TableCell = e.Item.Cells(1)
        Dim priceCell As TableCell = e.Item.Cells(2)
        Dim id As Integer = itemCell.Text
        Dim name As String = priceCell.Text
        If e.CommandSource.CommandName = "AddToCart" Then
            dr(0) = id
            dr(1) = name
            Cart.Rows.Add(dr)
        Else '从已经选定的人才里面去掉
            CartView.RowFilter = "姓名=" & name & ""
            If CartView.Count > 0 Then
                CartView.Delete(0)
            End If
            CartView.RowFilter = ""
        End If
        ShoppingCart.DataBind()
    End Sub
</script>

<title>在 DataGrid 使用模板</title></head>
<body>
<h3><font face="Verdana">在 DataGrid 使用模板</font></h3>
    <form runat=server>
        <table cellpadding="5">
            <tr valign="top">
                <td>
<b>人才列表</b>
<!--布置 DataGrid 控件，来显示数据 -->
<asp:DataGrid
id="MyDataGrid" 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_CartCommand"

```

```

>
<property name="Columns">
    <!--使用模板，来显示添加/删除-->
    <asp:TemplateColumn
        HeaderText="添加/删除">
        <template name="ItemTemplate">
            <asp:LinkButton ID=AddButton Text="添加"
                CommandName="AddToCart" ForeColor="blue"
                runat="server" />
            <asp:LinkButton ID=RemoveButton Text="去掉"
                CommandName="RemoveFromCart" ForeColor="blue"
                runat="server" />
        </template>
    </asp:TemplateColumn>
    <asp:BoundColumn HeaderText="序号" DataField="id"/>
    <!-- format Price as currency-->
    <asp:BoundColumn HeaderText="姓名" DataField="name"
        ItemStyle-HorizontalAlign="right" />
    </asp:TemplateColumn>
</property>
</asp:DataGrid>
</td><td>
<b>已经选定的人才</b>
    <!--布置 DataGrid 控件，来显示选定的人才-->
    <asp:DataGrid
        id="ShoppingCart" runat="server" BorderColor="black" BorderWidth="1"
        CellPadding="3" Font-Name="Verdana" Font-Size="8pt"
        HeaderStyle-BackColor="#aaaadd"/>
    </td>
</tr>
</table>
</form>
</body>

```

运行结果如图 1 - 8 所示。

所需要做的就是将数据捆绑到数据网格，然后就生成了一个整洁的 HTML 表格。

数据捆绑并不局限于来自数据库的数据，还可以捆绑到 hash 表格、数组、其它服务器控件、页面的适当层。如果默认的栏目不合适，还可以对其进行定制。



图 1-8 在 DataGrid 控件中使用模板的运行结果

使用 BoundColumn 控件选择一个简单的栏目，并指定栏的标题和此栏要捆绑到哪里。AutoGenerate="false"属性负责确保在创建数据网格时，不会为数据源的所有栏目建立网格，而是根据设计的需要有选择地显示栏目。

此前提到的 Repeater 和 DataList 控件也支持模板，这样就允许对控件的外观进行定制。Repeater 实际上并没有外观，必须要提供 UI，这就意味着必须要使用一个模板。相反，DataList 控件可以捆绑到数据列表，有一个默认的外观和丰富的行为。

1.5 本章小结

本章主要介绍了 ASP.Net 出现的背景，并和目前流行的 JSP、PHP 以及 ASP 等服务器端语言作了一个简单的比较。作为微软的战略产品，ASP.Net 的优势是显而易见的，在本章中还编写了第一个 .aspx 的页面。

第 2 章

.Net Framework 的安装及运行

主要内容

- | .Net Framework 的安装准备
- | .Net Framework 的安装过程
- | 进入 ASP.Net

本章主要介绍 .Net Framework 的安装(从下载软件、计算机配置、准备工作到 .Net Framework 的安装),并且介绍了在安装过程中出现的常见问题及其解决方法。

2.1 .Net Framework 的安装准备

Web 源于静态文本。静态文本的缺点就是维护难(文件数目多)、查询难、修改难。所以现在很多站点都采用 ASP 的动态页面。ASP(Active Server Pages)是一种类似于 VB 的面向对象的程序语言。ASP.Net 对于 ASP 来说有什么突破呢?

2.1.1 运行机制不同

ASP 属于一种解释型的编程框架,它的核心是 VBScript 和 JavaScript,受这两种脚本语言的限制,决定了 ASP 先天不足,它无法进行像传统编程语言那样的底层操作,所以如果进行一些诸如 SOCKET、文件等的操作时不得不借助于用其他传统编程语言如 C++、VB 和 Java 等编写的组件。并且由于它是解释执行的,所以在运行效率上大打折扣。而 ASP.Net 是一种编译型的编程框架,它的核心是 NGWS RunTime,除了和 ASP 一样可以采用 VBScript 和 JavaScript 作为编程语言外,还可以用 VB 和 C# 来编写。这就决定了它功能的强大,可以进行很多低层操作而不必借助于其它编程语言。

2.1.2 要求的配置不同

由于它是编译后运行的,所以执行效率要比 ASP 高得多。要安装 ASP.Net,电脑必须满足以下配置:

1. 硬件要求

- ? ? CPU : Intel Pentium II-class 300 MHz(最好 Intel Pentium III-class 600 MHz)。
- ? 内存 : 96 MB(最好 128 MB)。
- ? 磁盘空间 : 250 MB(完全安装) /155 MB(快速安装)。
- ? 显示器 : 800 × 600, 256 colors 。
- ? CD-ROM : 必需。

2. 软件要求

- ? Microsoft Windows 2000 + sp1。
- ? Microsoft Internet Explorer 5.5 。
- ? IIS 5.0。
- ? 其他 : MDAC 2.6。

ASP.Net 的安装过程很简单, 只需按照简单提示安装即可。但是, 如果安装了 Office 2000, 在此建议安装 ASP+之前先备份 Microsoft Office\Office\mso9.dll 这个文件, 因为安装完 ASP+后, Office 会提示用户注册, 否则 Office 2000 就会出现限制使用 50 次。此时将备份的 mso9.dll 文件覆盖掉原来的文件即可。

3. ASP+(NGWS SDK) 的下载地址

<http://download.microsoft.com/download/platformsdk/Trial/1812.10full/NT5/EN-US/Se-tup>

.exe 安装微软的 VisualStudio.Net 和安装 ASP+有很多相似之处。

4. 必须先安装的程序

安装 .Net Framework 必须先安装以下内容 :

- ? Windows 2000 sp1。
- ? 安装 IE 5.5。
- ? 必须要装有 IIS, 而且 IIS 要带 Front Page 扩展。
- ? Front Page 服务扩展的补丁 QFE。

相同地, 安装完 VisualStudio.Net 后同样会出现 Office 2000 的 50 次限制, 所以可以用同样的方法, 先备份 mso9.dll 文件, 然后安装完后覆盖掉原来的文件。

2.2 .Net Framework 的安装过程

2.2.1 Internet Explorer 5.5 的安装

运行安装程序, 就会出现如图 2 - 1 所示的界面。如果用户使用的是 Windows 2000 操作系统, 就要下载升级补丁, 来将 Internet Explorer 5.0 升级到 Internet Explorer 5.5。

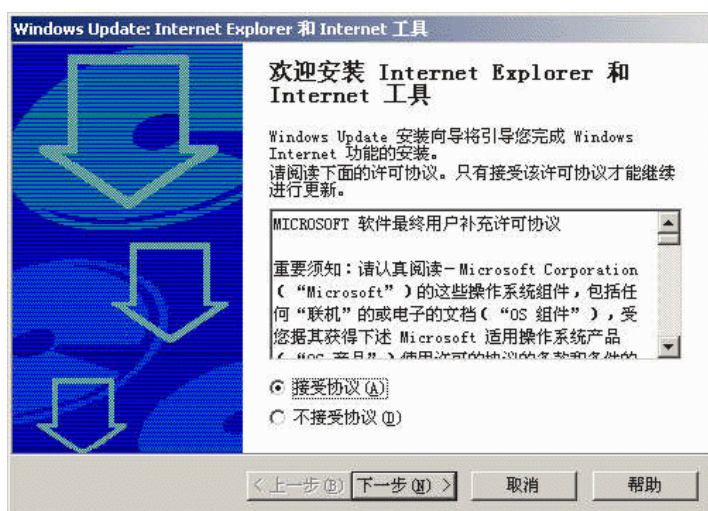


图 2-1 Internet Explorer 5.5 的安装界面

2.2.2 .Net Framework 的安装

下载之后双击如图 2-2 所示的文件，运行安装程序，则会出现如图 2-3 所示的对话框。



图 2-2 安装程序

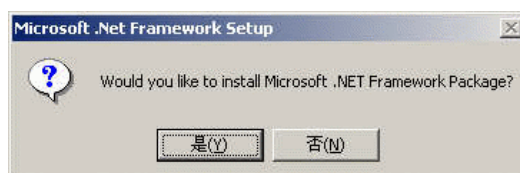


图 2-3 安装过程的对话框

然后，弹出如图 2-4 所示的界面，此时正在解压缩安装文件，计算机处理速度会减慢。

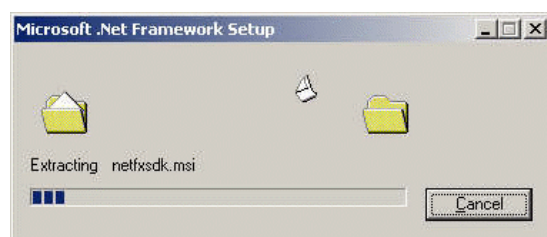


图 2-4 解压缩文件

图 2-5 是正在准备安装 .Net Framework。



图 2 - 5 .Net Framework 安装进程

注意：如果有 Internet Explorer 的活动窗口，安装程序会弹出如图 2 - 6 所示的对话框，此时建议最好把活动的 Internet Explorer 窗口关闭，然后继续安装。

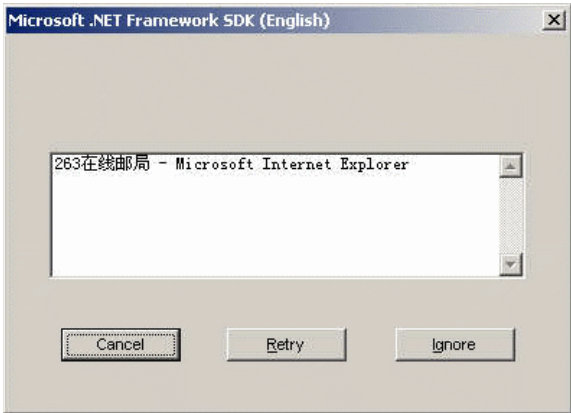


图 2 - 6 安装程序出现的提示对话框

接着进行 .Net Framework 的安装，如图 2 - 7 所示。



图 2 - 7 正式的安装过程

安装完成以后，桌面上将会弹出如图 2 - 8 所示的 .Net Framework SDK Overview 快捷方式。

双击鼠标左键，就会出现如图 2 - 9 所示的界面。

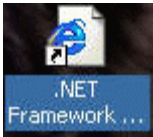


图 2 - 8 .Net Framework SDK Overview 快捷方式

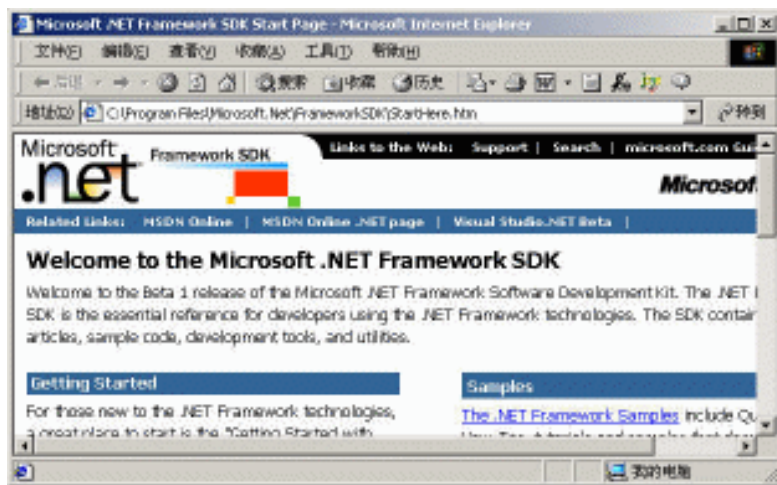


图 2-9 .Net Framework SDK 文件

.Net Framework 安装成功，所有以 .aspx 为扩展名的文件可在本地服务器上执行。

2.3 进入 ASP.NET

2.3.1 安装步骤

1. 下载和安装 .Net SDK

ASP.NET 是作为 .Net SDK 的一部分提供的，它是创建、生成和测试基于 .Net Framework 的应用程序所需要的所有技术的集合。要下载和安装 .Net SDK，请在 MSDN Online .Net 开发人员中心单击 Download the .Net Framework SDK（在上一节已经讲述了此步骤）。下载内容大约有 90 MB，不过，有一个选项可以将 SDK 分为一系列小文件进行下载。

提示：在安装 ASP.NET 之前，需要安装 Internet Explorer 5.5，可以从 <http://www.microsoft.com/Windows/ie/default.htm> 下载它。

2. 安装 .Net Framework 的快速入门示例

如果已经安装了 SDK，那么也应该安装 .Net Framework 的快速入门示例，包括在 .Net SDK 内的快速入门示例、各种 ASP.NET 特性和其他 .Net 技术的详细示例。安装快速入门示例，应先打开 SDK Overview 链接（.Net Framework SDK Overview），该链接已在 SDK 安装过程中添加至桌面，HTML 页面打开后，单击 .Net Framework Samples 链接，按照安装步骤来安装示例。

注意：在浏览默认文档时，可以选择页面顶端的 ASP.NET 链接。此链接包含了 ASP.NET 示例的列表。

3. 加入 ASPNG 讨论列表、参与在线讨论

Charles Carroll 作为不断壮大的 ASP.Net 社团的领导者之一，一直尽心尽力地主持着基于电子邮件的 ASP.Net 讨论列表。事实上，从七月初专业开发人员会议(PDC)的第一天开始，这个讨论列表就一直存在着。

此讨论列表是学习和发布日益丰富的 ASP.Net 开发人员知识的最佳选择。Charles 会检查和筛选所有的消息，确保发布的文章都与此列表相关，从而使讨论不脱离主题。除了社区方面的列表以外，Microsoft .Net Framework 小组还定期参加讨论并发布示例代码。最近一次是 Scott Guthrie 发布了一个动态生成图像的 ASP.Net 示例。

提示：要加入讨论列表，只需访问 <http://www.asplists.com/asplists/aspng.asp>。

在此页面上，可以找到与 ASP.Net 相关的不同主题的电子邮件讨论列表的完整列表。最活跃的列表是 ASPNGBeta(笔者推荐读者加入这个列表)。请从列表中选择 ASPNGBeta，也可以直接访问 <http://www.asplists.com/asplists/aspngbeta.asp>，选择所需的接收电子邮件分发的选项。访问该网站后，会出现如图 2 - 10 所示的页面。

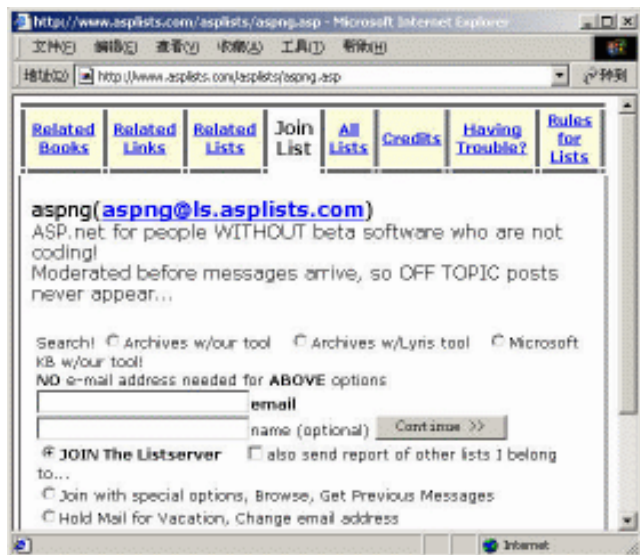


图 2 - 10 加入在线讨论

2.3.2 安装时的常见问题

(1) 如果安装 SDK 并且使用 ASP.Net，会不会中断现有的 ASP 应用程序？

如果安装 SDK 并且使用 ASP.Net，不会中断现有的 ASP 应用程序。ASP.Net 应用程序与现有的 ASP 应用程序可以同时运行。

ASP 使用的文件扩展名 (如 .asp 和 .asa) 和 ASP.Net 使用的文件扩展名 (如 .aspx、.asax、.ascx、.asmx 等) 是完全独立的，不会互相替代。而且，ASP.Net 应用程序的处理由一个独立的 ISAPI 扩展进行。

注意：尽管 ASP 和 ASP.NET 应用程序可以同时运行，但即使在同一目录下，它们也不能共享应用程序信息。例如，在 global.asa 中所作的设置不会影响 ASP.NET 应用程序，而在 global.asax 中所作的设置也不会影响 ASP 应用程序。类似地，应用程序和会话状态信息也不共享。ASP 应用程序中的某个用户无法使用在 ASP.NET 应用程序中为同一个用户添加的会话数据，除非重新创建该用户。

(2) 创建 ASP.NET 页面需要使用 Visual Studio 或其他工具吗？

创建 ASP.NET 页面不需要使用 Visual Studio 或其他工具。尽管 Visual Studio.NET 将提供惊人的设计、编码和调试环境，但创建 ASP.NET 应用程序并不需要使用它。ASP.NET 遵循与 ASP 相同的“只需保存”设计原则，即任何基于文本的编辑程序都可用于创建 ASP.NET 应用程序。

注意：在第一次访问 ASP.NET 页面时，该页面会自动编译，以后的请求将访问编译过的资源。

(3) ASP.NET 可以在 Windows 95、Windows 98 和 Windows Me 上运行吗？

ASP.NET 不可以在 Windows 95、Windows 98 和 Windows Me 上运行。用户确实可以利用这些平台之一来进行开发，但是 ASP.NET 应用程序必须运行在 Web 服务器上，或者运行在安装了 IIS 的 Windows 2000 或 Windows NT 4.0 上。

(4) ASP.NET 如何显示中文？

将 C:\WINNT\Microsoft.Net\Framework\v1.0.2204\config\Web 文件中的 iso-8859-1 更换为 GB 2312，就可以显示中文。其缺省设置如下：

```
...
<globalization
    requestencoding="iso-8859-1"
    responseencoding="iso-8859-1"
/>
```

需将其更改为：

```
...
<globalization
    requestencoding="GB 2312"
    responseencoding="GB 2312"
/>
...
```

2.4 本章小结

本章着重介绍了如何安装 .Net Framework，以及在安装的过程中应该注意的问题，需要提前做的安装准备等。然后针对常见的问题做了回答，解决了显示中文的问题。

第 3 章

ASP.Net 的 ASP Web Form 快速入门

主要内容

- | 什么是 ASP.Net 的 Web Forms
- | 编写第一个 Web Form 页面
- | 使用服务器控件编写 Web Form 页面

本章主要介绍 ASP.Net 的 ASP Web Forms 以及 ASP 转向 ASP.Net 非常重要的步骤。并且简单介绍如何使用服务器控件编写 Web Form 页面。

3.1 什么是 ASP.Net 的 Web Forms

ASP.Net 的 Web Forms 页面架构是一个可升级的公用语言运行程序模型,它可被用在服务器端动态地建立 Web 页面。

作为 ASP 的进化版本,ASP.Net 的 Web Forms 架构弥补了它的许多关键功能的不足。比如:能够创建并使用可再用用户界面控件,这些控件可以压缩到公用功能中,从而减少页面代码。Web Forms 能够建立清晰有序的程序结构和提供强大的 WYSIWYG 页面设计视图。

可以这样说,ASP.Net Web Forms 使创建的动态 Web 用户界面更简单、更强大,其页面无需脚本库程序或者 Cookie,能够满足任何浏览器客户端的需要。

?? ASP.Net 有如下的强大功能:

?? ASP.Net Web Form 页面兼容 ASP 的语法结构。

?? ASP.Net 服务器控件使得压缩合并公用功能简单易行。

?? ASP.Net 自带 45 个内置服务器控件,并且支持第三方产品。

?? ASP.Net 服务器控件可以创建适用于高、低版本浏览器的 HTML 内容。

?? ASP.Net 模板提供了定制列表服务器控件外观及行为的简单方法。

?? ASP.Net 验证控件可以容易地处理客户端或者服务器端的数据验证工作。

3.2 编写第一个 Web Form 页面

3.2.1 编写第一个 Web Form 页面

ASP.NET 的 Web Form 页面是纯文本文件，以.aspx 为文件扩展名。当客户端浏览器请求一个.aspx 文件时，ASP.NET 将解析并编译目标文件为.NET 架构类，接着这个类动态处理进来的请求。

注意：.aspx 文件只在第一次访问时被编译，随后的再次请求将重用这个类实例。

创建 ASP.NET 页面的最简单方法就是将现有的 HTML 文件的后缀名改为.aspx，不需要修改任何代码。下面的例程是执行收集用户名称和选择目录的功能。

ch03/test3_01.aspx 程序的源代码如下：

```
<html>
<head>
<title>最简单的 Web Form 页面</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
<body bgcolor="#FFFFFF" text="#000000">
<form name="form1"
      method="post"
      action="test3_01.aspx"
      >
<table width="256" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="161">请输入您的名字：
</td>
<td width="95">
<input type="text" name="name" size="15">
</td>
</tr>
<tr>
<td width="161">请选择您的国籍：
</td>
<td width="95">
<select name="Category">
<option selected>中国</option>
<option>日本</option>
<option>美国</option>
<option>俄罗斯</option>
```

```

        </select>
    </td>
</tr>
<tr>
    <td width="161">&nbsp;</td>
    <td width="95">
        <input type="submit"
            name="Submit"
            value="提交"
        >
    </td>
</tr>
</table>
</form>
</body>
</html>

```

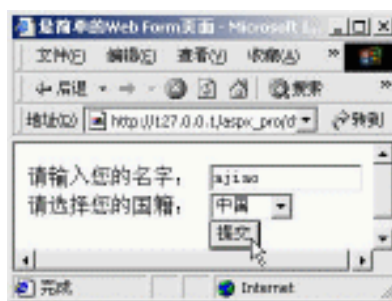


图 3 - 1 test3_01.aspx 的运行结果

运行结果如图 3 - 1 所示。

上例中，单击“提交”按钮后，看上去什么也没有发生，这是因为.aspx 文件只包含静态 HTML 内容。

3.2.2 使用 ASP 的<% %>块

ASP.Net 提供了兼容 ASP 的语法结构，其中包括支持<% %>代码块，使用它混合编写 HTML 以及.aspx 代码。这些代码段按照从上到下的顺序在页面中被执行。请看以下的例子。

使用 C# 语言编写的程序 ch03/test3_02c.aspx 的源代码如下：

```

<%@ Page Language="C#" %>
<html>
<head>
<title>最简单的 Web Form 页面</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
<body bgcolor="#FFFFFF" text="#000000">
<form name="form1" method="post" action="test3_02C#.aspx">
    <table width="256" border="0" cellspacing="0" cellpadding="0">
        <tr>
            <td width="161">请输入您的名字：</td>
            <td width="95">
                <input type="text" name="name" size="15">
            </td>
        </tr>
    </table>

```

```

<tr>
  <td width="161">请选择您的国籍 :
</td>
  <td width="95">
    <select name="Category">
      <option selected>中国</option>
      <option>日本</option>
      <option>美国</option>
      <option>俄罗斯</option>
    </select>
  </td>
</tr>
<tr>
  <td width="161">&nbsp;</td>
  <td width="95">
    <input type="submit" name="Submit" value="提交">
  </td>
</tr>
</table>
<p>
  <% for (int i=0; i <8; i++) { %>
    <font size="<%=i%>"> 欢迎使用 ASP.Net </font> <br>
  <% }%>
</p>
</form>
</body>
</html>

```

其中 C # 的语法为：

```
<%@ Page Language="C#" %>
```

此语句用于声明本页面使用的语言为：C # (读成 C sharp)。显示中文的语法是：

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
```

其中 charset=gb2312 是按照 GB 2312 编码来显示中文的。

以下是调用函数，循环输出变量：

```

<% for (声明变量;变量限制条件;变量递增条件) { %>
  //函数体
<% }%>

```

注意：在 C # 中声明变量的同时要养成赋初值的好习惯，这样可以有效地避免错误。

使用 VB 语言编写的程序 ch03/test3_02vb.aspx 的源代码如下：

```
<%@ Page Language="VB" %>
<html>
<head>
<title>最简单的 Web Form 页面</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
<body bgcolor="#FFFFFF" text="#000000">
<form name="form1" method="post" action="test3_01.aspx">
  <table width="256" border="0" cellspacing="0" cellpadding="0">
    <tr>
      <td width="161">请输入您的名字 : </td>
      <td width="95">
        <input type="text" name="name" size="15">
      </td>
    </tr>
    <tr>
      <td width="161">请选择您的国籍 :
      </td>
      <td width="95">
        <select name="Category">
          <option selected>中国</option>
          <option>日本</option>
          <option>美国</option>
          <option>俄罗斯</option>
        </select>
      </td>
    </tr>
    <tr>
      <td width="161">&nbsp;</td>
      <td width="95">
        <input type="submit" name="Submit" value="提交">
      </td>
    </tr>
  </table>
  <p>
    <% for i=0 to 7 %>
      <font size="<%=i%>">欢迎使用 ASP.Net
    </font>
    <br>
```

```

        <% next %>

    </form>

</body>

</html>

```

运行结果如图 3 - 2 所示。

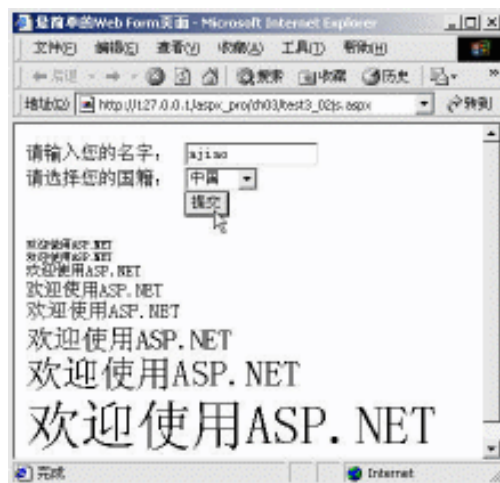


图 3 - 2 使用 ASP 的<% %>块举例运行结果

对于 VB 这种比较简单的语言，语法要求没有 C#那样严格，程序就可写得相对简便。使用 JavaScript 语言编写的程序，ch03/test3_02js.aspx 的源代码如下：

```

<%@ Page Language="JScript"%>

<html>

<head>

<title>最简单的 Web Form 页面</title>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

</head>

<body bgcolor="#FFFFFF" text="#000000">

<form name="form1" method="post" action="test3_01.aspx">

    <table width="256" border="0" cellspacing="0" cellpadding="0">

        <tr>

            <td width="161">请输入您的名字：

            </td>

            <td width="95">

                <input type="text" name="name" size="15">

            </td>

        </tr>

        <tr>

            <td width="161">请选择您的国籍：

```


使用 JavaScript 语言编写的程序，则是：

```
<% var i : int;

    for (i=0; i <8; i++) { %>

        <font size="<%=i%>"> 欢迎使用 ASP.Net </font> <br>

    <% }%>
```

通过将上面几个程序和下面静态的 HTML 程序进行对比，会发现使用几句程序语言就可以得到用 HTML 写半天的效果，执行的结果如下：

```
<font size="0"> 欢迎使用 ASP.Net </font> <br>
<font size="1"> 欢迎使用 ASP.Net </font> <br>
<font size="2"> 欢迎使用 ASP.Net </font> <br>
<font size="3"> 欢迎使用 ASP.Net </font> <br>
<font size="4"> 欢迎使用 ASP.Net </font> <br>
<font size="5"> 欢迎使用 ASP.Net </font> <br>
<font size="6"> 欢迎使用 ASP.Net </font> <br>
<font size="7"> 欢迎使用 ASP.Net </font> <br>
```

与 ASP 不同，aspx 文件中位于<% %>之间的代码是被编译过的，而不是使用脚本引擎解释执行的。这样处理后，就大大地提高了程序的运行效率。

3.2.3 编写可以显示结果的 Web Form

同时，在 ASP.Net 中可以像 ASP 一样使用<% %>代码块，可动态地修改 HTML 的输出内容。比如，运行以下程序，得到的结果如图 3 - 3 所示。



图 3 - 3 可以显示结果的举例运行结果

下面举一个显示提交结果的例子。

使用 VB 语言编写的程序 ch03/test3_03vb.aspx 的源代码如下：

```
<%@ Page Language="VB" %>

<html>
    <head>
        <title>显示提交结果</title></head>
    <body>
        <center>
            <form action="test3_03vb.aspx">
```

```

<table width="270" border="0" cellspacing="0" cellpadding="0" align="left">
  <tr>
    <td width="82">
      <div align="right">姓名 : </div>
    </td>
    <td width="188">
      <input name="Name" type="text" value="<%=Request.QueryString("Name")%>">
    </td>
  </tr>
  <tr>
    <td width="82">
      <div align="right">国家 :
    </div>
    </td>
    <td width="188">
      <select name="Category" size=1>
        <%
          Dim I As Integer
          Dim Values(4) As String
          Values(0) = "中国"
          Values(1) = "日本"
          Values(2) = "美国"
          Values(3) = "俄罗斯"
          For I = 0 To Values.Length - 1
            %>
            <% If (Request.QueryString("Category") = Values(i)) %>
              <option selected>
            <% Else %>
              <option>
            <% End If %>
            <%=Values(i)%> </option>
          <% Next %>
        </select>
      </td>
    </tr>
  <tr>
    <td width="82">
      <div align="right"></div>
    </td>

```

```

        <td width="188">
            <input type="submit" name="Lookup" value="提交">
        </td>
    </tr>
    <tr>
        <td colspan="2">
            <% If (Not Request.QueryString("Lookup") = Nothing) %>
                Hi <%=Request.QueryString("Name") %>,
                你是<%=Request.QueryString("Category") %>人
            <% End If %>
        </td>
    </tr>
</table>
<p align="left">
<BR>
</form>
</center>
</body>
</html>

```

上面的例子中，首先是建立一个一维数组，VB 建立数组的语法示例如下：

```

Dim Values(4) As String
Values(0) = "中国"
Values(1) = "日本"
Values(2) = "美国"
Values(3) = "俄罗斯"

```

在 VBScript 里，经常会用到 Array 函数来设定数组的初值，VB 7 中不像在 VB 6 中一样可以指定数组元素个数的范围，例如下面的例子：

Dim Values(1 to 4) 在 VB 6 中是正确的，但是在 VB 中却是错误的。

VB 7 的数组起始值都是从 0 开始的，且只能指定数组中元素的个数，不能指定数组元素的个数范围，所以在上面的例子中，Dim Values(4) As String 中的 4 表示数组元素的个数，范围是 0 ~ 3，而在 VB 6 中 Dim Values(4) As String 中的 4 表示该数组的最后一个元素代号，其范围是 0 ~ 4。

使用 JavaScript 语言编写的程序 ch03/test3_03js.aspx 的源代码如下：

```

<%@ Page Language="JScript" %>
<html>
    <head>
        <title>显示提交结果</title></head>
    <body>
        <center>

```

```
<form action="test3_03js.aspx">
  <table width="270" border="0" cellspacing="0" cellpadding="0" align="left">
    <tr>
      <td width="82">
        <div align="right">姓名 : </div>
      </td>
      <td width="188">
        <input name="Name"
          type="text"
          value="<%=Request.QueryString("Name")%>">
      </td>
    </tr>
    <tr>
      <td width="82">
        <div align="right">国家 : </div>
      </td>
      <td width="188">
        <select name="Category" size=1>
          <%
            var values : String [ ] = [ "中国", "日本", "美国", "俄罗斯" ];
            var i : int;
            for (i=0; i<values.Length; i++) {
              <%
                <option <% if (Request.QueryString("Category") == values[i])
{ Response.Write("selected"); } %>>
                  <%=values[i]%>
                </option>
              <% } %>
            </select>
          </td>
        </tr>
        <tr>
          <td width="82">
            <div align="right"></div>
          </td>
          <td width="188">
            <input type="submit" name="Lookup" value="提交">
          </td>
        </tr>
      </form>
```

```

<tr>
  <td colspan="2">
    <% if (Request.QueryString("Lookup") != null) { %>
      Hi <%=Request.QueryString("Name")%>,你是<%=Request.QueryString("Category") %>人。
    <% } %>
  </td>
</tr>
</table>
<p align="left">
<BR>
</form>
</center>
</body>
</html>

```

使用 C # 程序语言编写的 ch03/test3_03c.aspx 的源代码如下：

```

<%@ Page Language="C#" %>
<html>
  <head>
    <title>显示提交结果</title></head>
  <body>
    <center>
<form action="test3_03c.aspx">
  <table width="270" border="0" cellpadding="0" cellspacing="0" align="left">
    <tr>
      <td width="82">
        <div align="right">姓名：</div>
      </td>
      <td width="188">
        <input name="Name"
          type="text"
          value="<%=Request.QueryString["Name"]%>">
      </td>
    </tr>
    <tr>
      <td width="82">
        <div align="right">国家：</div>
      </td>
      <td width="188">
        <select name="Category" size=1>

```

```

        <%
            String [] values = { "中国", "日本", "美国", "俄罗斯" };
            for (int i=0; i<values.Length; i++) {
                %>
                <option <% if (Request.QueryString["Category"] == values[i])
{ Response.Write("selected"); } %>>
                    <%=values[i]%>
                </option>
                <% } %>
            </select>
        </td>
    </tr>
    <tr>
        <td width="82">
            <div align="right"></div>
        </td>
        <td width="188">
            <input type="submit" name="Lookup" value="提交">
        </td>
    </tr>
    <tr>
        <td colspan="2">
            <% if (Request.QueryString["Lookup"] != null) { %>
                Hi <%=Request.QueryString["Name"] %>, 你是
            <%=Request.QueryString["Category"] %>人。
            <% } %>
        </td>
    </tr>
</table>
<p align="left"><BR>
</form>
</center>
</body>
</html>

```

运行结果如图 3 - 4 所示。

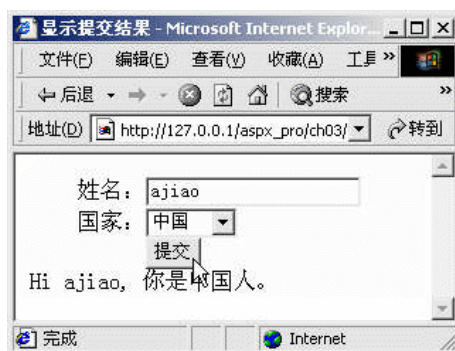


图 3-4 可以显示结果的举例运行结果

使用`<% %>`代码块虽然可以定制 ASP.Net 页面的输出内容，但却不能很好地建立一个清晰的 HTML 编程模型。这可以从上例的源代码中显而易见地看到。

3.3 使用服务器控件编写 Web Form 页面

3.3.1 使用服务器控件

除了使用`<% %>`代码块能编程实现动态内容外，ASP.Net 还提供了服务器控件来完成同样的功能。*.aspx 文件中，服务器控件以包含 `runat="server"` 属性设置的标记定义。

以下的例子中使用了 4 个服务器控件：`<form runat=server>`、`<asp:textbox runat=server>`、`<asp:dropdownlist runat=server>`、`<asp:button runat=server>`。运行时，这些服务器控件将动态建立 HTML 输出内容。

例如，使用 VB 程序语言编写的 ch03/test3_04vb.aspx 的源代码如下：

```
<%@ Page Language="VB" %>
<html>
  <head>
    <title>使用服务器控件编写 Web Form 页面</title>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312">
  </head>
  <body bgcolor="#FFFFFF" text="#000000">
    <center>
      <form action="test3_04vb.aspx" method="post" runat=server>
        姓名：<asp:textbox id="Name" runat="server"/>
        国家：<asp:dropdownlist id="Category" runat=server>
          <asp:listitem>中国</asp:listitem>
          <asp:listitem>日本</asp:listitem>
          <asp:listitem>美国</asp:listitem>
```

```
<asp:listitem>俄罗斯</asp:listitem>
</asp:dropdownlist>
<asp:button text="提交" runat="server"/>
</form>
</center>
</body>
</html>
```

使用 C# 语言编写的程序 ch03/test3_04vb.aspx 的源代码如下：

```
<%@ Page Language="C#"%>
<html>
  <head>
<title>使用服务器控件编写 Web Form 页面</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
  <body bgcolor="#FFFFFF" text="#000000">
    <center>
      <form action="test3_04vb.aspx" method="post" runat=server>
        姓名：<asp:textbox id="Name" runat="server"/>
        国家：<asp:dropdownlist id="Category" runat=server>
          <asp:listitem>中国</asp:listitem>
          <asp:listitem>日本</asp:listitem>
          <asp:listitem>美国</asp:listitem>
          <asp:listitem>俄罗斯</asp:listitem>
        </asp:dropdownlist>
        <asp:button text="提交" runat="server"/>
      </form>
    </center>
  </body>
</html>
```

使用 JavaScript 语言编写的程序 ch03/test3_04js.aspx 的源代码如下：

```
<%@ Page Language="JScript"%>
<html>
  <head>
<title>使用服务器控件编写 Web Form 页面</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
  <body bgcolor="#FFFFFF" text="#000000">
    <center>
      <form action="test3_04vb.aspx" method="post" runat=server>
```

```
    姓名：<asp:textbox id="Name" runat="server"/>  
    国家：<asp:dropdownlist id="Category" runat="server">  
        <asp:listitem>中国</asp:listitem>  
        <asp:listitem>日本</asp:listitem>  
        <asp:listitem>美国</asp:listitem>  
        <asp:listitem>俄罗斯</asp:listitem>  
    </asp:dropdownlist>  
<asp:button text="提交" runat="server"/>  
</form>  
</center>  
</body>  
</html>
```

通过上面的三个程序源代码可发现，无论使用什么样的程序语言，调用的服务器控件是不变的，变化的只是程序的语言，可以看到其程序良好的兼容性。

```
<form action="test3_04vb.aspx" method="post" runat="server">  
    姓名：<asp:textbox id="Name" runat="server"/>  
    国家：<asp:dropdownlist id="Category" runat="server">  
        <asp:listitem>中国</asp:listitem>  
        <asp:listitem>日本</asp:listitem>  
        <asp:listitem>美国</asp:listitem>  
        <asp:listitem>俄罗斯</asp:listitem>  
    </asp:dropdownlist>  
    <asp:button text="提交" runat="server"/>  
</form>
```

上面的程序运行结果如图 3 - 5 所示。

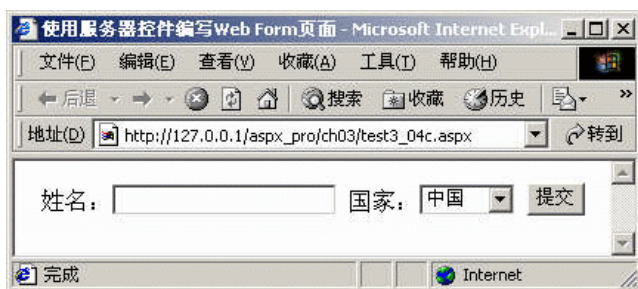


图 3 - 5 服务器控件举例运行结果

可以看到，使用服务器控件能够自动地保存客户端输入的任何数值，而不需考虑与服务器交互多少次。这些控件的状态不是保存在服务器端，而是在与服务器交互过程中保存在了一个<hidden>表单域中。同时，客户端不再需要客户端脚程序。

3.3.2 使用<asp:adrotator>控件举例

除了标准的 HTML 输出控件以外，ASP.Net 还可以让开发者在页面上使用丰富的定制控件。请试一试下面的例子，看看如何使用<asp:adrotator>控件在页面上动态显示旋转广告。广告信息设置文件 ch03/ads.xml 的源代码如下：

```
<Advertisements>
  <Ad>
    <ImageUrl>../images/banner1.gif</ImageUrl>
    <NavigateUrl>http://www.singnet.com.cn/music/</NavigateUrl>
    <AlternateText>世纪星之影音接触</AlternateText>
    <Keyword>影音接触</Keyword>
    <Impressions>80</Impressions>
  </Ad>
  <Ad>
    <ImageUrl>../images/banner2.gif</ImageUrl>
    <NavigateUrl>http://www.singnet.com.cn/bible/game/</NavigateUrl>
    <AlternateText>游戏天地</AlternateText>
    <Keyword>世纪星之游戏天地</Keyword>
    <Impressions>80</Impressions>
  </Ad>
  <Ad>
    <ImageUrl>../images/banner3.gif</ImageUrl>
    <NavigateUrl>http://www.singnet.com.cn/tianya/</NavigateUrl>
    <AlternateText>天涯社区</AlternateText>
    <Keyword>世纪星之天涯社区</Keyword>
    <Impressions>80</Impressions>
  </Ad>
  <Ad>
    <ImageUrl>../images/banner4.gif</ImageUrl>
    <NavigateUrl>http://www.singnet.com.cn/job/</NavigateUrl>
    <AlternateText>世纪星择业</AlternateText>
    <Keyword>世纪星择业</Keyword>
    <Impressions>80</Impressions>
  </Ad>
</Advertisements>
```

使用 VB 程序语言编写的 ch03/test3_05vb.aspx 的源代码如下：

```
<%@ Page Language="VB"%>
<html>
<head>
```

```

<title>使用<asp:adrotator>控件编写 Web Form 页面</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
<body bgcolor="#FFFFFF" text="#000000">
    <center>
<form action="test3_05vb.aspx" method="post" runat="server">
    <asp:adrotator AdvertisementFile="ads.xml"
        BorderColor="black"
        BorderWidth=1
        runat="server"
    />
    姓名: <asp:textbox id="Name"
        runat="server"
    />
    国家: <asp:dropdownlist id="Category" runat=server>
        <asp:listitem>中国</asp:listitem>
        <asp:listitem>日本</asp:listitem>
        <asp:listitem>美国</asp:listitem>
        <asp:listitem>俄罗斯</asp:listitem>
    </asp:dropdownlist>
    <asp:button text="提交" runat="server"/>
</form>
</center>
</body>
</html>

```

运行结果如图 3 - 6 所示。



图 3 - 6 <asp:adrotator>控件举例运行结果

使用 JavaScript 编写的程序 ch03/test3_05js.aspx 的源代码如下：

```
<%@ Page Language="JScript"%>
<html>
<head>
<title>使用<asp:adrotator>控件编写 Web Form 页面</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
<body bgcolor="#FFFFFF" text="#000000">
<center>
<form action="test3_05vb.aspx" method="post" runat="server">
<asp:adrotator AdvertisementFile="ads.xml"
BorderColor="black"
BorderWidth=1
runat="server"
/>
姓名: <asp:textbox id="Name"
runat="server"
/>
国家: <asp:dropdownlist id="Category" runat=server>
<asp:listitem>中国</asp:listitem>
<asp:listitem>日本</asp:listitem>
<asp:listitem>美国</asp:listitem>
<asp:listitem>俄罗斯</asp:listitem>
</asp:dropdownlist>
<asp:button text="提交" runat="server"/>
</form>
</center>
</body>
</html>
```

使用 C# 程序语言编写的 ch03/test3_05c.aspx 的源代码如下：

```
<%@ Page Language="C#"%>
<html>
<head>
<title>使用<asp:adrotator>控件编写 Web Form 页面</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
<body bgcolor="#FFFFFF" text="#000000">
<center>
<form action="test3_05vb.aspx" method="post" runat="server">
<asp:adrotator AdvertisementFile="ads.xml"
```

```
        BorderColor="black"
        BorderWidth=1
        runat="server"
    />
    姓名: <asp:textbox id="Name" runat="server"/>
    国家: <asp:dropdownlist id="Category" runat=server>
        <asp:listitem>中国</asp:listitem>
        <asp:listitem>日本</asp:listitem>
        <asp:listitem>美国</asp:listitem>
        <asp:listitem>俄罗斯</asp:listitem>
    </asp:dropdownlist>
    <asp:button text="提交" runat="server"/>
    </form>
    </center>
    </body>
</html>
```

上面的例子中都使用下面的语法来实现广告轮换：

```
<asp:adrotator AdvertisementFile="ads.xml" BorderColor="black" BorderWidth=1 runat="server"/>
```

有关服务器控件的使用将在以后的章节中讲述，在此不再赘述。

3.4 本章小结

本章着重介绍了 ASP.Net 的 ASP Web Form 快速入门；从 ASP 程序出发，通过实例讲解了如何从 ASP 升级到 ASP.Net；还介绍了如何使用服务器控件，并且制作了第一个使用服务器控件的网页。

第 4 章

ASP.Net 的内置服务器控件 ——HTML 控件

主要内容：

- ！ ASP.Net 的内置 HTML 控件简介
- ！ HTML 控件属性
- ！ HTML 控件运用实例

ASP.Net 的内置服务器控件，包括 HTML 控件(System.Web.UI.HtmlControls) 和 Web 控件(System.Web.UI.WebControls)。

HTML 控件(System.Web.UI.HtmlControls)，是以 HTML 标记为基础衍生出来的控件，此类控件通常只提供“属性”读取及设定的功能。HTML 标记(例如<html><body><head><table><image><td><tr>等)是构成 HTML 网页的要素，在 ASP.Net 中可以直接使用，但是在.aspx 文件中 HTML 标记会原原本本地下载给浏览器，不会加以编译。

4.1 HTML 控件简介

HTML 控件的对象的属性可以由 ASP.Net 程序来设定。通过 ASP.Net 来设定动态的 HTML 标记的属性，便可以产生动态的网页。

所有的 HTML 服务器控件都直接或者间接地由基本 System.Web.UI.HtmlControls.HtmlControl 类来产生。

用户可以通过扩展目前任何的 HTML 控件，或者通过结合目前使用的 HTML 控件，或者自己制作产生于基础 System.Web.UI.HtmlControls.HtmlControl 类的控件来开发客户自己的 HTML 控件。

HTML 控件的层次结构如图 4-1 所示。

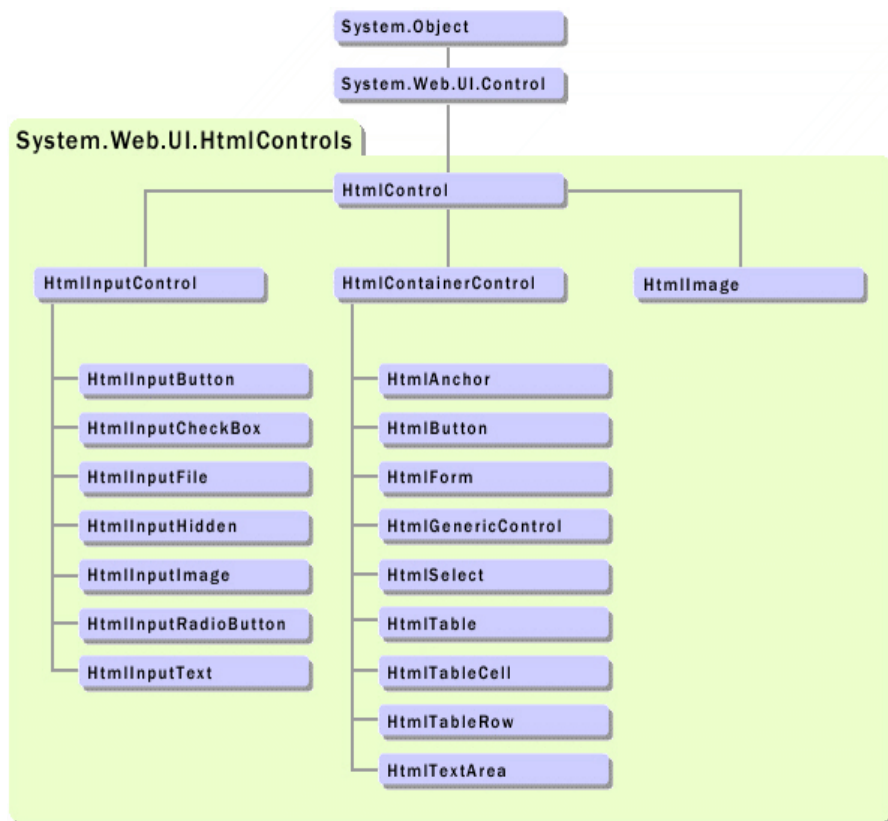


图 4 - 1 HTML 控件的层次结构

HTML 控件成员如表 4 - 1 所示。

表 4 - 1 HTML 控件成员列表

HtmlAnchor	HtmlButton
HtmlForm	HtmlGenericControl
HtmlImage	HtmlInputButton(Button)
HtmlInputButton(Reset)	HtmlInputButton(Submit)
HtmlInputCheckBox	HtmlInputFile
HtmlInputHidden	HtmlInputImage
HtmlInputRadioButton	HtmlInputText(Password)
HtmlInputText(Text)	HtmlSelect
HtmlTable	HtmlTableCell
HtmlTableRow	HtmlTextArea

HTML 控件常用的属性，包括 Attributes、Disable、Style、InnerHtml、InnerText 等。

4.1.1 Attributes 属性

Attributes 属性是用来设定 HTML 控件属性的，其语法如下：

HTML 控件名称.Attributes="属性值"

例如，使用 VB 语言编写的程序 ch04/test4_01vb.aspx 的源代码如下：

```
<html>
<head>
<script Language="VB" runat="server">
    Sub Page_Load(sender As Object, e As EventArgs)
        ' 设定背景颜色
        Body1.Attributes("Bgcolor") = "#FFFFFF"
        ' 设定字颜色
        Body1.Attributes("text")="#0000CC"
        ' 设定页面左边距离边框左边的宽度
        Body1.Attributes("leftmargin")="4"
        ' 设定页面顶部距离边框顶部的宽度
        Body1.Attributes("topmargin")="0"
        ' 设定页面空白的宽
        Body1.Attributes("marginwidth")="0"
        ' 设定页面空白的高
        Body1.Attributes("marginheight")="0"
    End Sub
</script>
<title>Attributes 属性举例</title></head>
<!--在服务器端执行，应用 Body1-->
<Body runat="server" id="Body1">
<h3>Attributes 属性举例</h3>
<h3>喜欢 ASP.Net ? ! </h3>
</Body></html>
```

其运行结果如图 4 - 2 所示。

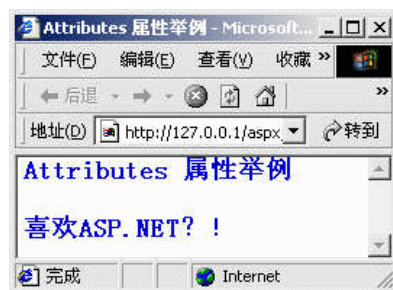


图 4 - 2 Attributes 属性举例运行结果

4.1.2 Disable 属性

Disable 属性适用于文字和按钮类的 HTML 控件，其值为 :True ,False。若设置为 True，则表示当前可用，否则为不可用。

4.1.3 InnerHtml 及 InnerText 属性

这两个属性用来设定<标记></标记>之间的 HTML 字段和纯文字。

使用 InnerHtml 的例子，如程序 ch04/test4_02vb.aspx，其源代码如下：

```
<html>
<head>
<script Language="VB" runat="server">
```

```
Sub Page_Load(sender As Object, e As EventArgs)
    插入 HTML 字段
    Body1.InnerHtml="<b>InnerHTML 及 InnerText 属性举例</b>"
End Sub

</script>

<title>InnerHTML 及 InnerText 属性举例</title></head>

<!--在服务器端执行，应用 Body1-->

<Body runat="server" id="Body1">

</Body></html>
```

运行结果如图 4 - 3 所示。



图 4 - 3 InnerHtml 属性举例运行结果

使用 Innertext 的例子，如程序 ch04/test4_03vb.aspx，其源代码如下：

```
<html>
<head>
<script Language="VB" runat="server">
    Sub Page_Load(sender As Object, e As EventArgs)
        插入文字
        Body1.InnerText="<b>InnerHTML 及 InnerText 属性举例</b>"
    End Sub
</script>

<title>InnerHTML 及 InnerText 属性举例</title></head>

<!--在服务器端执行，应用 Body1-->

<Body runat="server" id="Body1">

</Body></html>
```

运行结果如图 4 - 4 所示。

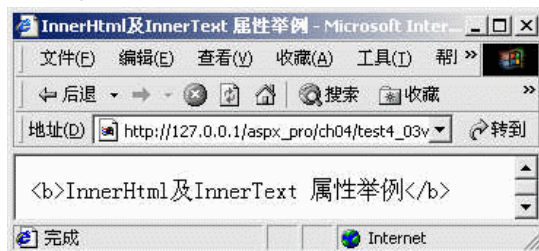


图 4 - 4 InnerText 属性举例运行结果

注意：综合上面的两个例子后会发现，使用 InnerHtml 属性时，其中的 HTML 标记会被浏览器编译，而使用 InnerText 属性时，如果含有 HTML 标记，浏览器也不会编译，将按照文字直接下载到客户端的浏览器。

4.1.4 Style 属性

Style 属性用来设定标记的样式。在程序 ch04/test4_04.htm 中使用样式设定文字属性，其源代码如下：

```
<html>
<head>
<title>Style 属性</title></head>
<Body BgColor="White">
<!--使用样式-->
<span style="color:white;
        background-color:blue;
        font-size:14pt;"
    >
Style 属性
</span>
</Body>
</Html>
```



图 4 - 5 Style 属性举例运行结果

在上面的例子中，设定了文字的颜色、背景色和字体大小，其效果如图 4 - 5 所示。如果在 ASP.Net 中使用 Style 样式来显示上面的结果。

例如，使用 VB 程序语言编写的程序 ch04/test4_04vb.aspx，其源代码如下：

```
<script Language="VB" runat="server">
    Sub Page_Load(sender As Object, e As EventArgs)
        Span1.Style("color") = "white"
        Span1.Style("background-color") = "blue"
        Span1.Style("font-size") = "14pt"
    End Sub
</script>
<html>
<head>
<title>Style 属性</title></head>
<Body BgColor="White">
<!--显示带有样式的内容-->
<span id="Span1" runat="server">
Style 属性
</span></Body></Html>
```

结果和上面是一样的，如图 4 - 6 所示。

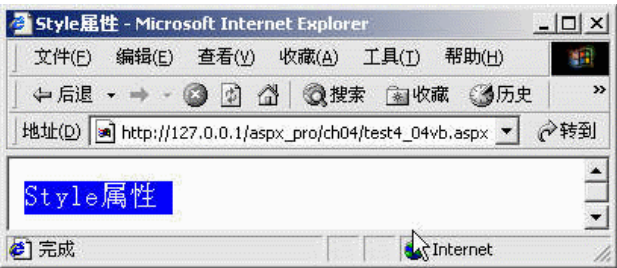


图 4 - 6 Style 属性举例运行结果

4.2 HtmlAnchor 控 件

HtmlAnchor 控件允许根据 HTML<a>标签编程，此控件必须有开始和解释标记。可以灵活地用此控件来修改<a>标签的属性，如用数据源来产生这些控件，或者用控件事件来控制 HtmlAnchor 控件。HtmlAnchor 控件，使用<a>标记可以设定链接，其语法如下：

```
<a
    runat="server"
    id="programmaticID"
    href="linkurl"
    name="bookmarkname"
    OnServerClick="onserverclickhandler"
    target="linkedcontentframeorwindow"
    title="titledisplayedbybrowser"
>
    linktext
</a>
```

其语法说明如表 4 - 2 所示。

表 4 - 2 HtmlAnchor 控件语法说明

runat="server"	在服务器端执行
id="programmaticID"	ID 序号
href="linkurl"	链接的网址
name="bookmarkname"	名称
OnServerClick="onserverclickhandler"	按下按钮所调用的程序
target="linkedcontentframeorwindow"	打开链接后出现的目标窗口
title="titledisplayedbybrowser"	
Linktext	链接的文字

例如，使用 VB 程序语言编写的 ch04/test4_05vb.html 的源代码如下：

```
<html>
```

```
<head>
<title>HTML 控件举例</title>
<meta http-equiv="Content-Type"
    content="text/html; charset=gb2312">
</head>
<body>
<h3><font face="宋体">单纯 HTML 例子</font></h3>
<!--加入链接-->
<a href="http://www.singnet.com.cn/">链接举例—世纪星</a>
</body>
</html>
```

运行结果如图 4 - 7 所示。



图 4 - 7 HTML 控件举例运行结果

在上面的例子中，使用了以下的语句来建立链接：

```
<a href="http://www.singnet.com.cn/">链接举例—世纪星 </a>
```

在 ASP.Net 中，HtmlAnchor 增加了 Href、Target、Title 等属性，这些属性的作用如下：

？ Href：指定链接的网址，与<a>标记的 Href 属性是一样的。

？ Target：用来显示链接的目标窗口，与<a>标记中的 Target 属性是一样的。

？ Title：当鼠标移动到已经加链接的文字上面，提示框里的文字就由此来指定。

使用 VB 程序语言编写的 ch04/test4_05vb.aspx 的源代码如下：

```
<html>
<head>
<title>HTML 控件举例</title>
<meta http-equiv="Content-Type"
    content="text/html; charset=gb2312">
</head>
<script language="VB" runat="server">
    Sub Page_Load(sender As Object, e As EventArgs)
        '显示的内容
        anchor1.Innerhtml="链接举例—世纪星"
        '链接的网址
```

```

        anchor1.HRef = "http://www.singnet.com.cn/"
'目标窗口
anchor1.Target="_blank"
'鼠标移动到上面时，显示的文字
anchor1.Title="世纪星"
End Sub
</script>
<body>
    <h3><font face="宋体">简单的 HtmlAnchor 例子</font></h3>
    <form runat=server>
        <p>
<!--调用在服务器运行的链接-->
        <a id=anchor1 runat="server">    </a>
        </form>
</body>
</html>

```

使用 C # 程序语言编写的 ch04 / test4_05c.aspx 的源代码如下：

```

<html>
<head>
<title>HTML 控件举例</title>
<meta http-equiv="Content-Type"
content="text/html; charset=gb2312">
</head>
<script language="C#" runat=server>
    void Page_Load(Object sender, EventArgs e) {
        //显示的内容
        anchor1.InnerHtml="链接举例 - 世纪星";
        ///链接的网址
        anchor1.HRef = "http://www.singnet.com.cn/";
        //目标窗口
        anchor1.Target="_blank";
        //鼠标移动到上面时，显示的文字
        anchor1.Title="世纪星";
    }
</script>
    <h3><font face="宋体">简单的 HtmlAnchor 例子</font></h3>
    <form runat=server>
        <p>
<!--调用在服务器运行的链接-->

```

```

        <a id=anchor1 runat="server">
        </a>
    </form>
</body>
</html>

```

注意：在 C# 程序代码中，属性名称一定要正确书写，否则会出现编译错误，因为它对源代码的要求比 VB 严格。

运行结果如图 4-8 所示。



图 4-8 HTML 控件举例运行结果

使用 JavaScript 语言编写的程序 ch04 / test4_05js.aspx 的源代码如下：

```

<html>
<head>
<title>HTML 控件举例</title>
<meta http-equiv="Content-Type"
content="text/html; charset=gb2312">
</head>
<script language="JScript" runat=server>
    protected function Page_Load(sender:Object, e:EventArgs) : void {
        //显示的内容
        anchor1.InnerHtml="链接举例 - 世纪星";
        ///链接的网址
        anchor1.HRef = "http://www.singnet.com.cn/";
        //目标窗口
        anchor1.Target="_blank";
        //鼠标移动到上面时，显示的文字
        anchor1.Title="世纪星";
    }
</script>
    <h3><font face="宋体">简单的 HtmlAnchor 例子</font></h3>
    <form runat=server> <p>
<!--调用在服务器运行的链接-->

```

```

<a id=anchor1 runat="server"> </a>
</form></body>
</html>

```

运行结果如图 4-9 所示，单击链接就会出现一个新窗口，所链接的网址便呈现出来。

下面的例子介绍如何在访问数据的同时使用 HtmlAnchor 控件，例子中包含两个文件：一个是显示详细信息的 DetailsPage.aspx 文件，另外一个显示数据的文件。

DetailsPage.aspx 的源代码如下：

```

<html>
<head>
    <script language="VB" runat="server">
        Dim m_id As String
        Sub Page_Load(sender As Object, e As EventArgs)
            m_id=Request.QueryString("id")
        End Sub
    </script>
<title>显示链接的详细内容</title></head>
<body>
    <h3><font face="宋体">HtmlAnchor
    所链接的详细内容</font></h3>
    <h4><font face="宋体">
    <!--显示提交的内容-->
    您刚才选择的是：<u><%= m_id %></u></font></h4>
</body>
</html>

```

本例中使用的语法如下，使用 Request 对象的 QueryString 方法来获得变量：

```
Request.QueryString("id")
```

显示内容则使用 Response 对象的 Write 方法，在例子中使用的是简写方法：

```
<%= m_id %>
```

使用 VB 程序语言编写的 ch04 / test4_06vb.aspx，其源代码如下：

```

<%@ Import Namespace="System.Data" %>
<html><title>DataBinding HtmlAnchor 控件举例</title>
<script language="VB" runat="server">
    Sub Page_Load(sender As Object, e As EventArgs)
        Dim dt As DataTable
        Dim dr As DataRow
        Dim i As Integer

```



图 4-9 HTML 控件举例运行结果

```
'创建数据表格
```

```
dt = New DataTable
```

```
'增加表格的列
```

```
dt.Columns.Add(New DataColumn("IntegerValue", GetType(Integer)))
```

```
dt.Columns.Add(New DataColumn("StringValue", GetType(String)))
```

```
dt.Columns.Add(New DataColumn("DateTimeValue", GetType(DateTime)))
```

```
dt.Columns.Add(New DataColumn("BoolValue", GetType(Boolean)))
```

```
dt.Columns.Add(New DataColumn("CurrencyValue", GetType(Double)))
```

```
'给表格赋值
```

```
For i = 1 To 9
```

```
    dr = dt.NewRow()
```

```
    dr(0) = i
```

```
    dr(1) = "单元 " + i.ToString()
```

```
    dr(2) = DateTime.Now.ToShortTimeString
```

```
    If (i Mod 2 <> 0) Then
```

```
        dr(3) = True
```

```
    Else
```

```
        dr(3) = False
```

```
    End If
```

```
    dr(4) = 1.23 * (i+1)
```

```
'增加数据到表格
```

```
dt.Rows.Add(dr)
```

```
Next
```

```
MyRepeater.DataSource=new DataView(dt)
```

```
'绑定数据
```

```
MyRepeater.DataBind
```

```
End Sub
```

```
</script>
```

```
<body>
```

```
<h3><font face="宋体">DataBinding
```

```
HtmlAnchor 举例</font></h3>
```

```
    <p>
```

```
    <form runat=server>
```

```
    <font face="宋体" size="2">
```

```
    <asp:Repeater id="MyRepeater" runat="server">
```

```
    <template name="ItemTemplate"> 数据链接<a href='<%# DataBinder.Eval(Container,"DataItem.  
    StringValue", "detailspage.aspx?id={0}") %>' runat="server">
```

```
    <%# DataBinder.Eval(Container, "DataItem.StringValue") %>
```

```
    </a>
```

```

<p>
        </template>
    </asp:Repeater>
</font>
</form>
</body>
</html>

```

使用 C# 语言编写的程序 ch04 / test4_06c.aspx 的源代码如下：

```

<%@ Import Namespace="System.Data" %>
<html>
<script language="C#" runat="server">
    void Page_Load(Object Src, EventArgs E) {
        DataTable dt = new DataTable();
        DataRow dr;

        dt.Columns.Add(new DataColumn("IntegerValue", typeof(Int32)));
        dt.Columns.Add(new DataColumn("StringValue", typeof(string)));
        dt.Columns.Add(new DataColumn("DateTimeValue", typeof(DateTime)));
        dt.Columns.Add(new DataColumn("BoolValue", typeof(bool)));
        dt.Columns.Add(new DataColumn("CurrencyValue", typeof(double)));

        for (int i = 0; i < 9; i++) {
            dr = dt.NewRow();
            dr[0] = i;
            dr[1] = "单元" + Int32.ToString(i);
            dr[2] = DateTime.Now;
            dr[3] = (i % 2 != 0) ? true : false;
            dr[4] = 1.23 * (i+1);
            dt.Rows.Add(dr);
        }
        MyRepeater.DataSource=new DataView(dt);
        MyRepeater.DataBind();
    }
</script>
<body>
<h3><font face="宋体">DataBinding HtmlAnchor
举例</font></h3>
    <p>
        <form runat=server>
            <font face="宋体" size="2">
<asp:Repeater id="MyRepeater" runat="server">

```

```

<template name="ItemTemplate">
    链接
    <a href='<%# DataBinder.Eval(Container,"DataItem.StringValue","detailspagec.aspx?id={0}") %>'
        runat="server">
    <%# DataBinder.Eval(Container, "DataItem.StringValue") %>
    </a>
    <p>
    </template>
</asp:Repeater>
</font>
</form>
</body>
</html>

```

程序 ch04\ DetailsPagec.aspx 的源代码如下：

```

<%@ Page Language="C#" %>
<html>
<head>
    <script runat="server">
        public string id;
        protected void Page_Load(Object Sender, EventArgs e)
        {
            id=Request.QueryString["id"];
        }
    </script>
<title>显示链接的详细内容</title></head>
<body>
    <h3><font face="宋体">HtmlAnchor
    所链接的详细内容</font></h3>
    <h4><font face="宋体">您刚才选择的是：
    <u><%= id %></u></font></h4>
</body>
</html>

```

使用 JavaScript 语言编写的程序 ch04 / test4_06js.aspx 的源代码如下：

```

<%@ Import Namespace="System.Data" %>
<html>
<script language="JScript" runat="server">
    public function Page_Load(Src:Object, E:EventArgs) : void {
        var dt:DataTable = new DataTable( );
        var dr:DataRow;
    }

```

```

dt.Columns.Add(new
DataColumn("IntegerValue", System.Type.GetType("System.Int32")));
    dt.Columns.Add(new
DataColumn("StringValue", System.Type.GetType("System.String")));
    dt.Columns.Add(new
DataColumn("DateTimeValue", System.Type.GetType("System.DateTime")));
    dt.Columns.Add(new
DataColumn("BoolValue", System.Type.GetType("System.Boolean")));
    dt.Columns.Add(new
DataColumn("CurrencyValue", System.Type.GetType("System.Double")));
    for (var i:int = 0; i < 9; i++) {
        dr = dt.NewRow();
        dr[0] = i;
        dr[1] = "单元 " + Int32.ToString(i);
        dr[2] = DateTime.Now;
        dr[3] = (i % 2 != 0) ? true : false;
        dr[4] = 1.23 * (i+1);
        dt.Rows.Add(dr);
    }
    MyRepeater.DataSource=new DataView(dt);
    MyRepeater.DataBind();
}
</script>
<body>
<h3><font face="宋体">DataBinding HtmlAnchor
举例</font></h3> <p>
    <form runat=server>
        <font face="宋体" size="2">
<asp:Repeater id="MyRepeater" runat="server">
    <template name="ItemTemplate"> 链接
<a href='<%# DataBinder.Eval(Container,"DataItem.StringValue","detailspage.aspx?id={0}") %>'
    runat="server">
<%# DataBinder.Eval(Container, "DataItem.StringValue") %>
        </a>
        <p>
    </template>
</asp:Repeater>
</font>
</form>

```

```
</body>
```

```
</html>
```

使用 JavaScript 语言编写的程序 ch04 / DetailsPagejs.aspx 的源代码如下：

```
<%@ Page Language="JScript" %>
```

```
<html>
```

```
<head>
```

```
    <script runat="server">
```

```
        public var id:String;
```

```
        protected function Page_Load(Sender:Object, e:EventArgs) : void
```

```
        {    id=Request.QueryString["id"];    }
```

```
    </script>
```

```
<title>显示链接的详细内容</title></head>
```

```
<body><h3>
```

```
    <font face="宋体">HtmlAnchor 所链接的详细内容
```

```
    </font>
```

```
    </h3>
```

```
    <h4><font face="宋体">您刚才选择的是：
```

```
    <u><%= id %>
```

```
    </u>
```

```
    </font>
```

```
    </h4>
```

```
</body>
```

```
</html>
```

运行结果如图 4 - 10 所示。

打开上面的页面链接，会看到链接的内容，结果如图 4 - 11 所示。



图 4 - 10 HTML 控件举例运行结果

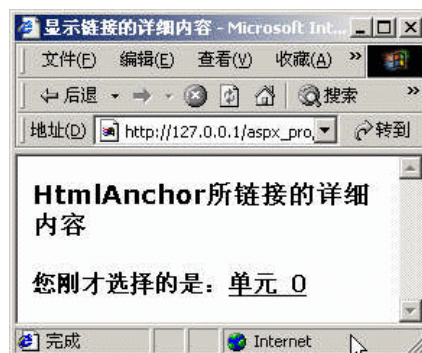


图 4 - 11 HTML 控件举例运行结果

在上面的例子中，首先是创建一个表格，规定表格的列，然后在表格里填充数据，这样组成一个数据的表格，也可使用数据库。（在以后的章节会专门介绍如何访问数据库。）

以<a>标记为例，HTML 控件比 HTML 标记多了“ runat="server" ”及“ id=控件名称 ”两种属性，其作用是：

“ runat="server" ”告诉 ASP.NET，这是一个在 Server 端执行的控件，所有 Server 控件都要加上这个属性。“ id=控件名称 ”用来指定这个控件的名称，此时这 本例中的 id=anchor1 中的 anchor1) 会成为 ASP.NET 程序的一个对象名称。

4.3 HtmlButton 控 件

HtmlButton 控件新增加了 OnServerClick 事件，此事件发生于按钮被按下时。如果想调用执行程序的话，那么可以将程序写在 OnServerClick 事件程序中，其语法如下：

```
<button
    runat="server"
    id="programmaticID"
    OnServerClick="onserverclickhandler"
>
    buttontext, image, or control
</button>
```

其说明如表 4 - 3 所示。

表 4 - 3 HtmlButton 控件语法说明

runat="server"	在服务器端执行
id="programmaticID"	ID 序号
onServerClick="onserverclickhandler"	所调用的事件程序
Buttontext, image ,or control	按钮的类型

下面来看一个例子。使用 VB 程序语言编写的 ch04 / test4_07vb.aspx 的源代码如下：

```
<html>
<head>
    <script language="VB" runat="server">
        Sub Button1_OnClick(sender As Object, e As EventArgs)
            Span1.InnerHtml="你点击按钮 1"
        End Sub
        Sub Button2_OnClick(sender As Object, e As EventArgs)
            Span1.InnerHtml="你点击按钮 2"
        End Sub
    </script>
<title>HTML 控件 HtmlButton 举例</title></head>
<body>
<h3><font face="宋体">HtmlButton 的例子</font></h3>
```

```

<form runat=server>
    <font face="宋体" size="2">
        <p>
<!--布置按钮 Button1，并且设定其样式-->
<button id=" " onServerClick="Button1_OnClick"
    style="font:8pt 宋体;background - color:lightgreen;
border - color:black;height=30;width:100" runat="server">
     点我! </button>
使用 HTML 的<img>标记的按钮
    <p>
<!--布置按钮 Button2，并且设定其样式，增加行为-->
<button id=Button2
    onServerClick="Button2_OnClick"
    style="font: 8pt 宋体;
    background - color:lightgreen;
    border - color:black;height=30;width:100"
    onmouseover="this.style.backgroundColor='yellow'"
    onmouseout="this.style.backgroundColor='lightgreen'"
    runat="server">
    也点击我一下呀!
    </button>
    可以闪烁的按钮 <p>
    <span id=Span1 runat=server />
</font>
</form>
</body>
</html>

```

使用 C# 程序语言编写的 ch04 / test4_07c.aspx 的源代码如下：

```

<html>
<head>
    <script language="C#" runat="server">
void Button1_OnClick(object Source, EventArgs e)
{
    Span1.InnerHtml="你点击按钮 1";
}

void Button2_OnClick(object Source, EventArgs e)
{
    Span1.InnerHtml="你点击按钮 2";
}

```

```

</script>
<title>HTML 控件 HtmlButton 举例</title>
</head>
<body>
<h3><font face="宋体">HtmlButton 的例子</font>
</h3>
<form runat=server>
<font face="宋体" size="2">
<p>
<!--布置按钮 Button1，并且设定其样式-->
<button id="Button1"
onServerClick="Button1_OnClick"
style="font: 8pt 宋体;
background-color:lightgreen;
border-color:black;
height=30;width:100"
runat="server"
>
 点我! </button>
使用 HTML 的<img>标记的按钮
<p>
<!--布置按钮 Button2，并且设定其样式，并且增加行为-->
<button id=Button2
onServerClick="Button2_OnClick"
style="font: 8pt 宋体;
background-color:lightgreen;
border-color:black;height=30;
width:100"
onmouseover="this.style.backgroundColor='yellow'"
onmouseout="this.style.backgroundColor='lightgreen'"
runat="server">
也点击我一下呀!
</button>
可以闪烁的按钮 <p>
<span id=Span1 runat=server />
</font>
</form>
</body>
</html>

```

使用 JavaScript 语言编写的程序 ch04 / test4_07js.aspx 源代码如下：

```
<html>
<head>
    <script language="JScript" runat="server">
        function Button1_OnClick(Source:Object, e:EventArgs) : void {
            Span1.InnerHtml="你点击按钮 1";
        }

        function Button2_OnClick(Source:Object, e:EventArgs) : void
        {
            Span1.InnerHtml="你点击按钮 2";
        }
    </script>
<title>HTML 控件 HtmlButton 举例</title>
</head>
<body>
<h3><font face="宋体">HtmlButton 的例子</font>
</h3>
    <form runat=server>
        <font face="宋体" size="2">
            <p>
<!--布置按钮 Button1，并且设定其样式-->
                <button id="Button1"
                    onServerClick="Button1_OnClick"
                    style="font: 8pt 宋体;
                    background-color:lightgreen;
                    border-color:black;
                    height=30;width:100"
                    runat="server"
                >
                    
                        点我!
            </button>
使用 HTML 的<img>标记的按钮
            <p>
<!--布置按钮 Button2，并且设定其样式，增加行为-->
                <button id=Button2
                    onServerClick="Button2_OnClick"
                    style="font: 8pt 宋体;
                    background-color:lightgreen;
```

```

border-color:black;
height=30;
width:100"
onmouseover="this.style.backgroundColor='yellow'"
onmouseout="this.style.backgroundColor='lightgreen'"
runat="server"
>
也点击我一下呀!
</button>
        可以闪烁的按钮
<p>
    <span id=Span1 runat=server />
</font>
</form>
</body>
</html>

```

运行结果如图 4 - 12 所示。

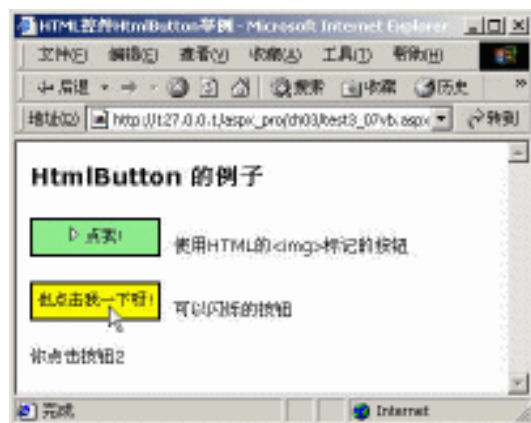


图 4 - 12 HtmlButton 控件举例运行结果

注意：HtmlButton 控件只适用于 IE 4.0 和 NetScape 6.0 以上的版本，在编写程序的时候必须将其安排在<Form runat="server">和</Form>之间。

在 test4_07vb.aspx、test4_07c.aspx 和 test4_07js.aspx 例子中，定义了两个 HtmlButton，同时还定制了它们的 Style 样式，以一个图片为背景。程序要点如下：

```

<button id="Button1" onServerClick="Button1_OnClick"
style="font:8pt 宋体;background-color:lightgreen;
border-color:black;height=30;width:100" runat="server">
     点我! </button>

```

第二个按钮不仅定制了它的样式，而且还增加了鼠标行为 onmouseover、onmouseout)，

使整个按钮制作得生动形象，其要点如下：

```
<button id=Button2 onServerClick="Button2_OnClick"
        style="font: 8pt 宋体;background - color:lightgreen;
border - color:black;height=30;width:100"
        onmouseover="this.style.backgroundColor='yellow'"
        onmouseout="this.style.backgroundColor='lightgreen'"
        runat="server"> 也点击我一下呀! </button>
```

提示：其中 Button1 的属性被设定为"Button1_OnClick"，只要按下 Button1 就会调用 Button1_OnClick 事件，同样地 Button1 的属性被设定为"Button2_OnClick"，按下 Button2 就会调用 Button2_OnClick 事件。

4.4 HtmlForm 控 件

制作网页时经常会遇到表单数据的处理，在 ASP 中采用数据上传模式，而在 ASP.Net 中采用事件驱动模式，其语法如下：

```
<form
    runat="server"
    id="programmaticID"
    method=POST | GET
    action="srcpageURL"
>
    Other controls, input forms, and so on.
</form>
```

语法说明如表 4 - 4 所示。

表 4 - 4 HtmlForm 控件语法说明

runat="server"	在服务器端执行
id="programmaticID"	ID 名称
method=POST GET	传送数据的方法
action="srcpageURL"	处理提交数据的页面
Other controls, input forms, and so on.	其他的控件等

下面通过实例来加以说明。使用 VB 程序语言编写的 ch04/tes4_08vb.aspx 的源代码如下：

```
<html>
<head>
<title>HtmlForm 简单举例</title>
<meta http - equiv="Content - Type" content="text/html;
charset=gb2312">
```

```
</head>
  <body>
    <h3><font face="宋体">HtmlForm 简单举例</font>
    </h3>
    <form runat="server">
      大名：<Input Type=Text
        id="Name"
        Size=20
        runat="server"
      />
      <P>
      爱好：<Input Type=Text
        id="Love"
        Size=20
        runat="server"
      />
      <P>
      <Input Type=Button
        Value="提交"
        id="Button1"
        OnServerClick="Button1_Click"
        runat="server"
      />
      <span id=Span1 runat=server />
    </form>
    <p>
    <!--显示提交的内容-->
    <asp:Label
      id="Label1"
      runat="server"
    />
  </body>
</html>
<script language="VB" runat="server">
  Sub Button1_Click(sender As Object, e As EventArgs)
    Span1.InnerHtml="你刚才按下了提交按钮"
    Label1.Text = Name.Value & "你好！你喜欢的是：" & Love.Value
  End Sub
</script>
```

使用 C# 程序语言编写的程序 ch04 / test4_08c.aspx 源代码如下：

```
<html>
<head>
    <script language="C#" runat="server">
        void Button1_Click(object Source, EventArgs e)
        {
            Span1.InnerHtml="你点击按钮 1";
        }
    </script>
<title>HtmlForm 简单举例</title>
<meta http-equiv="Content-Type"
content="text/html; charset=gb2312">
</head>
<body>
    <h3><font face="宋体">HtmlForm 简单举例
        </font>
    </h3>
    <form runat="server">
        大名：<Input Type=Text
            id="Name"
            Size=20
            runat="server"
        />
        <P>
        爱好：<Input Type=Text
            id="Love"
            Size=20
            runat="server"
        />
        <P>
        <Input Type=Button
            Value="提交"
            id="Button1"
            OnServerClick="Button1_Click"
            runat="server"
        />
        <!--显示提交的内容-->
        <span id=Span1 runat=server />
    </form>
```

```
<p>
<asp:Label id="Label1" runat="server"/>
</body>
</html>
```

使用 JavaScript 语言编写的程序源代码 ch04 / test4_08js.aspx 如下：

```
<html>
<head>
    <script language="JScript" runat="server">
        function Button1_Click(Source:Object, e:EventArgs) : void
        {
            Span1.InnerHtml="你点击按钮 1";
            Label1.Text = Name.Value + "你好！你喜欢的是：" + Love.Value;
        }
    </script>
<title>HtmlForm 简单举例</title>
<meta http-equiv="Content-Type" content="text/html;
charset=gb2312">
</head>
<body>
    <h3><font face="宋体">HtmlForm 简单举例
        </font>
    </h3>
    <form runat="server">
        大名：<Input Type=Text
            id="Name"
            Size=20
            runat="server"
        />
        <P>
        爱好：<Input Type=Text
            id="Love"
            Size=20
            runat="server"
        />
        <P>
        <Input Type=Button
            Value="提交"
            id="Button1"
            OnServerClick="Button1_Click"
```

```
runat="server"
/>
<!--显示提交的内容-->
<span id=Span1 runat=server />
</form>
<p>
<asp:Label
    id="Label1"
    runat="server"
/>
</body>
</html>
```

运行结果如图 4-13 所示。

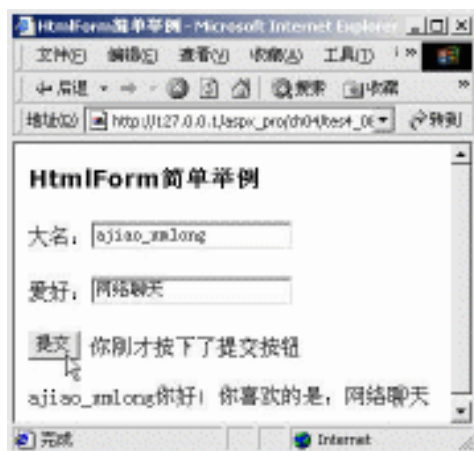


图 4-13 HtmlForm 控件举例运行结果

注意 :在 ASP 中 ,使用 Request 对象读取表单的内容 ,然后显示提交内容 ,而在 ASP.Net 中 ,表达及处理表单数据的程序属于同一个.aspx 网页 ,就没有必要使用 Request 对象来读取。因为按下提交按钮时 ,所调用的是同一个网页中的程序 ,读取的也是同一个页面的数据。

本节中的例子有如下的特点 :

<form runat="server"> : 不像 ASP ,不用设定其 Action 属性 ,由于表单和处理表单数据的程序在同一个页面里 ,所调用的程序就是它自己 ,没有必要设定其 Action 属性。

<Input Type=Button...> : 其中的 OnServerClick 属性被设定为 "Button1_Click" ,按下按钮就会调用 Button1_Click 事件。

<Input Type=Text id="..."> : 两个文本框的 id 分别是 Name 和 Love ,在显示提交数据的过程中 ,只有 Name.Value 和 Love.Value 才可以读取文本框的内容。

4.5 HtmlGenericControl 控件

HtmlGenericControl 控件的语法如下：

```
<span | body | div | font | others
    runat="server"
    id="programmaticID"
>
Contentbetweentags
</span | body | div | font | others>
```

其语法说明如表 4 - 5 所示。

表 4 - 5 HtmlGenericControl 控件语法说明

Runat="server"	在服务器端执行
id="programmaticID"	ID 序号

在本节的例子中就是改变<body></body>的 Bgcolor，根据客户端提交的选择结果，改变背景颜色。

使用 VB 语言编写的程序 ch04/test4_09vb.aspx 的源代码如下：

```
<html>
<head>
    <script language="VB" runat="server">
        按下确定按钮所调用的事件 SubmitBtn_Click
        Sub SubmitBtn_Click(sender As Object, e As EventArgs)
            Body.Attributes("bgcolor") = ColorSelect.Value
        End Sub
    </script>
<title>HtmlGenericControl 简单举例</title>
</head>
<!--显示 Body -->
<body id=Body runat=server>
<h3><font face="宋体">HtmlGenericControl 简单例子
    </font>
</h3>
<!--开始表单-->
    <form runat=server>
        <p> 请选择本页面的背景色：
    <!--布置一个下拉选单-->
        <p> <select id="ColorSelect" runat="server">
```

```
        <option value="White">白色</option>
        <option value="Wheat">麦黄</option>
        <option value="Gainsboro">浅蓝</option>
        <option value="LemonChiffon">桔黄</option>
    </select>
<!--布置"确定"按钮，其 OnServerClick 事件为 SubmitBtn_Click -->
    <input type="submit"
        runat="server"
        Value="确定"
        OnServerClick="SubmitBtn_Click">
<!--结束表单-->
</form>
</body>
</html>
```

使用 C # 语言编写的程序 ch04 / test4_09c.aspx 源代码如下：

```
<html>
<head>
    <script language="C#" runat="server">
        //按下确定按钮所调用的事件 SubmitBtn_Click()
        void SubmitBtn_Click(object Source, EventArgs e) {
            Body.Attributes["bgcolor"] = ColorSelect.Value;
        }    </script>
</head>
<!--显示 Body -->
<body id="Body" runat="server">
    <h3>
    <font face="宋体">HtmlGenericControl 简单例子
    </font>
    </h3>
    <!--开始表单-->
        <form runat="server">
            <p> 请选择本页面的背景色：  <p>
    <!--布置一个下拉选单-->
    <select id="ColorSelect" runat="server">
        <option value="White">白色</option>
        <option value="Wheat">麦黄</option>
        <option value="Gainsboro">浅蓝</option>
        <option value="LemonChiffon">桔黄</option>
```

```

        </select>
<!--布置"确定"按钮，其 OnServerClick 事件为 SubmitBtn_Click -->
        <input type="submit"
                runat="server"
                Value="确定"
                OnServerClick="SubmitBtn_Click">
<!--结束表单-->
</form>
</body>
</html>

```

使用 JavaScript 语言编写的程序 ch04/test4_09js.aspx 的源代码如下：

```

<html>
<head>
        <script language="JScript" runat="server">
                function SubmitBtn_Click(Source:Object, e:EventArgs) : void {
                        Body.Attributes["bgcolor"] = ColorSelect.Value; }
        </script>
<title>HtmlGenericControl 简单举例</title>
</head>
<!--显示 Body -->
<body id=Body runat=server>
<h3>
<font face="宋体">HtmlGenericControl 简单例子
</font>
</h3>
<!--开始表单-->
        <form runat=server>
                <p> 请选择本页面的背景色：
<!--布置一个下拉选单-->
                <p> <select id="ColorSelect" runat="server">
                        <option value="White">白色</option>
                        <option value="Wheat">麦黄</option>
                        <option value="Gainsboro">浅蓝</option>
                        <option value="LemonChiffon">桔黄</option>
                </select>
<!--布置"确定"按钮，其 OnServerClick 事件为 SubmitBtn_Click -->
                <input type="submit"
                        runat="server"
                        Value="确定"

```

```
OnServerClick="SubmitBtn_Click">
<!--结束表单-->
</form>
</body>
</html>
```

运行结果如图 4 - 14 所示。



图 4 - 14 HtmlGenericControl 控件举例运行结果

4.6 HtmlImage 控 件

HtmlImage 控件允许根据 HTML 标签编写程序,此控件允许设置图片的位置、宽、高度、边界宽度、鼠标移过时显示的提示文字和是否居中等图片属性。

注意：此控件不需要封闭标签。

HtmlImage 控件是用来显示图片的控件的，其语法如下：

```

```

语法说明如表 4 - 6 所示。

表 4 - 6 HtmlImage 控件语法说明

id="programmatic "	程序 ID 序号
runat="server"	在服务器端执行
alt="alttext"	alt 文字
align= top middle bottom left right	对齐方式
border="borderwidth"	边框宽度
height="imageheight"	图片的高度
Src="imageURI"	图片的网络地址
width="imagewidth"	图片的宽度

在 ch04/test4_10vb.aspx、ch04/test4_10c.aspx、ch04/test4_10js.aspx 中 ,使用了 HtmlImage 控件 , 根据客户端提交的结果来显示不同的图片。

使用 VB 语言编写的程序 ch04/test4_10vb.aspx 的源代码如下 :

```
<html>
<head>
    <script language="VB" runat="server">
        按下"应用"按钮所调用的程序
        Sub SubmitBtn_Click(sender As Object, e As EventArgs)
            调用图片
            Image1.Src = "../images/" & Select1.Value
        End Sub
    </script>
<title>HtmlImage 举例</title>
</head>
<body>
<h3><font face="宋体"> HtmlImage 举例
</font>
</h3>
<!--开始表单-->
    <form runat=server>
        
        <p> 请选择图片文件 :
    <!--布置一个下拉选单-->
        <select id="Select1" runat="server">
            <option Value="1.gif">1 号图片</option>
            <option Value="2.gif">2 号图片</option>
            <option Value="3.gif">3 号图片</option>
            <option Value="4.gif">4 号图片</option>
```

```
        <option Value="5.gif">5 号图片</option>
        <option Value="6.gif">6 号图片</option>
    </select>
<!--布置"应用"按钮，其 OnServerClick 事件为 SubmitBtn_Click -->
    <input type="submit"
        runat="server"
        Value="应用"
        OnServerClick="SubmitBtn_Click">
<!--结束表单-->
</form>
</body>
</html>
```

使用 C # 语言编写的程序 ch04 / test4_10c.aspx 的源代码如下：

```
<html>
<head>
    <script language="C#" runat="server">
        void SubmitBtn_Click(Object Sender, EventArgs e) {
            //调用图片
            Image1.Src="../images/" + Select1.Value;
        }
    </script>
<title>HtmlImage 举例</title>
</head>
<body>
<h3>
<font face="宋体"> HtmlImage 举例
</font>
</h3>
<!--开始表单-->
<form runat="server">
    
    <p> 请选择图片文件：
<!--布置一个下拉选单-->
        <select id="Select1" runat="server">
            <option Value="1.gif">1 号图片</option>
            <option Value="2.gif">2 号图片</option>
            <option Value="3.gif">3 号图片</option>
            <option Value="4.gif">4 号图片</option>
            <option Value="5.gif">5 号图片</option>
```

```

        <option Value="6.gif">6 号图片</option>
    </select>
    <!--布置"应用"按钮，其 OnServerClick 事件为 SubmitBtn_Click -->
    <input type="submit"
        runat="server"
        Value="应用"
        OnServerClick="SubmitBtn_Click">
    <!--结束表单-->
</form>
</body>
</html>

```

使用 JavaScript 语言编写的程序 ch04/test4_10js.aspx 的源代码如下：

```

<html>
<head>
    <script language="JScript" runat="server">
        function SubmitBtn_Click(Sender:Object, e:EventArgs) : void {
            //调用图片
            Image1.Src=" ../images/" + Select1.Value;
        }
    </script>
</head>
<title>HtmlImage 举例</title>
</head>
<body>
    <h3>
    <font face="宋体"> HtmlImage 举例
    </font>
    </h3>
    <!--开始表单-->
    <form runat=server>
        
        <p> 请选择图片文件：
    <!--布置一个下拉选单-->
        <select id="Select1" runat="server">
            <option Value="1.gif">1 号图片</option>
            <option Value="2.gif">2 号图片</option>
            <option Value="3.gif">3 号图片</option>
            <option Value="4.gif">4 号图片</option>
            <option Value="5.gif">5 号图片</option>
            <option Value="6.gif">6 号图片</option>

```

```
</select>
<!-- 布置"应用"按钮，其 OnServerClick 事件为 SubmitBtn_Click -->
<input type="submit"
       runat="server"
       Value="应用"
       OnServerClick="SubmitBtn_Click"
       >
<!-- 结束表单 -->
</form>
</body>
</html>
```

运行结果如图 4 - 15 所示。

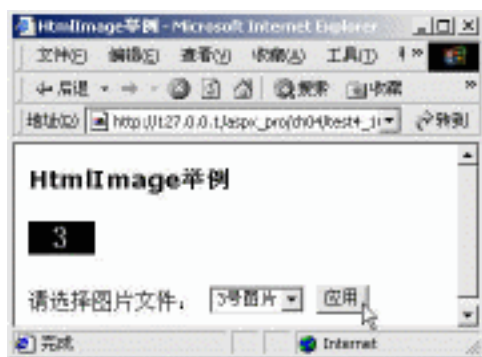


图 4 - 15 HtmlImage 控件举例运行结果

4.7 HtmlInputButton 控件

HtmlInputButton 控件用来根据 HTML 的 `<input type=button>`、`<input type=submit>` 和 `<input type=reset>` 标签编写程序。

当用户按下 HTML 提交按钮时，服务器会根据编写的程序处理用户的请求，或者显示提交的数据，然后把用户的请求输送给浏览器。

HtmlInputText 和 HtmlTextArea 控件，可以用来制作能在服务器上运行的检验用户输入的页面。

注意：此控件不需要封闭标签。

HtmlInputButton 控件的语法如下：

```
<input
  type=button | submit | reset
  runat="server"
  id="programmaticID"
```

```
OnServerClick="onserverclickhandler"
>
```

语法说明如表 4 - 7 所示。

表 4 - 7 HtmlInputButton 控件语法说明

type=button submit reset	按钮类型
runat="server"	在服务器端执行
id="programmaticID"	ID 序号
OnServerClick="onserverclickhandler"	按下按钮所调用的事件或程序

在本节的例子中，test4_11vb.aspx、test4_11c.aspx 和 test4_11js.aspx 中布置了两个按钮。一个用来提交，一个用来重置，两者的不同就是由<input type=...>中 type 设置的值来决定的。

4.7.1 HtmlInputButton 控件举例一

使用 VB 语言编写的程序 ch04/test4_11vb.aspx 的源代码如下：

```
<html>
<head>
  <script language="VB" runat="server">
    '按下提交按钮所调用的事件 SubmitBtn_Click
    Sub SubmitBtn_Click(sender As Object, e As EventArgs)
      '判断 Password 的值是否为 ASP.Net
      If Password.Value = "ASP.Net" Then
        Span1.InnerHtml = "口令正确！"
      Else
        Span1.InnerHtml="您的口令不正确！"
      End If
    End Sub
    '按下重置按钮所调用的事件 ResetBtn_Click
    Sub ResetBtn_Click(sender As Object, e As EventArgs)
      '重置 Name.Value 为空
      Name.Value = ""
      '重置 Password.Value 为空
      Password.Value = ""
    End Sub
  </script>
<title>HtmlInputButtons 举例</title>
</head>
<body>
```

```
<h3>
<font face="宋体">提交和重置
HtmlInputButtons 举例
</font>
</h3>

    <form runat=server>
        请输入帐号:
        <input id="Name"
            type=text
            size=24
            runat=server
        >

        <p> 请输入你的口令:
        <input id="Password"
            type=password
            size=24
            runat=server
        >

        (口令就是"ASP.Net")<p>
        <input type=submit
            value="提交"
            OnServerClick="SubmitBtn_Click"
            runat=server>

        <input type=reset
            OnServerClick="ResetBtn_Click"
            runat=server
            value="重置"
            name="Reset">

        <p>
        <span id="Span1"
            style="color:red"
            runat=server>

        </span>
    </form>
</body>
</html>
```

使用 C# 程序语言编写的 ch04/test4_11c.aspx 的源代码如下：

```
<html>
<head>
```

```
<script language="C#" runat="server">
    //按下提交按钮所调用的事件 SubmitBtn_Click()
    void SubmitBtn_Click(object Source, EventArgs e) {
        //判断 Password.Value 是否等于 ASP.Net
        if (Password.Value == "ASP.Net") {
            Span1.InnerHtml="口令正确！";
        }
        else {Span1.InnerHtml="您的口令不正确！";}
    }
    //按下重置按钮所调用的事件 ResetBtn_Click()
    void ResetBtn_Click(object Source, EventArgs e) {
        //重置 Name.Value 为空
        Name.Value = "";
        //重置 Password.Value 为空
        Password.Value = "";
    }
</script>
<title>HtmlInputButtons 举例</title></head>
<body>
<h3>
<font face="宋体" size="2">提交和重置 HtmlInputButtons 举例
</font>
</h3>
<form runat=server>
    <font size="2"> 请输入帐号:
    <input id="Name"
        type=text
        size=16
        runat=server>
    </font>
<p>
<font size="2"> 请输入你的口令:
    <input id="Password"
        type=password
        size=16
        runat=server>
    (口令就是"ASP.Net")
</font>
<p>
<font size="2">
    <input type=submit
        value="提交"
```

```

        OnServerClick="SubmitBtn_Click"
        runat=server>
<input type=reset
        OnServerClick="ResetBtn_Click"
        runat=server
        value="重置"
        name="Reset">
        </font>
    <p>
    <span id="Span1"
        style="color:red"
        runat=server>
    </span>
</form>
</body>
</html>

```

使用 JavaScript 语言编写的程序 ch04 / test4_11js.aspx 的源代码如下：

```

<html>
<head>
    <script language="JScript" runat="server">
        //按下提交按钮所调用的事件 SubmitBtn_Click()
        function SubmitBtn_Click(Source:Object, e:EventArgs) : void {
            //判断 Password.Value 值是否为 ASP.Net
            if (Password.Value == "ASP.Net") {
                Span1.InnerHtml="口令正确！";
            }
            else { Span1.InnerHtml="您的口令不正确！";}
        }
        //按下重置按钮所调用的事件 ResetBtn_Click()
        function ResetBtn_Click(Source:Object, e:EventArgs) : void {
            Name.Value = "";
            Password.Value = "";
        }
    </script>
    <title>HtmlInputButtons 举例</title>
</head>
<body>
    <h3>
    <font face="宋体" size="2">提交和重置 HtmlInputButtons 举例
    </font>
    </h3>
    <form runat=server>

```

```
<font size="2"> 请输入帐号:
<input id="Name"
      type=text
      size=16
      runat=server
    >
</font>
<p>
<font size="2"> 请输入你的口令:
<input id="Password"
      type=password
      size=16
      runat=server
    >
    (口令就是"ASP.Net")
</font>
<p>
<font size="2">
<input type=submit
      value="提交"
      OnServerClick="SubmitBtn_Click"
      runat=server
    >
<input type=reset
      OnServerClick="ResetBtn_Click"
      runat=server
      value="重置"
      name="Reset"
    >
</font>
<p>
<span id="Span1"
      style="color:red"
      runat=server
    >
</span>
</form>
</body>
</html>
```

上面的例子是利用程序来验证用户的输入内容，根据其是否符合程序设定的条件，来确定返回浏览器的内容。程序运行结果如图 4 - 16 所示。

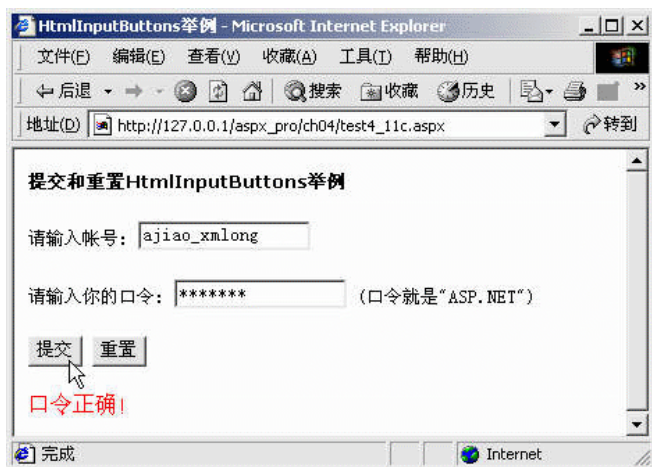


图 4 - 16 HtmlInputButton 控件举例运行结果

4.7.2 HtmlInputButton 控件举例二

在本例中，使用按钮控件来调用事件，然后输出事件结果。

使用 VB 语言编写的程序 ch04/test4_12vb.aspx 的源代码如下：

```
<html>
<head>
  <script language="VB" runat="server">
    按下确定按钮所调用的事件 Button1_Click
    Sub Button1_Click(sender As Object, e As EventArgs)
      Span1.InnerHtml = "刚才你点击了按钮！"
    End Sub
  </script>
</head>
<body>
  <h3>
    <font face="宋体">HtmlInputButton 按钮
  </font>
  </h3>
  <form runat=server>
    <p>
      <input type=button
```

```

        value="确定"
        onServerClick="Button1_Click"
        runat="server"
        name="Button"
    >
</form>
</body>
</html>

```

使用 C # 语言编写的程序 ch04/test4_12c.aspx 的源代码如下：

```

<html>
<head>
    <script language="C#" runat="server">
        //按下确定按钮所调用的事件 Button1_Click()
        void Button1_Click(object Source, EventArgs e)
        {
            Span1.InnerHtml="刚才你点击了按钮！";
        }
    </script>
<title>HtmlInputButton 举例二
</title>
</head>
<body>
<h3>
    <font face="宋体">HtmlInputButton 按钮
</font>
</h3>
<form runat=server>
    <p>
        <input type=button
            value="确定"
            onServerClick="Button1_Click"
            runat="server"
            name="Button"
        >
        <span id=Span1
            runat=server
        />
    </p>
</form>
</body>
</html>

```

使用 JavaScript 语言编写的程序 ch04 / test4_12js.aspx 的源代码如下：

```

<html>
<head>
    <script language="JScript" runat="server">
        //按下确定按钮所调用的事件 Button1_Click()
    function Button1_Click(Source:Object, e:EventArgs) :
    void { Span1.InnerHtml="刚才你点击了按钮！";}
    </script>
<title>HtmlInputButton 举例二</title>
</head>
<body>
<h3>
    <font face="宋体">HtmlInputButton 按钮
    </font>
</h3>
    <form runat=server>
    <p>
    <!--布置按钮，onServerClick 事件为 Button1_Click -->
    <input type=button
        value="确定"
        onServerClick="Button1_Click"
        runat="server"
        name="Button"
        >
    <!--显示交互信息-->
    <span id=Span1
        runat=server
        />
    </form>
</body>
</html>

```



图 4 - 17 HtmlInputButton 控件举例运行结果

在上面的例子中，利用在服务器端运行的程序，根据客户的操作来显示交互的信息，运行结果如图 4 - 17 所示。

4.8 HtmlInputCheckBox 控件

HtmlInputCheckBox 控件允许用户根据 HTML `<input type=checkbox>` 标签编写程序，可以用 HtmlInputButton 控件和事件来判断复选框是否被选定。

注意：这个控件不需要封闭标签。

其使用语法如下：

```
<input  
    type=checkbox  
    runat="server"  
    id="programmaticID"  
    checked  
>
```

语法说明如表 4 - 8 所示。

表 4 - 8 HtmlInputCheckBox 控件语法说明

type=checkbox	设定为复选框类型
runat="server"	在服务器端执行
id="programmaticID"	ID 序号
Checked	被选中

下面是 HtmlInputCheckBox 控件的例子。

使用 VB 语言编写的程序 ch04/test4_13vb.aspx 的源代码如下：

```
<html>  
<head>  
    <script language="VB" runat="server">  
        按下确定按钮所调用的事件 Button1_Click  
        Sub Button1_Click(sender As Object, e As EventArgs)  
            判断复选框是否被选定  
            If Check1.Checked Then  
                Span1.InnerHtml = "复选框被选中!"  
            Else  
                Span1.InnerHtml = "复选框没有被选中!"  
            End If  
        End Sub  
    </script>  
<title>HtmlInputCheckBox 举例  
</title>  
</head>  
<body>  
    <h3>  
    <font face="宋体">HtmlInputCheckBox 举例  
</font>  
</h3>  
<!--布置表单-->  
    <form runat=server>
```

```
<!--布置复选框-->
    <input id="Check1"
           type=checkbox
           runat="server"
    >
复选框 1<span id=Span1
           style="color:red"
           runat=server
    />

<p>
<!--布置按钮-->
    <input type=button
           id="Button1"
           value="确定"
           OnServerClick="Button1_Click"
           runat=server
    >

<!--结束表单-->
    </form>
</body>
</html>
```

使用 C# 语言编写的程序 ch04/test4_13c.aspx 的源代码如下：

```
<html>
<head>
    <script language="C#" runat="server">
        //按下确定按钮所调用的事件 Button1_Click()
        void Button1_Click(object Source, EventArgs e) {
            //判断复选框是否被选定
            if (Check1.Checked == true) {
                Span1.InnerHtml = "复选框被选中!";
            }
            else { Span1.InnerHtml = "复选框没有被选中!"; }
        }
    </script>
<title>HtmlInputCheckBox 举例
</title>
</head>
<body>
    <h3>
    <font face="宋体">
```

HtmlInputCheckBox 举例

```
</font>
```

```
</h3>
```

```
<!--布置表单-->
```

```
    <form runat=server>
```

```
<!--布置复选框-->
```

```
    <input id="Check1"
```

```
        type=checkbox
```

```
        runat="server"
```

```
    >
```

```
    复选框 1<span id=Span1
```

```
        style="color:red"
```

```
        runat=server
```

```
    />
```

```
<p>
```

```
<!--布置按钮-->
```

```
    <input type=button
```

```
        id="Button1"
```

```
        value="确定"
```

```
        OnServerClick="Button1_Click"
```

```
        runat=server
```

```
    >
```

```
    </span>
```

```
<!--结束表单-->
```

```
</form>
```

```
</body></html>
```

使用 JavaScript 语言编写的程序 ch04/test4_13js.asp 的源代码如下：

```
<html>
```

```
<head>
```

```
    <script language="JScript" runat="server">
```

```
        //按下确定按钮所调用的事件 Button1_Click()
```

```
        function Button1_Click(Source:Object, e:EventArgs : void {
```

```
            //判断复选框是否被选定
```

```
            if (Check1.Checked == true) {
```

```
                Span1.InnerHtml = "复选框被选中!"; }
```

```
            else { Span1.InnerHtml = "复选框没有被选中!"; }}
```

```
        </script>
```

```
<title>HtmlInputCheckBox 举例
```

```
</title>
```

```
</head>
<body> <h3>
<font face="宋体">
HtmlInputCheckBox 举例
</font></h3>
<!--布置表单-->
    <form runat=server>
<!--布置复选框-->
        <input id="Check1"
            type=checkbox
            runat="server"
            >
        复选框 1<span id=Span1
            style="color:red"
            runat=server
            />
    <p>
<!--布置按钮-->
        <input type=button
            id="Button1"
            value="确定"
            OnServerClick="Button1_Click"
            runat=server
            >
        </span>
<!--结束表单-->
    </form></body></html>
```

运行结果如图 4 - 18 所示。

按下“ 确定 ”按钮后，显示的结果如图 4 - 19 所示。

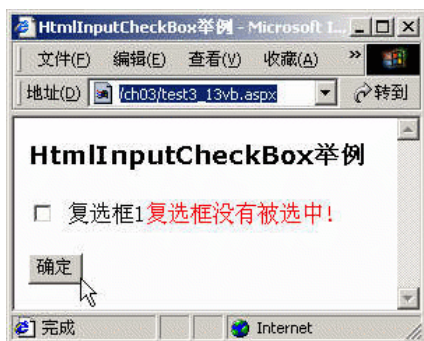


图 4 - 18 HtmlInputCheckBox 控件没有被选中时的运行结果

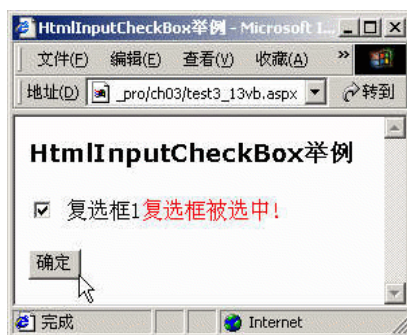


图 4 - 19 HtmlInputCheckBox 控件被选中时的运行结果

4.9 HtmlInputFile 控件

HtmlInputFile 控件，用来实现向网络服务器上指明的文件夹上传文件。HtmlInputFile 控件允许对 HTML<input type=file>标签编写程序。使用这个控件，可以从浏览器将二进制或文本文件上传到服务器。

HtmlInputFile 控件实现文件上传，适用于 HTML 3.2 版及版本更高的浏览器。

其使用语法如下：

```
<input
  type=file
  runat="server"
  id="programmaticID"
  accept="MIMEencodings"
  maxlength="maxfilepathlength"
  size="widthoffilepathtextbox"
  postedfile="uploadedfile"
>
```

语法说明如表 4 - 9 所示。

表 4 - 9 HtmlInputFile 控件语法说明

type=file	文件类型
runat="server"	在服务器端执行
id="programmaticID"	ID 序号
accept="MIMEencodings"	编码类型
Maxlength="maxfilepathlength"	文件路径框中最大的字节数
size="widthoffilepathtextbox"	路径框的宽度
postedfile="uploadedfile"	传输方式

下面举例来说明 HtmlInputFile 控件的用法。

使用 VB 语言编写的程序 ch04/test4_14vb.aspx 的源代码如下：

```
<html>
<head>
  <script language="VB" runat="server">
    Sub Button1_Click(sender As Object, e As EventArgs)
      '判断 Text1.Value 的值是否为空
      If Text1.Value = "" Then
        Span1.InnerHtml = "错误: 你必须输入文件名！"
        Return
      End If
    End Sub
  </script>
</head>
```

```

If Not IsNothing(File1.PostedFile) Then
    Try
        File1.PostedFile.SaveAs("c:\temp\"+Text1.Value)
        Span1.InnerHtml = "文件顺利地上传到 Web 服务器的 c:\temp\" & Text1.Value &
            "</b> "
    Catch exc As Exception
        Span1.InnerHtml = "文件上传失败<b>c:\temp\" & Text1.Value & "</b>失败原因 :
            <br>" & exc.ToString
    End Try
End If
End Sub
</script>
<title>HtmlInputFile 举例</title>
</head>
<body>
<h3>
    <font face="宋体">HtmlInputFile 举例</font>
</h3>
<form enctype="multipart/form - data" runat="server">
    <!--布置 HtmlInputFile 控件-->
    选择上传文件:
        <input id="File1"
            type=file
            runat="server"
        >

    <p>
        存储时的文件名: <input id="Text1"
            type="text"
            runat="server"
        >

    <p>
    <span id=Span1
        style="font:12pt 宋体;"
        runat="server"
    />

    <p>
    <input type=button
        id="Button1"
        value="上传"

```

```

        OnServerClick="Button1_Click"
        runat="server"
    >
</span>
</form>
</body>
</html>

```

使用 C # 程序语言编写的 ch04/test4_14c.aspx 的源代码如下：

```

<html>
<head>
    <script language="C#" runat="server">
        void Button1_Click(object Source, EventArgs e) {
            if (Text1.Value == "") {
                Span1.InnerHtml = "错误: 你必须输入文件名 ! ";
                return;
            }
            if (File1.PostedFile != null) {
                try {
                    File1.PostedFile.SaveAs("c:\\temp\\"+Text1.Value);
                    Span1.InnerHtml = "文件顺利地上传到 Web 服务器的 c:\\temp\\"+Text1.Value;
                }
                catch (Exception exc) {
                    Span1.InnerHtml = "文件上传失败<b>c:\\temp\\"+Text1.Value+"</b><br>" +
                        exc.ToString();
                }
            }
        }
    </script>
    <title>HtmlInputFile 举例</title>
</head>
<body>
    <h3>
        <font face="宋体">HtmlInputFile 举例
        </font>
    </h3>
    <form enctype="multipart/form - data" runat="server">
<!-- 布置 HtmlInputFile 控件-->
        选择上传文件: <input id="File1"
                        type=file

```

```

        runat="server"
    >

    <p>
    存储时的文件名: <input id="Text1"
        type="text"
        runat="server"
    >

    <p>
    <span id=Span1
        style="font:12pt 宋体;"
        runat="server"
    />

    <p>
    <input type=button
        id="Button1"
        value="上传"
        OnServerClick="Button1_Click"
        runat="server"
    >

    </span>
</form>
</body>
</html>

```

使用 JavaScript 程序语言编写的程序 ch04/test4_14js.aspx 的源代码如下：

```

<html>
<head>
    <script language="JScript" runat="server">
        function Button1_Click(Source:Object, e:EventArgs) : void {
            if (Text1.Value == "") {
                Span1.InnerHtml = "错误: 你必须输入文件名！";
                return;
            }
            if (File1.PostedFile != null) {
                try {
                    File1.PostedFile.SaveAs("c:\\temp\\"+Text1.Value);
                    Span1.InnerHtml = "文件顺利地被上传到 Web 服务器的 <b>c:\\temp\\"+Text1.
                        Value;
                }
                catch (exc:Exception) {

```

```

        Span1.InnerHtml = "文件上传失败<b>c:\\temp\\"+Text1.Value+"</b><br>" +
            exc.ToString();
    }
}
}
</script>
<title>HtmlInputFile 举例</title>
</head>
<body>
<h3>
    <font face="宋体">HtmlInputFile 举例
</font>
</h3>
    <form enctype="multipart/form - data" runat="server">
<!--布置 HtmlInputFile 控件-->
        选择上传文件: <input id="File1"
                                type=file
                                runat="server"
                                >

        <p>
            存储时的文件名: <input id="Text1"
                                    type="text"
                                    runat="server"
                                    >

        <p>
            <span id=Span1
                style="font:12pt 宋体;"
                runat="server"
            />

        <p>
            <input type=button
                id="Button1"
                value=" 上传"
                OnServerClick="Button1_Click"
                runat="server"
            >

        </span>
    </form></body></html>

```

运行结果如图 4 - 20 所示。

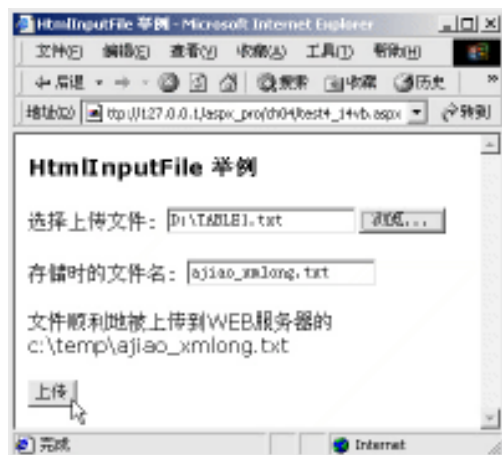


图 4 - 20 HtmlInputFile 控件举例运行结果

在 C:\temp 目录中会看到刚才上传的文件，如图 4 - 21 所示。

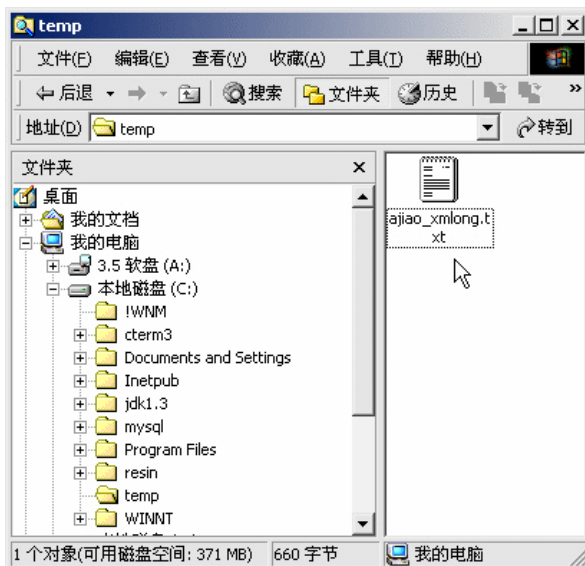


图 4 - 21 HtmlInputFile 控件举例运行结果查看

注意：上述的例子中，表单的 `Enctype`（编码类型）必须设置成 `enctype="multipart/form - data"`。

4.10 HtmlInputHidden 控件

HtmlInputHidden 控件允许用户参考 HTML 的 `<input type=hidden>` 标签编写程序。使用 HtmlInputHidden、HtmlInputButton 和 HtmlInputText 控件，可对特定的用户创建、储备和检查隐含的信息。HtmlInputHidden 控件使用的语法如下：

```

<input
  type="hidden"
  runat="server"
  id="programmaticID"
  value="contentsofhiddenfield"
>

```

语法说明如表 4 - 10 所示。

表 4 - 10 HtmlInputHidden 控件语法说明

type="hidden"	设定为隐藏类型
runat="server"	在服务器端执行
id="programmaticID"	ID 序号
value="contentsofhiddenfield"	隐藏的内容的值

注意：HtmlInputHidden 控件不需要封闭标签。

在此举例说明 HtmlInputHidden 控件的具体使用。下面的例子都是利用隐藏控件存储、显示上一次用户提交的内容，而本次提交的数据先保存，等到下一次才显示。

使用 VB 语言编写的程序 ch04/test4_15vb.aspx 的源代码如下：

```

<html>
<head>
  <script language="VB" runat="server">
    Sub Page_Load(sender As Object, e As EventArgs)
      '判断是否为用户提交数据
      If IsPostBack Then
        Span1.InnerHtml="隐藏控件的值: <b>" & HiddenValue.Value & "</b>"
      End If
    End Sub
  '按下确定按钮的调用的事件或者程序
  Sub SubmitBtn_Click(sender As Object, e As EventArgs)
    HiddenValue.Value = StringContents.Value
  End Sub
  </script>
<title>HtmlInputHidden 举例</title>
</head>
<body>
<h3>
  <font face="宋体">HtmlInputHidden 举例
  </font>
</h3>

```

```
<form runat=server>
<!--布置隐藏控件-->
    <input id="HiddenValue"
        type=hidden
        value="Initial Value"
        runat=server
    >

    请输入:
<!--布置文本框-->
    <input id="StringContents"
        type=text
        size=20
        runat=server
    >
    <p>
<!--布置确定按钮-->
    <input type=submit
        value="确定"
        OnServerClick="SubmitBtn_Click"
        runat=server
    >
    <p>
<!--显示隐藏内容-->
<span id=Span1
    runat=server>这里将显示上次您输入的内容。
</span>
</form>
</body>
</html>
```

使用 C# 语言编写的程序 ch04/test4_15c.aspx 的源代码如下：

```
<html>
<head>
    <script language="C#" runat="server">
        void Page_Load(object Source, EventArgs e) {
//判断是否为提交数据
            if (Page.IsPostBack) {
                Span1.InnerHtml="隐藏控件的值: <b>" + HiddenValue.Value + "</b>";
            }
        }
    }
</script>
</head>
<body>
    <form runat=server>
        <input id="HiddenValue" type="hidden" value="Initial Value" runat="server" />
        请输入:
        <input id="StringContents" type="text" size="20" runat="server" />
        <p>
            <input type="submit" value="确定" OnServerClick="SubmitBtn_Click" runat="server" />
        <p>
        <span id="Span1" runat="server">这里将显示上次您输入的内容。
    </form>
</body>
</html>
```

```
//按下按钮所调用的事件或者程序
void SubmitBtn_Click(object Source, EventArgs e) {
    HiddenValue.Value=StringContents.Value;
}
</script>
<title>HtmlInputHidden 举例</title>
</head>
<body>
<h3>
    <font face="宋体">HtmlInputHidden 举例
    </font>
</h3>
    <form runat=server>
<!--布置隐藏控件-->
        <input id="HiddenValue"
            type=hidden
            value="Initial Value"
            runat=server
        >

        请输入:
<!--布置文本框-->
        <input id="StringContents"
            type=text
            size=20
            runat=server
        >
        <p>
<!--布置确定按钮-->
        <input type=submit
            value="确定"
            OnServerClick="SubmitBtn_Click"
            runat=server
        >
        <p>
<!--显示隐藏控件的内容-->
        <span id=Span1 runat=server>
            这里将显示上次您输入的内容。
        </span>
```

```
</form>
```

```
</body>
```

```
</html>
```

使用 JavaScript 语言编写的程序 ch04/test4_15js.aspx 的源代码如下：

```
<html>
```

```
<head>
```

```
<script language="JScript" runat="server">
```

```
function Page_Load(Source:Object, e:EventArgs) : void {
```

```
//判断是否为提交数据操作
```

```
if (Page.IsPostBack) {
```

```
Span1.InnerHtml="隐藏控件值: <b>" + HiddenValue.Value + "</b>";
```

```
}
```

```
}
```

```
//暗线按钮所调用的事件或者程序
```

```
function SubmitBtn_Click(Source:Object, e:EventArgs) : void {
```

```
HiddenValue.Value=StringContents.Value; }
```

```
</script>
```

```
<title>HtmlInputHidden 举例
```

```
</title>
```

```
</head>
```

```
<body>
```

```
<h3>
```

```
<font face="宋体">
```

```
HtmlInputHidden 举例
```

```
</font></h3>
```

```
<form runat=server>
```

```
<!--布置隐藏控件-->
```

```
<input id="HiddenValue"
```

```
type=hidden
```

```
value="Initial Value"
```

```
runat=server
```

```
>
```

请输入:

```
<!--布置文本框-->
```

```
<input id="StringContents"
```

```
type=text
```

```
size=20
```

```
runat=server
```

```

        >
        <p>
        <!--布置确定按钮-->
        <input type=submit
            value="确定"
            OnServerClick="SubmitBtn_Click"
            runat=server
        >
        <p>
        <!--显示隐藏控件的内容-->
        <span id=Span1
            runat=server
        >
        这里将显示上次您输入的内容。
        </span>    </form>
    </body></html>

```

运行结果如图 4 - 22 所示。

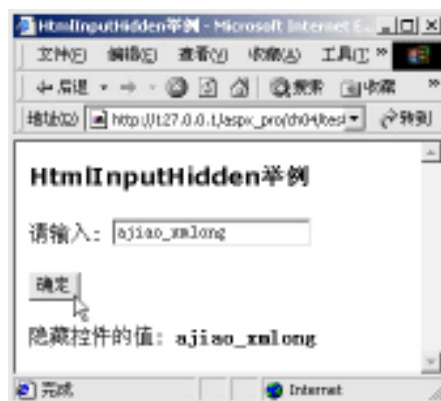


图 4 - 22 HtmlInputHidden 控件举例运行结果

4.11 HtmlInputImage 控件

HtmlInputImage 控件允许用户参考 HTML<input type=image>标签编写程序。可以使用 HtmlInputImage 控件和 HtmlInputText、HtmlTextArea 等其它的控件一起，来组成提供用户输入信息的表单。因为此控件是在服务器上运行的<input type=image>标签，所以它提供像 HTML 中按钮一样的功能，可以由用户来定制其样式。HtmlInputImage 控件为低版本的浏览器提供比 HtmlButton 控件更大的支持。其使用语法如下：

```

<input
    type=image
    runat="server"
    id="programmaticID"
    src="imagepath"
    align="imagealignment"
    alt="alttext"
    OnServerClick="onserverclickhandler"
    width="borderwidthinpixels"
>

```

语法说明如表 4 - 11 所示。

表 4 - 11 HtmlInputImage 控件语法说明

type=image	image 类型
runat="server"	在服务器端执行
id="programmaticID"	ID 序号
src="imagepath"	图片的网络地址
align="imagealignment"	对齐方式
alt="alttext"	鼠标移动到图片上面时所显示的文字
OnServerClick="onserverclickhandler"	按下按钮所调用的事件或者程序
width="borderwidthinpixels"	图片的边框宽度

本节的例子中，布置了两个图片按钮，在按钮 2 中定制了响应鼠标行为的效果。使用 VB 程序语言编写的程序 ch04/test4_16vb.aspx 的源代码如下：

```
<html>
<head>
  <script language="VB" runat="server">
按下按钮 1 所调用的程序 Button1_Click
    Sub Button1_Click(sender As Object, e As ImageClickEventArgs)
      Span1.InnerHtml="你刚才点击了按钮 1"
    End Sub
按下按钮 2 所调用的程序 Button2_Click
    Sub Button2_Click(sender As Object, e As ImageClickEventArgs)
      Span1.InnerHtml="你刚才点击了按钮 2"
    End Sub
  </script>
<title>HtmlInputImage 举例
</title>
</head>
<body>
  <h3>
    <font face="宋体">HtmlInputImage 举例
    </font>
  </h3>
  <form runat=server>
<!--布置图片控件按钮 1-->
  <input type=image
    id="InputImage1"
    src="../images/1.gif"
    onserverclick="Button1_Click"
```

```

        runat="server"
        width="50"
        height="25"
        name="image">

<p>
<!--布置图片控件按钮 2，并且增加鼠标的行为-->
    <input type=image
        id="InputImage2"
        src="../images/2.gif"
        onmouseover="this.src='../images/2.gif';"
        onmouseout="this.src='../images/3.gif';"
        OnServerClick="Button2_Click"
        runat="server"
        width="50"
        height="25">
    可以翻转图片的按钮
<p>
<!--显示程序运行提示-->
    <span id=Span1
        runat=server
    />

</form>
</body>
</html>

```

使用 C # 语言编写的程序 ch04/test4_16c.aspx 的源代码如下：

```

<html>
<head>
    <script language="C#" runat="server">
//按下图片按钮 1 所调用的程序 Button1_Click
        void Button1_Click(object Source, ImageClickEventArgs e) {
            Span1.InnerHtml="你刚才点击了按钮 1";
        }
//按下图片按钮 2 所调用的程序 Button2_Click
        void Button2_Click(object Source, ImageClickEventArgs e) {
            Span1.InnerHtml="你刚才点击了按钮 2";
        }
    </script>
<title>HtmlInputImage 举例
</title>

```

```
</head>
<body>
    <h3>
        <font face="宋体">HtmlInputImage 举例
        </font>
    </h3>
    <form runat=server>
<!--布置图片控件按钮 1-->
        <input type=image
            id="InputImage1"
            src="../images/1.gif"
            onserverclick="Button1_Click"
            runat="server"
            width="50"
            height="25"
            name="image">

        <p>
<!--布置图片控件按钮 2-->
            <input type=image
                id="InputImage2"
                src="../images/2.gif"
                onmouseover="this.src='../images/2.gif';"
                onmouseout="this.src='../images/3.gif';"
                OnServerClick="Button2_Click"
                runat="server"
                width="50"
                height="25"
            >

            可以翻转图片的按钮

        <p>
<!--显示程序运行结果提示-->
            <span id=Span1
                runat=server
            />

        </form>
    </body>
</html>
```

使用 JavaScript 语言编写的程序 ch04/test4_16js.aspx 的源代码如下：

```
<html>
<head>
    <script language="JScript" runat="server">
//按下图片按钮 1 所调用的程序 Button1_Click
        function Button1_Click(Source:Object, e:ImageClickEventArgs) : void {
            Span1.InnerHtml="你刚才点击了按钮 1";
        }
//按下图片按钮 2 所调用的程序 Button2_Click
        function Button2_Click(Source:Object, e:ImageClickEventArgs) : void {
            Span1.InnerHtml="你刚才点击了按钮 2";
        }
    </script>
<title>
    HtmlInputImage 举例
</title>
</head>
<body>
    <h3>
        <font face="宋体">
            HtmlInputImage 举例
        </font>
    </h3>
    <form runat=server>
<!-- 布置图片控件按钮 1-->
        <input type=image
            id="InputImage1"
            src="../images/1.gif"
            onserverclick="Button1_Click"
            runat="server"
            width="50"
            height="25"
            name="image"
        >
    <p>
<!-- 布置图片控件按钮 2-->
        <input type=image
            id="InputImage2"
            src="../images/2.gif"
            onmouseover="this.src='../images/2.gif';"
```

```

onmouseout="this.src='../images/3.gif';"
OnServerClick="Button2_Click"
runat="server"
width="50"
height="25"
>

```

可以翻转图片的按钮

<p>

<!--显示程序运行结果提示-->

```

<span id=Span1
runat=server
/>

```

</form></body></html>

当单击图片按钮 1 时,效果如图 4-23 所示。

当单击图片按钮 2 时,效果如图 4-24 所示。



图 4-23 HtmlInputImage 控件举例运行结果

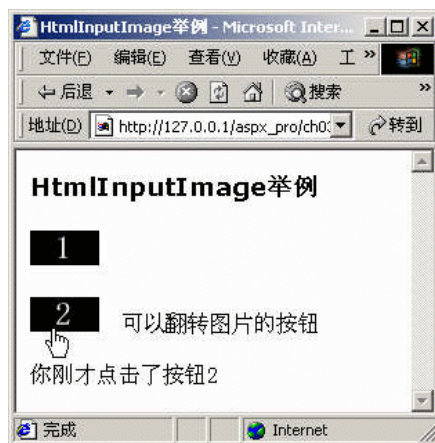


图 4-24 HtmlInputImage 控件举例运行结果

4.12 HtmlInputRadioButton 控件

HtmlInputRadioButton 控件允许用户参考 HTML<input type=radio>标签编写程序。客户选择时只有一个选项能被选中,不能提供多选。HtmlInputRadioButton 控件的任务是对客户所做的选择做出响应。其使用语法如下:

```

<input
type=radio
runat="server"
id="programmaticID"
checked

```

```
name="radiobuttongroup"
>
```

语法说明如表 4 - 12 所示。

表 4 - 12 HtmlInputRadioButton 控件语法说明

type=radio	单选框类型
runat="server"	在服务器端执行
id="programmaticID"	ID 序号
Checked	被选中
name="radiobuttongroup"	单选框组的名称

注意：此控件不需要封闭标签。

使用 VB 语言编写的程序 ch04/test4_17vb.aspx 的源代码如下：

```
<html>
<head>
    <script language="VB" runat="server">
        按下确定按钮所调用的程序 Button1_Click
        Sub Button1_Click(sender As Object, e As EventArgs)
            If Radio1.Checked = True Then
                Span1.InnerHtml = "选项 1 被选中"
            Else If Radio2.Checked = True Then
                Span1.InnerHtml = "选项 2 被选中"
            Else If Radio3.Checked = true Then
                Span1.InnerHtml = "选项 3 被选中"
            End If
        End Sub
    </script>
    <title>HtmlInputRadioButton 举例
</title>
</head>
<body>
    <h3>
        <font face="宋体"> HtmlInputRadioButton 举例
        </font>
    </h3>
    <form runat=server>
<!--布置单选框-->
<input type="radio"
    id="Radio1"
```

```
        name="Mode"
        runat="server"
    />
    选项 1
    <br>
    <input type="radio"
        id="Radio2"
        name="Mode"
        runat="server"
    />
    选项 2
    <br>
    <input type="radio"
        id="Radio3"
        name="Mode"
        runat="server"
    />
    选项 3
    <p>
        <span id=Span1 runat=server />
    <p>
    <!--布置确定按钮-->
    <input type=button
        id="Button1"
        value="确定"
        OnServerClick="Button1_Click"
        runat=server
    >
        </span>
    </form>
</body>
</html>
```

使用 C # 语言编写的程序 ch04/test4_17c.aspx 的源代码如下：

```
<html>
<head>
    <script language="C#" runat="server">
//按下确定按钮所调用的程序 Button1_Click
        void Button1_Click(object Source, EventArgs e) {
            if (Radio1.Checked == true)
```

```
        Span1.InnerHtml = "选项 1 被选中";
    else if (Radio2.Checked == true)
        Span1.InnerHtml = "选项 2 被选中";
    else if (Radio3.Checked == true)
        Span1.InnerHtml = "选项 3 被选中";
    }
</script>
<title>HtmlInputRadioButton 举例
</title>
</head>
<body>
    <h3><font face="宋体"> HtmlInputRadioButton 举例
        </font>
    </h3>
    <form runat=server>
<!--布置单选框-->
<input type="radio"
    id="Radio1"
    name="Mode"
    runat="server"
    />
    选项 1
    <br>
<input type="radio"
    id="Radio2"
    name="Mode"
    runat="server"
    />
    选项 2
    <br>
<input type="radio"
    id="Radio3"
    name="Mode"
    runat="server"
    />
    选项 3
    <p><span id=Span1 runat=server />
        <p>
<!--布置确定按钮-->
```

```
<input type=button
      id="Button1"
      value="确定"
      OnServerClick="Button1_Click"
      runat=server>
</form>
</body>
</html>
```

使用 JavaScript 语言编写的 ch04/test4_17js.aspx 的源代码如下：

```
<html>
<head>
  <script language="JScript" runat="server">
<!--按下确定按钮所调用的程序-->
    function Button1_Click(Source:Object, e:EventArgs) : void {
        if (Radio1.Checked == true)
            Span1.InnerHtml = "选项 1 被选中";
        else if (Radio2.Checked == true)
            Span1.InnerHtml = "选项 2 被选中";
        else if (Radio3.Checked == true)
            Span1.InnerHtml = "选项 3 被选中";
    }
  </script>
  <title>HtmlInputRadioButton 举例
</title>
</head>
<body>
  <h3>
    <font face="宋体"> HtmlInputRadioButton 举例
    </font>
  </h3>
  <form runat=server>
<!--布置单选框-->
    <input type="radio"
          id="Radio1"
          name="Mode"
          runat="server"
        />
    选项 1
    <br>
```

```

<input type="radio"
      id="Radio2"
      name="Mode"
      runat="server"/>

选项 2<br>
<input type="radio"
      id="Radio3"
      name="Mode"
      runat="server"/>

选项 3
<p> <span id=Span1 runat=server /> <p>
<!--布置确定按钮-->
<input type=button
      id="Button1"
      value="确定"
      OnServerClick="Button1_Click"
      runat=server>

</form>
</body>
</html>

```

程序运行结果如图 4 - 25 所示。



图 4 - 25 HtmlInputRadioButton 控件举例运行结果

4.13 HtmlInputText(Password、Text) 控件

HtmlInputText 控件能够和 HtmlInputButton、HtmlInputImage 或 HtmlButton 控件一起，在服务器端处理用户输入的数据。可以给 HtmlInputText 控件赋初始值、设定文本框的最大字节数及其类型。HtmlInputText 控件的作用就是通过文本框，使客户输入的数据传送到服务器上，同时验证客户的输入。

```

<input
  type=text | password
  runat="server"
  id="programmaticID"
  maxlength="max#ofcharacters"
  size="widthoftextbox"
  value="defaulttextboxcontents"
>

```

语法说明如表 4 - 13 所示。

表 4 - 13 HtmlInputText 控件语法说明

type=text password	框的类型
runat="server"	在服务器端执行
id="programmaticID"	ID 序号
maxLength="max#ofcharacters"	该文本框的最大字节数
size="widthoftextbox"	文本框的宽度
value="defaulttextboxcontents"	缺省的值

注意：HtmlInputText 控件不需要封闭标签。

使用 VB 语言编写的程序 ch04/test4_18vb.aspx 的源代码如下：

```
<html>
<head>
    <script language="VB" runat="server">
        按下按钮所调用的事件 SubmitBtn_Click
        Sub SubmitBtn_Click(sender As Object, e As EventArgs)
            If Password.Value = "ASP.Net" Then
                Span1.InnerHtml = "您的密码完全正确！"
            Else
                Span1.InnerHtml = "您输入的密码不正确！"
            End If
        End Sub
    </script>
    <title>HtmlInputText 控件填充文本和密码
</title>
</head>
<body>
    <h3>
        <font face="宋体"> HtmlInputText 控件填充文本和密码
    </font>
    </h3>
    <form runat=server>
        <!--布置文本框-->
        请您输入姓名:
        <input id="Name"
            type=text
            size=12
            runat=server
            maxLength="12"
```

```

        >
        <p> 请您输入密码:
<!--布置密码框-->
<input id="Password"
        type=password
        size=12
        runat=server
        maxlength="12"
        >
        (密码就是"ASP.Net")
    <p>
<!--布置确定按钮-->
    <input type=submit
        value="确定"
        OnServerClick="SubmitBtn_Click"
        runat=server
        >

    <p>
<!--显示程序运行的结果-->
    <span id="Span1"
        style="color:red"
        runat=server>

    </span>
</form>
</body>
</html>

```

使用 C # 语言编写的程序 ch04/test4_18c.aspx 的源代码如下：

```

<html>
<head>
    <head>
    <script language="C#" runat="server">
//按下按钮所调用的程序
    void SubmitBtn_Click(object Source, EventArgs e) {
        if (Password.Value == "ASP.Net")
            Span1.InnerHtml="您的密码完全正确！";
        else
            Span1.InnerHtml="您输入的密码不正确！";
    }

```

```
</script>
<title>HtmlInputText 控件填充文本和密码
</title>
</head>
<body>
    <h3>
        <font face="宋体"> HtmlInputText 控件填充文本和密码
        </font>
    </h3>
    <form runat=server>
        请您输入姓名:
        <!--布置文本框-->
        <input id="Name"
            type=text
            size=12
            runat=server
            maxlength="12"
            >
        <p> 请您输入密码:
        <!--布置密码框-->
        <input id="Password"
            type=password
            size=12
            runat=server
            maxlength="12"
            >
        (密码就是"ASP.Net")
        <p>
        <!--布置按钮-->
        <input type=submit
            value="确定"
            OnServerClick="SubmitBtn_Click"
            runat=server
            >
        <p>
        <!--显示程序运行结果提示-->
        <span id="Span1"
            style="color:red"
            runat=server
```

```

        >
    </span>
</form>
</body>
</html>

```

使用 JavaScript 语言编写的程序 ch04/test4_18js.aspx 的源代码如下：

```

<html>
<head>
    <head>
    <script language="JScript" runat="server">
//按下确定按钮所调用的程序 SubmitBtn_Click
        function SubmitBtn_Click(Source:Object, e:EventArgs) : void {
            if (Password.Value == "ASP.Net")
                Span1.InnerHtml="您的密码完全正确！";
            else
                Span1.InnerHtml="您输入的密码不正确！";
        }
    </script>
    <title>HtmlInputText 控件填充文本和密码
    </title>
</head>
<body>
    <h3>
        <font face="宋体"> HtmlInputText 控件填充文本和密码
        </font>
    </h3>
    <form runat=server>
        请您输入姓名:
        <!--布置文本框-->
        <input id="Name"
            type=text
            size=12
            runat=server
            maxlength="12"
            >
        <p> 请您输入密码:
        <!--布置密码框-->
        <input id="Password"
            type=password

```

```

        size=12
        runat=server
        maxlength="12"
    >
    (密码就是"ASP.Net")
    <p>
<!--布置按钮-->
    <input type=submit
        value="确定"
        OnServerClick="SubmitBtn_Click"
        runat=server
    >

    <p>
<!--显示程序运行结果-->
        <span id="Span1"
            style="color:red"
            runat=server>
        </span>
    </form>
</body>
</html>

```



图 4 - 26 HtmlInputText 控件举例运行结果

程序运行结果如图 4 - 26 所示。

4.14 HtmlSelect 控 件

HtmlSelect 控件允许用户参考 HTML<select>标签编写程序。可以添加下拉选单的选项，可以和数据绑定在一起，显示选单中的选项。其语法如下：

```

<select
    runat="server"
    id="programmaticID"
    OnServerChange="onserverchangehandler"
    DataSource="databindingsource"
    DataTextField="fieldtodatabindoptiontext"
    DataValueField="fieldtodatabindoptionvalue"
    Multiple
    Items="collectionofoptionelements"
    SelectedIndex="indexofcurrentlyselecteditem"
    Size="#ofvisibleitems"

```

```

Value="currentitemvalue"
>
<option>value1</option>
<option>value2</option>
</select>

```

语法说明如表 4 - 14 所示。

表 4 - 14 HtmlSelect 控件语法说明

runat="server"	在服务器端执行
id="programmaticID"	ID 序号
OnServerChange="onserverchangehandler"	改变选项所调用的程序
DataSource="databindingsource"	数据源
DataTextField="fieldtodatabindoptiontext"	文字字段
DataValueField="fieldtodatabindoptionvalue"	值字段
Multiple	多样
Items="collectionofoptionelements"	选项
SelectedIndex="indexofcurrentlyselecteditem"	选项索引号码
Size="#ofvisibleitems"	可以看见的选项的长度
Value="currentitemvalue"	当前选定值

在 test4_19vb.aspx、test4_19c.aspx 和 test4_19js.aspx 这三个例子中，可以自行添加下拉选单的选项，然后选定，再按下“应用”按钮查看其效果。

使用 VB 语言编写的程序 ch04/test4_19vb.aspx 的源代码如下：

```

<html>
<head>
    <script language="VB" runat="server">
        '按下应用按钮所调用的程序 Apply_Click
        Sub Apply_Click(sender As Object, e As EventArgs)
            Span1.Style("background - color") = ColorSelect.Value
        End Sub
        '按下添加按钮所调用的程序 AddToList_Click
        Sub AddToList_Click(sender As Object, e As EventArgs)
            ColorSelect.Items.Add(Text1.Value)
        End Sub
    </script>
    <title>HtmlSelect 实例
</title>
</head>
<body>

```

```
<h3>
    <font face="宋体"> HtmlSelect 实例
</font>
</h3>

<form runat=server>
    请选择您喜欢的颜色:
<br>
<!--布置下拉选单-->
    <select id="ColorSelect" runat="server">
        <option>SkyBlue</option>
        <option>LightGreen</option>
        <option>Gainsboro</option>
        <option>LemonChiffon</option>
    </select>
<!--布置应用按钮-->
    <input type="button"
        runat="server"
        Value="应用"
        OnServerClick="Apply_Click"
    >

    <p> 上面没有看见的颜色你可以通过下面来添加:
<br>
<!--布置一个文本框-->
    <input type="text"
        id="Text1"
        runat="server">
<!--布置一个添加按钮-->
    <input type="button"
        runat="server"
        Value="添加到列表"
        OnServerClick="AddToList_Click">

    <p>
<!--显示结果-->
<span id="Span1" runat="server">
    按下应用键，调用你喜欢的颜色。
</span>
</form>
</body>
</html>
```

使用 C# 语言编写的程序 ch04/test4_19c.aspx 的源代码如下：

```
<html>
<head>
    <script language="C#" runat="server">
//按下应用按钮所调用的程序 Apply_Click
        void Apply_Click(object Source, EventArgs e) {
            Span1.Style["background - color"] = ColorSelect.Value;
        }
//按下添加到列表按钮所调用的程序 AddToList_Click
        void AddToList_Click(object Source, EventArgs e) {
            ColorSelect.Items.Add(Text1.Value);
        }
    </script>
    <title>HtmlSelect 实例</title>
</head>
<body>
<h3>
    <font face="宋体"> HtmlSelect 实例
    </font>
</h3>
    <form runat=server>
        请选择您喜欢的颜色:
        <br>
<!--布置一个下拉选单-->
        <select id="ColorSelect"
            runat="server">
            <option>SkyBlue</option>
            <option>LightGreen</option>
            <option>Gainsboro</option>
            <option>LemonChiffon</option>
        </select>
<!--布置一个应用按钮-->
        <input type="button"
            runat="server"
            Value="应用"
            OnServerClick="Apply_Click"
            >
        <p> 上面没有看见的颜色你可以通过下面来添加:
<br>
<!--布置一个文本框-->
```

```
<input type="text"
      id="Text1"
      runat="server"
      >
<!--布置一个添加到列表按钮-->
<input type="button"
      runat="server"
      Value="添加到列表"
      OnServerClick="AddToList_Click"
      >

<p>
<!--显示程序运行结果-->
<span id="Span1" runat="server">
    按下应用键，调用你喜欢的颜色。
</span>
</form>
</body>
</html>
```

使用 JavaScript 语言编写的程序 ch04/test4_19js.aspx 的源代码如下：

```
<html>
<head>
    <script language="JScript" runat="server">
//按下应用按钮所调用的程序 Apply_Click
        function Apply_Click(Source:Object, e:EventArgs) : void {
            Span1.Style["background - color"] = ColorSelect.Value;
        }
//按下添加到列表按钮所调用的程序 AddToList_Click
        function AddToList_Click(Source:Object, e:EventArgs) : void {
            ColorSelect.Items.Add(Text1.Value);
        }
    </script>
    <title>HtmlSelect 实例
</title>
</head>
<body>
    <h3>
        <font face="宋体"> HtmlSelect 实例
    </font>
    </h3>
```

```
<form runat=server>
    请选择您喜欢的颜色:
    <br>
    <!--布置一个下拉选单-->
        <select id="ColorSelect"
            runat="server">
                <option>SkyBlue</option>
                <option>LightGreen</option>
                <option>Gainsboro</option>
                <option>LemonChiffon</option>
            </select>
    <!--布置应用按钮-->
        <input type="button"
            runat="server"
            Value="应用"
            OnServerClick="Apply_Click"
            >

    <p>
    上面没有看见的颜色你可以通过下面来添加:
    <br>
    <!--布置一个文本框-->
        <input type="text"
            id="Text1"
            runat="server"
            >
    <!--布置添加到列表按钮-->
        <input type="button"
            runat="server"
            Value="添加到列表"
            OnServerClick="AddToList_Click"
            >

    <p>
    <!--显示运行结果-->
        <span id="Span1"
            runat="server"
            >

        按下应用键，调用你喜欢的颜色。
        </span>
    </form>
```

```
</body>
```

```
</html>
```

运行结果如图 4 - 27 所示。

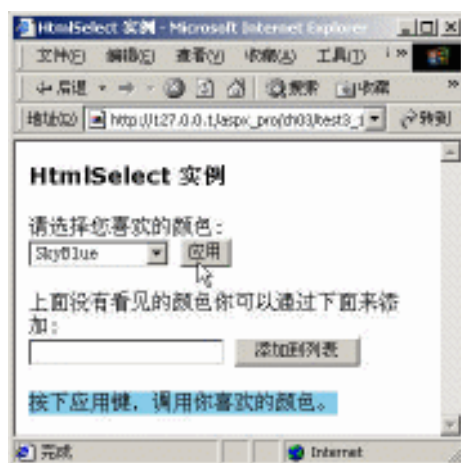


图 4 - 27 HtmlSelect 控件举例运行结果

在 ch04/test4_20vb.aspx、ch04/test4_20c.aspx 和 ch04/test4_20js.aspx 例子中，利用程序代码建立数组，使用数据绑定来绑定数组中的值当作下拉选单的选项，以供选定，这就是 DataBinding HtmlSelect。

使用 VB 程序语言编写的 ch04/test4_20vb.aspx 的源代码如下：

```
<html>
<head>
    <script language="VB" runat="server">
'当页面下载到客户端浏览器是所执行的程序 Page_Load
    Sub Page_Load(sender As Object, e As EventArgs)
        If Not IsPostBack Then
'建立一个数组
            Dim values as ArrayList= new ArrayList()
            values.Add ("IN")
            values.Add ("KS")
            values.Add ("MD")
            values.Add ("MI")
            values.Add ("OR")
            values.Add ("TN")
'建立数据源
            StateSelect.DataSource = values
'绑定数据
            StateSelect.DataBind
```

```

        End If
    End Sub
    按下提交按钮所调用的程序 SubmitBtn_Click
    Sub SubmitBtn_Click(sender As Object, e As EventArgs)
        Span1.InnerHtml = "您的选择: " & StateSelect.Value
    End Sub
</script>
<title>DataBinding HtmlSelect 举例
</title>
</head>
<body>
    <h3>
        <font face="宋体">DataBinding HtmlSelect
    </font>
    </h3>
    <form runat=server>
        请您选择一个:
        <br>
        <!--布置一个下拉选单-->
        <select id="StateSelect" runat="server" />
        <!--布置一个按钮-->
        <input type="submit"
            value="在下面显示出来"
            OnServerClick="SubmitBtn_Click"
            runat="server"
        >
        <p>
        <!--显示运行结果-->
        <span id="Span1"
            runat="server"
        />
    </form>
</body>
</html>

```

使用 C# 程序语言编写的 ch04/test4_20c.aspx 的源代码如下：

```

<html>
<head>
    <script language="C#" runat="server">
        //当页面下载到客户端浏览器时所执行的程序 Page_Load

```

```
void Page_Load(Object Sender, EventArgs E) {
    if (!IsPostBack) {
        //建立选项数组
        ArrayList values = new ArrayList();
        values.Add ("IN");
        values.Add ("KS");
        values.Add ("MD");
        values.Add ("MI");
        values.Add ("OR");
        values.Add ("TN");
        //建立数据源
        StateSelect.DataSource = values;
        //绑定数据
        StateSelect.DataBind();
    }
}

//按下按钮所调用的程序 SubmitBtn_Click
void SubmitBtn_Click(Object sender, EventArgs e) {
    Span1.InnerHtml = "您的选择: " + StateSelect.Value;
}

</script>
<title>DataBinding HtmlSelect 举例
</title>
</head>
<body>
    <h3>
        <font face="宋体">DataBinding HtmlSelect
        </font>
    </h3>
    <form runat=server>
        请您选择一个:
        <br>
        <!--布置一个下拉选单-->
        <select id="StateSelect" runat="server" />
        <!--布置按钮-->
        <input type="submit"
            value="在下面显示出来"
            OnServerClick="SubmitBtn_Click"
            runat="server">
```

```

        >
        <p>
<!--显示结果-->
        <span id="Span1" runat="server" />
    </form>
</body>
</html>

```

使用 JavaScript 程序语言编写的 ch04/test4_20js.aspx 的源代码如下：

```

<html>
<head>
    <script language="JScript" runat="server">
//当页面 下载到客户端的浏览器时所执行的程序 Page_Load
        function Page_Load(Sender:Object, E:EventArgs) : void {
            if (!IsPostBack) {
                //建立选项数组
                var values:ArrayList = new ArrayList();
                values.Add ("IN");
                values.Add ("KS");
                values.Add ("MD");
                values.Add ("MI");
                values.Add ("OR");
                values.Add ("TN");
                //建立数据源
                StateSelect.DataSource = values;
                //绑定数据
                StateSelect.DataBind();
            }
        }
//按下按钮时所调用的程序 SubmitBtn_Click
        function SubmitBtn_Click(sender:Object, e:EventArgs) : void {
            Span1.InnerHtml = "您的选择: " + StateSelect.Value;
        }
    </script>
    < title>DataBinding HtmlSelect 举例
</title>
</head>
<body>
    <h3>
        <font face="宋体">DataBinding HtmlSelect

```

```

</font>
</h3>
<form runat=server>
    请您选择一个:
    <br>
<!--布置下拉选单-->
    <select id="StateSelect"
        runat="server"
    />
<!--布置按钮-->
    <input type="submit"
        value="在下面显示出来"
        OnServerClick="SubmitBtn_Click"
        runat="server"
    >
    <p>
<!--显示程序运行结果-->
    <span id="Span1"
        runat="server"
    />
</form>
</body>
</html>

```



图 4 - 28 HtmlSelect 控件举例运行结果

程序执行结果如图 4 - 28 所示。

4.15 HtmlTable、HtmlTableCell 和 HtmlTableRow 控件

HtmlTable 控件可以参考 HTML <table> 标签编写程序。HtmlTable 控件给表格控件绑定数据，把数据行和数据单元灵活地加到使用 HtmlTableRowCollection 或 HtmlTableCellCollection 提供的方法而形成的数据表格上。使用 HtmlTable 控件可灵活地产生数据表格的行和单元。

HtmlTable 控件的语法如下：

```

<table
    runat="server"
    id="programmaticID"
    align=left | center | right
    bgcolor="bgcolor"
    border="borderwidthinpixels"

```

```

bordercolor="bordercolor"
cellpadding="spacingwithincellsinpixels"
cellspacing="spacingbetweenencellsinpixels"
height="tableheight"
rows="collectionofrows"
width="tablewidth"
>
</table>

```

语法说明如表 4 - 15 所示。

表 4 - 15 HtmlTable 控件语法说明

runat="server"	在服务器端执行
id="programmaticID"	ID 序号
align=left center right	对齐方式
bgcolor="bgcolor"	背景颜色
border="borderwidthinpixels"	边框宽度
bordercolor="bordercolor"	边框颜色
Cellpadding="spacingwithincellsinpixels"	单元格填充区宽度
cellspacing="spacingbetweenencellsinpixels"	单元格相互距离宽度
height="tableheight"	表格高度
rows="collectionofrows"	表格的行数
width="tablewidth"	表格的宽度

在 test4_21vb.aspx、test4_21c.aspx 和 test4_21js.aspx 程序中，使用 HtmlTable 控件，根据客户的要求，建立相应形式的表格。

使用 VB 程序语言编写的 ch04/test4_21vb.aspx 的源代码如下：

```

<html>
<head>
    <script language="VB" runat="server">
        '页面下载到客户端的浏览器时所执行的程序 Page_Load
        Sub Page_Load(sender As Object, e As EventArgs)
            '声明变量
            Dim numrows As Integer
            Dim numcells As Integer
            Dim i As Integer = 0
            Dim j As Integer = 0
            Dim Row As Integer = 0
            Dim r As HtmlTableRow
            Dim c As HtmlTableCell

```

```
' 根据客户提交的数据创建表格
numrows = CInt(Select1.Value)
numcells = CInt(Select2.Value)
For j = 0 To numrows - 1
    r = new HtmlTableRow()
    If (row Mod 2 <> 0) Then
        r.BgColor = "Gainsboro"
    End If
    row += 1
    For i = 0 To numcells - 1
        c = new HtmlTableCell()
        c.Controls.Add(new LiteralControl("行数" & j & ", 列数 " & i))
        r.Cells.Add(c)
    Next i
    Table1.Rows.Add(r)
Next j
End Sub
</script>
<title>HtmlTable 举例
</title>
</head>
<body>
    <h3>
        <font face="宋体">HtmlTable 举例
    </font>
    </h3>
    <form runat=server>
        <font face="宋体" size=" - 1">
            <p>
<!--布置一个表格控件-->
                <table id="Table1"
                    CellPadding=5
                    CellSpacing=0
                    Border="1"
                    runat="server"
                />
            <p> 选择表格的行数:
            <!--布置一个下拉选单确定表格的行数-->
                <select id="Select1" runat="server">
```

```

        <option Value="1">1</option>
        <option Value="2">2</option>
        <option Value="3">3</option>
        <option Value="4">4</option>
        <option Value="5">5</option>
    </select>
    <br>

```

选择表格的列数:

<!-- 布置一个下拉选单，根据客户的选择确定表格的列数-->

```

    <select id="Select2" runat="server">
        <option Value="1">1</option>
        <option Value="2">2</option>
        <option Value="3">3</option>
        <option Value="4">4</option>
        <option Value="5">5</option>
    </select>

```

<!-- 布置一个按钮-->

```

    <input type="submit"
        value="创建表格"
        runat="server"
    >
    </font>
</form>

```

</body>

</html>

使用 C# 语言编写的程序 ch04/test4_21c.aspx 的源代码如下：

```

<html>
<head>
    <script language="C#" runat="server">
//页面下载到客户端的浏览器时所执行的程序 Page_Load
void Page_Load(Object sender, EventArgs e) {
    //声明变量，并且赋值
    int row = 0;
    //创建表格
    int numRows = int.FromString(Select1.Value);
    int numcells = int.FromString(Select2.Value);
    for (int j=0; j<numrows; j++) {
        HtmlTableRow r = new HtmlTableRow();
        // set bgcolor on alternating rows
    }
}
    </script>

```

```
        if (row%2 == 1)
            r.BgColor="Gainsboro";
        row++;
        for (int i=0; i<numcells; i++) {
            HtmlTableCell c = new HtmlTableCell();
            c.Controls.Add(new LiteralControl("行数" +j.ToString( ) + ", 列数" + i.ToString( )));
            r.Cells.Add(c);
        }
        Table1.Rows.Add(r);
    }
}
</script>
<title>HtmlTable 举例
</title>
</head>
<body>
    <h3>
        <font face="宋体">HtmlTable 举例
        </font>
    </h3>
    <form runat=server>
        <font face="宋体" size=" - 1">
            <p>
                <!--布置一个表格控件-->
                <table id="Table1"
                    CellPadding=5
                    CellSpacing=0
                    Border="1"
                    runat="server"
                />
                <p> 选择表格的行数:
                <!--布置下拉选单，让客户决定表格的行数-->
                <select id="Select1" runat="server">
                    <option Value="1">1</option>
                    <option Value="2">2</option>
                    <option Value="3">3</option>
                    <option Value="4">4</option>
                    <option Value="5">5</option>
                </select>
```

```

        <br>
        选择表格的列数:
        <!--布置下拉菜单，让客户选择表格的列数-->
        <select id="Select2" runat="server">
            <option Value="1">1</option>
            <option Value="2">2</option>
            <option Value="3">3</option>
            <option Value="4">4</option>
            <option Value="5">5</option>
        </select>
        <!--布置按钮-->
        <input type="submit"
            value="创建表格"
            runat="server"
            >
        </font>
    </form>
</body>
</html>

```

使用 JavaScript 语言编写的程序 ch04/test4_21js.aspx 的源代码如下：

```

<html>
<head>
    <script language="JScript" runat="server">
        //当页面下载到客户端时，所执行的程序 Page_Load
        function Page_Load(sender:Object, e:EventArgs) : void {
            //定义变量，并赋值
            var row:int = 0;
            //generate rows and cells
            var numRows:int = int.FromString(Select1.Value);
            var numcells:int = int.FromString(Select2.Value);
            for (var j:int=0; j<numRows; j++) {
                var r:HtmlTableRow = new HtmlTableRow();
                // set bgcolor on alternating rows
                if (row%2 == 1)
                    r.BgColor="Gainsboro";
                row++;
                for (var i:int=0; i<numcells; i++) {
                    var c:HtmlTableCell = new HtmlTableCell( );
                    c.Controls.Add(new LiteralControl("行数 " + j.ToString( ) + ", 列数 " + i.ToString( )));

```

```
        r.Cells.Add(c);
    }
    Table1.Rows.Add(r);
}
}
</script>
<title>HtmlTable 举例
</title>
</head>
<body>
    <h3>
        <font face="宋体">HtmlTable 举例
        </font>
    </h3>
    <form runat=server>
        <font face="宋体"
            size=" - 1">
            <p>
<!--布置一个表格控件-->
                <table id="Table1"
                    CellPadding=5
                    CellSpacing=0
                    Border="1"
                    runat="server"
                />
            <p> 选择表格的行数:
<!--布置一个下拉选单，让客户选择表格的行数-->
                <select id="Select1" runat="server">
                    <option Value="1">1</option>
                    <option Value="2">2</option>
                    <option Value="3">3</option>
                    <option Value="4">4</option>
                    <option Value="5">5</option>
                </select>
                <br>
                选择表格的列数:
<!--布置一个下拉选单，让客户选择表格的列数-->
                <select id="Select2" runat="server">
                    <option Value="1">1</option>
```

```

        <option Value="2">2</option>
        <option Value="3">3</option>
        <option Value="4">4</option>
        <option Value="5">5</option>
    </select>
<!-- 布置一个表格控件-->
    <input type="submit"
        value="创建表格"
        runat="server"
    >
</font>
</form>
</body>
</html>

```

其运行结果如图 4 - 29 所示。

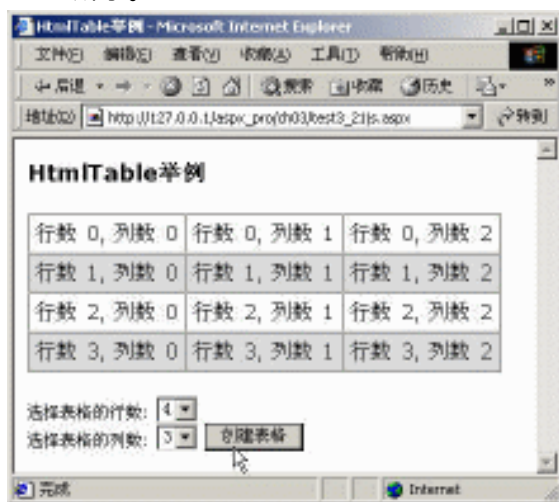


图 4 - 29 HtmlTable 控件举例运行结果

4.16 HtmlTextArea 控件

HtmlTextArea 控件可以参考 HTML <textarea> 标签编写程序。列属性决定控件的宽度，行属性决定控件的高度。如果想从用户处收集反馈资料，就可以使用 HtmlTextArea 控件。

```

<textarea
    runat="server"
    id="programmaticID"
    cols="numberofcolsintextarea"
    name="namepassedtobrowser"

```

```
        rows="numberofrowsintextarea"
        onserverchange="onserverchangehandler"
    >
    textareacontent
</textarea>
```

语法说明如表 4 - 16 所示。

表 4 - 16 HtmlTextArea 控件语法说明

runat="server"	在服务器端执行
id="programmaticID"	ID 序号
cols="numberofcolsintextarea"	文本区域的列数
name="namepassedtobrowser"	名字
rows="numberofrowsintextarea"	文本区域的行数
onserverchange="onserverchangehandler"	
textareacontent	文本区域的内容

在 test4_22vb.aspx、test4_22c.aspx 和 test4_22js.aspx 例子中，使用 HtmlTextArea 控件得到有关 ASP.Net 反馈信息，在用户提交以后显示在浏览器中。

使用 VB 语言编写的程序 ch04/test4_22vb.aspx 的源代码如下：

```
<html>
<head>
    <script language="VB" runat="server">
        按下按钮时调用的程序 SubmitBtn_Click
        Sub SubmitBtn_Click(sender As Object, e As EventArgs)
            Span1.InnerHtml = "您提交了：<br>" & TextArea1.Value
        End Sub
    </script>
</title>HtmlTextArea 举例</title></head>
<body>
    <h3>
        <font face="宋体">HtmlTextArea 举例
    </font>
    </h3>
    <form runat=server>
        <font face="宋体" size=" - 1">
            请写出您最喜欢 ASP.Net 的哪一点?:
            <br>
        </font>
    <!--布置一个 textarea 控件-->
    <textarea id="TextArea1"
```

```

        cols=40
        rows=4
        runat=server
    />
<font face="宋体"
    size=" - 1">
<!--布置一个按钮控件-->
<input type=submit
    value="提交"
    OnServerClick="SubmitBtn_Click"
    runat=server
>
<p>
<!--显示结果-->
    <span id="Span1"
        runat="server"
    />

</span>
</font>
</form>
</span>
</textarea>
</body>
</html>

```

使用 C # 语言编写的程序 ch04/test4_22c.aspx 的源代码如下：

```

<html>
<head>
    <script language="C#" runat="server">
//按下按钮所调用的程序 SubmitBtn_Click
        void SubmitBtn_Click(Object sender, EventArgs e) {
            Span1.InnerHtml = "您提交了: <br>" + TextArea1.Value;
        }
    </script>
    <title>HtmlTextArea 举例
</title>
</head>
<body>
    <h3>
        <font face="宋体">HtmlTextArea 举例

```

```
</font>
</h3>
<form runat=server>
    <font face="宋体" size=" - 1">
        你最喜欢 ASP.Net 的哪一点?:
    <br>
    <!--布置一个 textarea 控件-->
        <textarea id="TextArea1"
            cols=40
            rows=4
            runat=server
        />
    <!--布置一个按钮控件-->
    <input type=submit
        value="提交"
        OnServerClick="SubmitBtn_Click"
        runat=server
    >
    <p>
    <!--显示结果-->
        <span id="Span1"
            runat="server"
        />
    </font>
</form>
</body>
</html>
```

使用 JavaScript 语言编写的程序 ch04/test4_22js.aspx 的源代码如下：

```
<html>
<head>
    <script language="JScript" runat="server">
//按下按钮所调用的程序 SubmitBtn_Click
        function SubmitBtn_Click(sender:Object, e:EventArgs) {
            Span1.InnerHtml = "您提交了: <br>" + TextArea1.Value;
        }
    </script>
    <title>HtmlTextArea 举例
    </title>
</head>
```

```

<body>
    <h3>
        <font face="宋体">HtmlTextArea 举例
    </font>
    </h3>
    <form runat=server>
        <font face="宋体" size=" - 1">
            你最喜欢 ASP.Net 的哪一点?:
        <br>
        <!--布置一个 textarea 控件-->
        <textarea id="TextArea1"
            cols=40
            rows=4
            runat=server
        />
        <!--布置按钮控件-->
        <input type=submit
            value="提交"
            OnServerClick="SubmitBtn_Click"
            runat=server
        >
        <p>
        <!--显示结果-->
        <span id="Span1" runat="server" />
        </font>
    </form>
</body>
</html>

```

运行结果如图 4 - 30 所示。

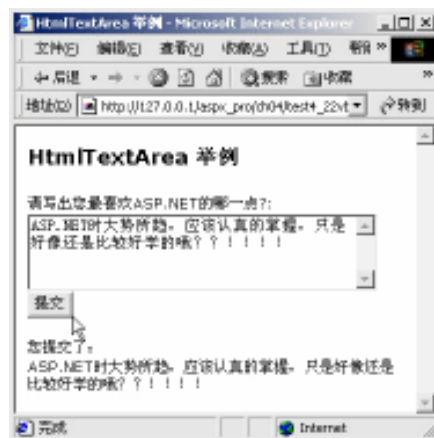


图 4 - 30 HtmlTextArea 控件运行结果

4.17 本章小结

本章着重介绍了 ASP.Net 内置服务器控件 HTML 控件。按照拼音排序，详细地讲解了 20 个 HTML 控件，并且列举了详实的实例，来说明其使用方法。

第 5 章

ASP.Net 的内置服务器控件——Web 控件

主要内容

- ！ Web 控件简介
- ！ Web 控件中各个控件的具体实例

本章将着重介绍 ASP.Net 服务器控件的 Web 控件,通过实例和程序说明,讲解 ASP.Net 提供的 Web 控件的使用方法。并且对于涉及到数据库方面的控件,做了详细的介绍。

5.1 Web 控件简介

所有的 Web 控件都是直接或间接地由 System.Web.UI.WebControls 或 WebControl 类产生。Web 控件的结构层次如图 5 - 1 所示。左边显示的是 HTML 标签之间的关系,中间的控

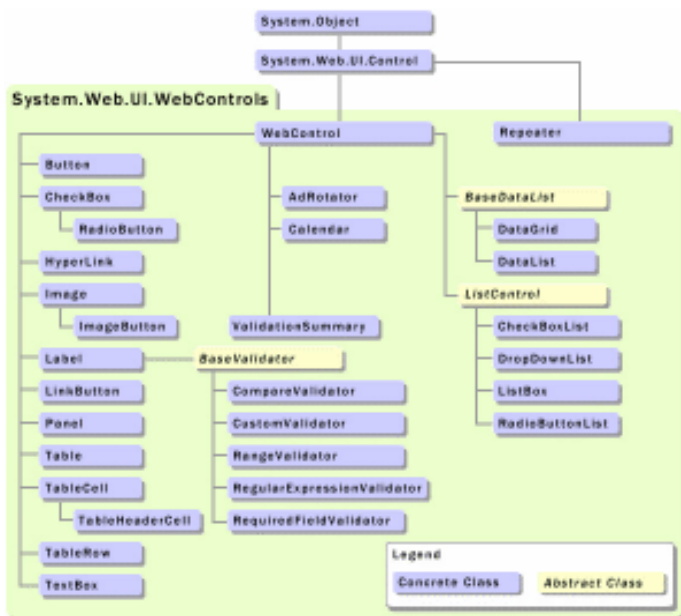


图 5 - 1 Web 控件的结构

件则确认提交的表单，右边的控件可以提供数据绑定支持。此外，在中间显示了功能强大的 AdRotator 广告控件和 Calendar 日历控件。

通过扩展可以使目前的 Web 控件，或者结合目前的 Web 控件，或者制作基于基础 System.Web.UI.WebControls 和 WebControl 类的控件，来开发客户自己的 Web 控件。

Web 控件的组成如表 5 - 1 所示。

表 5 - 1 Web 控件的组成

AdRotator	Button
Calendar	CheckBox
CheckBoxList	CompareValidator
CustomValidator	DataGrid
DataList	DropDownList
HyperLink	Image
ImageButton	Label
LinkButton	ListBox
Panel	RadioButton
RadioButtonList	RangeValidator
RegularExpressionValidator	Repeater
RequiredFieldValidator	Table
TableCell	TableRow
TextBox	ValidationSummary

5.2 AdRotator 控 件

AdRotator Web 控件用于广告轮显，当页面变化时显示不同的广告。其使用语法如下：

```
<asp:AdRotator
    id=value
    runat="SERVER"
    AdvertisementFile="AdvertisementFile"
    propertyname=propertyvalue
    OnAdCreated="OnAdCreatedMethod"
></asp:AdRotator>
```

AdRotator Web 控件的语法说明如表 5 - 2 所示。

表 5 - 2 AdRotator Web 控件语法说明

id=value	ID 序号
runat="SERVER"	在服务器端执行
AdvertisementFile="AdvertisementFile"	引用广告文件
propertyname=propertyvalue	设定属性及其属性值
OnAdCreated="OnAdCreatedMethod"	

广告文件(Advertisement File) 的语法如下：

```
<?xml xmlns="Ad Rotator Schedule" ?>
<Advertisements>
  <Ad>
    <ImageUrl>URL of image to display</ImageUrl>
    <TargetUrl> URL of page to link to </TargetUrl>
    <AlternateText>
      Text to show as ToolTip
    </AlternateText>
    <Keyword>keyword used to filter</Keyword>
    <Impressions>relative weighting of ad</Impressions>
  </Ad>
</Advertisements>
```

在使用 AdRotator Web 控件的过程中，首先按照上面的语法建立广告文件，然后在布置 AdRotator Web 控件的时候，按照 AdRotator Web 控件语法引用广告文件。广告文件语法说明如表 5 - 3 所示。

表 5 - 3 AdRotator Web 控件语法说明

<ImageUrl>URL of image to display</ImageUrl>	广告图片的网络地址
<TargetUrl> URL of page to link to </TargetUrl>	图片所链接的网络地址
<AlternateText> Text </AlternateText>	在图片显示前所显示的文字
<Keyword>keyword used to filter</Keyword>	关键字
<Impressions>relative weighting of ad</Impressions>	加权值

在 test5_01vb.aspx 程序中，只是简单地使用 AdRotator Web 控件来随机显示广告条。使用 VB 程序语言编写的 ch05/test5_01vb.aspx 的源代码如下：

```
<html>
<title>AdRotator 举例</title>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
<body>
<h3><font face="Verdana">AdRotator 举例</font></h3>
  <form runat=server>
    <asp:AdRotator id="ar1"
      AdvertisementFile="Ads.xml"
      BorderWidth="1"
      runat=server />
  </form>
```

```
</body>
```

```
</html>
```

其中最重要的组成部分是广告文件 Ads.xml，源代码如下：

```
<Advertisements>
```

```
<Ad>
```

```
<ImageUrl>../images/banner1.gif</ImageUrl>
```

```
<NavigateUrl>http://www.singnet.com.cn/music/</NavigateUrl>
```

```
<AlternateText>世纪星之影音接触</AlternateText>
```

```
<Keyword>Music</Keyword>
```

```
<Impressions>80</Impressions>
```

```
</Ad>
```

```
<Ad>
```

```
<ImageUrl>../images/banner2.gif</ImageUrl>
```

```
<NavigateUrl>http://www.singnet.com.cn/bible/game/</NavigateUrl>
```

```
<AlternateText>游戏天地</AlternateText>
```

```
<Keyword>Game</Keyword>
```

```
<Impressions>80</Impressions>
```

```
</Ad>
```

```
<Ad>
```

```
<ImageUrl>../images/banner3.gif</ImageUrl>
```

```
<NavigateUrl>http://www.singnet.com.cn/tianya/</NavigateUrl>
```

```
<AlternateText>天涯社区</AlternateText>
```

```
<Keyword>Tianya</Keyword>
```

```
<Impressions>80</Impressions>
```

```
</Ad>
```

```
<Ad>
```

```
<ImageUrl>../images/banner4.gif</ImageUrl>
```

```
<NavigateUrl>http://www.singnet.com.cn/job/</NavigateUrl>
```

```
<AlternateText>世纪星择业</AlternateText>
```

```
<Keyword>Singet</Keyword>
```

```
<Impressions>80</Impressions>
```

```
</Ad>
```

```
</Advertisements>
```

注意：在上面的广告文件(Ads.xml)中，4 个广告出现的机率都是 $80/320 = 1/4$ 。出现的机率是指在网页被浏览无限多次以后，4 幅图片中，每张图片平均起来会有一次出现的机会，但是不一定每 4 次中就有一次机会给特定的一张图片，可能在 4 次里面根本它就没有出现过，或者在 4 次中出现了两次，但是经历多次之后，比方说 10000 次，每幅图片出现的次数在都 2500 次左右。

运行结果如图 5 - 2 所示。

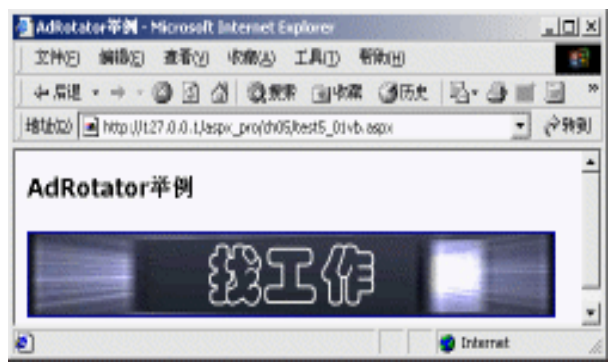


图 5 - 2 AdRotator Web 控件举例运行结果

5.3 Button Web 控件

Button Web 控件提供按钮控件，使得通过表单提交的数据传送给服务器。其使用语法如下：

```
<asp:Button id="MyButton" runat="server"
    Text="label"
    Command="command"
    CommandArgument="commandArgument"
    OnClick="OnClickMethod"
/>
```

语法说明如表 5 - 4 所示。

表 5 - 4 Button Web 控件的语法说明

id="MyButton" runat="server"	ID
Text="label"	按钮表面的文字
Command="command"	命令
CommandArgument="commandArgument"	命令状态
OnClick="OnClickMethod"	按下按钮调用的方法

Button 举例一：使用 Button Web 控件形成的按钮，通过客户端的操作，来显示响应的结果。使用 VB 程序语言编写的 ch05/test5_02vb.aspx 的源代码如下：

```
<html>
<head>
    <script language="VB" runat="server">
        '按下按钮所调用的程序 Button1_Click
        Sub Button1_Click(sender As Object, e As EventArgs)
            Label1.Text="您刚才点击了按钮"
        End Sub
    </script>
<title>Button 举例一</title>
</head>
<body>
<h3><font face="Verdana">Button 举例一</font>
</h3>
<form runat=server>
<!--布置按钮控件-->
    <asp:Button
id=Button1
Text="点击我一下。"
onclick="Button1_Click"
runat="server"
/>
    <!--显示运行结果提示信息-->
<asp:Label
id=Label1
runat=server />
</form>
</body>
</html>
```

使用 C# 程序语言编写的 ch05/test5_02c.aspx 的源代码如下：

```
<html>
<head>
    <script language="C#" runat="server">
        //按下按钮所调用的程序 Button1_Click
        void Button1_Click(object Source, EventArgs e) {
            Label1.Text="您刚才点击了按钮";
        }
    </script>
<title>Button 举例一</title>
</head>
```

```

<body>
<h3><font face="Verdana">Button 举例一</font>
</h3>

<form runat=server>
    <!-- 布置按钮控件-->
    <asp:Button
id=Button1
Text="点击我一下。"
onclick="Button1_Click"
runat="server"
/>

    <!-- 显示运行结果提示信息-->
    <asp:Label
id=Label1
runat=server />
    </form>
</body>
</html>

```

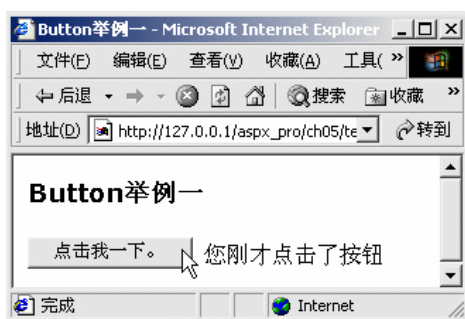


图 5-3 Button Web 控件举例运行结果

运行结果如图 5-3 所示。

Button 举例二：通过客户端提交的信息，交互地显示运行结果。

使用 VB 语言编写的程序 ch05/test5_03vb.aspx 的源代码如下：

```

<html>
<head>
    <script language="VB" runat="server" Debug="true" >
        ' 页面被浏览时所调用的程序 Page_Load
        Sub Page_Load(sender As Object, e As EventArgs)
            if Not IsPostBack Then
                dim values as ArrayList
                values = new ArrayList()
                values.Add(new PositionData("Microsoft", "Msft", "150 股"))
                values.Add(new PositionData("Intel", "Intc", "25 股"))
                values.Add(new PositionData("Dell", "Dell", "115 股"))
                repeater1.DataSource = values
                repeater1.DataBind
            End If
        End Sub
    Sub
        Repeater1_ItemCommand(sender As Object, e As RepeaterCommandEventArgs)
            lblResult.Text = "<u>命令</u>是: <b>" & e.CommandName & "</b>, <u>参考条件</u>"

```

```

是: <b>" & e.CommandArgument & "</b>"
End Sub
class PositionData
    Dim m_name As String
    Dim m_ticker As String
    Dim m_shares As String
    Public Sub New(name As String, ticker As String, shares As String)
        MyBase.New
        m_name = name
        m_ticker = ticker
        m_shares = shares
    End Sub
    ReadOnly Property Name As String
        Get
            Return m_name
        End Get
    End Property
    ReadOnly Property Ticker As String
        Get
            Return m_ticker
        End Get
    End Property
    ReadOnly Property Shares As String
        Get
            Return m_shares
        End Get
    End Property
End Class
</script>
<title>Button 举例二</title></head>
<body>
<h3><font face="Verdana"> Button 举例二</font></h3>
    <form runat=server>
        <font face="Verdana" size="-1">
            <asp:Repeater
id=peater1
onitemcommand="Repeater1_ItemCommand"
runat="server">
                <template name="ItemTemplate">

```

```

        <asp:Button id=btnBuy Text="买股票"
CommandName="买"
CommandArgument='<%# DataBinder.Eval(Container.DataItem, "Ticker") %>'
runat="server"
/>

        <asp:Button id=btnSell Text="卖股票"
CommandName="卖"
CommandArgument='<%# DataBinder.Eval(Container.DataItem, "Shares") %>'
runat="server" />

        <asp:Label
id=lblCompany
Text='<%# DataBinder.Eval(Container.DataItem, "Name") %>'
Font-Bold="true"
runat=server
/>

        <p>
        </template>
    </asp:Repeater>
    <asp:Label
id=lblResult
runat="server"
/>

    </font>
</form>
</body>
</html>

```

使用 C # 程序语言编写的程序 ch05/test5_03c.aspx 的源代码如下：

```

<html>
<head>
    <script language="C#" runat="server" Debug="true">
        void Page_Load(Object Sender, EventArgs e) {
            if (!Page.IsPostBack) {
                ArrayList values = new ArrayList();
                values.Add(new PositionData("Microsoft", "Msft", "150 股"));
                values.Add(new PositionData("Intel", "Intc", "25 股"));
                values.Add(new PositionData("Dell", "Dell", "115 股"));
                repeater1.DataSource = values;
                repeater1.DataBind();
            }
        }
    }

```

```

    }
    void Repeater1_ItemCommand(object sender, RepeaterCommandEventArgs e)
    { lblResult.Text = "<u><u>命令</u>是: <b>" + e.CommandName + "</b>, <u>
      参考条件</u>是: <b>" + e.CommandArgument + "</b>";
    }
    public class PositionData {
        private string name;
        private string ticker;
        private string shares;
        public PositionData(string name, string ticker, string shares) {
            this.name = name;
            this.ticker = ticker;
            this.shares = shares;
        }
        public string Name {
            get {
                return name;
            }
        }
        public string Ticker {
            get {
                return ticker;
            }
        }
        public string Shares {
            get {
                return shares;
            }
        }
    }
}
</script>
<title>Button 举例二</title></head>
<body>
<h3><font face="Verdana"> Button 举例二</font></h3>
    <form runat=server>
        <font face="Verdana" size="-1">
            <asp:Repeater
id=repeater1
onitemcommand="Repeater1_ItemCommand"

```

```

runat="server"
/>

<template name="ItemTemplate">
    <asp:Button

id=btnBuy
Text="买股票"
CommandName="买"
CommandArgument='<%# DataBinder.Eval(Container.DataItem, "Ticker") %>'
runat="server" />

    <asp:Button

id=btnSell
Text="卖股票"
CommandName="卖"
CommandArgument='<%# DataBinder.Eval(Container.DataItem, "Shares") %>'
runat="server" />

    <asp:Label

id=lblCompany
Text='<%# DataBinder.Eval(Container.DataItem, "Name") %>'
Font-Bold="true"
runat=server />

    <p>
    </template>
</asp:Repeater>
<asp:Label

id=lblResult
runat="server" />
    </font>
</form>
</body>
</html>

```



图 5 - 4 Button Web 控件举例运行结果

运行结果如图 5 - 4 所示。

Button 举例三：在提供强大功能的同时，实现丰富多彩的按钮界面。

使用 VB 程序语言编写的程序 ch05/test5_04vb.aspx 的源代码如下：

```

<%@ page language="VB" Debug="true"%>
<html>
<head>
    <script runat="server" >
        按钮 1 按下所调用的程序 Button1_Click
        Sub Button1_Click(sender As Object, e As EventArgs)

```

```

        Label1.Text="您刚才点击按钮 1"
    End Sub
    '按下按钮 2 所调用的程序 Button2_Click
    Sub Button2_Click(sender As Object, e As EventArgs)
        Label1.Text="您刚才点击按钮 2"
    End Sub
</script>
<title>Button 举例三鼠标动作</title></head>
<body>
<h3><font face="Verdana"> Button 举例三鼠标动作</font></h3>
    <p>把您的鼠标移动到按钮上面观察效果：</p>
    <form runat=server>
        <font face="Verdana" size="-1">
<!--布置按钮 1，添加鼠标行为，同时应用 CSS 样式-->
            <asp:Button
id=Button1
runat="server"
                Text="按钮 1"
                Width="100px"
                onmouseover="this.style.backgroundColor='yellow'"
                onmouseout="this.style.backgroundColor='buttonface'"
onclick="Button1_Click" />
应用 CSS 样式(按钮背景)...
            <p>
<!--布置按钮 2，添加鼠标行为-->
            <asp:Button
id=Button2
runat="server"
                Text="按钮 2"
                Width="100px"
                onmouseover="this.style.fontWeight='bold'"
                onmouseout="this.style.fontWeight='normal'"
onclick="Button2_Click" />
应用 CSS 样式(按钮字体)...
            <p>
<!--显示运行结果的提示信息-->
            <asp:Label
id=Label1
runat=server />

```

```

        </font>
    </form>
</body>
</html>

```

使用 C# 程序语言编写的程序 ch05/test5_04c.aspx 的源代码如下：

```

<%@ Page Language="C#" Debug="true" %>
<html>
<head>
    <script runat="server">
//按下按钮 1 时所调用的程序 Button1_Click
        void Button1_Click(object s, EventArgs e) {
            Label1.Text="您刚才点击按钮 1";
        }
//按下按钮 2 时所调用的程序 Button2_Click
        void (object s, EventArgs e) {
            Label1.Text="您刚才点击按钮 1";
        }
    </script>
<title>Button 举例三鼠标动作</title></head>
<body>
<h3><font face="Verdana"> Button 举例三鼠标动作</font></h3>
    <p>把您的鼠标移动到按钮上面观察效果：</p>
    <form runat=server>
        <font face="Verdana" size="-1">
<!--布置按钮 1，添加鼠标行为，同时使用 CSS 样式-->
        <asp:Button
            id=Button1
            runat="server"
                Text="Button1"
                Width="100px"
                onmouseover="this.style.backgroundColor='yellow'"
                onmouseout="this.style.backgroundColor='buttonface'"
                onclick="Button1_Click" /> 应用 CSS 样式( 按钮背景颜色 ) ...
        <p>
<!--布置按钮 2，使用鼠标行为-->
        <asp:Button
            id=Button2
            runat="server"
                Text="Button2"

```

```

Width="100px"
onmouseover="this.style.fontWeight='bold'"
onmouseout="this.style.fontWeight='normal'"
onclick="Button2_Click" /> 应用 CSS 样式(按钮字体)...
</p>
<asp:Label
id=Label1
runat=server />
</font>
</form>
</body>
</html>

```

运行结果如图 5 - 5 所示。

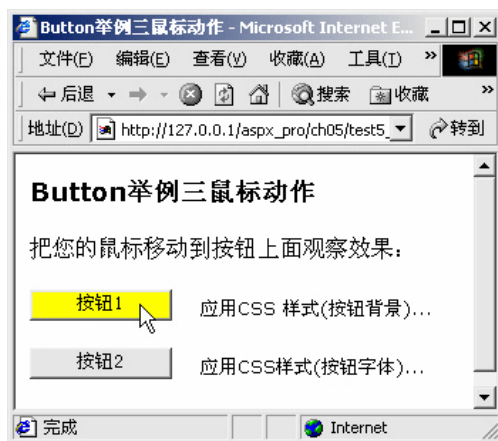


图 5 - 5 Button Web 控件举例运行结果

5.4 Calendar Web 控 件

Calendar Web 控件可方便地显示日历，也可查看特定的一天、一个星期或者一个月。

```

<asp:Calendar id="Calendar1" runat="server"
CellPadding="pixels"
CellSpacing="pixels"
DayNameFormat="FirstLetter|FirstTwoLetters|Full|Short"
FirstDayOfWeek="Default|Monday|Tuesday|Wednesday|
Thursday|Friday|Saturday|Sunday"
NextMonthText="HTML text"
NextPrevFormat="ShortMonth|FullMonth|CustomText"
PrevMonthText="HTML text"

```

```

SelectedDate="date"
SelectionMode="None|Day|DayWeek|DayWeekMonth"
SelectMonthText="HTML text"
SelectWeekText="HTML text"
ShowDayHeader="True|False"
ShowGridLines="True|False"
ShowNextPrevMonth="True|False"
ShowTitle="True|False"
TitleFormat="Month|MonthYear"
TodaysDate="date"
VisibleDate="date"
TodayDayStyle-property="value"
DayHeaderStyle-property="value"
DayStyle-property="value"
NextPrevStyle-property="value"
OtherMonthDayStyle-property="value"
SelectedDayStyle-property="value"
SelectorStyle-property="value"
StyeDayHeaderStyle-property="value"
TitleStyle-property="value"
TodayDayStyle-property="value"
WeekendDayStyle-property="value"
/>

```

Calendar 日历控件举例一：在客户端选择日期，然后在日历下面显示选定的日期。
使用 VB 程序语言编写的程序 ch05/test5_05vb.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true" %>
<html>
<head>
    <script runat="server">
        '当客户选择日期变化时，所调用的程序 Date_Selected
        Sub Date_Selected(sender As Object, e As EventArgs)
            Label1.Text =
                "被选择的日期是: " + Calendar1.SelectedDate.ToShortDateString
        End Sub
    </script>
</head>
<body>
    <h3><font face="Verdana">Calendar ( 日历 ) 举例</font></h3>
    <form runat="server">

```

```

        <!--布置日历控件-->
        <!--当客户变换选择时( onselectionchanged ) , 调用程序 Date_Selected -->
        <asp:Calendar
            id=Calendar1
            onselectionchanged ="Date_Selected"
            runat="server" />
        <p>
        <!--显示运行结果-->
        <asp:Label id=Label1 runat="server" />
    </form>
</body>
</html>

```

使用 C # 程序语言编写的程序 ch05/test5_05c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true" %>
<html>
<head>
    <script runat="server">
        //当客户选择日期变化时所调用的程序 Date_Selected
        void Date_Selected(object s, EventArgs e) {
            Label1.Text =
            "被选择的日期是: " + Calendar1.SelectedDate.ToShortDateString();
        }
    </script>
    <title>Calendar ( 日历 ) 举例</title>
</head>
<body>
    <h3><font face="Verdana">Calendar ( 日历 ) 举例</font></h3>
    <form runat=server>
        <!--布置日历控件-->
        <asp:Calendar
            id=Calendar1
            onselectionchanged="Date_Selected"
            runat="server" />
        <p>
        <!--显示运行结果-->
        <asp:Label
            id=Label1
            runat="server" />
    </form>

```

```
</body>
```

```
</html>
```

运行结果如图 5 - 6 所示。



图 5 - 6 Calendar Web 控件举例运行结果

Calendar 日历控件举例二：客户可以选择以天、星期或者月作为单位。
使用 VB 语言编写的程序 ch05/test5_06vb.aspx 的源代码如下：

```
<%@ Page language="VB" Debug="true" %>
<html>
<head>
<script runat="server">
    '当页面被浏览时，所调用的程序 Page_Load
    Sub Page_Load(sender As Object, e As EventArgs)
        Calendar1.SelectionMode = 1stSelMode.SelectedIndex
        If Calendar1.SelectionMode = CalendarSelectionMode.None Then
            Calendar1.SelectedDates.Clear
        End If
    End Sub

    '客户选择日期改变时，所调用的程序 Date_Selected
    Sub Date_Selected(sender As Object, e As EventArgs)
        Select (Calendar1.SelectedDates.Count)
            Case 0: 'None
                Label1.Text = "当前没有选定的日期："
            Case 1: 'Day
                Label1.Text = "您选择的日期是：" & Calendar1.SelectedDate.ToShortDateString
            Case 7: 'Week
```

```

        Label1.Text = "你选择的是从星期开始的： " &
            Calendar1.SelectedDate.ToShortDateString
    Case Else: 'Month
        Label1.Text = "您选择的日期从月开始的：" &
            Calendar1.SelectedDate.ToShortDateString
    End Select
End Sub
</script>
<title>Date 选择模式</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
<body><h3><font face="Verdana">Date 选择模式</font></h3>
    <p>
        <form runat=server>
<!--布置一个下拉选单-->
        选择一个日期模式: <asp:DropDownList
            id="lstSelMode"
            runat=server
            AutoPostBack=true>
<asp:ListItem Value="None" >无</asp:ListItem>
<asp:ListItem Selected Value="Day" >天</asp:ListItem>
        <asp:ListItem Value="DayWeek" >天、星期</asp:ListItem>
<asp:ListItem Value="DayWeekMonth" >天、星期、月</asp:ListItem>
        </asp:DropDownList>
    <p>
<!--布置一个日历控件，同时设定其属性-->
        <asp:Calendar
            id=Calendar1
            runat="server"

            onselectionchanged="Date_Selected"
            DayNameFormat="FirstLetter"
            Font-Name="Arial" Font-Size="12px"
            Height="180px" Width="200px"
            SelectorStyle-BackColor="gainsboro"
            TodayDayStyle-BackColor="gainsboro"
            DayHeaderStyle-BackColor="gainsboro"
            OtherMonthDayStyle-ForeColor="gray"
            TitleStyle-BackColor="gray"
            TitleStyle-Font-Bold="True"

```

```
TitleStyle-Font-Size="12px"
SelectedDayStyle-BackColor="Navy"
SelectedDayStyle-Font-Bold="True"
/>

<p>
<!--显示其结果-->
    <asp:Label
id=Label1
runat="server" />
```

```
</form></body></html>
```

使用 C # 程序语言编写的程序 ch05/test5_06c.aspx 的源代码如下：

```
<%@ Page language="C#" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script runat="server">
//在页面被调用的同时执行的程序 Page_Load
void Page_Load(Object Sender, EventArgs e) {
    Calendar1.SelectionMode = (CalendarSelectionMode)lstSelMode.SelectedIndex;
    if (Calendar1.SelectionMode == CalendarSelectionMode.None)
        Calendar1.SelectedDates.Clear();
}
//日期被选定时，执行的程序 Date_Selected
void Date_Selected(object s, EventArgs e) {
    switch (Calendar1.SelectedDates.Count) {
        case (0):    //None
            Label1.Text = "当前没有选定的日期：";
            break;
        case (1):    //Day
            Label1.Text = "您选择的日期是：" +
                Calendar1.SelectedDate.ToShortDateString();
            break;
        case (7):    //Week
            Label1.Text = "你选择的是从星期开始的：" +
                Calendar1.SelectedDate.ToShortDateString();
            break;
        default:     //Month
            Label1.Text = "您选择的日期从月开始的：" +
                Calendar1.SelectedDate.ToShortDateString();
```

```

    }
}
</script>
<title>Date 选择模式</title></head>
<body><h3><font face="Verdana">Date 选择模式</font></h3>
<p>
<form runat=server>
    选择一个选择模式 :
<!--布置一个下拉单选-->
    <asp:DropDownList
id="lstSelMode"
runat=server
AutoPostBack=true>
        <asp:ListItem Value="None" >无</asp:ListItem>
        <asp:ListItem Selected Value="Day" >天</asp:ListItem>
        <asp:ListItem Value="DayWeek" >天、星期</asp:ListItem>
        <asp:ListItem Value="DayWeekMonth" >天、星期、月</asp:ListItem>
    </asp:DropDownList>
<p>
<!--布置一个日历控件，同时设定其属性-->
    <asp:Calendar
id=Calendar1
runat="server"
        onselectionchanged="Date_Selected"
        DayNameFormat="FirstLetter"
        Font-Name="Arial" Font-Size="12px"
        Height="180px" Width="200px"
        SelectorStyle-BackColor="gainsboro"
        TodayDayStyle-BackColor="gainsboro"
        DayHeaderStyle-BackColor="gainsboro"
        OtherMonthDayStyle-ForeColor="gray"
        TitleStyle-BackColor="gray"
        TitleStyle-Font-Bold="True"
        TitleStyle-Font-Size="12px"
        SelectedDayStyle-BackColor="Navy"
        SelectedDayStyle-Font-Bold="True"
    />
<p>
<!--显示运行结果-->

```

```
<asp:Label
id=Label1
runat="server" />

</form></body></html>
```

选择日期模式为天，客户只能选择一天，结果如图 5 - 7 所示。



图 5 - 7 选择日期模式为天时的运行结果

选择日期模式为天、星期时，客户不仅可以像图 5 - 7 中一样选择单独的一天，也可以选择一个星期(七天)，其效果如图 5 - 8 所示。



图 5 - 8 选择日期模式为星期时的运行结果

选择日期模式为天、星期、月时，客户不仅可以选择一天、一周，还可选择一个月，其运行结果如图 5-9 所示。



图 5-9 选择日期模式为天、星期、月时的运行结果

Calendar 日历控件举例三：用图片来链接日历。

使用 VB 程序语言编写的程序 ch05/test5_07vb.aspx 的源代码如下：

```
<%@ Page language="VB" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script runat="server">
'当日期被选定时所调用的程序 Date_Selected
Sub Date_Selected(sender As Object, e As EventArgs)
    Select (Calendar1.SelectedDates.Count)
        Case 0: 'None
            Label1.Text = "当前没有选定的日期"
        Case 1: 'Day
            Label1.Text = "选定的日期为：" & Calendar1.SelectedDate.ToShortDateString
        Case 7: 'Week
            Label1.Text = "这个选择是以周为单位的：" &
                Calendar1.SelectedDate.ToShortDateString
        Case Else: 'Month
            Label1.Text = "这个选择是以月为单位的：" &
                Calendar1.SelectedDate.ToShortDateString
    End Select
End Sub
```

```

        End Sub
    </script>
<title>日历图片链接</title></head>
<body> <h3><font face="Verdana">日历图片链接</font></h3>
    <p>
        <form runat=server>
<!-- 布置日历控件，同时使用图片做链接-->
<asp:Calendar
id=Calendar1
runat="server"
    onselectionchanged="Date_Selected"
    DayNameFormat="Short"
    SelectionMode="DayWeekMonth"
    Font-Name="Verdana;Arial" Font-Size="12px"
    Height="180px" Width="230px"
    TodayDayStyle-Font-Bold="True"
    DayHeaderStyle-Font-Bold="True"
    OtherMonthDayStyle-ForeColor="gray"
    TitleStyle-BackColor="#3366ff"
    TitleStyle-ForeColor="white"
    TitleStyle-Font-Bold="True"
    SelectedDayStyle-BackColor="#ffcc66"
    SelectedDayStyle-Font-Bold="True"
    NextMonthText = "<img src='../images/monthright.gif' border=0>"
    PrevMonthText = "<img src='../images/monthleft.gif' border=0>"
    SelectorStyle-BackColor="#99ccff"
    SelectWeekText = "<img src='../images/selweek.gif' border=0 onmouseover=
        this.style.backgroundColor=#ffcc66' onmouseout=this.style.backgroundColor=
        '#99ccff'"
    SelectMonthText = "<img src='../images/selmonth.gif' border=0 onmouseover=
        this.style.backgroundColor=#ffcc66' onmouseout=this.style.backgroundColor=
        '#99ccff'"
    />
    <p>
<!-- 显示运行结果-->
<asp:Label
id=Label1
runat="server" />
    </form>
</body>

```

```
</html>
```

使用 C# 程序语言编写的 ch05/test5_07c.aspx 的源代码如下：

```
<%@ Page language="C#" Debug="true"%>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
```

```
<script language="C#" runat="server">
```

```
//日期被选定时，所调用的程序 Date_Selected
```

```
void Date_Selected(object sender, EventArgs e) {
```

```
    switch (Calendar1.SelectedDates.Count) {
```

```
        case (0):    //None
```

```
            Label1.Text = "当前没有选定的日期";
```

```
            break;
```

```
        case (1):    //Day
```

```
            Label1.Text = "选定的日期为：" + Calendar1.SelectedDate.ToShortDateString();
```

```
            break;
```

```
        case (7):    //Week
```

```
            Label1.Text = "这个选择是以周为单位的：" +
```

```
                Calendar1.SelectedDate.ToShortDateString();
```

```
            break;
```

```
        default:    //Month
```

```
            Label1.Text = "这个选择是以月为单位的：" +
```

```
                Calendar1.SelectedDate.ToShortDateString();
```

```
    }
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<h3><font face="Verdana">Selection Link Graphics</font></h3>
```

```
<p>
```

```
<form runat=server>
```

```
<!--布置日历控件，同时设定其属性-->
```

```
<asp:Calendar
```

```
id=Calendar1
```

```
runat="server"
```

```
onselectionchanged="Date_Selected"
```

```
DayNameFormat="Short"
```

```
SelectionMode="DayWeekMonth"
```

```
Font-Name="Verdana;Arial" Font-Size="12px"
```

```
Height="180px" Width="230px"
```

```

TodayDayStyle-Font-Bold="True"
DayHeaderStyle-Font-Bold="True"
OtherMonthDayStyle-ForeColor="gray"
TitleStyle-BackColor="#3366ff"
TitleStyle-ForeColor="white"
TitleStyle-Font-Bold="True"
SelectedDayStyle-BackColor="#ffcc66"
SelectedDayStyle-Font-Bold="True"
NextMonthText = "<img src='../images/monthright.gif' border=0>"
PrevMonthText = "<img src='../images/monthleft.gif' border=0>"
SelectorStyle-BackColor="#99ccff"
SelectWeekText = "<img src='../images/selweek.gif' border=0 onmouseover=
    this.style.backgroundColor='#ffcc66' nmouseout=this.style.backgroundColor=
    '#99ccff'>"
SelectMonthText = "<img src='../images/selmonth.gif' border=0 onmouseover=
    this.style.backgroundColor='#ffcc66' onmouseout=this.style.backgroundColor=
    '#99ccff'>"
/>
<p>
<asp:Label
id=Label1
runat="server" />
</form>
</body>
</html>

```

运行结果如图 5 - 10 所示，当天的日期是用黑体显示。



图 5 - 10 选择一天时的运行结果

选择整周和选择整月时，运行结果如图 5-11 所示。

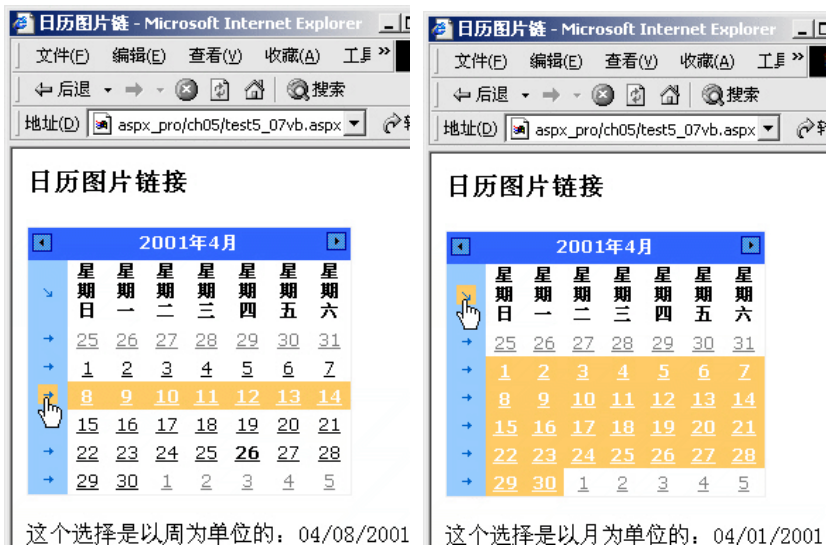


图 5-11 选择整周和选择整月时的运行结果

Calendar 控件日历举例四：使用文字来建立链接。

使用 VB 程序语言编写的程序 ch05/test5_08vb.aspx 的源代码如下：

```
<%@ Page language="VB" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script runat="server">
'当日期被选定时，调用的程序 Date_Selected
Sub Date_Selected(sender As Object, e As EventArgs)
    Select (Calendar1.SelectedDates.Count)
        Case 0: 'None
            Label1.Text = "当前没有选定日期"
        Case 1: 'Day
            Label1.Text = "当前选定的日期为：" &
                Calendar1.SelectedDate.ToShortDateString
        Case 7: 'Week
            Label1.Text = "这次的选择是以星期为单位的：" &
                Calendar1.SelectedDate.ToShortDateString
        Case Else: 'Month
            Label1.Text = "这次的选择是以月为单位的：" &
                Calendar1.SelectedDate.ToShortDateString
    End Select
End Sub
```

```
</script>
<title>用文本链接</title></head>
<body>
<h3><font face="Verdana">用文本链接</font></h3>
  <p>
    <form runat=server>
      <p>
<!--布置日历控件，并且设定其属性-->
      <asp:Calendar
id=Calendar1
runat="server"
      onselectionchanged="Date_Selected"
      DayNameFormat="Short"
      SelectionMode="DayWeekMonth"
      Font-Name="Verdana;Arial" Font-Size="12px"
      Height="180px" Width="230px"
      TodayDayStyle-Font-Bold="True"
      DayHeaderStyle-Font-Bold="True"
      OtherMonthDayStyle-ForeColor="gray"
      TitleStyle-BackColor="#3366ff"
      TitleStyle-ForeColor="white"
      TitleStyle-Font-Bold="True"
      SelectedDayStyle-BackColor="#ffcc66"
      SelectedDayStyle-Font-Bold="True"
      NextPrevFormat="ShortMonth"
      NextPrevStyle-ForeColor="white"
      NextPrevStyle-Font-Size="10px"
      SelectorStyle-BackColor="#99ccff"
      SelectorStyle-ForeColor="navy"
      SelectorStyle-Font-Size="9px"
      SelectWeekText = " 星期"
      SelectMonthText = " 月"
      />
    <p>
      <asp:Label
id=Label1
runat="server" />
    </form>
  </p>
</body>
```

```
</html>
```

使用 C#语言编写的程序 ch05/test5_08c.aspx 的源代码如下：

```
<%@ Page language="C#" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
//日期被选定时，所调用的程序 Date_Selected
void Date_Selected(object s, EventArgs e) {
    switch (Calendar1.SelectedDates.Count) {
        case (0):    //None
            Label1.Text = "当前没有选择日期";
            break;
        case (1):    //Day
            Label1.Text = "您选择的日期为：" +
                Calendar1.SelectedDate.ToShortDateString();
            break;
        case (7):    //Week
            Label1.Text = "这个选择以星期为单位" +
                Calendar1.SelectedDate.ToShortDateString();
            break;
        default:     //Month
            Label1.Text = "这个选择以月为单位：" +
                Calendar1.SelectedDate.ToShortDateString();
    }
}
</script>
<title>使用文本链接</title></head>
<body>
<h3><font face="Verdana">使用文本链接</font></h3>
<p>
<form runat=server>
<p>
<!--布置日历控件，并且设定其属性-->
<asp:Calendar
id=Calendar1
runat="server"
onselectionchanged="Date_Selected"
DayNameFormat="Short"
```

```

SelectionMode="DayWeekMonth"
Font-Name="Verdana;Arial" Font-Size="12px"
Height="180px" Width="230px"
TodayDayStyle-Font-Bold="True"
DayHeaderStyle-Font-Bold="True"
OtherMonthDayStyle-ForeColor="gray"
TitleStyle-BackColor="#3366ff"
TitleStyle-ForeColor="white"
TitleStyle-Font-Bold="True"
SelectedDayStyle-BackColor="#ffcc66"
SelectedDayStyle-Font-Bold="True"
NextPrevFormat="ShortMonth"
NextPrevStyle-ForeColor="white"
NextPrevStyle-Font-Size="10px"
SelectorStyle-BackColor="#99ccff"
SelectorStyle-ForeColor="navy"
SelectorStyle-Font-Size="9px"
SelectWeekText = "星期"
SelectMonthText = "月"
/>
<p>
<!--显示运行的结果-->
<asp:Label
id=Label1
runat="server" />
</form>
</body>
</html>

```



图 5 - 12 使用文字做链接的举例运行结果

运行结果如图 5 - 12 所示。

Calendar 日历控件举例五：在特定的日期中加入提示，如节日或者生日。

使用 VB 语言编写的程序 ch05/test5_09vb.aspx，其源代码如下：

```

<%@ Page language="VB" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    '定义一个数组
    Dim holidays(13,32) as String
    '当页面被浏览时，所调用的程序 Page_Load

```

```

Sub Page_Load(sender As Object, e As EventArgs)
    '给数组赋值
    holidays(1,1) = "元旦"
        holidays(1,17) = "Martin Luther King's Birthday"
        holidays(1,26) = "Australia Day"
        holidays(2,2) = "Groundhog Day"
        holidays(2,14) = "Valentine's Day"
        holidays(2,19) = "MPope's Birthday"
        holidays(2,21) = "President's Day"
        holidays(3,17) = "St. Patrick's Day"
        holidays(4,1) = "愚人节"
        holidays(4,25) = "ANZAC Day"
    holidays(5 , 19) = "ajiao 的生日"
    holidays(6,19) = "xmlong 的生日"
        holidays(7,4) = "Independence Day"
        holidays(9,17) = "SWarren's Birthday"
        holidays(12,25) = "圣诞节"
        holidays(12,26) = "Boxing Day"
End Sub

'显示日历的同时调用的程序 Calendar1_DayRender
Sub Calendar1_DayRender(sender As Object, e As DayRenderEventArgs)
    Dim d as CalendarDay
    Dim c as TableCell
    d = e.Day
    c = e.Cell
    If d.IsOtherMonth Then
        c.Controls.Clear
    Else
        Try
            Dim Hol As String = holidays(d.Date.Month,d.Date.Day)
            If Hol <> "" Then
                c.Controls.Add(new LiteralControl("<br>" + Hol))
            End If
        Catch exc as Exception
            Response.Write (exc.ToString())
        End Try
    End If
End Sub

'客户选择日期时，调用的程序 Date_Selected

```

```
Sub Date_Selected(sender As Object, e As EventArgs)
    Label1.Text
    = "选择的日期是: " + Calendar1.SelectedDate.ToShortDateString
End Sub

</script>
<title>在日历上加入节日提示 : </title></head>
<body>
<h3><font face="Verdana">在日历上加入节日提示 : </font></h3>
    <p><p>
    <form runat=server>
<!-- 布置日历控件，同时设定其属性-->
    <asp:Calendar
id=Calendar1
runat="server"

        ondayrender="Calendar1_DayRender"
        onselectionchanged="Date_Selected"
        ShowGridLines="true"
        BorderWidth="1"
        Font-Name="Verdana"
        Font-Size="9px"
        Width="500px"
        VisibleDate="01/01/2000"
        TitleStyle-BackColor="Gainsboro"
        TitleStyle-Font-Size="12px"
        TitleStyle-Font-Bold="true"
        DayStyle-VerticalAlign="Top"
        DayStyle-Height="50px"
        DayStyle-Width="14%"
        SelectedDate="1/1/0001"
        SelectedDayStyle-BackColor="Navy"
    />

    <p>
<!-- 显示运行结果-->
    <asp:Label
id=Label1
runat="server" />
    </form>
</body>
</html>
```

使用 C # 语言编写的程序 ch05/test5_09c.aspx 的源代码如下：

```
<%@ Page language="C#" Debug="true"%>
<%@ Page language="C#" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
    <script language="C#" runat="server">
        //建议一个数组
String [ ][ ] holidays;
        //浏览网页的同时，执行的程序 Page_Load
void Page_Load(Object Sender, EventArgs e) {
    //把节日放在二维数组里面
        holidays = new String[13][];
        for (int n=0; n<13 ;n++)
            holidays[n] = new String[32];
        holidays[1][1] = "元旦";
        holidays[1][17] = "Martin Luther King's Birthday";
        holidays[1][26] = "Australia Day";
        holidays[2][2] = "Groundhog Day";
        holidays[2][14] = "Valentine's Day";
        holidays[2][19] = "MPope's Birthday";
        holidays[2][21] = "President's Day";
        holidays[3][17] = "St. Patrick's Day";
        holidays[4][1] = "愚人节";
        holidays[4][25] = "ANZAC Day";
        holidays[5][19] = "ajiao 的生日";
        holidays[6][19] = "xmlong 的生日";
        holidays[7][4] = "Independence Day";
        holidays[9][17] = "SWarren's Birthday";
        holidays[12][25] = "圣诞节";
        holidays[12][26] = "Boxing Day";
    }
    //显示日历的程序 Calendar1_DayRender
void Calendar1_DayRender(object sender, DayRenderEventArgs e) {
        CalendarDay d = ((DayRenderEventArgs)e).Day;
        TableCell c = ((DayRenderEventArgs)e).Cell;
        if (d.IsOtherMonth) {
            c.Controls.Clear();
        }
    }
```

```
else {
    try {
        string Hol = holidays[d.Date.Month][d.Date.Day];

        if (Hol != string.Empty)
            c.Controls.Add(new LiteralControl("<br>" + Hol));
    }
    catch (Exception exc) {
        Response.Write (exc.ToString());
    }
}

//当日期被选定时，执行的程序 Date_Selected
void Date_Selected(object s, EventArgs e) {
    Label1.Text = "选择的日期是:" + Calendar1.SelectedDate.ToShortDateString( );
}
</script>
</head>
<body>
<h3><font face="Verdana">在日历上加入节日提示</font></h3>
    <p><p>
<form runat=server>
<!--布置日历控件，同时设定其属性-->
    <asp:Calendar
id=Calendar1
runat="server"

    ondayrender="Calendar1_DayRender"
    onselectionchanged="Date_Selected"
    ShowGridLines="true"
    Font-Name="Verdana;Arial"
    Font-Size="9px"
    Width="500px"
    VisibleDate="01/01/2000"
    TitleStyle-BackColor="Gainsboro"
    TitleStyle-Font-Size="12px"
    TitleStyle-Font-Bold="true"
    DayStyle-VerticalAlign="Top"
    DayStyle-Height="50px"
    DayStyle-Width="14%"
```

```

        SelectedDate="1/1/0001"
        SelectedDayStyle-BackColor="Navy"
    />
    <p>
    <!--布置一个 label 控件，显示程序执行结果-->
    <asp:Label
    id=Label1
    runat="server" />
    </form>
</body>
</html>>

```

运行结果如图 5 - 13 所示。



图 5 - 13 在日历上加入节日提示的举例运行结果

5.5 CheckBox Web 控件

CheckBox Web 控件用在网页中，以让客户选择相关的选项。

注意：也可以使用 CheckBoxList 控件，使其更加容易地和数据捆绑在一起。

CheckBox Web 控件的使用语法如下：

```

<asp:CheckBox
    id="CheckBox1"
    runat="server"
    AutoPostBack="True|False"
    Text="label"

```

```
TextAlign="RightToLeft"
Checked="TrueIfFalse"
OnCheckedChanged="OnCheckedChangedMethod"
/>
```

CheckBox Web 控件的语法说明如表 5 - 5 所示。

表 5 - 5 CheckBox Web 控件的语法说明

语 句	说 明
id="CheckBox1"	ID 序号
runat="server"	在服务器端执行
AutoPostBack="TrueIfFalse"	自动提交数据
Text="label"	文字
TextAlign="RightToLeft"	文字对齐方式
Checked="TrueIfFalse"	是否被选中
OnCheckedChanged="OnCheckedChangedMethod"	选定以后调用的方法

下面是有关 CheckBox Web 控件的使用举例。

使用 VB 语言编写的程序 ch05/test5_10vb.aspx 的源代码如下：

```
<%@ Page language="VB" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
    <script language="VB" runat="server">
        按下提交按钮所调用的程序 SubmitBtn_Click
        Sub SubmitBtn_Click(sender As Object, e As EventArgs)
            If Check1.Checked = true Then
                Label1.Text = "复选框被选择!"
            Else
                Label1.Text = "复选框没有被选择!"
            End If
        End Sub
    </script>
<title>CheckBox 举例</title></head>
<body>
<h3><font face="Verdana">CheckBox 举例</font></h3>
    <form runat=server>
        <!--布置 CheckBox Web 控件, -->
        <asp:CheckBox
id=Check1
Text="复选框"
```

```

runat="server" />
    <!--布置按钮控件，当客户按下提交时，调用 SubmitBtn_Click 程序-->
    <asp:button
text="提交"
OnClick="SubmitBtn_Click"
runat=server/>
    <p>
    <!--布置一个 label 控件，来显示运行结果-->
    <asp:Label
id=Label1
font-name="arial"
font-size="10pt"
runat="server"/>
    </form>
</body>
</html>

```

使用 C # 程序语言编写的程序 ch05/test5_10c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
    <script language="C#" runat="server">
        //按下提交按钮所调用的程序
        void SubmitBtn_Click(Object Sender, EventArgs e) {
            if (Check1.Checked == true) {
                Label1.Text = "复选框被选择!";
            }
            else {
                Label1.Text = "复选框没有被选择!";
            }
        }
    </script>
<title>CheckBox 举例</title>
</head>
<body>
<h3><font face="Verdana">CheckBox 举例</font></h3>
    <form runat=server>
        <!--布置复选框-->
        <asp:CheckBox

```

```

id=Check1
Text="复选框"
runat="server" />
    <!--布置按钮，按下时所调用的程序为 SubmitBtn_Click -->
    <asp:button
text="提交"
OnClick="SubmitBtn_Click"
runat=server/>
    <p>
    <!--布置 label 控件来显示结果-->
    <asp:Label
id=Label1
font-name="arial"
font-size="10pt"
runat="server"/>
    </form>
</body>
</html>

```

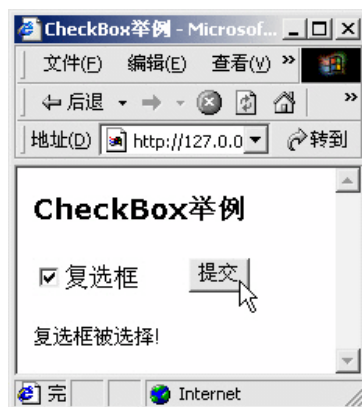


图 5 - 14 CheckBox Web 控件举例运行结果

运行结果如图 5 - 14 所示。

5.6 CheckBoxList Web 控件

使用 CheckBoxList Web 控件，可以制作多个复选框组，提供选项列表供客户多选。并且可以比较容易地使用数据绑定，动态地显示选项列表中的选项。

注意：也可以使用 CheckBox Web 控件，CheckBoxList Web 控件更加容易捆绑数据，CheckBox Web 控件则更加容易控制版面设计。

CheckBoxList Web 控件的使用语法如下：

```

<asp:CheckBoxList
id="CheckBoxList1"
runat="server"
AutoPostBack="True|False"
CellPadding="Pixels"
DataSource='<% databindingexpression %>'
DataTextField="DataSourceField"
DataValueField="DataSourceField"
RepeatColumns="ColumnCount"
RepeatDirection="Vertical|Horizontal"
RepeatLayout="Flow|Table"

```

```

TextAlign="Right|Left"
OnSelectedIndexChanged="OnSelectedIndexChangedMethod"
/>
<asp:ListItem text="label" value="value" selected="True|False" />
</asp:CheckBoxList>

```

其语法介绍如表 5 - 6 所示。

表 5 - 6 CheckBoxList Web 控件的语法说明

id="CheckBoxList1"	ID 序号
runat="server"	在服务器端执行
AutoPostBack="True False"	是否自动提交
CellPadding="Pixels"	单元格
DataSource=<% databindingexpression %>	数据源
DataTextField="DataSourceField"	数据文字
DataValueField="DataSourceField"	数据值
RepeatColumns="ColumnCount"	列数
RepeatDirection="Vertical Horizontal"	显示方向(垂直 水平)
RepeatLayout="Flow Table"	布置方式(顺序 表格)
TextAlign="Right Left"	文字对齐方式
OnSelectedIndexChanged="OnSelectedIndexChangedMethod"	选择之后调用的程序
<asp:ListItem text="label" value="value" selected="True False" />	选项列表

在下面的例子中布置了两个控件 , 一个是 CheckBoxList Web 控件 , 另外一个也是 CheckBox Web 控件。通过程序对比 , 可以看出这两个复选框控件的差别。

使用 VB 程序语言编写的程序 ch05/test5_11vb.aspx 的源代码如下 :

```

<%@ Page language="VB" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    '按下提交按钮所调用的程序
    Sub Button1_Click(sender As Object, e As EventArgs)
        Dim s As String = "你的选项:<br>"
        Dim i As Int32
        For i = 0 to Check1.Items.Count-1
            If Check1.Items(i).Selected Then
                ' 列出选项
                s = s & Check1.Items(i).Text
                s = s & "<br>"
            End If
        Next i
    End Sub

```

```

        End If
    Next
    Label1.Text = s
End Sub

'当布局复选框的选择变化是，所调用的程序 chkLayout_CheckedChanged
Sub chkLayout_CheckedChanged(sender As Object, e As EventArgs)
    If chkLayout.Checked = true Then
        '使用表格显示
        Check1.RepeatLayout = RepeatLayout.Table
    Else
        '顺序显示
        Check1.RepeatLayout = RepeatLayout.Flow
    End If
End Sub

'当选择排列方向变化时：所调用的程序 chkDirection_CheckedChanged
Sub chkDirection_CheckedChanged(sender As Object, e As EventArgs)
    If chkDirection.Checked = true Then
        '水平显示
        Check1.RepeatDirection = RepeatDirection.Horizontal
    Else
        '垂直显示
        Check1.RepeatDirection = RepeatDirection.Vertical
    End If
End Sub

</script>
<title>CheckBoxList 举例</title></head>
<body>
<h3><font face="Verdana">CheckBoxList 举例</font></h3>
    <form runat=server>
        <!--布置 CheckBoxList Web 控件-->
        <asp:CheckBoxList
            id=Check1
            runat="server">
            <asp:ListItem>选项 1</asp:ListItem>
            <asp:ListItem>选项 2</asp:ListItem>
            <asp:ListItem>选项 3</asp:ListItem>
            <asp:ListItem>选项 4</asp:ListItem>
            <asp:ListItem>选项 5</asp:ListItem>
            <asp:ListItem>选项 6</asp:ListItem>

```

```

</asp:CheckBoxList>
<p>
    <!--布置 CheckBox Web 控件，来控制显示方式，同时设定其属性-->
    <asp:CheckBox
id=chkLayout
OnCheckedChanged="chkLayout_CheckedChanged"
Text="以表格显示"
Checked=true
AutoPostBack="true"
runat="server" />
    <br>
    <!--布置 CheckBox Web 控件，来控制显示方向，同时设定其属性-->
    <asp:CheckBox
id=chkDirection
OnCheckedChanged="chkDirection_CheckedChanged"
Text="水平显示"
AutoPostBack="true"
runat="server" />
    <p>
    <!--布置按钮控件-->
    <asp:Button
id=Button1
Text="提交"
onclick="Button1_Click"
runat="server"/>
    <p>
    <!--布置 label 控件，来显示运行结果-->
    <asp:Label
id=Label1
font-name="Verdana"
font-size="8pt"
runat="server"/>
    </form>
</body>
</html>

```

使用 C # 程序语言编写的 ch05/test5_11c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true"%>
<html>
<head>

```

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
    //按下提交按钮所调用的程序 Button1_Click
    void Button1_Click(object Source, EventArgs e) {
        String s = "您的选项:<br>";
        for (int i=0; i < Check1.Items.Count; i++)
        {
            if ( Check1.Items[ i ].Selected )
            {
                //列出选项
                s = s + Check1.Items[i].Text;
                s = s + "<br>";
            }
        }
        Label1.Text = s;
    }

    //布局方式改变时，所调用的程序 chkLayout_CheckedChanged
    void chkLayout_CheckedChanged(Object sender, EventArgs e) {
        if (chkLayout.Checked == true) {
            //在表格中显示
            Check1.RepeatLayout = RepeatLayout.Table;
        }
        else {
            //顺序显示
            Check1.RepeatLayout = RepeatLayout.Flow;
        }
    }

    //显示方向改变时所调用的程序 chkDirection_CheckedChanged
    void chkDirection_CheckedChanged(Object sender, EventArgs e) {
        if (chkDirection.Checked == true) {
            //水平显示
            Check1.RepeatDirection = RepeatDirection.Horizontal;
        }
        else {
            //垂直显示
            Check1.RepeatDirection = RepeatDirection.Vertical;
        }
    }
}
</script>
```

```
<title>CheckBoxList 举例</title></head>
<body>
<h3><font face="Verdana">CheckBoxList 举例</font></h3>
    <form runat=server>
<!--布置 CheckBoxList Web 控件-->
        <asp:CheckBoxList
id=Check1
runat="server">
            <asp:ListItem>选项 1</asp:ListItem>
            <asp:ListItem>选项 2</asp:ListItem>
            <asp:ListItem>选项 3</asp:ListItem>
            <asp:ListItem>选项 4</asp:ListItem>
            <asp:ListItem>选项 5</asp:ListItem>
            <asp:ListItem>选项 6</asp:ListItem>
        </asp:CheckBoxList>
    <p>
<!--布置 CheckBox Web 控件，来控制显示方式-->
        <asp:CheckBox
id=chkLayout
OnCheckedChanged="chkLayout_CheckedChanged"
Text="在表格中显示"
Checked=true
AutoPostBack="true"
runat="server" />
        <br>
<!--布置 CheckBox Web 控件，来控制显示方式-->
        <asp:CheckBox
id=chkDirection
OnCheckedChanged="chkDirection_CheckedChanged"
Text="水平显示"
AutoPostBack="true"
runat="server" />
    <p>
<!--布置按钮-->
        <asp:Button
id=Button1
Text="提交"
onclick="Button1_Click"
runat="server"/>
```

```

<p>
    <!--布置 label 控件，来显示结果-->
    <asp:Label
id=Label1
font-name="Verdana"
font-size="8pt"
runat="server"/>
</form>
</body>
</html>

```

程序运行结果如图 5 - 15 和 5 - 16 所示。

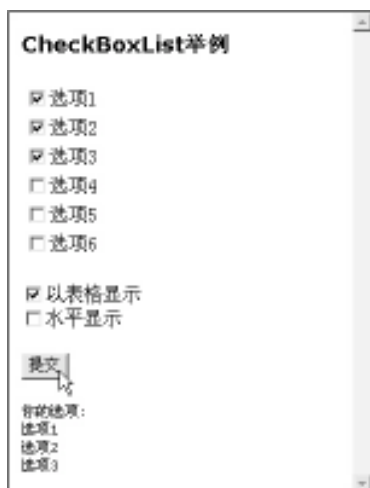


图 5 - 15 CheckBoxList Web 控件举例运行结果

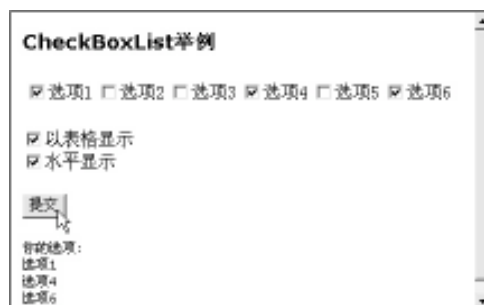


图 5 - 16 CheckBoxList Web 控件举例运行结果

5.7 CompareValidator Web 控件

CompareValidator Web 控件，提供在客户端或者服务器端对有关内容验证的功能。下面来看例子。

使用 VB 语言编写的程序 ch05/test5_12vb.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true" clienttarget="downlevel"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    '按下提交按钮所调用的程序 Button1_OnSubmit
    Sub Button1_OnSubmit(sender As Object, e As EventArgs)
        If Page.IsValid Then

```

```

        lblOutput.Text = "结果：正确!"
    Else
        lblOutput.Text = "结果：不正确!"
    End If
End Sub

'选择规则变化时，所调用的程序 lstOperator_SelectedIndexChanged
Sub lstOperator_SelectedIndexChanged(sender As Object, e As EventArgs)
    comp1.Operator = lstOperator.SelectedIndex
    comp1.Validate
End Sub
</script>
<title>CompareValidator 举例</title></head>
<body>
<h3><font face="Verdana">CompareValidator 举例</font></h3>
<p>在两个文本框里面都输入字符串，然后选择比较规则，按下验证按钮进行比较。
</p>
<form runat=server>
    <table bgcolor="#eeeeee" cellpadding=10>
        <tr valign="top">
            <td>
                <h5><font face="Verdana">字符串 1:</font></h5>
                <!--布置文本框-->
                <asp:TextBox
                Selected
                id="txtComp"
                runat="server">
            </asp:TextBox>
            </td>
            <td>
                <h5><font face="Verdana">比较规则:</font></h5>
                <!--布置选项列表。列举比较规则-->
                <asp:ListBox
                id="lstOperator"
                OnSelectedIndexChanged="lstOperator_SelectedIndexChanged"
                runat="server">
                    <asp:ListItem
                    Selected
                    Value="Equal" >相等
                </asp:ListItem>

```

```

<asp:ListItem
Value="NotEqual" >不相等
</asp:ListItem>
<asp:ListItem
Value="GreaterThan" >GreaterThan
</asp:ListItem>
<asp:ListItem
Value="GreaterThanEqual" >GreaterThanEqual
</asp:ListItem>
<asp:ListItem
Value="LessThan" >少于
</asp:ListItem>
<asp:ListItem
Value="LessThanEqual" >LessThanEqual
</asp:ListItem>
    </asp:ListBox>
    </td>
    <td>
        <h5><font face="Verdana">字符串 2:</font></h5>
<!--布置文本框-->
<asp:TextBox
id="txtCompTo"
runat="server">
</asp:TextBox>
<p>
<!--布置按钮-->
<asp:Button
runat=server
Text="验证"
ID="Button1"
onclick="Button1_OnSubmit"
/>
        </td>
    </tr>
</table>

<!--布置 CompareValidator Web 控件，来比较两个文本框的内容-->
<asp:CompareValidator
id="comp1"
ControlToValidate="txtComp"

```

```

ControlToCompare = "txtCompTo"
Type="String"
runat="server"
/>
        <br>
<!--布置 label 标签，来显示运行结果-->
<asp:Label
ID="lblOutput"
Font-Name="verdana"
Font-Size="10pt"
runat="server"/>
    </form>
</body>
</html>

```

使用 C # 语言编写的程序 ch05/test5_12c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true" clienttarget="downlevel"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
    <script language="C#" runat="server">
        //按下提交按钮所调用的程序 Button1_OnSubmit
        void Button1_OnSubmit(Object sender, EventArgs e) {
            if (Page.IsValid) {
                lblOutput.Text = "结果：正确!";
            }
            else {
                lblOutput.Text = "结果：不正确!";
            }
        }
        //当比较规则变化时，所调用的程序 lstOperator_SelectedIndexChanged
        void lstOperator_SelectedIndexChanged(Object sender, EventArgs e) {
            comp1.Operator = (ValidationCompareOperator) lstOperator.SelectedIndex;
            comp1.Validate();
        }
    </script>
<title>CompareValidator 举例</title>
</head>
<body>
<h3><font face="Verdana">CompareValidator 举例</font>

```

```

</h3>
<p>在两个文本框里面都输入字符串，然后选择比较规则，按下验证按钮进行比较。
</p>
<form runat=server>
    <table bgcolor="#eeeeee" cellpadding=10>
        <tr valign="top">
            <td>
                <h5><font face="Verdana">字符串 1:</font>
</h5>
<!--布置文本框-->
<asp:TextBox
Selected
id="txtComp"
runat="server">
</asp:TextBox>
            </td>
            <td>
                <h5><font face="Verdana">比较规则:</font>
</h5>
<!--布置选项列表，选择比较规则-->
                <asp:ListBox
id="lstOperator"
OnSelectedIndexChanged="lstOperator_SelectedIndexChanged"
runat="server">
                    <asp:ListItem
Selected
Value="Equal" >相等
</asp:ListItem>
                    <asp:ListItem
Value="NotEqual" >不相等
</asp:ListItem>
                    <asp:ListItem
Value="GreaterThan" >GreaterThan
</asp:ListItem>
                    <asp:ListItem
Value="GreaterThanEqual" >GreaterThanEqual
</asp:ListItem>
                    <asp:ListItem
Value="LessThan" >少于

```

```

</asp:ListItem>
<asp:ListItem
Value="LessThanEqual" >LessThanEqual
</asp:ListItem>
</asp:ListBox>
</td>
<td>
<h5><font face="Verdana">字符串 2:</font></h5>
<!--布置文本框-->
<asp:TextBox
id="txtCompTo"
runat="server">
</asp:TextBox>
<p>
<!--布置按钮-->
<asp:Button
runat=server
Text="验证"
ID="Button1"
onclick="Button1_OnSubmit"
/>
</td>
</tr>
</table>
<!--布置 CompareValidator Web 控件，来比较两个文本框的内容-->
<asp:CompareValidator
id="comp1"
ControlToValidate="txtComp"
ControlToCompare = "txtCompTo"
Type="String"
runat="server"
/>
<br>
<!--布置 label 控件，来显示运行的结果-->
<asp:Label
ID="lblOutput"
Font-Name="verdana"
Font-Size="10pt"
runat="server"

```

```
</>  
</form>  
</body>  
</html>
```

程序运行结果如图 5 - 17 所示。

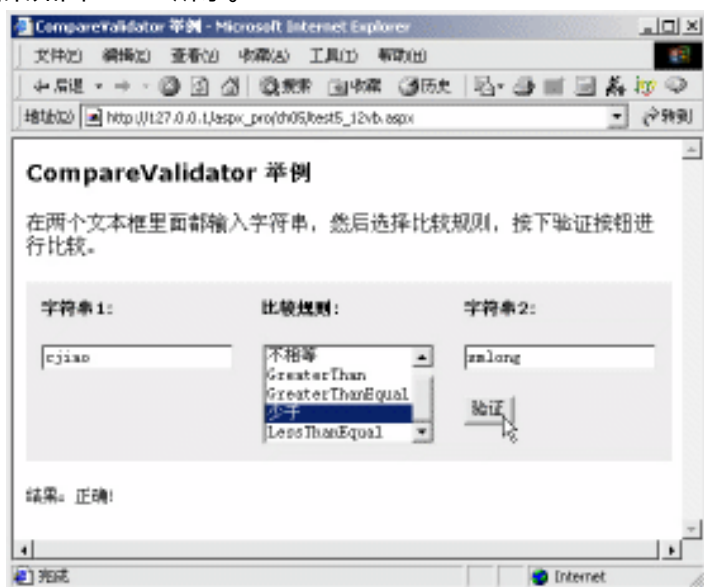


图 5 - 17 CompareValidator Web 控件举例运行结果

5.8 CustomValidator Web 控件

CustomValidator Web 控件提供对客户端提交的内容进行验证的功能。下面举例加以说明。

例如，使用 VB 语言编写的程序 ch05/test5_13vb.aspx 的源代码如下：

```
<%@ Page language="VB" Debug="true"%>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">  
<script language="VB" runat=server>
```

按下验证按钮所调用的程序 ValidateBtn_OnClick

```
Sub ValidateBtn_OnClick(sender As Object, e As EventArgs)  
    If (Page.IsValid) Then  
        lblOutput.Text = "事件正确!"  
    Else  
        lblOutput.Text = "事件不正确! :-(  
    End If
```

```

        End Sub
    ' 服务器验证程序 ServerValidate
    Function ServerValidate (sender As Object, value As String) As Boolean
        Try
            Dim num As Int32 = Int32.FromString(value)
            If (num Mod 2) <> 0 Then
                ServerValidate = False
            else
                ServerValidate = True
            End If
        Catch E As Exception
            ' Do Nothing
        End Try
    End Function
</script>
<title>CustomValidator 举例</title>
</head>
<body>
<h3><font face="Verdana">CustomValidator 举例</font></h3>
<form runat="server">
<!--布置一个 label 标签，来显示运行结果-->
<asp:Label
id=lblOutput
runat="server"
    Text="请输入一个偶数:"
    Font-Name="Verdana"
    Font-Size="10pt" />
<br>
<p>
<!--布置一个文本框-->
<asp:TextBox
id=Text1
runat="server" />
    <!--布置一个 CustomValidator Web 控件，来验证客户提交的数据-->
    <asp:CustomValidator
id="CustomValidator1"
runat="server"
    ControlToValidate="Text1"

```

```
OnServerValidate="ServerValidate"
Display="Static"
Font-Name="verdana"
Font-Size="10pt">
    不是一个偶数!
</asp:CustomValidator>
<p>
<asp:Button
text="验证"
onclick="ValidateBtn_OnClick"
runat="server" />
</form>
</body>
</html>
```

使用 C# 语言编写的程序 ch05/test5_13c.aspx 的源代码如下：

```
<%@ Page language="C#" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat=server>
    //按下验证按钮所调用的程序 ValidateBtn_OnClick
    void ValidateBtn_OnClick(object sender, EventArgs e) {
        if (Page.IsValid) {
            lblOutput.Text = "事件正确!";
        }
        else {
            lblOutput.Text = "事件不正确! :-(";
        }
    }

    //验证提交内容程序
    bool ServerValidate (object source, string value) {
        try {
            int num = Int32.FromString(value);
            if (num%2 == 0)
                return true;
        }
        catch (Exception) {}
        return false;
    }
}
```

```
</script>
<title>CustomValidator 举例
</title>
</head>
<body>
<h3><font face="Verdana">CustomValidator 举例
</font>
</h3>
<form runat="server">
<!-- 布置一个 label 标签，来显示运行的结果-->
<asp:Label
id=lblOutput
runat="server"
    Text="请输入一个偶数:"
    Font-Name="Verdana"
    Font-Size="10pt" />
<br>
    <p>
<!-- 布置一个文本框-->
<asp:TextBox
id=Text1 runat="server" />
<!-- 布置一个 CustomValidator Web 控件，验证时调用的程序 ServerValidate -->
<asp:CustomValidator
id="CustomValidator1"
runat="server"
    ControlToValidate="Text1"
    OnServerValidate=" "
    Display="Static"
    Font-Name="verdana"
    Font-Size="10pt">
        不是偶数!
    </asp:CustomValidator>
    <p>
<!-- 布置按钮，调用 ValidateBtn_OnClick-->
<asp:Button
text="验证"
onclick="ValidateBtn_OnClick"
runat="server" />
</form>
```

```
</body>
```

```
</html>
```

运行结果如图 5 - 18 所示。



图 5 - 18 CustomValidator Web 控件举例运行结果

5.9 DataGrid Web 控件

DataGrid Web 控件是一个可以绑定数据 ,同时显示多列数据的控件。设定 DataGrid Web 控件属性 ,而不必编写任何程序 ,就可以输出数据表格。

DataGrid Web 控件的使用语法如下 :

```
<asp:DataGrid id="programmaticID" runat=server
    DataSource='<%# DataBindingExpression %>'
    AllowPaging="True|False"
    AllowSorting="True|False"
    AutoGenerateColumns="True|False"
    BackImageUrl="url"
    CellPadding="pixels"
    CellSpacing="pixels"
    DataKeyField="DataSourceKeyField"
    GridLines="None|Horizontal|Vertical|Both"
    HorizontalAlign="Center|Justify|Left|NotSet|Right"
    PagedDataSource
    PageSize="ItemCount"
    ShowFooter="True|False"
    ShowHeader="True|False"
    VirtualItemCount="ItemCount"

    AlternatingItemStyle-property="value"
```

```

EditItemStyle-property="value"
FooterStyle-property="value"
HeaderStyle-property="value"
ItemStyle-property="value"
PagerStyle-property="value"
SelectedItemStyle-property="value"

OnCancelCommand="OnCancelCommandMethod"
OnDeleteCommand="OnDeleteCommandMethod"
OnEditCommand="OnEditCommandMethod"
OnItemCommand="OnItemCommandMethod"
OnItemCreated="OnItemCreatedMethod"
OnPageIndexChanged="OnPageIndexChangedMethod"
OnSortCommand="OnSortCommandMethod"
OnUpdateCommand="OnUpdateCommandMethod"
</asp:DataGrid>

```

或者使用下面的语法：

```

<asp:DataGrid id="programmaticID" runat=server
    DataSource=<%# DataBindingExpression %>
    AutoGenerateColumns="False"
    (other properties)

```

```

<property name="Columns">
    <asp:BoundColumn
        DataField="DataSourceField"
        DataFormatString="FormatString"
        FooterText="FooterText"
        HeaderImageUrl="url"
        HeaderText="HeaderText"
        ReadOnly="True|False"
        SortField="DataSourceFieldToSortBy"
        Visible="True|False"
        FooterStyle-property="value"
        HeaderStyle-property="value"
        ItemStyle-property="value"
    />

```

```

<asp:ButtonColumn
    ButtonType="LinkButton|PushButton"

```

```
Command="BubbleText"
DataTextField="DataSourceField"
DataTextFormatString="FormatString"
FooterText="FooterText"
HeaderImageUrl="url"
HeaderText="HeaderText"
ReadOnly="True|False"
SortField="DataSourceFieldToSortBy"
Text="ButtonCaption"
Visible="True|False"
/>
```

```
<asp:EditCommandColumn
    ButtonType="LinkButton|PushButton"
    CancelText="CancelButtonCaption"
    EditText="EditButtonCaption"
    FooterText="FooterText"
    HeaderImageUrl="url"
    HeaderText="HeaderText"
    ReadOnly="True|False"
    SortField="DataSourceFieldToSortBy"
    UpdateText="UpdateButtonCaption"
    Visible="True|False"
/>
```

```
<asp:HyperlinkColumn
    DataNavigateUrlField="DataSourceField"
    DataNavigateUrlFormatString="FormatExpression"
    DataTextField="DataSourceField"
    DataTextFormatString="FormatExpression"
    FooterText="FooterText"
    HeaderImageUrl="url"
    HeaderText="HeaderText"
    NavigateUrl="url"
    ReadOnly="True|False"
    SortField="DataSourceFieldToSortBy"
    Target="window"
    Text="HyperlinkText"
    Visible="True|False"
```

```

/>

<asp:TemplateColumn>
    FooterText="FooterText"
    HeaderImageUrl="url"
    HeaderText="HeaderText"
    ReadOnly="True|False"
    SortField="DataSourceFieldToSortBy"
    Visible="True|False"
    <template name="HeaderTemplate">
        Header template HTML
    </template>
    <template name="ItemTemplate">
        ItemTemplate HTML
    </template>
    <template name="EditItemTemplate">
        EditItem template HTML
    </template>
    <template name="FooterTemplate">
        Footer template HTML
    </template>
</asp:TemplateColumn>
</property>
</asp:DataGrid>

```

5.9.1 使用 DataGrid Web 控件浏览数据

举例一：使用 DataGrid Web 控件浏览数据。

使用 VB 语言编写的程序 ch05/test5_14vb.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    '创建数据视图
    Function CreateDataView(DB as string,table as string) As ICollection
        dim connstr as string

```

```

'链接数据库
connstr="DBQ=" & Server.MapPath(Db)&" ;Driver={Microsoft Access Driver (*.mdb)};"
dim cmd as ADODatasetCommand
cmd=new ADODatasetCommand("Select id,name,sex,college,class from "& table,Connstr )
dim ds as DataSet = new DataSet()
cmd.FillDataSet(ds,table)
CreateDataView= new DataView(ds.tables(table))
End Function
'页面被浏览是调用的程序 Page_Load
Sub Page_Load(sender As Object, e As EventArgs)
    '打开表格
    MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
    '绑定数据源
    MyDataGrid.DataBind()
End Sub
</script>
<title>DataGrid 举例</title>
<body>
<h3><font face="Verdana"> DataGrid 举例
</font>
</h3>
    <form runat=server>
<!--布置 DataGrid Web 控件，来显示数据-->
<ASP:DataGrid
id="MyDataGrid"
runat="server"
    BorderColor="black"
    BorderWidth="1"
    GridLines="Both"
    CellPadding="3"
    CellSpacing="0"
    Font-Name="Verdana"
    Font-Size="8pt"
    HeaderStyle-BackColor="#aaaadd"
/>
</form>
</body>
</html>

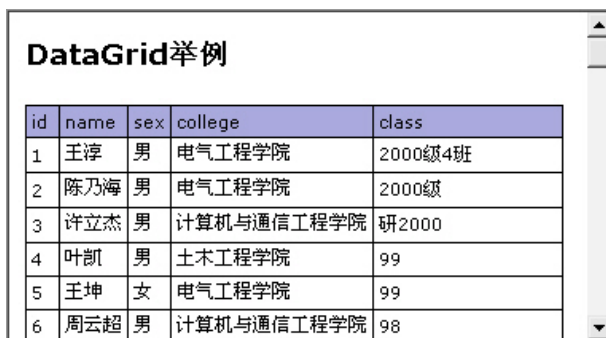
```

使用 C # 语言编写的程序 ch05/test5_14c.aspx 的源代码如下：

```
<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
    //创建函数数据视图 CreateDataview
    ICollection CreateDataview(string Db,string table) {
        string connstr;
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
        ADODatasetCommand cmd;
        cmd=new ADODatasetCommand("Select id,name,sex,college,class from "+table,connstr);
        DataSet ds = new DataSet();
        cmd.FillDataSet(ds,table);
        DataView dv = new DataView(ds.Tables[table]);
        return dv;
    }
    //在页面被浏览时，所调用的程序 Page_Load
    void Page_Load(Object sender, EventArgs e) {
        MyDataGrid.DataSource = CreateDataview("sample.mdb","new");
        MyDataGrid.DataBind();
    }
</script>
<title>DataGrid 举例</title><body>
<h3><font face="Verdana"> DataGrid 举例
</font>
</h3>
<form runat=server>
<!--布置 DataGrid Web 控件-->
<ASP:DataGrid
id="MyDataGrid"
runat="server"
    BorderColor="black"
    BorderWidth="1"
    GridLines="Both"
    CellPadding="3"
    CellSpacing="0"
    Font-Name="Verdana"
```

```
Font-Size="8pt"
HeaderStyle-BackColor="#aaaadd"
/>
</form>
</body>
</html>
```

程序运行结果如图 5 - 19 所示。



id	name	sex	college	class
1	王淳	男	电气工程学院	2000级4班
2	陈乃海	男	电气工程学院	2000级
3	许立杰	男	计算机与通信工程学院	研2000
4	叶凯	男	土木工程学院	99
5	王坤	女	电气工程学院	99
6	周云超	男	计算机与通信工程学院	98

图 5 - 19 DataGrid Web 控件举例一运行结果

5.9.2 使用 DataGrid Web 控件浏览数据更改表头

举例二：使用 DataGrid Web 控件浏览数据，把表头改掉。

使用 VB 语言编写的程序 ch05/test5_37vb.aspx 的源代码如下：

```
<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    '创建数据视图 CreateDataView
    Function CreateDataView(DB as string,table as string) As ICollection
        dim connstr as string
        connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
        dim cmd as ADODatasetCommand
        cmd=new ADODatasetCommand("Select id,name,sex,college,class from "& table,Connstr )
        dim ds as DataSet = new DataSet()
        cmd.FillDataSet(ds,table)
        CreateDataView= new DataView(ds.Tables(table))
    End Function
```

```
'当页面被浏览时所调用的程序 Page_Load
Sub Page_Load(sender As Object, e As EventArgs)
    MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
    MyDataGrid.DataBind()
End Sub
</script>
<title>在 DataGrid 特殊的列</title></head>
<body>

<h3><font face="Verdana">在 DataGrid 特殊的列</font></h3>
<form runat=server>
    <!--布置 DataGrid Web 控件来显示数据-->
    <ASP:DataGrid
id="MyDataGrid"
runat="server"
    BorderColor="black"
    BorderWidth="1"
    GridLines="Both"
    CellPadding="3"
    CellSpacing="0"
    Font-Name="Verdana"
    Font-Size="8pt"
    HeaderStyle-BackColor="#aaaadd"
    AutoGenerateColumns="false">
    <!--制作表头-->
    <property name="Columns">
        <asp:BoundColumn
            HeaderText="序号"
            DataField="id"
        />
        <asp:BoundColumn
            HeaderText="姓名"
            DataField="name"
        />
        <asp:BoundColumn
            HeaderText="性别"
            DataField="sex"
            ItemStyle-HorizontalAlign="center"
        />
```

```
<asp:BoundColumn
    HeaderText="学院、系"
    DataField="college"
/>

<!-- format Price as currency-->
<asp:BoundColumn
    HeaderText="班级"
    DataField="class"
    ItemStyle-HorizontalAlign="right"
/>
</property>
</asp:DataGrid>
</form>
</body>
</html>
```

使用 C # 语言编写的程序 ch05/test5_15c.aspx 的源代码如下：

```
<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
    //创建数据视图 CreateDataview
    ICollection CreateDataview(string Db,string table) {
        string connstr;
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
        ADODatasetCommand cmd;
        cmd=new ADODatasetCommand("Select id,name,sex,college,class from "+table,connstr);
        DataSet ds = new DataSet();
        cmd.FillDataSet(ds,table);
        DataView dv = new DataView(ds.Tables[table]);
        return dv;
    }
    //在页面被浏览的同时，所调用的程序 Page_Load
    void Page_Load(Object sender, EventArgs e) {
        MyDataGrid.DataSource = CreateDataview("sample.mdb","new");
        MyDataGrid.DataBind();
    }
}
```

```
</script>
<title>在 DataGrid 特殊的列</title></head>
<body>
<h3><font face="Verdana">在 DataGrid 特殊的列</font></h3>
  <form runat=server>
    <!-- 布置 DataGrid Web 控件，来显示数据-->
    <ASP:DataGrid
      id="MyDataGrid"
      runat="server"
      BorderColor="black"
      BorderWidth="1"
      GridLines="Both"
      CellPadding="3"
      CellSpacing="0"
      Font-Name="Verdana"
      Font-Size="8pt"
      HeaderStyle-BackColor="#aaaadd"
      AutoGenerateColumns="false">
      <property name="Columns">
        <!-- 设定数据表格的表头-->
        <asp:BoundColumn
          HeaderText="序号"
          DataField="id"
        />
        <asp:BoundColumn
          HeaderText="姓名"
          DataField="name"
        />
        <asp:BoundColumn
          HeaderText="性别"
          DataField="sex"
          ItemStyle-HorizontalAlign="center"
        />
        <asp:BoundColumn
          HeaderText="学院、系"
          DataField="college"
        />
        <!-- format Price as currency-->
        <asp:BoundColumn
```

```

        HeaderText="班级"
        DataField="class"
        ItemStyle-HorizontalAlign="right"
    />
</property>
</asp:DataGrid>
</form>
</body>
</html>

```

运行结果如图 5 - 20 所示。可以与 DataGrid Web 控件举例一的运行结果进行对比。

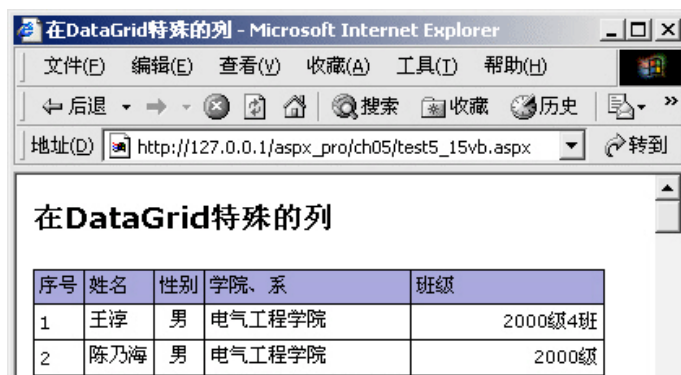


图 5 - 20 DataGrid Web 控件举例二运行结果

5.9.3 使用 DataGrid Web 在数据中加入链接

举例三：使用 DataGrid Web 在数据中加入链接。

使用 VB 语言编写的程序 ch05/test5_16vb.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    '建立一个数据视图 CreateDataView
    Function CreateDataView(DB as string,table as string) As ICollection
        dim connstr as string
        connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
        dim cmd as ADODatasetCommand
        cmd=new ADODatasetCommand("Select id,name,sex,college,class from "& table,Connstr )

```

```
dim ds as DataSet = new DataSet()
cmd.FillDataSet(ds,table)
CreateDataView= new DataView(ds.Tables(table))
End Function
Sub Page_Load(sender As Object, e As EventArgs)
    MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
    MyDataGrid.DataBind()
End Sub
</script>
<title>在 DataGrid 使用链接
</title>
<body>

<h3><font face="Verdana">在 DataGrid 使用链接
</font>
</h3>
    <form runat=server>
<!--布置 DataGrid Web 控件，显示数据库中的数据-->
<ASP:DataGrid
id="MyDataGrid"
runat="server"
    BorderColor="black"
    BorderWidth="1"
    GridLines="Both"
    CellPadding="3"
    CellSpacing="0"
    Font-Name="Verdana"
    Font-Size="8pt"
    HeaderStyle-BackColor="#aaaadd"
    AutoGenerateColumns="false"
>
    <!--设定各个列的属性-->
    <property name="Columns">
        <asp:BoundColumn
            HeaderText="序号"
            DataField="id"
        />
        <!--在此列上面建立链接-->
        <asp:HyperLinkColumn
```

```

        HeaderText="姓名"
        DataNavigateUrlField="id"
        DataNavigateUrlFormatString="detailspage.aspx?id={0}"
        Target="_new"
        DataTextField="name"
    />
    <asp:BoundColumn
        HeaderText="性别"
        DataField="sex"
        ItemStyle-HorizontalAlign="center"
    />
    <asp:BoundColumn
        HeaderText="学院、系"
        DataField="college"
    />
    <!-- format Price as currency-->
    <asp:BoundColumn
        HeaderText="班级"
        DataField="class"
        ItemStyle-HorizontalAlign="right"
    />
</property>
</asp:DataGrid>
</form>
</body>
</html>

```

使用 C # 语言编写的程序 ch05/test5_16c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
    //建立数据视图 CreateDataview
    ICollection CreateDataview(string Db,string table) {
        string connstr;
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
        ADODatasetCommand cmd;

```

```

cmd=new ADODataSetCommand("Select id,name,sex,college,class from "+table,connstr);
DataSet ds = new DataSet();
cmd.FillDataSet(ds,table);
    DataView dv = new DataView(ds.Tables[table]);
    return dv;
}
//在页面被调用时，所调用的程序 Page_Load
void Page_Load(Object sender, EventArgs e) {
    MyDataGrid.DataSource = CreateDataview("sample.mdb","new");
    MyDataGrid.DataBind();
}
</script>
<title>在 DataGrid 使用链接</title><body>
<h3><font face="Verdana">在 DataGrid 使用链接</font></h3>
    <form runat=server>
<!--布置 Datagrid Web 控件-->
<ASP:DataGrid
id="MyDataGrid"
runat="server"
    BorderColor="black"
    BorderWidth="1"
    GridLines="Both"
    CellPadding="3"
    CellSpacing="0"
    Font-Name="Verdana"
    Font-Size="8pt"
    HeaderStyle-BackColor="#aaaadd"
    AutoGenerateColumns="false"
>
    <!--设定数据表格各个列的属性-->
    <property name="Columns">
        <asp:BoundColumn
            HeaderText="序号"
            DataField="id"
        />
        <!--在此列加入链接-->
        <asp:HyperLinkColumn
            HeaderText="姓名"
            DataNavigateUrlField="id"

```

```

        DataNavigateUrlFormatString="detailspage.aspx?id={0}"
        Target="_new"
        DataTextField="name"
    />
    <asp:BoundColumn
        HeaderText="性别"
        DataField="sex"
        ItemStyle-HorizontalAlign="center"
    />
    <asp:BoundColumn
        HeaderText="学院、系"
        DataField="college"
    />
    <!-- format Price as currency-->
    <asp:BoundColumn
        HeaderText="班级"
        DataField="class"
        ItemStyle-HorizontalAlign="right"
    />
    </property>
</asp:DataGrid>
</form>
</body>
</html>

```

程序 ch05/ detailspage.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    Function CreateDataView(DB as string,table as string) As ICollection
        dim connstr as string
        connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
        dim cmd as ADODatasetCommand
        cmd=new ADODatasetCommand("Select name,addr,telephone,intro from "& table &" where id="&
            Request.QueryString("id"),Connstr )
        dim ds as DataSet = new DataSet()

```

```

cmd.FillDataSet(ds,table)
CreateDataView= new DataView(ds.Tables(table))
End Function
Sub Page_Load(sender As Object, e As EventArgs)
    MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
    MyDataGrid.DataBind()
End Sub
</script>
<title>在 DataGrid 中，显示详细情况
</title></head>
<body> <h3><font face="Verdana">在 DataGrid 中，显示详细情况</font></h3>
<form runat=server>
    <ASP:DataGrid
    id="MyDataGrid"
    runat="server"
    BorderColor="black"
    BorderWidth="1"
    GridLines="Both"
    CellPadding="3"
    CellSpacing="0"
    Font-Name="Verdana"
    Font-Size="8pt"
    HeaderStyle-BackColor="#aaaadd"
    AutoGenerateColumns="false">
        <property name="Columns">
            <asp:BoundColumn
            HeaderText="住址"
            DataField="addr"
            />
            <asp:BoundColumn
            HeaderText="电话"
            DataField="telephone"
            />
            <asp:BoundColumn
            HeaderText="简介"
            DataField="intro"
            ItemStyle-HorizontalAlign="left"
            />
        </property>

```

```

</asp:DataGrid>
</form></body></html>

```

运行结果如图 5 - 21 所示。

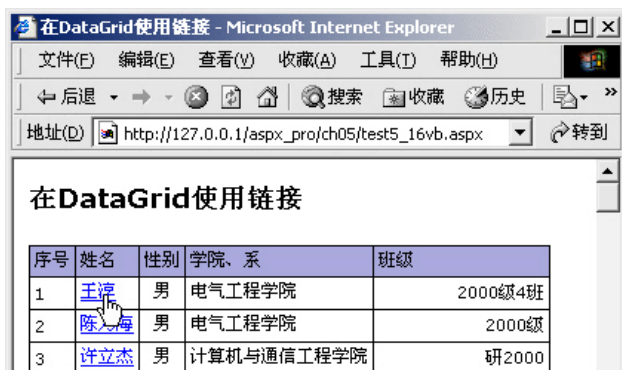


图 5 - 21 DataGrid Web 控件举例三运行结果

详细内容的页面如图 5 - 22 所示。

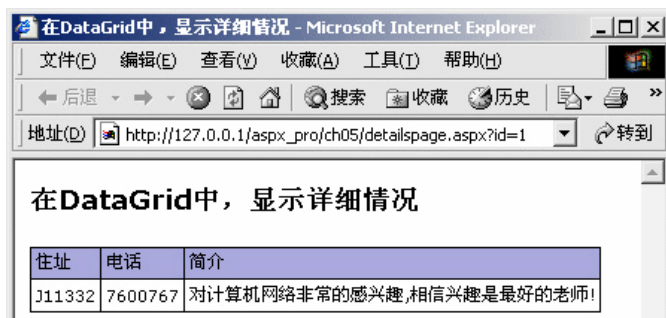


图 5 - 22 显示详细情况的运行结果

5.9.4 使用 DataGrid Web 控件用 Session 临时存储数据

举例四：使用 DataGrid Web 控件用 Session 临时存储数据。

使用 VB 语言编写的程序 ch05/test5_17vb.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    Dim Cart As DataTable
    Dim CartView As DataView
    '建立数据视图 CreateDataView
    Function CreateDataView(DB as string,table as string) As ICollection

```

```

    dim connstr as string
    connstr="DBQ=" & Server.MapPath(Db)&" ;Driver={Microsoft Access Driver (*.mdb)};"
    dim cmd as ADODatasetCommand
    cmd=new ADODatasetCommand("Select id,name from "& table,Connstr )
    dim ds as DataSet = new DataSet()
    cmd.FillDataSet(ds,table)
    CreateDataView= new DataView(ds.Tables(table))
    End Function

Sub Page_Load(sender As Object, e As EventArgs)
    '建立人才选定表格
    If Session("DG4VB_ShoppingCart") = Nothing Then
        Cart = New DataTable()
        Cart.Columns.Add(new DataColumn("序号", GetType(Integer)))
        Cart.Columns.Add(new DataColumn("姓名", GetType(String)))
        Session("DG4VB_ShoppingCart") = Cart
    Else
        Cart = Session("DG4VB_ShoppingCart")
    End If
    CartView = New DataView(Cart)
    ShoppingCart.DataSource = CartView
    ShoppingCart.DataBind
    If Not IsPostBack Then'
        MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
        MyDataGrid.DataBind
    End If
End Sub

'命令执行程序 Grid_CartCommand
Sub Grid_CartCommand(sender As Object, e As DataGridCommandEventArgs)
    Dim dr As DataRow = Cart.NewRow()
    ' e.Item 是要选择的行
    Dim itemCell As TableCell = e.Item.Cells(0)
    Dim priceCell As TableCell = e.Item.Cells(1)
    Dim id As String = itemCell.Text
    Dim name As String = priceCell.Text
    '添加到已经选定人才表格
    If e.CommandSource.CommandName = "AddToCart" Then
        dr(0) = id
        dr(1) = name
    End If
End Sub

```

```

        Cart.Rows.Add(dr)
    Else '从已经选定的人才表格里面去掉所选项
        CartView.RowFilter = "姓名=" & name & ""
        If CartView.Count > 0 Then
            CartView.Delete(0)
        End If
        CartView.RowFilter = ""
    End If
    ShoppingCart.DataBind()
End Sub
</script>
<title>在 DataGrid 中使用 ButtonColumn
</title>
<body>
<h3><font face="Verdana">在 DataGrid 中使用 ButtonColumn
</font>
</h3>
    <form runat=server>
        <table cellpadding="5">
            <tr valign="top">
                <td>
                    <b>人才列表</b>
                <!-- 布置 DataGrid 控件，来显示数据 -->
                <ASP:DataGrid
                    id="MyDataGrid"
                    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_CartCommand"
                >
                <property name="Columns">
                    <asp:BoundColumn

```

```

        HeaderText="序号"
        DataField="id"
    />
    <asp:BoundColumn
        HeaderText="姓名"
        DataField="name"
    />
    '使用 ButtonColumn 控件。执行按钮命令
    <asp:ButtonColumn
        HeaderText="选定人才"
        Text="选定"
        CommandName="AddToCart"
    />
    <!-- format Price as currency-->
    <asp:ButtonColumn
        HeaderText="去掉"
        text="去掉"
        CommandName="RemoveFromCart"
        ItemStyle-HorizontalAlign="right"
    />
    </property>
</asp:DataGrid>
</td><td>
<b>已经选定人才</b>
<!--显示选定人才的列表-->
    <ASP:DataGrid
        id="ShoppingCart"
        runat="server"
        BorderColor="black"
        BorderWidth="1"
        CellPadding="3"
        Font-Name="Verdana"
        Font-Size="8pt"
        HeaderStyle-BackColor="#aaaadd"
    />
</td>
</tr>
</table>
</form>

```

```
</body>
```

```
</html>
```

使用 C# 程序语言编写的 ch05/test5_17c.aspx 的源代码如下：

```
<%@ Page language="C#" Debug="true"%>
```

```
<%@ Import Namespace="System.Data" %>
```

```
<%@ Import Namespace="System.Data.ADO" %>
```

```
<%@ Import Namespace="System"%>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
```

```
<script runat="server">
```

```
    DataTable Cart;
```

```
    DataView CartView;
```

```
    //建立数据视图函数 CreateDataview
```

```
    ICollection CreateDataview(string Db,string table) {
```

```
        string connstr;
```

```
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
```

```
        ADODataSetCommand cmd;
```

```
        cmd=new ADODataSetCommand("Select name,sex from "+table,connstr);
```

```
        DataSet ds = new DataSet();
```

```
        cmd.FillDataSet(ds,table);
```

```
        DataView dv = new DataView(ds.Tables[table]);
```

```
        return dv;
```

```
    }
```

```
    //在页面被浏览时，所调用的程序 Page_Load
```

```
    void Page_Load(Object sender, EventArgs e) {
```

```
        //把临时的选项放在 Session 里面
```

```
        if (Session["DG4_ShoppingCart"] == null) {
```

```
            Cart = new DataTable();
```

```
            Cart.Columns.Add(new DataColumn("姓名", typeof(string)));
```

```
            Cart.Columns.Add(new DataColumn("性别", typeof(string)));
```

```
            Session["DG4_ShoppingCart"] = Cart;
```

```
        }
```

```
        else {
```

```
            Cart = (DataTable)Session["DG4_ShoppingCart"];
```

```
        }
```

```
        CartView = new DataView(Cart);
```

```
        ShoppingCart.DataSource = CartView;
```

```
        ShoppingCart.DataBind();
```



```
<ASP:DataGrid
id="MyDataGrid"
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_CartCommand"
>
<!-- 设定各个列的属性-->
<property name="Columns">
    <asp:BoundColumn
        HeaderText="姓名"
        DataField="name"
    />
    <asp:BoundColumn
        HeaderText="性别"
        DataField="sex"
    />
    <asp:ButtonColumn
        HeaderText="选定人才"
        Text="选定"
        CommandName="AddToCart"
    />
    <!-- format Price as currency-->
    <asp:ButtonColumn
        HeaderText="去掉"
        text="去掉"
        CommandName="RemoveFromCart"
        ItemStyle-HorizontalAlign="right"
    />
</property>
</asp:DataGrid>
</td><td>
```

```

<b>已经选定人才</b>
<!--布置 DataGrid 控件，显示选定的人才列表-->
<ASP:DataGrid
id="ShoppingCart"
runat="server"
BorderColor="black"
BorderWidth="1"
CellPadding="3"
Font-Name="Verdana"
Font-Size="8pt"
HeaderStyle-BackColor="#aaaadd"
/></td>    </tr>    </table>

</form>
</body>
</html>

```

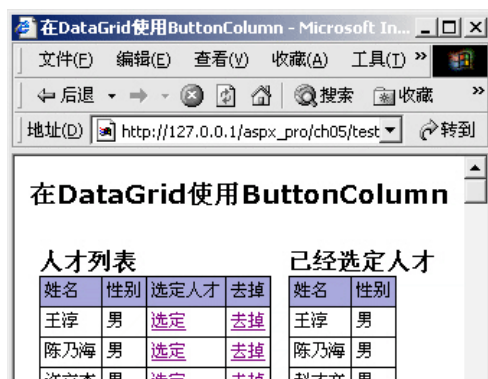


图 5 - 23 在 DataGrid 中使用 ButtonColumn 运行结果

运行结果如图 5 - 23 所示。

5.9.5 给 DataGrid Web 控件中加入模板

举例五：使用 DataGrid Web 控件浏览数据，使用模板来显示数据表格。

使用 VB 语言编写的程序 ch05/test5_18vb.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
'建立临时选定人才表格
Dim Cart As DataTable
Dim CartView As DataView
'创建数据视图 CreateDataView
Function CreateDataView(DB as string,table as string) As ICollection
    dim connstr as string
    connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
    dim cmd as ADODatasetCommand
    cmd=new ADODatasetCommand("Select id,name from "& table,Connstr )
    dim ds as DataSet = new DataSet( )
    cmd.FillDataSet(ds,table)

```

```
CreateDataView= new DataView(ds.Tables(table))
End Function
'当页面被流蓝色时所调用的程序 Page_Load
Sub Page_Load(sender As Object, e As EventArgs)
    If Session("DG5VB_ShoppingCart") = Nothing Then
        Cart = New DataTable()
        Cart.Columns.Add(new DataColumn("序号", GetType(Integer)))
        Cart.Columns.Add(new DataColumn("姓名", GetType(string)))
        Session("DG5VB_ShoppingCart") = Cart
    Else
        Cart = Session("DG5VB_ShoppingCart")
    End If
    CartView = New DataView(Cart)
    ShoppingCart.DataSource = CartView
    CartView.Sort=id
    ShoppingCart.DataBind
    MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
    MyDataGrid.DataBind
End Sub
Sub Grid_CartCommand(sender As Object, e As DataGridCommandEventArgs)
    Dim dr As DataRow = Cart.NewRow( )
    ' e.Item 是要选定的行
    Dim itemCell As TableCell = e.Item.Cells(1)
    Dim priceCell As TableCell = e.Item.Cells(2)
    Dim id As Integer = itemCell.Text
    Dim name As String = priceCell.Text
    If e.CommandSource.CommandName = "AddToCart" Then
        dr(0) = id
        dr(1) = name
        Cart.Rows.Add(dr)
    Else '从已经选定的人才里面去掉
        CartView.RowFilter = "姓名=" & name & ""
        If CartView.Count > 0 Then
            CartView.Delete(0)
        End If
        CartView.RowFilter = ""
    End If
    ShoppingCart.DataBind( )
End Sub
```

```

</script>
<title>在 DataGrid 使用模板</title></head>
<body>
<h3><font face="Verdana">在 DataGrid 使用模板</font></h3>
    <form runat=server>
        <table cellpadding="5">
            <tr valign="top">
                <td>
<b>人才列表</b>
<!--布置 DataGrid 控件。来显示数据-->
<asp:DataGrid
id="MyDataGrid"
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_CartCommand"
>
    <property name="Columns">
        <!--使用模板，来显示添加/删除-->
        <asp:TemplateColumn
            HeaderText="添加/删除">
                <template name="ItemTemplate">
                    <asp:LinkButton
                        ID=AddButton
                        Text="添加"
                        CommandName="AddToCart"
                        ForeColor="blue"
                        runat="server"
                    />
                    <asp:LinkButton
                        ID=RemoveButton
                        Text="去掉"

```

```

        CommandName="RemoveFromCart"
        ForeColor="blue"
        runat="server"
    />
</template>
</asp:TemplateColumn>
<asp:BoundColumn
    HeaderText="序号"
    DataField="id"/>
<!-- format Price as currency-->
<asp:BoundColumn
    HeaderText="姓名"
    DataField="name"
    ItemStyle-HorizontalAlign="right" />
</asp:TemplateColumn>
</property>
</asp:DataGrid>
</td><td>
<b>已经选定的人才</b>
<!--布置 DataGrid 控件，来显示选定的人才-->
<asp:DataGrid
id="ShoppingCart"
runat="server"
    BorderColor="black"
    BorderWidth="1"
    CellPadding="3"
    Font-Name="Verdana"
    Font-Size="8pt"
    HeaderStyle-BackColor="#aaaadd"
    />
</td>
</tr>
</table>
</form>
</body>
</html>

```

使用 C# 语言编写的程序 ch05/test5_18c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>

```

```
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script runat="server">
    DataTable Cart;
    DataView CartView;
    //创建数据视图 CreateDataview
    ICollection CreateDataview(string Db,string table) {
        string connstr;
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
        ADODatasetCommand cmd;
        cmd=new ADODatasetCommand("Select name,sex from "+table,connstr);
        DataSet ds = new DataSet();
        cmd.FillDataSet(ds,table);
        DataView dv = new DataView(ds.Tables[table]);
        return dv;
    }
    //当页面被浏览时，所调用的程序 Page_Load
    void Page_Load(Object sender, EventArgs e)
    {
        //通过 Session 变量来保存已经选定的人才的数据
        if (Session["DG5_ShoppingCart"] == null) {
            Cart = new DataTable();
            Cart.Columns.Add(new DataColumn("姓名", typeof(string)));
            Cart.Columns.Add(new DataColumn("性别", typeof(string)));
            Session["DG5_ShoppingCart"] = Cart;
        }
        else {
            Cart = (DataTable)Session["DG5_ShoppingCart"];
        }
        CartView = new DataView(Cart);
        ShoppingCart.DataSource = CartView;
        CartView.Sort="性别";
        ShoppingCart.DataBind( );
        MyDataGrid.DataSource = CreateDataview("sample.mdb","new");
        MyDataGrid.DataBind( );
    }
    void Grid_CartCommand(object sender, DataGridCommandEventArgs e) {
```

```

        DataRow dr = Cart.NewRow( );
        // e.Item 是要选定的行
        TableCell itemCell = e.Item.Cells[1];
        TableCell priceCell = e.Item.Cells[2];
        string name = itemCell.Text;
        string sex = priceCell.Text;
        if (((LinkButton)e.CommandSource).CommandName == "AddToCart") {
            dr[0] = name;
            dr[1] = sex;
            Cart.Rows.Add(dr);
        }
        else { //Remove from Cart
            CartView.RowFilter = "姓名='"+name+"'";
            if (CartView.Count > 0) {
                CartView.Delete(0);
            }
            CartView.RowFilter = "";
        }
        ShoppingCart.DataBind( );
    }
</script>
<title>在 DataGrid 中使用 Template</title><body>
<h3><font face="Verdana">在 DataGrid 中使用 Template</font></h3>
    <form runat=server>
        <table cellpadding="5">
            <tr valign="top">
                <td>
                    <b>人才列表</b>
                    <!-- 布置 DataGrid 控件，来显示数据-->
                    <asp:DataGrid
                        id="MyDataGrid"
                        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_CartCommand"
    >
        <!--设定各个列的属性-->
        <property name="Columns">
            <!--以模板形式来显示数据-->
            <asp:TemplateColumn
                HeaderText="添加/删除">
                <template name="ItemTemplate">
                    <asp:LinkButton
                        ID=AddButton
                        Text="添加"
                        CommandName="AddToCart"
                        ForeColor="blue"
                        runat="server"
                    />
                    <asp:LinkButton
                        ID=RemoveButton
                        Text="删除"
                        CommandName="RemoveFromCart"
                        ForeColor="blue"
                        runat="server"
                    />
                </template>
            </asp:TemplateColumn>
            <asp:BoundColumn
                HeaderText="姓名"
                DataField="name"
            />
            <!-- format Price as currency-->
            <asp:BoundColumn
                HeaderText="性别"
                DataField="sex"
                ItemStyle-HorizontalAlign="right"
            />
        </property>
    </asp:DataGrid>
</td>

```

```

<td>
    <b>已经选定的人才</b>
    <asp:DataGrid
        id="ShoppingCart"
        runat="server"
        BorderColor="black"
        BorderWidth="1"
        CellPadding="3"
        Font-Name="Verdana"
        Font-Size="8pt"
        HeaderStyle-BackColor="#aaaadd"
    />
</td>    </tr>
</table>    </form>
</body></html>

```

运行结果如图 5 - 24 所示。

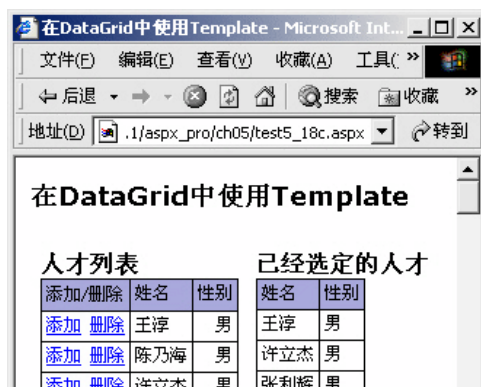


图 5 - 24 在 DataGrid 控件中使用模板的运行结果

5.9.6 使用 DataGrid Web 控件编辑数据

举例六：在 DataGrid 中使用编辑命令。

使用 VB 语言编写的程序 ch05/test5_19vb.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    Dim Cart As DataTable
    Dim CartView As DataView
    Sub Page_Load(sender As Object, e As EventArgs)
        Dim dr As DataRow
        If Session("DG6VB_ShoppingCart") = Nothing Then
            Cart = New DataTable( )
            Cart.Columns.Add(new DataColumn("Qty", GetType(String)))
            Cart.Columns.Add(new DataColumn("Item", GetType(String)))
            Cart.Columns.Add(new DataColumn("Price", GetType(String)))
            Session("DG6VB_ShoppingCart") = Cart
            '插入行同时写入数据
            For i = 1 To 6
                dr = Cart.NewRow( )

```

```

        If (i Mod 2 <> 0) Then
            dr(0) = "2"
        Else
            dr(0) = "1"
        End If
        dr(1) = "产品" & i.ToString
        dr(2) = (1.23 * (i + 1)).ToString
        Cart.Rows.Add(dr)
    Next
Else
    Cart = Session("DG6VB_ShoppingCart")
End If
CartView = New DataView(Cart)
CartView.Sort="Item"
If Not IsPostBack Then
    '每次的数据只需链接一次
    BindGrid
End If
End Sub

'进行数据编辑时所调用的程序 MyDataGrid_Edit
Sub MyDataGrid_Edit(sender As Object, e As DataGridCommandEventArgs)
    MyDataGrid.EditItemIndex = e.Item.ItemIndex
    BindGrid
End Sub

'按下取消所调用的程序 MyDataGrid_Cancel
Sub MyDataGrid_Cancel(sender As Object, e As DataGridCommandEventArgs)
    MyDataGrid.EditItemIndex = -1
    BindGrid
End Sub

'按下更改所调用的程序 MyDataGrid_Update
Sub MyDataGrid_Update(sender As Object, e As DataGridCommandEventArgs)
    ' 在更改行里面，被更改的值存放在文本框里面
    Dim qtyText As TextBox = e.Item.Cells(2).Controls(0)
    Dim priceText As TextBox = e.Item.Cells(3).Controls(0)
    Dim item As String = e.Item.Cells(1).Text
    Dim qty As String = qtyText.Text
    Dim price As String = priceText.Text
    '如果是数据库的话，将使用 uodate 语句，但是在这里只是对内存里面的数据进行修改
    '把原来的数据表格删掉，换上一个新的

```

```

        '把原来的数据表格删掉
        ' 使用 RowFilter 属性对数据进行筛选
        CartView.RowFilter = "Item=" & item & ""
        If CartView.Count > 0 Then
            CartView.Delete(0)
        End If
        CartView.RowFilter = ""
        'add new entry
        Dim dr As DataRow = Cart.NewRow( )
        dr(0) = qty
        dr(1) = item
        dr(2) = price
        Cart.Rows.Add(dr)
        MyDataGrid.EditItemIndex = -1
        BindGrid
    End Sub
Sub BindGrid( )
    '绑定数据
    MyDataGrid.DataSource = CartView
    MyDataGrid.DataBind
End Sub
</script>
<title>在 DataGrid 中使用编辑命令</title>
<body style="font: 10pt verdana">
    <form runat="server">
        <h3>
<font face="Verdana">在 DataGrid 中使用编辑命令</font>
</h3>
<!--布置 DataGrid 控件，来显示数据-->
<asp:DataGrid
id="MyDataGrid"
runat="server"
    BorderColor="black"
    BorderWidth="1"
    CellPadding="3"
    Font-Name="Verdana"
    Font-Size="8pt"
    HeaderStyle-BackColor="#aaaadd"
    OnEditCommand="MyDataGrid_Edit"

```

```

        OnCancelCommand="MyDataGrid_Cancel"
        OnUpdateCommand="MyDataGrid_Update"
        AutoGenerateColumns="false"
    >
    <!--设定各个列的属性-->
    <property name="Columns">
        <!-- EditCommandColumn 为命令列-->
    <asp:EditCommandColumn
        EditText="编辑"
        CancelText="取消"
        UpdateText="更改"
        ItemStyle-Wrap="false"
        HeaderText="编辑命令栏"
        HeaderStyle-Wrap="false"
    />
    <asp:BoundColumn
        HeaderText="产品"
        ReadOnly="true"
        DataField="Item"
    />
    <asp:BoundColumn
        HeaderText="需求量"
        DataField="Qty"
    />
    <asp:BoundColumn
        HeaderText="价格"
        DataField="Price"
    />
    </property>
    </asp:DataGrid>
</form>
</body>
</html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_19c.aspx 的源代码如下：

```

ch05/test5_41c.aspx
<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<html>
<head>

```

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
    DataTable Cart;
    DataView CartView;
//当页面被浏览时所调用的程序 Page_Load
    void Page_Load(Object sender, EventArgs e) {
        if (Session["DG6_ShoppingCart"] == null) {
            Cart = new DataTable();
            Cart.Columns.Add(new DataColumn("Qty", typeof(string)));
            Cart.Columns.Add(new DataColumn("Item", typeof(string)));
            Cart.Columns.Add(new DataColumn("Price", typeof(string)));
            Session["DG6_ShoppingCart"] = Cart;
            //页面被浏览时，建立数据表格
            for (int i=1; i<5; i++) {
                DataRow dr = Cart.NewRow();
                dr[0] = Int32.ToString((i%2)+1);
                dr[1] = "产品" + Int32.ToString(i);
                dr[2] = double.ToString(1.23 * (i+1));
                Cart.Rows.Add(dr);
            }
        }
        else {
            Cart = (DataTable)Session["DG6_ShoppingCart"];
        }
        CartView = new DataView(Cart);
        CartView.Sort = "Item";
        if (!IsPostBack) {
            BindGrid();
        }
    }
//编辑数据表格的程序 MyDataGrid_Edit
    public void MyDataGrid_Edit(Object sender, DataGridCommandEventArgs e) {
        MyDataGrid.EditItemIndex = (int)e.Item.ItemIndex;
        BindGrid();
    }
//取消操作时所调用的程序 MyDataGrid_Cancel
    public void MyDataGrid_Cancel(Object sender, DataGridCommandEventArgs e) {
        MyDataGrid.EditItemIndex = -1;
        BindGrid();
    }
}
```

```

    }
    //进行更改操作时所调用的程序 MyDataGrid_Update
    public void MyDataGrid_Update(Object sender, DataGridCommandEventArgs e) {
        //编辑时，要更改的数据在文本框里面
        string item = e.Item.Cells[1].Text;
        string qty = ((TextBox)e.Item.Cells[2].Controls[0]).Text;
        string price = ((TextBox)e.Item.Cells[3].Controls[0]).Text;
        //如果是数据库的话，使用 update 来更改数据，这里是一个临时的表格
        //在内存里面更改就可以
        //删掉原来的数据表里面的行，加入新的行
        //使用 RowFilter 属性来筛选数据
        CartView.RowFilter = "Item='" + item + "'";
        if (CartView.Count > 0) {
            //item exists in cart
            CartView.Delete(0);
        }
        CartView.RowFilter = "";
        //加入新的表格
        DataRow dr = Cart.NewRow();
        dr[0] = qty;
        dr[1] = item;
        dr[2] = price;
        Cart.Rows.Add(dr);
        MyDataGrid.EditItemIndex = -1;
        BindGrid();
    }
    public void BindGrid() {
        MyDataGrid.DataSource = CartView;
        MyDataGrid.DataBind();
    }
}
</script>
<title>在 DataGrid 中使用编辑命令</title><body style="font: 10pt verdana">
    <form runat="server">
        <h3><font face="Verdana">在 DataGrid 中使用编辑命令</font></h3>
        <asp:DataGrid
            id="MyDataGrid"
            runat="server"
                BorderColor="black"
                BorderWidth="1"

```

```
        CellPadding="3"
        Font-Name="Verdana"
        Font-Size="8pt"
        HeaderStyle-BackColor="#aaaaadd"
        OnEditCommand="MyDataGrid_Edit"
        OnCancelCommand="MyDataGrid_Cancel"
        OnUpdateCommand="MyDataGrid_Update"
        AutoGenerateColumns="false"
    >
    <property name="Columns">
        <asp:EditCommandColumn
            EditText="编辑"
            CancelText="取消"
            UpdateText="更改"
            ItemStyle-Wrap="false"
            HeaderText="命令栏"
            HeaderStyle-Wrap="false"
        />
    <asp:BoundColumn
        HeaderText="产品"
        ReadOnly="true"
        DataField="Item"
    />
    <asp:BoundColumn
        HeaderText="需求量"
        DataField="Qty"
    />
    <asp:BoundColumn
        HeaderText="价格"
        DataField="Price"
    />
    </property>
</asp:DataGrid>
</form>
</body>
</html>
```

运行结果如图 5 - 25 所示。

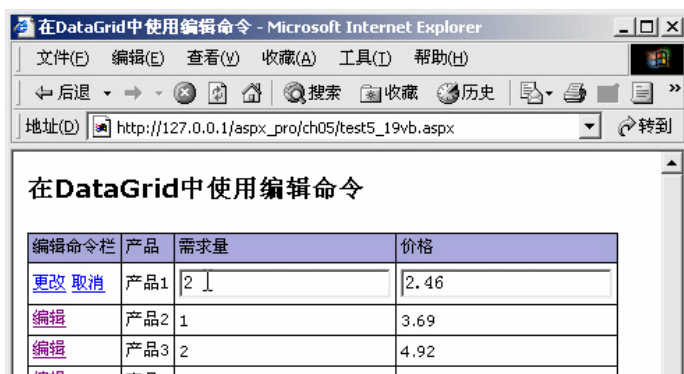


图 5 - 25 在 DataGrid 中使用编辑命令的运行结果

5.9.7 使用 DataGrid Web 控件隐藏一列数据

举例七：在 DataGrid 中隐藏一列数据。

使用 VB 语言编写的程序 ch05/test5_20vb.aspx 的源代码如下：

```
<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    '使用 ICollection 来建立数据视图
    Function CreateDataView(DB as string,table as string) As ICollection
dim connstr as string
'链接数据库
connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
dim cmd as ADODatasetCommand
cmd=new ADODatasetCommand("Select id,name,sex,college,class from "& table,Connstr )
dim ds as DataSet = new DataSet( )
cmd.FillDataSet(ds,table)
        '建立数据视图
CreateDataView= new DataView(ds.Tables(table))
    End Function
    '当页面被浏览时，所调用的程序 Page_Load
    Sub Page_Load(sender As Object, e As EventArgs)
        If Not IsPostBack Then
            MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
            MyDataGrid.DataBind
        End If
    End Sub
End Script
</head>
</html>
```

```

        End If
    End Sub

    '按下更改一列数据可见性的按钮，所调用的程序 Button1_Col1Vis
    Sub Button1_Col1Vis(sender As Object, e As EventArgs)
        MyDataGrid.Columns(2).Visible = Not MyDataGrid.Columns(2).Visible
        Label1.Text = "该列的可见属性为：" & MyDataGrid.Columns(2).Visible.ToString
    End Sub
</script>
<title>在 DataGrid 中隐藏列数据
</title>
<body>
<h3>
<font face="Verdana">
在 DataGrid 中隐藏列数据
</font>
</h3>
<form runat=server>
    <!--布置 DataGrid 控件，显示数据表格-->
    <ASP:DataGrid
id="MyDataGrid"
runat="server"
        BorderColor="black"
        BorderWidth="1"
        GridLines="Both"
        CellPadding="3"
        CellSpacing="0"
        Font-Name="Verdana"
        Font-Size="8pt"
        HeaderStyle-BackColor="#aaaadd"
        AutoGenerateColumns="false"
    >
        <!--设定数据表格各个列的属性-->
        <property name="Columns">
            <asp:BoundColumn
                HeaderText="序号"
                DataField="id"
            />
            <asp:BoundColumn
                HeaderText="姓名"

```

```

        DataField="name"
    />
    <asp:BoundColumn
        HeaderText="性别"
        DataField="sex"
    />
    <asp:BoundColumn
        HeaderText="学院"
        DataField="college"
    />
    <asp:BoundColumn
        HeaderText="班级"
        DataField="class"
        ItemStyle-HorizontalAlign="right"
    />
</property>
</asp:DataGrid>
<br>
<!-- 布置按钮控件来确定某列数据的显示或者隐藏-->
    <asp:Button
        id="Button1"
        Text="更改【 性别】列的可见性"
        OnClick="Button1_Col1Vis"
        runat="server"
    />
<br>
<!-- 布置 label 控件显示运行结果-->
    <asp:Label
        id="Label1"
        runat="server"
    />
</form></body></html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_20c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

```

```

<script runat="server">
    //使用 ICollection 来建立数据视图
    ICollection CreateDataview(string Db,string table) {
        string connstr;
        //连接数据库
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
        //建立 ADODatasetCommand 命令
        ADODatasetCommand cmd;
        cmd=new ADODatasetCommand("Select id,name,sex,college,class from "+table,connstr);
        DataSet ds = new DataSet( );
        cmd.FillDataSet(ds,table);
        DataView dv = new DataView(ds.Tables[table]);
        return dv;
    }
    //当页面被浏览时，所调用的程序 Page_Load
    void Page_Load(Object sender, EventArgs e) {
        if (!IsPostBack) {
            MyDataGrid.DataSource = CreateDataview("sample.mdb","new");
            MyDataGrid.DataBind( );
            Label1.Text = "列 3 的可见性：" + MyDataGrid.Columns[2].Visible.ToString( );
        }
    }
    //按下按钮所调用的程序 Button1_Col1Vis
    void Button1_Col1Vis(Object sender, EventArgs e) {
        MyDataGrid.Columns[2].Visible = !MyDataGrid.Columns[2].Visible;
        Label1.Text = "列 3 的可见性：" + MyDataGrid.Columns[2].Visible.ToString( );
    }
}
</script>
<title>在 DataGrid 在 DataGrid 中隐藏列数据
</title>
<body>
<h3>
<font face="Verdana">在 DataGrid 中隐藏列数据
</font></h3>
<form runat="server">
    <!--布置 DataGrid 控件来显示数据-->
    <ASP:DataGrid
    id="MyDataGrid"
    runat="server"

```

```
        BorderColor="black"
        BorderWidth="1"
        GridLines="Both"
        CellPadding="3"
        CellSpacing="0"
        Font-Name="Verdana"
        Font-Size="8pt"
        HeaderStyle-BackColor="#aaaadd"
        AutoGenerateColumns="false"
    >
    <!-- 设定各个数据列的属性-->
        <property name="Columns">
            <asp:BoundColumn
                HeaderText="序号"
                DataField="id"
            />
            <asp:BoundColumn
                HeaderText="姓名"
                DataField="name"
            />
            <asp:BoundColumn
                HeaderText="性别"
                DataField="sex"
            />
            <asp:BoundColumn
                HeaderText="学院"
                DataField="college"
            />
            <asp:BoundColumn
                HeaderText="班级"
                DataField="class"
                ItemStyle-HorizontalAlign="right"
            />
        </property>
    </asp:DataGrid>
    <br>
    <!-- 布置 Button 按钮控件-->
        <asp:Button
            id="Button1"
```

```
Text="更改【性别】栏的可见性"  
OnClick="Button1_Col1Vis"  
runat="server"  
/>  
<br>  
<!--布置 label 控件来显示运行的结果-->  
    <asp:Label  
        id="Label1"  
        runat="server"  
    />  
</form></body></html>
```

程序运行结果如图 5-26 所示。

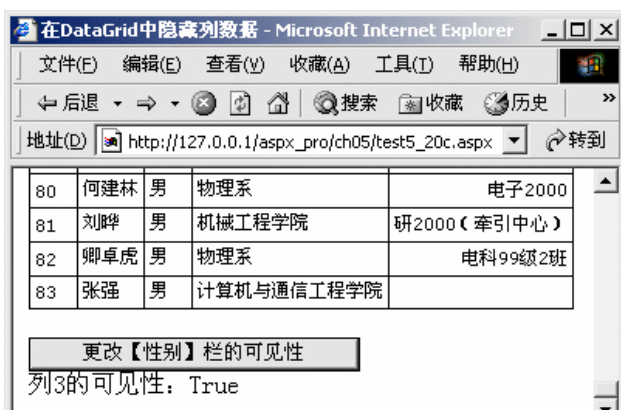


图 5-26 DataGrid 中隐藏一列数据当显示该列时的运行结果

不显示数据时的运行结果如图 5-27 所示。

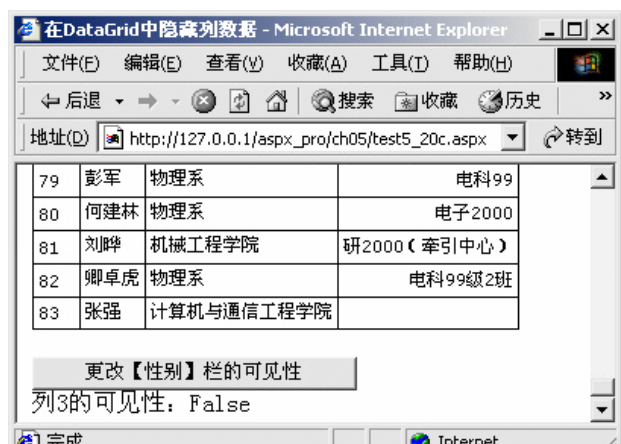


图 5-27 DataGrid 中隐藏一列数据当隐藏该列时的运行结果

5.9.8 使用 DataGrid Web 控件设置数据排序

举例八：在 DataGrid 中设置排序。

使用 VB 语言编写的程序 ch05/test5_21vb.aspx 的源代码如下：

```
<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    Dim SortField As String
    '函数 CreateDataView 使用 ICollection 返回数据视图
    Function CreateDataView(DB as string,table as string) As ICollection
        dim connstr as string
        connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
        dim cmd as ADODatasetCommand
        cmd=new ADODatasetCommand("Select id,name,sex,college,class from "& table &" order by
            id DESC",Connstr )
        dim ds as DataSet = new DataSet( )
        cmd.FillDataSet(ds,table)
        Dim dv as DataView
        dv=new DataView(ds.Tables(table))
        dv.Sort = SortField
        CreateDataView=dv
    End Function
    Sub Page_Load(sender As Object, e As EventArgs)
        If Not IsPostBack Then
            If SortField = "" Then
                SortField = "sex"
            End If
            BindGrid
        End If
    End Sub
    Sub MyDataGrid_Sort(sender As Object, e As DataGridSortCommandEventArgs)
        ' 根据字段排序
        SortField = e.SortField
        BindGrid
    End Sub
End Script
</head>
</html>
```

```

End Sub
Sub BindGrid( )
    '建立数据源
    MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
    '绑定数据
    MyDataGrid.DataBind
End Sub
</script>
<title>在 DataGrid 设置排序</title><body>
<h3><font face="Verdana">在 DataGrid 设置排序</font></h3>
    <form runat=server>
        <!--布置控件 DataGrid -->
        <ASP:DataGrid
            id="MyDataGrid"
            runat="server"
                AllowSorting="true"
                OnSortCommand="MyDataGrid_Sort"
                BorderColor="black"
                BorderWidth="1"
                CellPadding="3"
                Font-Name="Verdana"
                Font-Size="8pt"
                HeaderStyle-BackColor="#ccccff"
                HeaderStyle-ForeColor="black"
            />
    </form>
</body>
</html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_21c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
    <script language="C#" runat="server">
        string SortField;
        //使用 ICollection 来建立数据视图
        ICollection CreateDataview(string Db,string table) {

```

```

        string connstr;
//连接数据库
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
        ADODatasetCommand cmd;
        cmd=new ADODatasetCommand("Select id,name,sex,college,class from "+table,connstr);
        DataSet ds = new DataSet( );
        cmd.FillDataSet(ds,table);
        DataView dv = new DataView(ds.Tables[table]);
        //在数据中设置排序
        dv.Sort = SortField;
        //返回数据视图
        return dv;
    }
    void Page_Load(Object sender, EventArgs e) {
        if (!IsPostBack) {
            if (SortField == "") {
                SortField = "id";
            }
            BindGrid( );
        }
    }
//排序时调用的程序 MyDataGrid_Sort
    void MyDataGrid_Sort(Object sender, DataGridSortCommandEventArgs e) {
        SortField = (string)e.SortField;
        BindGrid();
    }
//绑定数据
    void BindGrid() {
        MyDataGrid.DataSource = CreateDataview("sample.mdb","new");
        MyDataGrid.DataBind( );
    }
</script>
<title>
在 DataGrid 排序
</title>
<body>
<h3>
<font face="Verdana">
    在 DataGrid 排序

```

```

</font>
</h3>
<form runat=server>
<!--布置 DataGrid 控件。来显示数据-->
    <ASP:DataGrid
    id="MyDataGrid"
    runat="server"

    <!--允许排序-->
        AllowSorting="true"
        OnSortCommand="MyDataGrid_Sort"
        BorderColor="black"
        BorderWidth="1"
        CellPadding="3"
        Font-Name="Verdana"
        Font-Size="8pt"
        HeaderStyle-BackColor="#ccccff"
        HeaderStyle-ForeColor="black"
    />

</form></body></html>

```

运行结果如图 5 - 28 和 5 - 29 所示。



图 5 - 28 以 id 排序时的运行结果



图 5 - 29 以 college 排序时的运行结果

5.9.9 使用 DataGrid Web 控件自定义排序

举例九：在 DataGrid 中自定义排序。

使用 VB 语言编写的程序 ch05/test5_22vb.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
Dim SortField As String
' 使用 ICollection 建立数据视图
Function CreateDataView(DB as string,table as string) As ICollection
dim connstr as string
'连接数据库
connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
dim cmd as ADODatasetCommand
cmd=new ADODatasetCommand("Select id,name,sex,college from "& table &" order by id
DESC",Connstr )
dim ds as DataSet = new DataSet( )
cmd.FillDataSet(ds,table)
Dim dv as DataView
'把执行命令后形成的数据加入到数据视图中
dv=new DataView(ds.Tables(table))
dv.Sort = SortField
' CreateDataView 函数返回值 dv , 数据视图
CreateDataView=dv
End Function
Sub Page_Load(sender As Object, e As EventArgs)
If Not IsPostBack Then
If SortField = "" Then
SortField = "id"
End If
BindGrid
End If
End Sub
'排序时调用的函数 MyDataGrid_Sort
Sub MyDataGrid_Sort(sender As Object, e As DataGridSortCommandEventArgs)
SortField = e.SortField
BindGrid
End Sub
'数据绑定函数 BindGrid
Sub BindGrid( )
MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
MyDataGrid.DataBind
End Sub
</script>

```

```
<title>在 DataGrid 中自定排序
</title>
<body>
<h3><font face="Verdana">在 DataGrid 中自定排序
</font>
</h3>
    <form runat=server>
<!--布置 DataGrid 控件，显示数据-->
<ASP:DataGrid
    id="MyDataGrid"
    runat="server"
    BorderColor="black"
    BorderWidth="1"
    CellPadding="3"
    Font-Name="Verdana"
    Font-Size="8pt"
    HeaderStyle-BackColor="#cccccc"
    HeaderStyle-ForeColor="black"
    AutoGenerateColumns="false"
    <!--设定可以排序模式-->
    AllowSorting="true"
    <!--排序时调用 MyDataGrid_Sort -->
    OnSortCommand="MyDataGrid_Sort"
    >
    <!--设定各个列的属性-->
<property name="Columns">
<!--可以排序的 -->
<asp:BoundColumn
    HeaderText="序号可排序"
    DataField="id"
    SortField="id"
/>
<!--不可以排序-->
<asp:BoundColumn
    HeaderText="姓名不可排序"
    DataField="name"
/>
<asp:BoundColumn
    HeaderText="性别不排序"
```

```

        DataField="sex"
    />
    <!--使用模板来定义列-->
    <asp:TemplateColumn
        SortField = "college">
        <template name="HeaderTemplate"> 按学院排序-->
            <!--布置一个图片按钮，来对 college 数据列排序-->
            <asp:ImageButton
                id=SortButton1
                runat="server"
                CommandName="sort"
                CommandArgument="college"
                ImageUrl=" ../images/arrow1.gif"
            />
        </template>
        <template name="ItemTemplate">
            <!--布置 label 控件来显示 college 列的数据-->
            <asp:Label
                runat="server"
                Text='<%# DataBinder.Eval(Container, "DataItem.college") %>'
            />
        </template>
    </asp:TemplateColumn>
</property>
</asp:DataGrid>
</form></body></html>

```

实现同样的功能，使用 C#语言编写的程序 ch05/test5_22c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
    string SortField;
    ICollection CreateDataview(string Db,string table) {
        string connstr;
        //连接数据库
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
        ADODatasetCommand cmd;

```

```

cmd=new ADODataSetCommand("Select id,name,sex,college,class from "+table,connstr);
DataSet ds = new DataSet( );
cmd.FillDataSet(ds,table);
    //把执行 SQL 命令得到的结果放入到 DataView 中
    DataView dv = new DataView(ds.Tables[table]);
dv.Sort = SortField;
    return dv;
}
//页面被浏览时所调用的程序 Page_Load
void Page_Load(Object sender, EventArgs e) {
    if (!IsPostBack) {
        if (SortField == "") {
            SortField = "id";
        }
        BindGrid( );
    }
}
//排序程序 MyDataGrid_Sort
protected void MyDataGrid_Sort(Object sender, DataGridSortCommandEventArgs e) {
    SortField = (string)e.SortField;
    Response.Write(SortField);
    BindGrid( );
}
//绑定数据 BindGrid
public void BindGrid( ) {
    MyDataGrid.DataSource = CreateDataview("sample.mdb","new");
    MyDataGrid.DataBind( );
}
</script>
<title>在 DataGrid 中自定义排序</title>
<body><h3>
在 DataGrid 中自定义排序
</h3>
    <form runat=server>
    <!--布置 DataGrid 控件，显示数据-->
    <ASP:DataGrid
        id="MyDataGrid"
        runat="server"
        BorderColor="black"

```

```
        BorderWidth="1"
        CellPadding="3"
        Font-Name="Verdana"
        Font-Size="8pt"
        HeaderStyle-BackColor="#cccccc"
        HeaderStyle-ForeColor="black"
        AutoGenerateColumns="false"
        <!--允许排序-->
        AllowSorting="true"
        <!--排序时调用的程序 MyDataGrid_Sort -->
        OnSortCommand="MyDataGrid_Sort"
    >
    <!--设定各个列的属性-->
    <property name="Columns">
    <!-- 可以排序的 -->
    <asp:BoundColumn
        HeaderText="序号可排序"
        DataField="id"
        SortField="id"
    />
    <!-- 不可以排序-->
    <asp:BoundColumn
        HeaderText="姓名不可排序"
        DataField="name"
    />
    <asp:BoundColumn
        HeaderText="性别不排序"
        DataField="sex"
    />
    <!--使用模板，来对 college 列排序-->
    <asp:TemplateColumn
        SortField = "college">
        <template name="HeaderTemplate"> 按学院排序-->
        <!--布置一个 ImageButton 控件来链接排序-->
    <asp:ImageButton
        id=SortButton1
        runat="server"
        CommandName="sort"
        CommandArgument="college"
```

```

        ImageUrl=" ../images/arrow1.gif"
    />
</template>
<template name="ItemTemplate">
    <!--使用 label 来显示 college 列的数据-->
    <asp:Label
runat="server"
Text='<%# DataBinder.Eval(Container, "DataItem.college") %>'
    />
</template>
</asp:TemplateColumn>
</property>
</asp:DataGrid>
</form></body></html>

```

自定义数据排序的运行结果如图 5-30 和 5-31 所示。



图 5-30 在 DataGrid 中自定义排序以学院排序时的运行结果

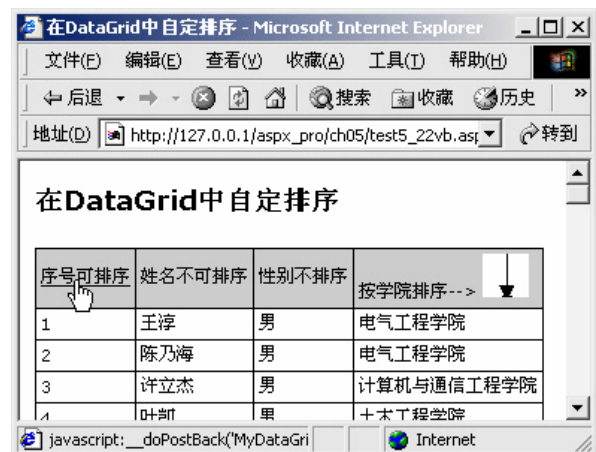


图 5-31 在 DataGrid 中自定义排序以序号排序时的运行结果

5.9.10 使用 DataGrid Web 控件分页显示数据

举例十：使用 DataGrid Web 控件分页显示数据。在 DataGrid 中分页显示，所有使用过 asp 的人都知道，分页是一件麻烦的事情，而在 ASP.Net 中却十分简单。下面来看使用 VB 语言编写的程序 ch05/test5_23vb.aspx，其源代码如下：

```
<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    ' 使用 ICollection 建立数据视图
    Function CreateDataView(DB as string,table as string) As ICollection
dim connstr as string
'建立数据库连接
        connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
        dim cmd as ADODatasetCommand
        cmd=new ADODatasetCommand("Select id,name,sex,college from "& table &" order by id
            DESC",Connstr )
        dim ds as DataSet = new DataSet( )
        '将数据加入到视图中去
        cmd.FillDataSet(ds,table)
        Dim dv as DataView
        dv=new DataView(ds.Tables(table))
        '函数 CreateDataView 返回值 dv
        CreateDataView=dv
    End Function
    '页面被浏览是所调用的程序 Page_Load
    Sub Page_Load(sender As Object, e As EventArgs)
        If chk1.Checked Then
            MyDataGrid.PagerStyle.Mode = PagerMode.NumericPages
        Else
            MyDataGrid.PagerStyle.Mode = PagerMode.NextPrev
        End If
        BindGrid
    End Sub
    '分页时所调用的程序 MyDataGrid_Page
```

```
Sub MyDataGrid_Page(sender As Object, e As DataGridPageChangedEventArgs)
    BindGrid
End Sub
Sub BindGrid( )
    MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
    MyDataGrid.DataBind
    ShowStats
End Sub
Sub ShowStats( )
    lblCurrentIndex.Text = "当前页序号是：" & MyDataGrid.CurrentPageIndex
    lblPageCount.Text = "页面总数：" & MyDataGrid.PageCount
End Sub
</script>
<title>在 DataGrid 中分页显示
</title>
<body>
<h3>
<font face="Verdana">在 DataGrid 中分页显示
</font>
</h3>
<form runat=server>
<!-- 布置 DataGrid 控件来显示数据-->
    <ASP:DataGrid
        id="MyDataGrid"
        runat="server"
        AllowPaging="True"
    <!-- 设定每页显示的数据数目为 10-->
        PageSize="10"
        PagerStyle-Mode="NumericPages"
        PagerStyle-HorizontalAlign="Right"
    <!-- 建立数据页面之间的连接-->
        PagerStyle-NextPageText="下一页"
        PagerStyle-PrevPageText="上一页"
        OnPageIndexChanged="MyDataGrid_Page"
        BorderColor="black"
        BorderWidth="1"
        GridLines="Both"
        CellPadding="3"
        CellSpacing="0"
```

```

        Font-Name="Verdana"
        Font-Size="8pt"
        HeaderStyle-BackColor="#aaaadd"
        AlternatingItemStyle-BackColor="#eeeeee"
    />
<p>
<!--布置一个复选框，来让客户选定分页模式-->
<asp:Checkbox
    id="chk1"
    runat="server"
    Text="以分页显示"
    Font-Name="Verdana"
    Font-Size="8pt"
    AutoPostBack="true"
    />
<p>
<table bgcolor="#eeeeee" cellpadding="6">
    <tr>
    <td nowrap>
    <font face="Verdana" size="-2">
        <!--布置 label 控件显示当前页数和总的页数-->
        <asp:Label
            id="lblCurrentIndex"
            runat="server"
            />
        <br>
        <asp:Label
            id="lblPageCount"
            runat="server"
            />
        <br>
    </font>
    </td>
    </tr>
    </table>
</form>
</body>
</html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_23c.aspx 的源代码如下：

```
<%@ Page language="C#" Debug="true"%>
<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
    ICollection CreateDataview(string Db,string table) {
        string connstr;
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
        ADODatasetCommand cmd;
        cmd=new ADODatasetCommand("Select id,name,sex,college,class from "+table,connstr);
        DataSet ds = new DataSet( );
        cmd.FillDataSet(ds,table);
        DataView dv = new DataView(ds.Tables[table]);
        return dv;
    }
    void Page_Load(Object sender, EventArgs e) {
        if (chk1.Checked) {
            MyDataGrid.PagerStyle.Mode=PagerMode.NumericPages;
        }
        else {
            MyDataGrid.PagerStyle.Mode=PagerMode.NextPrev;
        }
        BindGrid( );
    }
    void MyDataGrid_Page(Object sender, DataGridPageChangedEventArgs e) {
        BindGrid( );
    }
    void BindGrid( ) {
        MyDataGrid.DataSource = CreateDataview("sample.mdb","new");
        MyDataGrid.DataBind( );
        ShowStats( );
    }
    void ShowStats( ) {
        lblCurrentIndex.Text = "当前页是 : " + MyDataGrid.CurrentPageIndex;
        lblPageCount.Text = "页的总数为 : " + MyDataGrid.PageCount;
    }
}
```

```
</script><title>在 DataGrid 中分页显示 </title> <body>
```

```
<h3>
```

在 DataGrid 中分页显示

```
</h3>
```

```
<form runat=server>
```

```
<ASP:DataGrid
```

```
id="MyDataGrid"
```

```
runat="server"
```

```
AllowPaging="True"
```

```
PageSize="10"
```

```
PagerStyle-Mode="NumericPages"
```

```
PagerStyle-HorizontalAlign="Right"
```

```
PagerStyle-NextPageText="下一页"
```

```
PagerStyle-PrevPageText="上一页"
```

```
OnPageIndexChanged="MyDataGrid_Page"
```

```
BorderColor="black"
```

```
BorderWidth="1"
```

```
GridLines="Both"
```

```
CellPadding="3"
```

```
CellSpacing="0"
```

```
Font-Name="Verdana"
```

```
Font-Size="8pt"
```

```
HeaderStyle-BackColor="#aaaadd"
```

```
AlternatingItemStyle-BackColor="#eeeeee"
```

```
/>
```

```
<p>
```

```
<asp:Checkbox
```

```
id="chk1"
```

```
runat="server"
```

```
Text="以分页显示"
```

```
Font-Name="Verdana"
```

```
Font-Size="8pt"
```

```
AutoPostBack="true"
```

```
/>
```

```
<p>
```

```
<table bgcolor="#eeeeee" cellpadding="6">
```

```
<tr>
```

```
<td nowrap>
```

```
<font face="Verdana" size="-2">
```

```

<asp:Label
    id="lblCurrentIndex"
    runat="server"
/>
<br>
<asp:Label
    id="lblPageCount"
    runat="server"
/> <br>
</font> </td></tr></table>

</form>

</body>
</html>

```

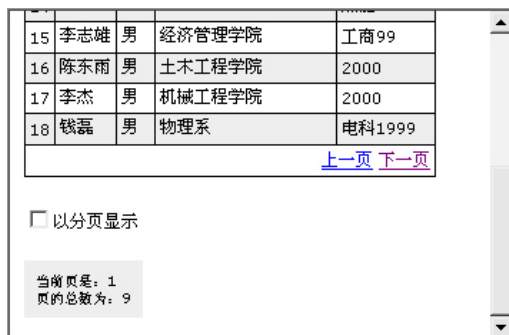


图 5-32 在 DataGrid 中分页显示运行结果一

程序运行结果如图 5-32 和 5-33 所示。



图 5-33 在 DataGrid 中分页显示运行结果二

5.9.11 使用 DataGrid Web 控件详细地数据分页

举例十一：在数据表格里详细地分页。

使用 VB 语言编写的程序 ch05/test5_24vb.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>

```

```

<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    ' 使用 ICollection 建立数据视图
    Function CreateDataView(DB as string,table as string) As ICollection
dim connstr as string
'连接数据库
        connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
        dim cmd as ADODatasetCommand
cmd=new ADODatasetCommand("Select id,name,sex,college,class from "& table,Connstr )
        dim ds as DataSet = new DataSet( )
        cmd.FillDataSet(ds,table)
        Dim dv as DataView
            '在数据视图 DataView 中填充数据
            dv=new DataView(ds.Tables(table))
            CreateDataView=dv
    End Function
'在页面被浏览时所调用的程序 Page_Load
    Sub Page_Load(sender As Object, e As EventArgs)
        If chk1.Checked Then
            '当复选框被选中时，分页显示 1，2，3，...
            MyDataGrid.PagerStyle.Visible = True
        Else
            MyDataGrid.PagerStyle.Visible = False
        End If
        BindGrid
    End Sub
'按下连接时所调用的程序 PagerButtonClick
    Sub PagerButtonClick(sender As Object, e As EventArgs)
        Dim arg As String = sender.CommandArgument
        Select arg
            Case "next":
                If (MyDataGrid.CurrentPageIndex < (MyDataGrid.PageCount - 1)) Then
                    MyDataGrid.CurrentPageIndex += 1
                End If
            Case "prev":
                If (MyDataGrid.CurrentPageIndex > 0) Then

```

```

        MyDataGrid.CurrentPageIndex -= 1
    End If
    Case "last":
        MyDataGrid.CurrentPageIndex = (MyDataGrid.PageCount - 1)
    Case Else:
        '页数
        MyDataGrid.CurrentPageIndex = arg.ToInt32()
    End Select
    BindGrid
End Sub
Sub MyDataGrid_Page(sender As Object, e As DataGridPageChangedEventArgs)
    BindGrid
End Sub
'绑定数据
Sub BindGrid( )
    MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
    MyDataGrid.DataBind
    ShowStats
End Sub
Sub ShowStats( )
    lblCurrentIndex.Text = "当前页序号是：" & MyDataGrid.CurrentPageIndex
    lblPageCount.Text = "页面总数：" & MyDataGrid.PageCount
End Sub
</script>
<title>数据表格详细分页
</title>
<body>
<h3><font face="Verdana">数据表格详细分页
</font>
</h3>
<form runat=server>
<!--布置 DataGrid 控件，来显示数据-->
    <ASP:DataGrid
        id="MyDataGrid"
        runat="server"
        AllowPaging="True"
        PageSize="10"
        PagerStyle-Mode="NumericPages"
        PagerStyle-HorizontalAlign="Right"

```

```
OnPageIndexChanged="MyDataGrid_Page"
BorderColor="black"
BorderWidth="1"
GridLines="Both"
CellPadding="3"
CellSpacing="0"
Font-Name="Verdana"
Font-Size="8pt"
HeaderStyle-BackColor="#aaaadd"
AlternatingItemStyle-BackColor="#eeeeee"
/>
<p>
<!--使用 LinkButton 控件，来建立各个数据页面之间的关系-->
    <asp:LinkButton
    id="btnPrev"
    runat="server"
        Text="上一页"
        CommandArgument="prev"
        ForeColor="navy"
        Font-Name="verdana" Font-size="8pt"
        OnClick="PagerButtonClick"
    />
    <asp:LinkButton
    id="btnNext"
    runat="server"
        Text="下一页"
        CommandArgument="next"
        ForeColor="navy"
        Font-Name="verdana" Font-size="8pt"
        OnClick="PagerButtonClick"
    />
    <!--使用 LinkButton 来实现到第六页的连接-->
    <asp:LinkButton
    id="btnPage6"
    runat="server"
        Text="到第六页"
        CommandArgument="6"
        ForeColor="navy"
        Font-Name="verdana" Font-size="8pt"
```

```
        OnClick="PagerButtonClick"
    />
    <asp:LinkButton
    id="btnFirst"
    runat="server"
        Text="到首页"
        CommandArgument="0"
        ForeColor="navy"
        Font-Name="verdana" Font-size="8pt"
        OnClick="PagerButtonClick"
    />
    <asp:LinkButton
    id="btnLast"
    runat="server"
        Text="到尾页"
        CommandArgument="last"
        ForeColor="navy"
        Font-Name="verdana" Font-size="8pt"
        OnClick="PagerButtonClick"
    />
    <p>
    <asp:Checkbox
    id="chk1"
    runat="server"
        Text="显示分页"
        Font-Name="Verdana"
        Font-Size="8pt"
        AutoPostBack="true"
    />
    <p>
    <table bgcolor="#eeeeee" cellpadding="6">
    <tr>
    <td nowrap>
    <font face="Verdana" size="-2">
    <!--显示当前的页面和总的页数-->
    <asp:Label
    id="lblCurrentIndex"
    runat="server"
    />
```

```

        <br>
        <asp:Label
            id="lblPageCount"
            runat="server"
        />
        <br>
    </font>
</td>
</tr>
</table>
</form>
</body>
</html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_24c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
    //使用 ICollection 来建立数据视图
    ICollection CreateDataView(string Db,string table) {
        string connstr;
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
        ADODatasetCommand cmd;
        cmd=new ADODatasetCommand("Select id,name,sex,college,class from "+table,connstr);
        DataSet ds = new DataSet( );
        cmd.FillDataSet(ds,table);
        //把数据填充到数据视图里面去
        DataView dv = new DataView(ds.Tables[table]);
        return dv;
    }
    void Page_Load(Object sender, EventArgs e) {
        if (chk1.Checked) {
            MyDataGrid.PagerStyle.Visible=true;
        }
        else {
            MyDataGrid.PagerStyle.Visible=false;
        }
    }
}

```

```
    }
    BindGrid( );
}

void PagerButtonClick(Object sender, EventArgs e) {
    //使用客户界面分页
    String arg = ((LinkButton)sender).CommandArgument;
    switch(arg)
    {
        case ("next"):
            if (MyDataGrid.CurrentPageIndex < (MyDataGrid.PageCount - 1))
                MyDataGrid.CurrentPageIndex ++;
            break;
        case ("prev"):
            if (MyDataGrid.CurrentPageIndex > 0)
                MyDataGrid.CurrentPageIndex --;
            break;
        case ("last"):
            MyDataGrid.CurrentPageIndex = (MyDataGrid.PageCount - 1);
            break;
        default:
            //页数
            MyDataGrid.CurrentPageIndex = arg.ToInt32();
            break;
    }
    BindGrid( );
}

void MyDataGrid_Page(Object sender, DataGridPageChangedEventArgs e) {
    //当前页码的代号已经有了
    BindGrid( );
}

void BindGrid( ) {
    MyDataGrid.DataSource = CreateDataview("sample.mdb","new");
    MyDataGrid.DataBind( );
    ShowStats( );
}

void ShowStats( ) {
    lblCurrentIndex.Text = "当前页是 : " + MyDataGrid.CurrentPageIndex;
    lblPageCount.Text = "总页数是 : " + MyDataGrid.PageCount;
}
```

```
</script>
<title>数据表格详细分页
</title>
<body>
<h3>
<font face="Verdana">数据表格详细分页
</font>
</h3>
    <form runat=server>
<!--布置 DataGrid 控件，来显示数据-->
        <ASP:DataGrid
            id="MyDataGrid"
            runat="server"
                AllowPaging="True"
                PageSize="10"
                PagerStyle-Mode="NumericPages"
                PagerStyle-HorizontalAlign="Right"
                OnPageIndexChanged="MyDataGrid_Page"
                BorderColor="black"
                BorderWidth="1"
                GridLines="Both"
                CellPadding="3"
                CellSpacing="0"
                Font-Name="Verdana"
                Font-Size="8pt"
                HeaderStyle-BackColor="#aaaadd"
                AlternatingItemStyle-BackColor="#eeeeee"
            />
    <p>
<!--使用 LinkButton 控件，来建立各个数据页面之间的关系-->
        <asp:LinkButton
            id="btnPrev"
            runat="server"
                Text="上一页"
                CommandArgument="prev"
                ForeColor="navy"
                Font-Name="verdana" Font-size="8pt"
                OnClick="PagerButtonClick"
            />
```

```
<asp:LinkButton
id="btnNext"
runat="server"
    Text="下一页"
    CommandArgument="next"
    ForeColor="navy"
    Font-Name="verdana" Font-size="8pt"
    OnClick="PagerButtonClick"
/>
<!--使用 LinkButton 来实现到第六页的连接-->
<asp:LinkButton
id="btnPage6"
runat="server"
    Text="到第六页"
    CommandArgument="6"
    ForeColor="navy"
    Font-Name="verdana" Font-size="8pt"
    OnClick="PagerButtonClick"
/>
<asp:LinkButton
id="btnFirst"
runat="server"
    Text="到首页"
    CommandArgument="0"
    ForeColor="navy"
    Font-Name="verdana" Font-size="8pt"
    OnClick="PagerButtonClick"
/>
<asp:LinkButton
id="btnLast"
runat="server"
    Text="到尾页"
    CommandArgument="last"
    ForeColor="navy"
    Font-Name="verdana" Font-size="8pt"
    OnClick="PagerButtonClick"
/>
<p>
<asp:Checkbox
```

```

        id="chk1"
        runat="server"
        Text="显示分页"
        Font-Name="Verdana"
        Font-Size="8pt"
        AutoPostBack="true"
    />
</p>
<table bgcolor="#eeeeee" cellpadding="6">
<tr>
<td nowrap>
<font face="Verdana" size="-2">
        <!--显示当前的页面和总的页数-->
        <asp:Label
            id="lblCurrentIndex"
            runat="server"
        />
        <br>
        <asp:Label
            id="lblPageCount"
            runat="server"
        />
        <br>
        </font>
    </td>
</tr>
</table>
</form>
</body>
</html>

```

运行程序结果如图 5 - 34 所示。



图 5 - 34 数据表格详细分页运行结果

5.10 DataList Web 控件

数据访问的形式除了 DataGrid 之外, 还有 DataList 和 Repeater, 用户可以灵活地使用上述几个控件。使用 DataList Web 控件, 需要提供模板, 即用控件 Template。属性 DataSource 可以赋予一个 ICollection 对象。ItemTemplate 控件指定每一个记录显示的形式, 例如

```
<asp:DataList id="DataList1" runat="server">
<template name="ItemTemplate">
    <%# DataBinder.Eval(Container.DataItem, "name") %>
</template>
</asp:DataList>
```

另外，还可以使用 HeaderTemplate 和 FooterTemplate，它们来自定义显示数据表格的头部和底部。DataList 有更强劲的控制能力，RepeatDirection 用来指定显示数据的方向。

插入 LinkButton 控件可以引起 Select 事件，DataList 中可设 OnClick 属性指定事件 Button1_Click。例如

```
<asp:LinkButton
    id=Button1
    Text="刷新数据列表"
    OnClick="Button1_Click"
    runat="server"
/>
```

使用语法如下：

```
<asp:DataList
    id="DataList1"
    runat="server"
    CellPadding="pixels"
    CellSpacing="pixels"
    DataKeyField="DataSourceKeyField"
    DataSource='<% databindingexpression %>'
    ExtractTemplateRows="To be supplied"
    GridLines="None|Horizontal|Vertical|Both"
    RepeatColumns="ColumnCount"
    RepeatDirection="Vertical|Horizontal"
    RepeatLayout="Flow|Table"
    ShowFooter="True|False"
    ShowHeader="True|False"

    OnCancelCommand="OnCancelCommandMethod"
    OnDeleteCommand="OnDeleteCommandMethod"
    OnEditCommand="OnEditCommandMethod"
    OnItemCommand="OnItemCommandMethod"
    OnItemCreated="OnItemCreatedMethod"
    OnUpdateCommand="OnUpdateCommandMethod"

    <template name="Header">
```

```

        Header template HTML
    </template >
    <template name="Item">
        Item template HTML
    </template >
    <template name="AlternatingItem">
        Alternating item template HTML
    </template >
    <template name="EditItem">
        Edited item template HTML
    </template >
    <template name="SelectedItem">
        Selected item template HTML
    </template >
    <template name="Separator">
        Separator template HTML
    </template >
    <template name="Footer">
        Footer template HTML
    </template >

    AlternatingItemStyle-property="value"
    EditItemStyle-property="value"
    FooterStyle-property="value"
    HeaderStyle-property="value"
    ItemStyle-property="value"
    SelectedItemStyle-property="value"
    SeparatorStyle-property="value"
/>

```

在使用 DataGrid Web 控件时，通过设定的模板，可以制定出数据表格显示的样式。同时 DataGrid Web 控件还支持几种命令，通过在 DataGrid Web 控件中布置命令来实现许多功能。

实例一：使用 DataList Web 控件浏览数据。使用 VB 语言编写的程序 ch05/test5_25vb.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    ' 使用 ICollection 建立数据视图
    Function CreateDataView(DB as string,table as string) As ICollection
        dim connstr as string
        '连接数据库
        connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
        dim cmd as ADODatasetCommand
        cmd=new ADODatasetCommand("Select TOP 50 id,name from "& table,Connstr )
        dim ds as DataSet = new DataSet( )
        cmd.FillDataSet(ds,table)
        Dim dv as DataView
        dv=new DataView(ds.Tables(table))
        '函数返回值 dv
        CreateDataView=dv
    End Function
    Sub Page_Load(sender As Object, e As EventArgs)
        If Not IsPostBack Then
            DataList1.DataSource =CreateDataView("sample.mdb","new")
            DataList1.DataBind
        End If
    End Sub
    '客户按下刷新数据列表时，所调用的程序 Button1_Click
    Sub Button1_Click(sender As Object, e As EventArgs)
        If DropDown1.SelectedIndex = 0
            '使用 DataList 控件的 RepeatDirection 属性数据连确定显示的方向
            DataList1.RepeatDirection = RepeatDirection.Horizontal
        Else
            DataList1.RepeatDirection = RepeatDirection.Vertical
        End If
        If DropDown2.SelectedIndex = 0
            ' 使用 DataList 控件的 RepeatLayout 来确定数据显示的排版方式
            DataList1.RepeatLayout = RepeatLayout.Table
        Else
            DataList1.RepeatLayout = RepeatLayout.Flow
        End If
        If Check1.Checked And DataList1.RepeatLayout = RepeatLayout.Table Then
            DataList1.BorderWidth = Unit.Pixel(1)
            DataList1.GridLines = GridLines.Both
        End If
    End Sub

```

```

        Else
            DataList1.BorderWidth = Unit.Pixel(0)
            DataList1.GridLines = GridLines.None
        End If
        DataList1.RepeatColumns = DropDown3.SelectedIndex + 1
    End Sub
</script>
<title>asp:DataList 举例
</title>
<body>
<h3>
<font face ="Verdana">asp:DataList 举例
</font>
</h3>
    <form runat = server>
    <font face ="Verdana" size ="-1">
        <!--布置 DataList 控件来显示数据-->
        <asp:DataList
            id ="DataList1 "
            runat ="server"
            BorderColor ="black"
            CellPadding ="3"
            Font-Name ="Verdana"
            Font-Size ="8pt"
            HeaderStyle-BackColor ="#aaaadd"
            AlternatingItemStyle-BackColor ="Gainsboro"
            >
            <template name ="HeaderTemplate"> 人才列表
            </template>
            <!--使用模板来设定数据显示的样式-->
            <template name ="ItemTemplate">
            <%# DataBinder.Eval(Container.DataItem, "name") %>
            </template>
            </asp:DataList>
            <p>
            <hr noshade align ="left" width ="300px">
排序显示方向:
            <!--布置下拉选单提供数据排列方式的选项-->
            <asp:DropDownList

```

```
id=DropDown1
runat="server" >
<asp:ListItem>水平
</asp:ListItem>
<asp:ListItem>竖直
</asp:ListItem>
</asp:DropDownList><br>
以表格显示吗:
<asp:DropDownList
    id=DropDown2
    runat="server"
    >
    <asp:ListItem>表格
    </asp:ListItem>
    <asp:ListItem>
        Flow
    </asp:ListItem>
</asp:DropDownList>
<br>
显示列数:
<asp:DropDownList
    id=DropDown3
    runat="server" >
    <asp:ListItem>1
    </asp:ListItem>
    <asp:ListItem>2
    </asp:ListItem>
    <asp:ListItem>3
    </asp:ListItem>
    <asp:ListItem>4
    </asp:ListItem>
    <asp:ListItem>5
    </asp:ListItem>
</asp:DropDownList>
<br>
显示边框吗:
<asp:CheckBox
    id=Check1
    runat="server"
```

```

        />
        <p>
        <!--布置 LinkButton 控件，当客户按下时，调用 Button1_Click -->
        <asp:LinkButton
            id=Button1
            Text="刷新数据列表"
            OnClick="Button1_Click"
            runat="server"
        />
    </font>
</form>
</body>
</html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_25c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
    //使用 Icollection 建立数据视图
    ICollection CreateDataview(string Db,string table) {
        string connstr;
        //连接数据库
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
        ADODataSetCommand cmd;
        cmd=new ADODataSetCommand("Select TOP 50 id,name from "+table,connstr);
        DataSet ds = new DataSet( );
        cmd.FillDataSet(ds,table);
        DataView dv = new DataView(ds.Tables[table]);
        return dv;
    }
    void Page_Load(Object Sender, EventArgs E) {
        if (!IsPostBack) {
            DataList1.DataSource = CreateDataview("sample.mdb","new");
            DataList1.DataBind( );
        }
    }
}

```

//当客户按下刷新数据列表时所调用的程序 Button1_Click

```
void Button1_Click(Object Sender, EventArgs E) {
    if (DropDown1.SelectedIndex == 0)
        DataList1.RepeatDirection = RepeatDirection.Horizontal;
    else
        DataList1.RepeatDirection = RepeatDirection.Vertical;
    if (DropDown2.SelectedIndex == 0)
        DataList1.RepeatLayout = RepeatLayout.Table;
    else
        DataList1.RepeatLayout = RepeatLayout.Flow;
    DataList1.RepeatColumns=DropDown3.SelectedIndex+1;
    if ((Check1.Checked==true) && (DataList1.RepeatLayout == RepeatLayout.Table)) {
        DataList1.BorderWidth = Unit.Pixel(1);
        DataList1.GridLines = GridLines.Both;
    }
    else {
        DataList1.BorderWidth = Unit.Pixel(0);
        DataList1.GridLines = GridLines.None;
    }
}
}
</script>
<title>asp:DataList 举例
</title>
<body>
<h3>asp:DataList 举例
</h3>
<form runat=server>
<font face="Verdana" size="-1">
<!--布置 DataList 控件，来显示数据-->
<asp:DataList
    id="DataList1"
    runat="server"
    BorderColor="black"
    CellPadding="3"
    Font-Name="Verdana"
    Font-Size="8pt"
    HeaderStyle-BackColor="#aaaadd"
    AlternatingItemStyle-BackColor="Gainsboro"
>
```

```

<template name="HeaderTemplate"> 人才列表
</template>
<template name="ItemTemplate">
    <!--使用 DataBinder 来显示数据-->
    <%# DataBinder.Eval(Container.DataItem, "name") %>
</template>
</asp:DataList>
<p>
<hr noshade align="left" width="300px">
排序显示方向:
<asp:DropDownList
    id=DropDown1
    runat="server" >
    <asp:ListItem>水平
</asp:ListItem>
    <asp:ListItem>竖直
</asp:ListItem>
</asp:DropDownList>
<br>
以表格显示吗:
<asp:DropDownList
    id=DropDown2
    runat="server" >
    <asp:ListItem>表格
</asp:ListItem>
    <asp:ListItem>
        Flow
    </asp:ListItem>
</asp:DropDownList>
<br>
显示列数:
<asp:DropDownList
    id=DropDown3
    runat="server">
    <asp:ListItem>1
</asp:ListItem>
    <asp:ListItem>2
</asp:ListItem>
    <asp:ListItem>3

```

```

</asp:ListItem>
<asp:ListItem>4
</asp:ListItem>
<asp:ListItem>5
</asp:ListItem>
</asp:DropDownList>
<br>
显示边框吗:
<asp:CheckBox
    id=Check1
    runat="server"
/>
<p>
    <!--布置 LinkButton 按钮，当客户按下时调用 Button1_Click -->
<asp:LinkButton
    id=Button1
    Text="刷新数据列表"
    OnClick="Button1_Click"
    runat="server"
/>
</font>
</form>
</body>
</html>

```

运行结果如图 5 - 35 所示。

实例二：在 DataList 中直接显示详细的情况。使用 VB 语言编写的程序 ch05/test5_26vb.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">

```

‘使用 ICollection 来建立数据视图

Function CreateDataView(DB as string,table as string) As ICollection

dim connstr as string

‘连接数据库

```
connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
```



图 5 - 35 DataList 控件举例运行结果

```

        dim cmd as ADODatasetCommand
        cmd=new ADODatasetCommand("Select TOP 5 id,name,sex,college,class from "&
            table,Connstr )
        dim ds as DataSet = new DataSet( )
        cmd.FillDataSet(ds,table)
        Dim dv as DataView
        dv=new DataView(ds.Tables(table))
        CreateDataView=dv
    End Function
    '当页面被浏览时执行的程序 Page_Load
    Sub Page_Load(sender As Object, e As EventArgs)
        If Not IsPostBack Then
            BindList
        End If
    End Sub
    Sub BindList
        DataList1.DataSource= CreateDataView("sample.mdb","new")
        DataList1.DataBind
    End Sub
    Sub DataList_ItemCommand(sender As Object, e As DataListCommandEventArgs)
        Dim cmd As String = e.CommandSource.CommandName
        If cmd = "select" Then
            DataList1.SelectedIndex = e.Item.ItemIndex
        End If
        BindList
    End Sub
</script>
<title>在 DataList 中直接显示详情
</title>
</head>
<body>
<h3><font face="Verdana">在 DataList 中直接显示详情</font></h3>
    <form runat=server>
        <font face="Verdana" size="-1">
            <!--布置 DataList 控件来显示数据-->
            <asp:DataList
id="DataList1"
runat="server"
        BorderColor="black"

```

```
BorderWidth="1"
GridLines="Both"
CellPadding="3"
Font-Name="Verdana"
Font-Size="8pt"
Width="150px"
HeaderStyle-BackColor="#aaaadd"
AlternatingItemStyle-BackColor="Gainsboro"
SelectedItemStyle-BackColor="yellow"
OnItemCommand="DataList_ItemCommand"
>
    <template name="HeaderTemplate">
        人才列表
    </template>
    <template name="ItemTemplate">
        <!-- 布置 LinkButton 按下详细说明时显示详细数据-->
        <asp:LinkButton
            id="button1"
            runat="server"
            Text="详细说明"
            CommandName="select"
        />
        <%=# DataBinder.Eval(Container.DataItem, "name") %>
    </template>
    <template name="SelectedItemTemplate">
        姓名:
        <%=# Container.DataItem("name") %>
        <br>
        性别 :
        <%=# DataBinder.Eval(Container.DataItem, "sex") %>
        <br>
        学院 :
        <%=# DataBinder.Eval(Container.DataItem, "college") %>
        <br>
        班级 :
        <%=# DataBinder.Eval(Container.DataItem, "class") %>
        <br>
    </template>
</asp:DataList>
```

```

</font>
</form>
</body>
</html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_26c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
    //使用 Icollection 建立数据视图
    ICollection CreateDataview(string Db,string table) {
        string connstr;
        //连接数据库
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
        ADODatasetCommand cmd;
        cmd=new ADODatasetCommand("Select TOP 5 id,name,sex,college,class from "+table,connstr);
        DataSet ds = new DataSet( );
        cmd.FillDataSet(ds,table);
        DataView dv = new DataView(ds.Tables[table]);
        return dv;
    }
    void Page_Load(Object Sender, EventArgs e) {
        if (!IsPostBack)
            BindList( );
    }
    void BindList( ) {
        DataList1.DataSource= CreateDataview("sample.mdb","new");
        DataList1.DataBind( );
    }
    void DataList_ItemCommand(object Sender, DataListCommandEventArgs e) {
        string cmd = ((LinkButton)e.CommandSource).CommandName;
        if (cmd == "select")
            DataList1.SelectedIndex = e.Item.ItemIndex;
        BindList( );
    }
</script>

```

```
<title>DataList 中直接显示详
</title>
</head>
<body>
<h3><font face="Verdana">DataList 中直接显示详情
</font>
</h3>
    <form runat=server>
<font face="Verdana" size="-1">
    <!--布置 DataList 控件来显示数据-->
        <asp:DataList id="DataList1" runat="server"
            BorderColor="black"
            BorderWidth="1"
            GridLines="Both"
            CellPadding="3"
            Font-Name="Verdana"
            Font-Size="8pt"
            Width="150px"
            HeaderStyle-BackColor="#aaaadd"
            AlternatingItemStyle-BackColor="Gainsboro"
            SelectedItemStyle-BackColor="yellow"
            OnItemCommand="DataList_ItemCommand"
        >
            <template name="HeaderTemplate">
                人才列表
            </template>
        <template name="ItemTemplate">
            <!--布置 LinkButton 建立链接-->
                <asp:LinkButton
                    id="button1"
                    runat="server"
                    Text="详细说明"
                    CommandName="select"
                />
                <%=# DataBinder.Eval(Container.DataItem, "name") %>
            </template>
            <!--使用模板 template 来显示数据-->
                <template name="SelectedItemTemplate">
                    姓名:
```

```

<%# ((DataRowView)Container.DataItem)["name"] %>
<br>
性别:
<%# DataBinder.Eval(Container.DataItem, "sex") %>
<br>
学院:
<%# DataBinder.Eval(Container.DataItem, "college") %>
<br>
班级 :
<%# DataBinder.Eval(Container.DataItem, "class") %>
<br>
</template>
</asp:DataList>
</font>    </form>
</body></html>

```

运行结果如图 5 - 36 所示。

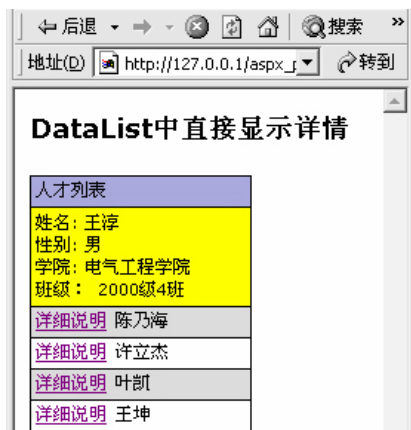


图 5 - 36 使用 DataList 直接显示详细情况运行结果

5.11 DropDownList Web 控件

DropDownList Web 控件的使用语法如下：

```

<asp:DropDownList id="DropDownList1" runat="server"
    DataSource="<% databindingexpression %>"
    DataTextField="DataSourceField"
    DataValueField="DataSourceField"
    AutoPostBack="True|False"
    OnSelectedIndexChanged="OnSelectedIndexChangedMethod"

```

```
/>
    <asp:Listitem value="value" selected="True|False">
        Text
    </asp:Listitem>
</asp:DropDownList>
```

实例一：通过使用 DropDownList Web 控件，可以在网页中布置一个单选的下拉选单。
使用 VB 语言编写的程序 ch05/test5_27vb.aspx 的源代码如下：

```
<%@ Page language="VB" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
    <script language="VB" runat="server">
        Sub SubmitBtn_Click(sender As Object, e As EventArgs)
            Label1.Text="您选择的是：" + DropDownList1.SelectedItem.Text
        End Sub
    </script>
<title>DropDownList 举例</title></head>
<body>
<h3><font face="Verdana">DropDownList 举例</font></h3>
<form runat=server>
<!--布置下拉选单-->
    <asp:DropDownList id=DropDown1 runat="server">
        <asp:ListItem>产品 1</asp:ListItem>
        <asp:ListItem>产品 2</asp:ListItem>
        <asp:ListItem>产品 3</asp:ListItem>
        <asp:ListItem>产品 4</asp:ListItem>
        <asp:ListItem>产品 5</asp:ListItem>
        <asp:ListItem>产品 6</asp:ListItem>
    </asp:DropDownList>
    <asp:button
text="提交"
OnClick="SubmitBtn_Click"
runat=server
/>
<p>
<asp:Label
id=Label1
font-name="Verdana"
```

```
font-size="10pt"
runat="server">
    请选择</asp:Label>
</form>
</body>
</html>
```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_27c.aspx 的源代码如下：

```
<%@ Page language="C#" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
    <script language="C#" runat="server">
        void SubmitBtn_Click(Object Sender, EventArgs e) {
            Label1.Text="您选择的是：" + DropDown1.SelectedItem.Text;
        }
    </script>
<title>DropDownList 举例</title></head>
<body>
<h3><font face="Verdana">DropDownList 举例</font></h3>
    <form runat=server>
        <asp:DropDownList id=DropDown1 runat="server">
            <asp:ListItem>产品 1</asp:ListItem>
            <asp:ListItem>产品 2</asp:ListItem>
            <asp:ListItem>产品 3</asp:ListItem>
            <asp:ListItem>产品 4</asp:ListItem>
            <asp:ListItem>产品 5</asp:ListItem>
            <asp:ListItem>产品 6</asp:ListItem>
        </asp:DropDownList>
        <asp:button text="提交"
OnClick="SubmitBtn_Click"
runat=server
/>
<p>
<asp:Label
id=Label1
font-name="Verdana"
font-size="10pt"
runat="server"
```

```
>
    请选择</asp:Label>
</form>
</body>
</html>
```



图 5 - 37 DropDownList Web 控件举例运行结果

运行结果如图 5 - 37 所示。

5.12 Hyperlink Web 控件

Hyperlink Web 控件可以用来链接网址，可以在文字上链接，也可以在图片上面链接。其使用语法如下：

```
<asp:Hyperlink id="Hyperlink1" runat="server"
    NavigateUrl="url"
    Text="HyperlinkText"
    ImageUrl="url"
    Target="window"
/>
or
<asp:Hyperlink id="Hyperlink1" runat="server"
    NavigateUrl="url"
    ImageUrl="url"
    Target="window"
/>
Text
</asp:Hyperlink>
```

使用 VB 语言编写的程序 ch05/test5_28vb.aspx (该例是在文字上面加入链接)，其源代码如下：

```
<%@ Page language="VB" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="VB" runat="server">
    Sub Page_Load(sender As Object, e As EventArgs)
        HyperLink1.NavigateUrl = "http://www.singnet.com.cn/"
    End Sub
</script>
<title>asp:hyperlink 举例</title>
<body>
```

```

<h3><font face="Verdana"> asp:hyperlink 举例</font>
</h3>
    <form runat=server>
        <p>
            <!--布置 hyperlink 控件，建立链接-->
            <asp:hyperlink id=HyperLink1 runat="server">
                到世纪星
            </asp:hyperlink>
        </form>
    </body>
</html>

```

使用 C# 语言编写的程序 ch05/test5_28c.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat=server>
    void Page_Load (Object sender, EventArgs e) {
        HyperLink1.NavigateUrl = "http://www.singnet.com.cn";
    }
</script>
<title>asp:hyperlink 举例</title>
<body>
<h3><font face="Verdana"> asp:hyperlink 举例</font>
</h3>

```

```

    <form runat=server>
        <p>
            <!--布置 hyperlink 控件，建立链接-->
            <asp:hyperlink id=HyperLink1 runat="server">
                到世纪星
            </asp:hyperlink>
        </form>
    </body>
</html>

```



图 5 - 38 Hyperlink Web 控件举例
运行结果

运行结果如图 5 - 38 所示。

实例二：使用图片链接网址。

使用 VB 语言编写的程序 ch05/test5_29vb.aspx 的源代码如下：

```

<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>

```

```

<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
  <script language="VB" runat="server">
'使用 ICollection 来建立数据视图
    Function CreateDataView(DB as string,table as string) As ICollection
        dim connstr as string
        connstr="DBQ=" & Server.MapPath(Db)&" ; Driver={Microsoft Access Driver (*.mdb)};"
        dim cmd as ADODatasetCommand
        cmd=new ADODatasetCommand("Select top 10 id,name from "& table,Connstr )
        dim ds as DataSet = new DataSet( )
        cmd.FillDataSet(ds,table)
        CreateDataView= new DataView(ds.Tables(table))
    End Function
    Sub Page_Load(sender As Object, e As EventArgs)
        MyRepeater.DataSource = CreateDataView("sample.mdb","new")
        MyRepeater.DataBind( )
    End Sub
  </script>
<title> asp:hyperlink 举例</title>
<body>
<h3><font face="Verdana"> asp:hyperlink 举例</font>
</h3>
  <p>
<form runat=server>
<!--使用 Repeater 控件来显示数据-->
    <asp:Repeater
id="MyRepeater"
runat="server">
        <template name="ItemTemplate">
            链接的是 :
            <!--布置 HyperLink 建立链接-->
<asp:HyperLink
id=HyperLink1
Text='<%# DataBinder.Eval(Container.DataItem, "name") %>' NavigateUrl='<%# DataBinder.Eval
(Container.DataItem,"id","detailspage.aspx?id={0}") %>' runat="server" />
        <p>
    </template>

```

```
</asp:Repeater>
</form>
</body>
</html>
```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_29c.aspx 的源代码如下：

```
<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
    //使用 Icollection 来建立数据视图
    ICollection CreateDataview(string Db,string table) {
        string connstr;
        connstr="DBQ="+Server.MapPath(Db)+";Driver={Microsoft Access Driver (*.mdb)};";
        ADODataSetCommand cmd;

        //链接数据库

        cmd=new ADODataSetCommand("Select id,name,sex,college,class from "+table,connstr);
        DataSet ds = new DataSet( );
        cmd.FillDataSet(ds,table);

        DataView dv = new DataView(ds.Tables[table]);

        //返回数据视图
        return dv;
    }

    void Page_Load(Object sender, EventArgs e) {
        MyRepeater.DataSource = CreateDataview("sample.mdb","new");
        MyRepeater.DataBind( );
    }
</script>
<title> asp:hyperlink 举例</title>
<body>
<h3><font face="Verdana"> asp:hyperlink 举例</font>
</h3>

<p>

<form runat=server>
    <!--布置 Repeater 控件来显示数据-->
        <asp:Repeater
id="MyRepeater"
```

```

runat="server">
    <template name="ItemTemplate">
        链接的是：
        <!--布置 HyperLink 控件来建立链接-->
        <asp:HyperLink
            id=HyperLink1
            Text='<%=# DataBinder.Eval(Container.DataItem, "name") %>' NavigateUrl='<%=# DataBinder.Eval
                (Container.DataItem,"id","detailspage.aspx?id={0}") %>' runat="server" />
        <p>
    </template>
</asp:Repeater>
</form>
</body>
</html>

```

运行结果如图 5 - 39 所示。



图 5 - 39 asp:hyperlink 举例二运行结果

5.13 Image Web 控件

Image Web 控件用来布置一个图片。其使用语法如下：

```

<asp:Image id="Image1" runat="server"
    ImageUrl="string"
    AlternateText="string"
    ImageAlign="NotSet|AbsBottom|AbsMiddle|BaseLine|Bottom|Left|Middle
        Right|TextTop|Top"
/>

```

下面的例子中，根据客户的选择而显示不同的图片。

使用 VB 语言编写的程序 ch05/test5_30vb.aspx 的源代码如下：

```

<html>
<head>
    <script language="VB" runat="server">
        Sub SubmitBtn_Click(sender As Object, e As EventArgs)
            Image1.ImageUrl = "../images/" & DropDown1.SelectedItem.Value
            Image1.AlternateText = DropDown1.SelectedItem.Text
        End Sub
    </script>
</head>
<body>
    <h3><font face="Verdana">简单 HtmlImage 举例</font>
</h3>
    <form runat="server">
        <asp:Image
            ID="Image1"
            ImageUrl="../images/banner1.gif"
            AlternateText="影音接触"
            runat="server"
        />
        <p>
            请选择图片:
            <asp:DropDownList
                id=DropDown1
                runat="server">
                    <asp:ListItem Value="banner1.gif">影音接触</asp:ListItem>
                    <asp:ListItem Value="banner2.gif">游戏</asp:ListItem>
                    <asp:ListItem Value="banner3.gif">贺卡</asp:ListItem>
                    <asp:ListItem Value="banner4.gif">世纪星</asp:ListItem>
                </asp:DropDownList>
            <asp:button
                text="应用"
                OnClick="SubmitBtn_Click"
                runat="server"
            />
        </form>
    </body>
</html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_30c.aspx 的源代码如下：

```

<html>

```

```
<head>
    <script language="C#" runat="server">
        void SubmitBtn_Click(Object sender, EventArgs e) {
            Image1.ImageUrl = "../images/" + DropDown1.SelectedItem.Value;
            Image1.AlternateText = DropDown1.SelectedItem.Text;
        }
    </script>
</head>
<body>
    <h3><font face="Verdana">简单 HtmlImage 举例</font>
</h3>
    <form runat="server">
        <asp:Image
            ID="Image1"
            ImageUrl="../images/banner1.gif"
            AlternateText="影音接触"
            runat="server"
        />
        <p>
            请选择图片:
            <asp:DropDownList
                id="DropDown1"
                runat="server">
                <asp:ListItem Value="banner1.gif">影音接触</asp:ListItem>
                <asp:ListItem Value="banner2.gif">游戏</asp:ListItem>
                <asp:ListItem Value="banner3.gif">贺卡</asp:ListItem>
                <asp:ListItem Value="banner4.gif">世纪星</asp:ListItem>
            </asp:DropDownList>
            <asp:button
                text="应用"
                OnClick="SubmitBtn_Click"
                runat="server"
            />
        </form>
    </body>
</html>
```

运行结果如图 5 - 40 所示。



图 5 - 40 Image Web 控件举例运行结果

5.14 ImageButton Web 控件

ImageButton Web 控件可以布置一个图片按钮。其使用语法如下：

```
<asp:ImageButton id="ImageButton1" runat="server"
    ImageUrl="string"
    Command="command"
    CommandArgument="commandArgument"
    OnClick="OnClickMethod"
/>
```

举例一：布置一个 ImageButton Web 控件，当客户按下图片按钮时，显示提示信息。

使用 VB 语言编写的程序 ch05/test5_31vb.aspx 的源代码如下：

```
<html>
<head>
    <script language="VB" runat="server">
        Sub ImageButton1_OnClick(sender As Object, e As ImageClickEventArgs)
            Label1.Text = "你按下了图片按钮"
        End Sub
    </script>
</head>
<body>
    <h3><font face="Verdana">ImageButton 举例</font>
</h3>
    <form runat="server">
        <asp:ImageButton
            id=Button1
```

```

        ImageUrl="../images/sample.jpg"
        BorderWidth="2px"
        onclick="ImageButton1_OnClick"
        runat="server"
    />
<asp:Label
    id=Label1
    runat=server
/>
</form>
</body>
</html>

```

实现同样的功能，使用 C# 程序语言编写的程序 ch05 / test5_31c.aspx，其源代码如下：

```

<html>
<head>
    <script language="C#" runat="server">
        void ImageButton1_OnClick(object Source, ImageClickEventArgs e) {
            Label1.Text="你按下了图片按钮";
        }
    </script>
</head>
<body>
    <h3><font face="Verdana">ImageButton 举例</font>
</h3>
    <form runat=server>
        <asp:ImageButton
            id=Button1
            ImageUrl="../images/sample.jpg"
            BorderWidth="2px"
            onclick="ImageButton1_OnClick"
            runat="server"
        />
        <asp:Label
            id=Label1
            runat=server />
    </form>
</body>
</html>

```

运行结果如图 5 - 41 所示。



图 5 - 41 ImageButton 举例运行结果

举例二：像 Image Map 一样使用 ImageButton，可以把图片按钮分成不同的区域。客户按下不同的区域，调用不同的处理程序。

使用 VB 语言编写的程序 ch05/test5_32vb.aspx 的源代码如下：

```
<html>
<head>
    <script language="VB" runat="server">
        Sub ImageButton1_OnClick(sender As Object, e As ImageClickEventArgs)
            '获得鼠标的位置
            Dim x As Integer = e.X
            Dim y As Integer = e.Y
            Label1.Text = "X: " + x.ToString
            Label2.Text = "Y: " + y.ToString
            Button1.ImageUrl = "../images/billg.gif "
            '根据鼠标的位置判断到底是那个部位被按下
            If x>20 And x<35 And y>72 And y<88 Then
                Label3.Text = "你按 Bill 的耳朵。"
            Else if x>42 And x<58 And y>48 And y<64 Then
                Label3.Text = "你按 Bill 的眼睛。"
            Else if x>62 And x<90 Then
                If y>46 And y<56 Then
                    Label3.Text = "你按 Bill 的眼睛。"
                Else if y<72 And y>60 Then
                    Label3.Text = "你按 Bill 的鼻子。"
                Else if y<92 And y>74 Then
                    Button1.ImageUrl = "../images/billg_tongue.gif"
                    Label3.Text = "yeah!...."
                Else
                    whoops( )
                End If
            Else if x>100 And x<108 And y>56 And y<72 Then
                Label3.Text = "你按 Bill 的耳朵。"
            Else
                whoops( )
            End If
        End Sub
        Sub whoops( )
            Label3.Text = "错误！应该按 Bill 的眼睛、耳朵、鼻子、或者嘴巴！"
        End Sub
    </script>
</head>
<body>
    <div>
        <img alt="Bill Clinton" data-bbox="180 180 250 250"/>
        <div>
            <div>X: <span></span> Y: <span></span>
            <div>你按 Bill 的耳朵。
            <div>你按 Bill 的眼睛。
            <div>你按 Bill 的鼻子。
            <div>yeah!....
            <div>whoops( )
        </div>
    </div>
</body>
</html>
```

```
</script>
</head>
<body>
  <h3><font face="Verdana">像 Image Map 一样使用 ImageButton</font>
</h3>
  <form runat=server>
    <table width="100%" border=0>
      <tr>
        <td width="25%">
          <asp:ImageButton
            id=Button1
            ImageUrl="~/images/billg.gif"
            BorderWidth="2px"
            onclick="ImageButton1_OnClick"
            runat="server"
          />
        </td>
        <td>
          <asp:Label
            id=Label1
            Text="X:"
            runat="SERVER"
          />
          <br>
          <asp:Label
            id=Label2
            Text="Y:"
            runat="SERVER"
          />
        </td>
      </tr>
    </table>
    <asp:Label
      id=Label3
      font-bold="true"
      Text=""
      runat="SERVER"
    />
  </form>
```

```
</body>
```

```
</html>
```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_32c.aspx 的源代码如下：

```
<html>
```

```
<head>
```

```
<script language="C#" runat="server">
```

```
void ImageButton1_OnClick(object Source, ImageClickEventArgs e) {
```

```
    //获得鼠标的位置
```

```
    int x=e.X;
```

```
    int y=e.Y;
```

```
    Label1.Text = "X: " + x.ToString();
```

```
    Label2.Text = "Y: " + y.ToString();
```

```
    Button1.ImageUrl = "../images/billg.gif";
```

```
    // 根据鼠标的位置判断到底是那个部位被按下
```

```
    if (x>20 && x<35 && y>72 && y<88) {
```

```
        Label3.Text = "你按 Bill 的耳朵。";
```

```
    }
```

```
    else if (x>42 && x<58 && y>48 && y<64) {
```

```
        Label3.Text = "你按 Bill 的眼睛。";
```

```
    }
```

```
    else if (x>62 && x<90) {
```

```
        if (y>46 && y<56) {
```

```
            Label3.Text = "你按 Bill 的眼睛。";
```

```
        }
```

```
        else if (y<72 && y>60) {
```

```
            Label3.Text = "你按 Bill 的鼻子。";
```

```
        }
```

```
        else if (y<92 && y>74) {
```

```
            Button1.ImageUrl = "../images/billg_tongue.gif";
```

```
            Label3.Text = "yeah!....";
```

```
        }
```

```
        else {
```

```
            whoops();
```

```
        }
```

```
    }
```

```
    else if (x>100 && x<108 && y>56 && y<72) {
```

```
        Label3.Text = "你按 Bill 的耳朵。";
```

```
    }
```

```
    else {
```

```
        whoops( );
    }
}

void whoops() {
    Label3.Text = "错误！应该按 Bill 的眼睛、耳朵、鼻子、或者嘴巴！";
}

</script>
</head>
<body>
    <h3><font face="Verdana">像 Image Map 一样使用 ImageButton</font>
</h3>
    <form runat=server>
        <table width="100%" border=0>
            <tr>
                <td width="25%">
                    <asp:ImageButton
                        id=Button1
                        ImageUrl="../images/billg.gif"
                        BorderWidth="2px"
                        onclick="ImageButton1_OnClick"
                        runat="server"
                    />
                </td>
                <td>
                    <asp:Label
                        id=Label1
                        Text="X:"
                        runat="SERVER"
                    />
                    <br>
                    <asp:Label
                        id=Label2
                        Text="Y:"
                        runat="SERVER"
                    />
                </td>
            </tr>
        </table>
        <asp:Label
```

```

        id=Label3
        font-bold="true"
        Text=""
        runat="SERVER"
    />
</form>
</body>
</html>

```

程序运行结果如图 5 - 42 所示。



图 5 - 42 像 Image Map 一样使用 ImageButton 运行结果

举例三：添加有关鼠标的行为。

使用 VB 语言编写的程序 ch05/test5_33vb.aspx 的源代码如下：

```

<html>
<head>
    <style>
        .applyBorder { border:0.1cm solid blue }
        .removeBorder { border:0.1cm solid gainsboro }
    </style>
    <script language="VB" runat="server">
        Sub Button1_Click(sender As Object, e As ImageClickEventArgs)
            Label1.Text="你按第一个按钮"
        End Sub
        Sub Button2_Click(sender As Object, e As ImageClickEventArgs)
            Label1.Text="你按第二个按钮"
        End Sub
        Sub Button3_Click(sender As Object, e As ImageClickEventArgs)
            Label1.Text="你按第三个按钮"
        End Sub
    </script>
</head>
<body>
    <div style="border: 1px solid blue; padding: 10px;">
        <img alt="Bill Clinton" style="width: 100px; height: 100px; border: 1px solid blue;"/>
        X: 85  
Y: 48  
Go ahead... Click me!  
你按Bill的眼睛。
    </div>
</body>
</html>

```

```
End Sub
</script>
</head>
<body>
    <h3><font face="Verdana">在 ImageButton 控件中使用 Mouse-Over 行为</font>
</h3>
    <form runat=server>
    <!-- 布置 ImageButton 控件，并且设定其属性-->
        <asp:ImageButton
            id=Button1
            runat="server"
                ImageUrl="../images/banana.jpg"
                BorderWidth="0.1cm"
                BorderColor="Gainsboro"
                onmouseover="this.style.border='0.1cm solid black';"
                onmouseout="this.style.border='0.1cm solid gainsboro';"
                onclick="Button1_Click"
            />
        在 ImageButton 直接应用样式
    <br>
        <asp:ImageButton
            id=Button2
            runat="server"
                ImageUrl="../images/banana.jpg"
                CssClass="removeBorder"
                onmouseover="this.className='applyBorder'"
                onmouseout="this.className='removeBorder'"
                onclick="Button2_Click"
            />
        在 ImageButton 中使用已经定义的样式
    <br>
    <!-- 布置第三个 ImageButton 控件，实现变换图片的功能-->
        <asp:ImageButton
            id=Button3
            runat="server"
                ImageUrl="../images/banana.jpg"
                BorderWidth="0.1cm"
                BorderColor="Gainsboro"
```

```

        onmouseover="this.src='../images/mango.jpg';"
        onmouseout="this.src='../images/banana.jpg';"
        onclick="Button3_Click"
    />
    变换图片
<p>
    <asp:Label id=Label1 runat=server />
</form>
</body>
</html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_33c.aspx 的源代码如下：

```

<html>
<head>
    <style>
        .applyBorder { border:0.1cm solid blue }
        .removeBorder { border:0.1cm solid gainsboro }
    </style>
    <script language="C#" runat="server">
        void Button1_Click(object Source, ImageClickEventArgs e) {
            Label1.Text="你按第一个按钮";
        }
        void Button2_Click(object Source, ImageClickEventArgs e) {
            Label1.Text="你按第二个按钮";
        }
        void Button3_Click(object Source, ImageClickEventArgs e) {
            Label1.Text="你按第三个按钮";
        }
    </script>
</head>
<body>
    <h3><font face="Verdana">在 ImageButton 控件中使用 Mouse-Over 行为</font>
</h3>
    <form runat=server>
        <asp:ImageButton
            id=Button1
            runat="server"
            ImageUrl="../images/banana.jpg"
            BorderWidth="0.1cm"

```

```
BorderColor="Gainsboro"
onmouseover="this.style.border='0.1cm solid black';"
onmouseout="this.style.border='0.1cm solid gainsboro';"
onclick="Button1_Click"
/>
```

在 ImageButton 直接应用样式

```
<br>
<asp:ImageButton
id=Button2
runat="server"
ImageUrl="~/images/banana.jpg"
CssClass="removeBorder"
onmouseover="this.className='applyBorder'"
onmouseout="this.className='removeBorder'"
onclick="Button2_Click"
/>
```

在 ImageButton 中使用已经定义的样式

```
<br>
<!--布置第三个 ImageButton 控件，实现变换图片的功能-->
<asp:ImageButton
id=Button3
runat="server"
ImageUrl="~/images/banana.jpg"
BorderWidth="0.1cm"
BorderColor="Gainsboro"
onmouseover="this.src='~/images/mango.jpg';"
onmouseout="this.src='~/images/banana.jpg';"
onclick="Button3_Click" />
变换图片
```

```
<p>
<asp:Label id=Label1 runat=server />
```

```
</form>
```

```
</body>
```

```
</html>
```

运行结果如图 5 - 43 所示。



图 5 - 43 在 ImageButton 控件中使用 Mouse-Over 行为的运行结果

5.15 Label Web 控件

Label 控件用来显示文字。其使用语法如下：

```
<asp:Label id="Label1" runat="server"
    Text="label"
/>
or
<asp:Label id="Label1" runat="server">
    Text
</asp:Label>
```

下面将在文本框里填写文字，并用 Label 控件显示出来。

使用 VB 语言编写的程序 ch05/test5_34vb.aspx 的源代码如下：

```
<html>
<head>
    <script language="VB" runat="server">
        Sub Button1_Click(sender As Object, e As EventArgs)
            Label1.Text = Text1.Text
        End Sub
    </script>
</head>
<body>
    <h3><font face="Verdana">Label 控件举例</font>
</h3>
    <form runat="server">
```

```
<asp:Label
    id="Label1"
    Text="这是 Label1"
    Font-Name="Verdana"
    Font-Size="10pt"
    Width="200px"
    BorderStyle="solid"
    BorderColor="#cccccc"
    runat="server"
/>
<p>
<asp:TextBox
    id="Text1"
    Text="将本文本框里面的文字复制到上面的 Label"
    Width="200px"
    runat="server"
/>
<asp:Button
    id="Button1"
    Text="复制"
    OnClick="Button1_Click"
    Runat="server"
/>
</form>
</body>
</html>
```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_34c.aspx 的源代码如下：

```
<html>
<head>
    <script language="C#" runat="server">
        void Button1_Click(Object Sender, EventArgs e) {
            Label1.Text = Text1.Text;
        }
    </script>
</head>
<body>
    <h3><font face="Verdana">Label 控件举例</font>
</h3>
    <form runat="server">
```

```

<asp:Label
    id="Label1"
    Text="这是 Label1"
    Font-Name="Verdana"
    Font-Size="10pt"
    Width="200px"
    BorderStyle="solid"
    BorderColor="#cccccc"
    runat="server"
/>
<p>
<asp:TextBox
    id="Text1"
    Text="将本文本框里面的文字复制到上面的 Label"
    Width="200px"
    runat="server"
/>
<asp:Button
    id="Button1"
    Text="复制"
    OnClick="Button1_Click"
    Runat="server"
/>
</form>
</body>
</html>

```

运行结果如图 5 - 44 所示。



图 5 - 44 Label Web 控件举例运行结果

5.16 LinkButton Web 控件

LinkButton Web 控件是用来在文字上建立链接的。其使用语法如下：

```

<asp:LinkButton id="LinkButton1" runat="server"
    Text="label"
    Command="command"
    CommandArgument="commandArgument"
    OnClick="OnClickMethod"
/>
or

```

```

<asp:LinkButton id="LinkButton1" runat="server"
    Command="command"
    CommandArgument="commandArgument"
    OnClick="OnClickMethod"
/>
    Text
</asp:LinkButton>

```

在本例中只要按下链接，就会出现相应的处理结果。

使用 VB 语言编写的程序 ch05/test5_35vb.aspx 的源代码如下：

```

<html>
<head>
    <script language="VB" runat="server">
        Sub LinkButton1_Click(sender As Object, e As EventArgs)
            Label1.Text="你按链接按钮"
        End Sub
    </script>
</head>
<body>
    <h3><font face="Verdana">LinkButton 举例</font>
</h3>
    <form runat=server>
        <!--布置 LinkButton ，来建立链接-->
        <asp:LinkButton
            Text="按我！"
            Font-Name="Verdana"
            Font-Size="14pt"
            onclick="LinkButton1_Click"
            runat="server"
        />
        <!--布置 label 控件。来显示运行结果-->
        <asp:Label
            id=Label1
            runat=server
        />
    </form>
</body>
</html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05 / test5_35c.aspx 的源代码如下：

```

<html>

```

```

<head>
    <script language="C#" runat="server">
        void LinkButton1_Click(Object sender, EventArgs e) {
            Label1.Text="你刚才按链接按钮";
        }
    </script>
</head>
<body>
    <h3><font face="Verdana">LinkButton 举例</font>
</h3>
<form runat=server>
    <!--布置 LinkButton ，来建立链接-->
        <asp:LinkButton
            Text="按我！"
            Font-Name="Verdana"
            Font-Size="14pt"
            onclick="LinkButton1_Click"
            runat="server"
        />
    <!--布置 label 控件。来显示运行结果-->
    <asp:Label
        id=Label1
        runat=server
    />
</form>
</body>
</html>

```



图 5 - 45 LinkButton Web 控件举例运行结果

运行结果如图 5 - 45 所示。

5.17 ListBox Web 控 件

ListBox Web 控件可以用来提供一个单选或者多选的列表框，如果想提供多选功能，则要设定 SelectionMode 属性为 Multiple。其使用语法如下：

```

<asp:ListBox id="Listbox1" runat="server"
    DataSource="<% databindingexpression %>"
    DataTextField="DataSourceField"
    DataValueField="DataSourceField"
    AutoPostBack="True|False"

```

```

        Rows="rowcount"
        SelectionMode="Single|Multiple"
        OnSelectedIndexChanged="OnSelectedIndexChangedMethod"
    />
    <asp:Listitem value="value" selected="True|False">
        Text
    </asp:Listitem>
</ASP:ListBox>

```

ListBox Web 控件应用举例一：提供一个可以单选列表框。

使用 VB 语言编写的程序 ch05/test5_36vb.aspx 的源代码如下：

```

<html>
<head>
    <script language="VB" runat="server">
        Sub SubmitBtn_Click(sender As Object, e As EventArgs)
            If ListBox1.SelectedIndex > -1 Then
                Label1.Text="您选择的是：" + ListBox1.SelectedItem.Text
            End If
        End Sub
    </script>
</head>
<body>
    <h3><font face="Verdana">ListBox 举例</font>
</h3>
    <p>
    <form runat=server>
    <!--布置 ListBox 控件，来提供选择列表-->
        <asp:ListBox
            id=ListBox1
            Width="100px"
            runat="server"
        >
            <asp:ListItem>选项 1</asp:ListItem>
            <asp:ListItem>选项 2</asp:ListItem>
            <asp:ListItem>选项 3</asp:ListItem>
            <asp:ListItem>选项 4</asp:ListItem>
            <asp:ListItem>选项 5</asp:ListItem>
            <asp:ListItem>选项 6</asp:ListItem>
        </asp:ListBox>
    <!--布置 button 控件，当按下按钮时调用 SubmitBtn_Click-->

```

```

        <asp:button
            Text="提交"
            OnClick="SubmitBtn_Click"
            runat="server"
        />
    <p>
<!-- 布置 label 控件来显示运行结果-->
        <asp:Label
            id=Label1
            font-name="Verdana"
            font-size="10pt"
            runat="server"
        />
    </form>
</body>
</html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_36c.aspx 的源代码如下：

```

<html>
<head>

    <script language="C#" runat="server">
        void SubmitBtn_Click(Object Sender, EventArgs e) {
            if (ListBox1.SelectedIndex > -1) {
                Label1.Text="你的选择是：" + ListBox1.SelectedItem.Text;
            }
        }
    </script>
</head>
<body>
    <h3><font face="Verdana">ListBox 举例</font>
</h3>
    <p>
    <form runat="server">
    <!-- 布置 ListBox 控件，来提供选择列表-->
        <asp:ListBox
            id=ListBox1
            Width="100px"
            runat="server"
        >

```

```

        <asp:ListItem>选项 1</asp:ListItem>
        <asp:ListItem>选项 2</asp:ListItem>
        <asp:ListItem>选项 3</asp:ListItem>
        <asp:ListItem>选项 4</asp:ListItem>
        <asp:ListItem>选项 5</asp:ListItem>
        <asp:ListItem>选项 6</asp:ListItem>
    </asp:ListBox>
    <!--布置 button 控件，当按下按钮时调用 SubmitBtn_Click-->
    <asp:button
        Text="提交"
        OnClick="SubmitBtn_Click"
        runat="server"
    />
    <p>
    <!--布置 label 控件来显示运行结果-->
    <asp:Label
        id=Label1
        font-name="Verdana"
        font-size="10pt"
        runat="server"
    />
    </form>
</body>
</html>

```

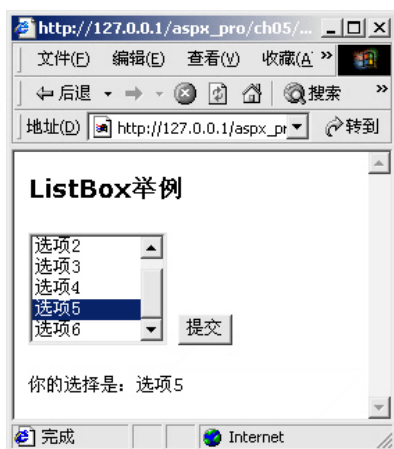


图 5 - 46 ListBox Web 控件举例一运行结果

运行结果如图 5 - 46 所示。

ListBox Web 控件应用举例二：使用数组来制定列表框的选项。

使用 VB 语言编写的程序 ch05/test5_37vb.aspx 的源代码如下：

```

<html>
<head>
    <script language="VB" runat="server">
        Sub Page_Load(sender As Object, e As EventArgs)
            If Not IsPostBack Then
                Dim values as ArrayList= new ArrayList()
                values.Add ("选项 1")
                values.Add ("选项 2")
                values.Add ("选项 3")
                values.Add ("选项 4")
                values.Add ("选项 5")
                values.Add ("选项 6")
            }
        }
    </script>

```

```

        ListBox1.DataSource = values
        ListBox1.DataBind
    End If
End Sub

Sub SubmitBtn_Click(sender As Object, e As EventArgs)
    If ListBox1.SelectedIndex > -1 Then
        Label1.Text = "您的选择是：" + ListBox1.SelectedItem.Text
    End If
End Sub
</script>
</head>
<body>
    <h3><font face="Verdana">DataBinding ListBox</font>
</h3>
    <form runat=server>
        <!-- 布置 ListBox 控件，来显示下拉列表-->
        <asp:ListBox
            id="ListBox1"
            Width="100px"
            runat="server"
        />
        <!-- 布置按钮，当客户按下时调用 SubmitBtn_Click-->
        <asp:button
            Text="提交"
            OnClick="SubmitBtn_Click"
            runat=server
        />
        <p>
        <!-- 布置 Label 控件，来显示运行结果-->
        <asp:Label
            id=Label1
            font-name="Verdana"
            font-size="10pt"
            runat="server"
        />
    </form>
</body>
</html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_37c.aspx 的源代码如下：

```
<html>
<head>
    <script language="C#" runat="server">
        void Page_Load(Object Sender, EventArgs E) {
            if (!IsPostBack) {
                ArrayList values = new ArrayList();
                values.Add ("选项 1");
                values.Add ("选项 2");
                values.Add ("选项 3");
                values.Add ("选项 4");
                values.Add ("选项 5");
                values.Add ("选项 6");
                ListBox1.DataSource = values;
                ListBox1.DataBind();
            }
        }
        void SubmitBtn_Click(Object sender, EventArgs e) {
            if ( ListBox1.SelectedIndex > -1 ) {
                Label1.Text = "您的选择是：" + ListBox1.SelectedItem.Text;
            }
        }
    </script>
</head>
<body>
    <h3><font face="Verdana">DataBinding ListBox</font>
</h3>
    <form runat="server">
        <!-- 布置 ListBox 控件，来显示下拉列表-->
        <asp:ListBox
            id="ListBox1"
            Width="100px"
            runat="server"
        />
        <!-- 布置按钮，当客户按下时调用 SubmitBtn_Click-->
        <asp:button
            Text="提交"
            OnClick="SubmitBtn_Click"
            runat="server"
        />
    </form>
</body>
</html>
```

```

    <p>
    <!--布置 Label 控件，来显示运行结果-->
    <asp:Label
        id=Label1
        font-name="Verdana"
        font-size="10pt"
        runat="server"
    />
</form>
</body>
</html>

```

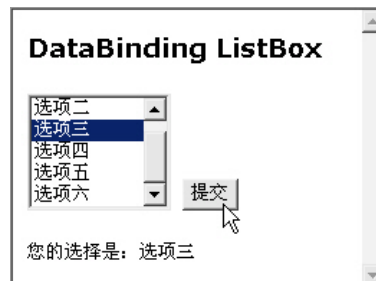


图 5 - 47 ListBox Web 控件举例二运行结果

运行结果如图 5 - 47 所示。

5.18 Panel Web 控件

Panel Web 控件，可以像使用 HTML 里的<DIV>一样编写程序。其使用语法如下：

```

<asp:Panel id="Panel1" runat="server"
    BackImageUrl="url"
    HorizontalAlign="Center|Justify|Left|NotSet|Right"
    Wrap="True|False"
/>
(Other controls declared here)
</asp:Panel>

```

本例中，使用 VB 语言编写的程序 cho5/test5_38vb.aspx 的源代码如下：

```

<html>
<head>
    <script language="VB" runat="server">
        Sub Page_Load(sender As Object, e As EventArgs)
            Dim numlabels as Integer
            Dim numtexts as Integer
            Dim i as Integer
            '显示/隐藏 Panel 内容
            If Check1.Checked Then
                Panel1.Visible = False
            Else
                Panel1.Visible = True
            End If
            '建立 label 控件

```

```
numlabels = CInt(DropDown1.SelectedItem.Value)
For i = 1 To numlabels
    Dim l As Label = New Label()
    l.Text = "Label" & i
    l.ID = "Label" & i
    Panel1.Controls.Add(l)
    Panel1.Controls.Add(new LiteralControl("<br>"))
Next
'建立 textbox 控件
numtexts = CInt(DropDown2.SelectedItem.Value)
For i = 1 To numtexts
    Dim t As TextBox = New TextBox()
    t.Text = "文本框" & i
    t.ID = "文本框" & i
    Panel1.Controls.Add(t)
    Panel1.Controls.Add(new LiteralControl("<br>"))
Next
End Sub
</script>
</head>
<body>
    <h3><font face="Verdana">Panel 举例</font>
</h3>
    <form runat=server>
<!--布置 Panel 控件-->
        <asp:Panel
            id="Panel1"
            runat="server"
            BackColor="gainsboro"
            Height="200px"
            Width="300px">
            Panel1: 这是静态的内容
        <p>
        </asp:Panel>
        <p>
        建立 Labels:
<!--布置 DropDownList 控件, 让客户选择产生 Label 的个数-->
        <asp:DropDownList
            id=DropDown1
```

```

        runat="server"
    >
        <asp:ListItem Value="0">0</asp:ListItem>
        <asp:ListItem Value="1">1</asp:ListItem>
        <asp:ListItem Value="2">2</asp:ListItem>
        <asp:ListItem Value="3">3</asp:ListItem>
        <asp:ListItem Value="4">4</asp:ListItem>
    </asp:DropDownList>
    <br>
    建立文本框:
    <asp:DropDownList
    id=DropDown2
    runat="server"
    >
        <asp:ListItem Value="0">0</asp:ListItem>
        <asp:ListItem Value="1">1</asp:ListItem>
        <asp:ListItem Value="2">2</asp:ListItem>
        <asp:ListItem Value="3">3</asp:ListItem>
        <asp:ListItem Value="4">4</asp:ListItem>
    </asp:DropDownList>
    <p>
    <!--布置 CheckBox，决定 Panel 的显示还是隐藏-->
    <asp:CheckBox
    id="Check1"
    Text="隐藏 Panel"
    runat="server"
    />
    <p>
    <!--布置 Button 按钮，按下时刷新页面-->
    <asp:Button
    Text="刷新 Panel"
    runat="server"
    />
    </font>    </form>
</body></html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/ test5_38c.aspx 的源代码如下：

```

<html>
<head>
    <script language="C#" runat="server">

```

```
void Page_Load(Object sender, EventArgs e) {
    //显示/隐藏 Panel 内容
    if (Check1.Checked) {
        Panel1.Visible=false;
    }
    else {
        Panel1.Visible=true;
    }
    //建立 label 控件
    int numlabels = int.FromString(DropDown1.SelectedItem.Value);
    for (int i=1; i<=numlabels; i++) {
        Label l = new Label();
        l.Text = "Label" + Int32.ToString(i);
        l.ID = "Label" + Int32.ToString(i);
        Panel1.Controls.Add(l);
        Panel1.Controls.Add(new LiteralControl("<br>"));
    }
    //建立 textbox 控件
    int numtexts = int.FromString(DropDown2.SelectedItem.Value);
    for (int i=1; i<=numtexts; i++) {
        TextBox t = new TextBox();
        t.Text = "文本框" + Int32.ToString(i);
        t.ID = "文本框" + Int32.ToString(i);
        Panel1.Controls.Add(t);
        Panel1.Controls.Add(new LiteralControl("<br>"));
    }
}

</script></head>
<body>
    <h3><font face="Verdana">Panel 举例</font>
</h3>
    <form runat=server>
<!--布置 Panel 控件-->
        <asp:Panel
            id="Panel1"
            runat="server"
            BackColor="gainsboro"
            Height="200px"
            Width="300px">
```

```
Panel1: 这是静态的内容
<p>
</asp:Panel>
<p>
建立 Labels:
<!--布置 DropDownList 控件，让客户选择产生 Label 的个数-->
<asp:DropDownList
    id=DropDown1
    runat="server"
>
    <asp:ListItem Value="0">0</asp:ListItem>
    <asp:ListItem Value="1">1</asp:ListItem>
    <asp:ListItem Value="2">2</asp:ListItem>
    <asp:ListItem Value="3">3</asp:ListItem>
    <asp:ListItem Value="4">4</asp:ListItem>
</asp:DropDownList>
<br>
建立文本框:
<asp:DropDownList
    id=DropDown2
    runat="server"
>
    <asp:ListItem Value="0">0</asp:ListItem>
    <asp:ListItem Value="1">1</asp:ListItem>
    <asp:ListItem Value="2">2</asp:ListItem>
    <asp:ListItem Value="3">3</asp:ListItem>
    <asp:ListItem Value="4">4</asp:ListItem>
</asp:DropDownList>
<p>
<!--布置 CheckBox，决定 Panel 的显示还是隐藏-->
<asp:CheckBox
    id="Check1"
    Text="隐藏 Panel"
    runat="server"
/>
<p>
<!--布置 Button 按钮，按下时刷新页面-->
<asp:Button
    Text="刷新 Panel"
```

```

        runat="server"
    />
</font>    </form>
</body></html>

```

举例运行结果如图 5 - 48 所示。



图 5 - 48 Panel Web 控件举例运行结果

5.19 RadioButton Web 控件

RadioButton Web 控件,可以在网页中布置单选框,通过使用一个 GroupName 来建立一组 RadioButton。其使用语法如下:

```

<asp:RadioButton id="RadioButton1" runat="server"
    AutoPostBack="True|False"
    Checked="True|False"
    GroupName="GroupName"
    Text="label"
    TextAlign="Right|Left"
    OnCheckedChanged="OnCheckedChangedMethod"
/>

```

RadioButton Web 控件举例: 根据提交的内容显示不同的结果。
使用 VB 语言编写的程序 ch05/test5_39vb.aspx 的源代码如下:

```

<html>
<head>
    <script language="VB" runat="server">
        Sub SubmitBtn_Click(sender As Object, e As EventArgs)

```

```
        If Radio1.Checked Then
            Label1.Text = "你选择的是：" & Radio1.Text
        Else If Radio2.Checked Then
            Label1.Text = "你选择的是：" & Radio2.Text
        Else If Radio3.Checked Then
            Label1.Text = "你选择的是：" & Radio3.Text
        End If
    End Sub
End Sub
</script>
</head>
<body>
    <h3><font face="Verdana">RadioButton 举例</font>
</h3>
    <form runat=server>
        <h4>选择下面的选项</h4>
        <!--布置 RadioButton 控件，使用 RadioGroup1 建立一个组-->
        <asp:RadioButton
            id=Radio1
            Text="选项一"
            Checked="True"
            GroupName="RadioGroup1"
            runat="server"
        />    这个选项是第一个了。
        <br>
        <asp:RadioButton
            id=Radio2
            Text="选项二"
            GroupName="RadioGroup1"
            runat="server"
        />    这是第二个选项了。
        <br>
        <asp:RadioButton
            id=Radio3
            runat="server"
            Text="选项三"
            GroupName="RadioGroup1"
        />    这是第三个选项。
        <br>
        <!--布置按钮，按下按钮时调用 SubmitBtn_Click-->
```

```
<asp:button
    text="提交"
    OnClick="SubmitBtn_Click"
    runat=server
/>
<br>
<!--布置 Label 控件，显示运行结果-->
<asp:Label
    id=Label1
    font-bold="true"
    runat="server"
/>
</form></body></html>
```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_39c.aspx 的源代码如下：

```
<html>
<head>
    <script language="C#" runat="server">
        void SubmitBtn_Click(Object Sender, EventArgs e) {
            if (Radio1.Checked) {
                Label1.Text = "你选择的是：" + Radio1.Text;
            }
            else if (Radio2.Checked) {
                Label1.Text = "你选择的是：" + Radio2.Text;
            }
            else if (Radio3.Checked) {
                Label1.Text = "你选择的是：" + Radio3.Text;
            }
        }
    </script></head>
<body>
    <h3><font face="Verdana">RadioButton 举例</font>
</h3>
    <form runat=server>
        <h4>选择下面的选项</h4>
        <!--布置 RadioButton 控件，使用 RadioGroup1 建立一个组-->
        <asp:RadioButton
            id=Radio1
            Text="选项一"
            Checked="True"
```

```

        GroupName="RadioGroup1"
        runat="server"
    />    这个选项是第一个了。
    <br>
    <asp:RadioButton
        id=Radio2
        Text="选项二"
        GroupName="RadioGroup1"
        runat="server"
    />    这是第二个选项了。
    <br>
    <asp:RadioButton
        id=Radio3
        runat="server"
        Text="选项三"
        GroupName="RadioGroup1"
    />    这是第三个选项
    <br>
    <!--布置按钮，按下按钮时调用 SubmitBtn_Click-->
    <asp:button
        text="提交"
        OnClick="SubmitBtn_Click"
        runat=server
    />
    <br>
    <!--布置 Label 控件，显示运行结果-->
    <asp:Label
        id=Label1
        font-bold="true"
        runat="server"
    />
    </form></body></html>

```

运行结果如图 5 - 49 所示。



图 5 - 49 RadioButton Web 控件举例运行结果

5.20 RadioButtonList Web 控件

RadioButtonList Web 控件可以用来制作单选列表，像使用其它的列表控件一样。哪一个被选定取决于每个单选框的 Selected 属性。其使用语法如下：

```
<asp:RadioButtonList id="RadioButtonList1" runat="server"
    AutoPostBack="True|False"
    CellPadding="Pixels"
    DataSource="<% databindingexpression %>"
    DataTextField="DataSourceField"
    DataValueField="DataSourceField"
    RepeatColumns="ColumnCount"
    RepeatDirection="Vertical|Horizontal"
    RepeatLayout="Flow|Table"
    TextAlign="Right|Left"
    OnSelectedIndexChanged="OnSelectedIndexChangedMethod"
/>
<asp:ListItem text="label" value="value" selected="True|False" />
</asp:RadioButtonList>
```

可以通过设定其 RepeatDirection 属性来确定其排列方式是顺序还是在表格里面。
使用 VB 语言编写的程序 ch05/test5_40vb.aspx 的源代码如下：

```
<html>
<head>
    <script language="VB" runat="server">
        Sub Button1_Click(sender As Object, e As EventArgs)
            If RadioButtonList1.SelectedIndex > -1 Then
                Label1.Text = "您选择的是：" & RadioButtonList1.SelectedItem.Text
            End If
        End Sub

        Sub chkLayout_CheckedChanged(sender As Object, e As EventArgs)
            If chkLayout.Checked = True Then
                RadioButtonList1.RepeatLayout = RepeatLayout.Table
            Else
                RadioButtonList1.RepeatLayout = RepeatLayout.Flow
            End If
        End Sub

        Sub chkDirection_CheckedChanged(sender As Object, e As EventArgs)
            If chkDirection.Checked = True Then
                RadioButtonList1.RepeatDirection = RepeatDirection.Horizontal
            Else
                RadioButtonList1.RepeatDirection = RepeatDirection.Vertical
            End If
        End Sub
    </script>
</head>
<body>
```

```
</script></head>
<body>
    <h3><font face="Verdana">RadioButtonList 举例</font> </h3>
    <form runat=server>
<!--布置 RadioButtonList 控件-->
        <asp:RadioButtonList
            id=RadioButtonList1
            runat="server"
        >
            <asp:ListItem>选项 1</asp:ListItem>
            <asp:ListItem>选项 2</asp:ListItem>
            <asp:ListItem>选项 3</asp:ListItem>
            <asp:ListItem>选项 4</asp:ListItem>
            <asp:ListItem>选项 5</asp:ListItem>
            <asp:ListItem>选项 6</asp:ListItem>
        </asp:RadioButtonList>
        <p>
<!--布置 CheckBox 来让客户选择是否在表格中显示-->
            <asp:CheckBox
                id=chkLayout
                OnCheckedChanged="chkLayout_CheckedChanged"
                Text="在表格中显示"
                Checked=true
                AutoPostBack="true"
                runat="server"
            />
            <br>
<!--布置 CheckBox 来让客户选择排列方向-->
            <asp:CheckBox
                id=chkDirection
                OnCheckedChanged="chkDirection_CheckedChanged"
                Text="水平排列"
                AutoPostBack="true"
                runat="server"
            />
            <p>
            <asp:Button
                id=Button1
                Text="提交"
```

```
        onclick="Button1_Click"
        runat="server"
    />
<p>
<!--布置 Label 控件，显示运行结果-->
    <asp:Label
        id=Label1
        font-name="Verdana"
        font-size="8pt"
        runat="server"
    />
</form></body></html>
```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_40c.aspx 的源代码如下：

```
<html>
<head>
    <script language="C#" runat="server">
        void Button1_Click(object Source, EventArgs e) {
            if (RadioButtonList1.SelectedIndex > -1) {
                Label1.Text = "您的选择是：" + RadioButtonList1.SelectedItem.Text;
            }
        }
        void chkLayout_CheckedChanged(Object sender, EventArgs e) {
            if (chkLayout.Checked == true) {
                RadioButtonList1.RepeatLayout = RepeatLayout.Table;
            }
            else {
                RadioButtonList1.RepeatLayout = RepeatLayout.Flow;
            }
        }
        void chkDirection_CheckedChanged(Object sender, EventArgs e) {
            if (chkDirection.Checked == true) {
                RadioButtonList1.RepeatDirection = RepeatDirection.Horizontal;
            }
            else {
                RadioButtonList1.RepeatDirection = RepeatDirection.Vertical;
            }
        }
    }
</script></head>
<body>
```

```
<h3><font face="Verdana">RadioButtonList 举例</font> </h3>
```

```
<form runat=server>
```

```
<!--布置 RadioButtonList 控件-->
```

```
<asp:RadioButtonList
    id=RadioButtonList1
    runat="server"
>
    <asp:ListItem>选项 1</asp:ListItem>
    <asp:ListItem>选项 2</asp:ListItem>
    <asp:ListItem>选项 3</asp:ListItem>
    <asp:ListItem>选项 4</asp:ListItem>
    <asp:ListItem>选项 5</asp:ListItem>
    <asp:ListItem>选项 6</asp:ListItem>
```

```
</asp:RadioButtonList>
```

```
<p>
```

```
<!--布置 CheckBox 来让客户选择是否在表格中显示-->
```

```
<asp:CheckBox
    id=chkLayout
    OnCheckedChanged="chkLayout_CheckedChanged"
    Text="在表格中显示"
    Checked=true
    AutoPostBack="true"
    runat="server"
/>
```

```
<br>
```

```
<!--布置 CheckBox 来让客户选择排列方向-->
```

```
<asp:CheckBox
    id=chkDirection
    OnCheckedChanged="chkDirection_CheckedChanged"
    Text="水平排列"
    AutoPostBack="true"
    runat="server"
/>
```

```
<p>
```

```
<asp:Button
    id=Button1
    Text="提交"
    onclick="Button1_Click"
    runat="server"
```

```

/>
<p>
<!--布置 Label 控件，显示运行结果-->
<asp:Label
    id=Label1
    font-name="Verdana"
    font-size="8pt"
    runat="server"
/>
</form></body></html>

```

其运行结果如图 5 - 50 所示。



图 5 - 50 RadioButtonList Web 控件举例运行结果

5.21 RangeValidator Web 控件

RangeValidator Web 控件用来在客户端验证输入的值是否在特定的范围之内。下面对 RangeValidator Web 控件举例。

使用 VB 语言编写的程序 ch05/test5_41vb.aspx 的源代码如下：

```

<%@ Page clienttarget=downlevel %>
<html>
<head>
<script language="VB" runat="server">
    Sub Button1_Click(sender As Object, e As EventArgs)
        If Page.IsValid Then
            lblOutput.Text = "结果：正确！"
        Else
            lblOutput.Text = "结果：不正确！"
        End If
    End Sub
End Sub

```

```

Sub lstOperator_SelectedIndexChanged(sender As Object, e As EventArgs)
    rangeVal.Type = lstType.SelectedIndex
    rangeVal.Validate
End Sub
</script></head>
<body>
    <h3><font face="Verdana">RangeValidator 举例</font></h3>
    <p>
        <form runat="server">
            <table bgcolor="#eeeeee" cellpadding=10>
                <tr valign="top">
                    <td> <h5><font face="Verdana">要检查的值：</font></h5>
                        <!--布置文本框-->
                        <asp:TextBox
                            Selected
                            id="txtComp"
                            runat="server"
                        />
                    </td>
                    <td> <h5><font face="Verdana">数据类型</font></h5>
                        <!--布置下拉选单，来选择比较的类型-->
                        <asp:DropDownList
                            id="lstType"
                            OnSelectedIndexChanged="lstOperator_SelectedIndexChanged"
                            runat=server
                        >
                            <asp:ListItem Selected Value="String" >字符串型</asp:ListItem>
                            <asp:ListItem Value="Integer" >整数型</asp:ListItem>
                            <asp:ListItem Value="Double" >双整型</asp:ListItem>
                            <asp:ListItem Value="Date" >日期型</asp:ListItem>
                            <asp:ListItem Value="Currency" >货币型</asp:ListItem>
                        </asp:DropDownList>
                    </td>
                    <td> <h5><font face="Verdana">最小值：</font></h5>
                        <!--输入最小值-->
                        <asp:TextBox
                            id="txtMin"
                            runat="server"
                        /></td>
                </tr>
            </table>
        </form>
    </p>

```

```

<td><h5><font face="Verdana">最大值：</font></h5>
<!--输入最大值-->
    <asp:TextBox
        id="txtMax"
        runat="server"
    />
    <p>
    <!--按钮，按下是调用 Button1_Click-->
    <asp:Button
        Text="验证"
        ID="Button1"
        onclick="Button1_Click"
        runat="server"
    />
</td> </tr> </table>
<!--布置 RangeValidator 用来比较客户端的输入-->
    <asp:RangeValidator
        id="rangeVal"
        Type="String"
        ControlToValidate="txtComp"
        MaximumControl="txtMax"
        MinimumControl="txtMin"
        runat="server"
    />
    <br>
<!--布置 label 控件用来显示运行结果-->
    <asp:Label
        id="lblOutput"
        Font-Name="verdana"
        Font-Size="10pt"
        runat="server"
    />
</form></body></html>

```

实现同样的功能，使用 C# 语言编写 ch05/test5_41c.aspx 的程序的源代码如下：

```

<%@ Page clienttarget=downlevel %>
<html>
<head>
    <script language="C#" runat="server">
        void Button1_Click(Object sender, EventArgs e) {

```

```

        if (Page.IsValid) {
            lblOutput.Text = "结果：正确！";
        }
        else {
            lblOutput.Text = "结果：不正确！";
        }
    }
}

void lstOperator_SelectedIndexChanged(Object sender, EventArgs e) {

    rangeVal.Type = (ValidationDataType) lstType.SelectedIndex;
    rangeVal.Validate();
}
}
</script></head>
<body>
    <h3><font face="Verdana">RangeValidator 举例</font></h3>
    <p>
        <form runat="server">
            <table bgcolor="#eeeeee" cellpadding=10>
                <tr valign="top">
                    <td> <h5><font face="Verdana">要检查的值：</font></h5>
                    <!-- 布置文本框 -->
                    <asp:TextBox
                        Selected
                        id="txtComp"
                        runat="server"
                    />
                </td>
                <td> <h5><font face="Verdana">数据类型</font></h5>
                <!-- 布置下拉选单，来选择比较的类型 -->
                <asp:DropDownList
                    id="lstType"
                    OnSelectedIndexChanged="lstOperator_SelectedIndexChanged"
                    runat="server"
                >
                    <asp:ListItem Selected Value="String" >字符串型</asp:ListItem>
                    <asp:ListItem Value="Integer" >整数型</asp:ListItem>
                    <asp:ListItem Value="Double" >双整型</asp:ListItem>
                    <asp:ListItem Value="Date" >日期型</asp:ListItem>
                    <asp:ListItem Value="Currency" >货币型</asp:ListItem>
                </asp:DropDownList>
            </table>
        </form>
    </p>

```

```
</asp:DropDownList>
</td>
<td><h5><font face="Verdana">最小值：</font></h5>
<!--输入最小值-->
    <asp:TextBox
        id="txtMin"
        runat="server"
    />
</td>
<td> <h5><font face="Verdana">最大值：</font></h5>
<!--输入最大值-->
    <asp:TextBox
        id="txtMax"
        runat="server"
    />
    <p>
        <!--按钮，按下是调用 Button1_Click-->
        <asp:Button
            Text="验证"
            ID="Button1"
            onclick="Button1_Click"
            runat="server"
        />
    </td> </tr> </table>
<!--布置 RangeValidator 用来比较客户端的输入-->
<asp:RangeValidator
    id="rangeVal"
    Type="String"
    ControlToValidate="txtComp"
    MaximumControl="txtMax"
    MinimumControl="txtMin"
    runat="server"
/>
<br>
<!--布置 label 控件用来显示运行结果-->
<asp:Label
    id="lblOutput"
    Font-Name="verdana"
    Font-Size="10pt"
```

```

runat="server"
/> </form></body></html>

```

运行结果如图 5 - 51 所示。

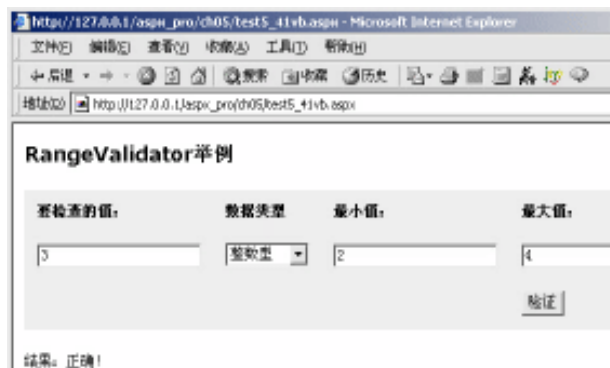


图 5 - 51 RangeValidator Web 控件举例运行结果

5.22 RegularExpressionValidator Web 控件

RegularExpressionValidator Web 控件, 用来检验输入的数据是否符合 Regular Expression 规则。下面通过举例来说明。

使用 VB 语言编写的程序 ch05/test5_42vb.aspx 的源代码如下:

```

<html>
<head>
  <script language="VB" runat="server">
    Sub ValidateBtn_Click(sender As Object, e As EventArgs)
      If Page.IsValid Then
        lblOutput.Text = "输入正确!"
      Else
        lblOutput.Text = "输入错误: -( "
      End If
    End Sub
  </script>
</head>
<body>
  <h3><font face="Verdana">简单 RegularExpressionValidator 举例</font>
</h3>
  <form runat="server">

    <table bgcolor="#eeeeee" cellpadding=10>
      <tr valign="top">

```

```
<td colspan=3>
    <!--布置 label，显示提示信息-->
    <asp:Label
        ID="lblOutput"
        Text="请输入你的 ID 号码、学号"
        Font-Name="Verdana"
        Font-Size="10pt"
        runat="server"
    />
</td>
</tr>
<tr>
    <td> <font face=Verdana size=2> 输入 ID 号码 :( 必须是 2 - 10 位字母 ) </font>
</td>
    <td>
<!--布置文本框-->
<ASP:TextBox
    id=TextBox1
    runat=server
    />
</td>
    <td>
<!--确定验证规则 RegularExpressionValidator-->
<asp:RegularExpressionValidator
    id="RegularExpressionValidator1"
    runat="server"
    ControlToValidate="TextBox1"
    ValidationExpression="[a-zA-Z]{2,10}"
    Display="Static"
    Font-Name="verdana"
    Font-Size="10pt"> 必须是 2 - 10 位英文字母 ! ! ! !
</asp:RegularExpressionValidator>
</td>
</tr>
<tr>
    <td align=right> <font face=Verdana size=2>输入您的学号 :( 必须是 8 位数字 ) </font>
</td>
    <td>
<!--布置文本框-->
```

```

        <ASP:TextBox
            id=TextBox2
            runat=server
        /> </td>
    <td>
        <!--确定验证规则 RegularExpressionValidator-->
        <asp:RegularExpressionValidator
            id="RegularExpressionValidator2"
            runat="server"
            ControlToValidate="TextBox2"
            ValidationExpression="\d{8}$"
            Display="Static"
            Font-Name="verdana"
            Font-Size="10pt"> 学号必须是 7 位数 ! ! ! </asp:RegularExpressionValidator>
    </td>
</tr>
<tr>
    <td>
    </td>
    <td>
        <!--布置按钮，按下时调用 ValidateBtn_Click-->
        <ASP:Button
            text="验证"
            OnClick="ValidateBtn_Click"
            runat=server
        />
    </td>
    <td>
    </td>
</tr>
</table>
</form>
</body>
</html>

```

实现同样的功能，使用 C# 语言编写的程序 ch05/test5_42c.aspx 的源代码如下：

```

<html>
<head>
    <script language="C#" runat=server>
        void ValidateBtn_Click(Object Src, EventArgs E) {

```

```
        if (Page.IsValid) {
            lblOutput.Text = "输入正确！";
        }
        else {
            lblOutput.Text = "输入不正确！：-(";
        }
    }
}
</script>
</head>
<body>
<h3><font face="Verdana">简单 RegularExpressionValidator 举例</font>
</h3>
<form runat="server">

<table bgcolor="#eeeeee" cellpadding=10>
    <tr valign="top">
        <td colspan=3>
            <!--布置 label，显示提示信息-->
            <asp:Label
                ID="lblOutput"
                Text="请输入你的 ID 号码、学号"
                Font-Name="Verdana"
                Font-Size="10pt"
                runat="server"
            />
        </td>

    </tr>
    <tr>
        <td><font face=Verdana size=2> 输入 ID 号码：( 必须是 2 - 10 位字母 ) </font>
        </td>
        <td>
            <!--布置文本框-->
            <ASP:TextBox
                id=TextBox1
                runat=server
            />
        </td>
        <td>
            <!--确定验证规则 RegularExpressionValidator-->
```

```

<asp:RegularExpressionValidator
    id="RegularExpressionValidator1"
    runat="server"
        ControlToValidate="TextBox1"
        ValidationExpression="[a-zA-Z]{2,10}"
        Display="Static"
        Font-Name="verdana"
        Font-Size="10pt"> 必须是 2 - 10 位英文字母 !!!
</asp:RegularExpressionValidator>
</td>
</tr>
<tr>
    <td align=right> <font face=Verdana size=2>输入您的学号 : ( 必须是 8 位数字 ) </font>
</td>
    <td>
        <!--布置文本框-->
        <ASP:TextBox
            id=TextBox2
            runat=server
        /> </td>
    <td>
        <!--确定验证规则 RegularExpressionValidator-->
        <asp:RegularExpressionValidator
            id="RegularExpressionValidator2"
            runat="server"
                ControlToValidate="TextBox2"
                ValidationExpression="^\d{8}$"
                Display="Static"
                Font-Name="verdana"
                Font-Size="10pt"> 学号必须是 7 位数 !!! </asp:RegularExpressionValidator>
        </td>
    </tr>
<tr>
    <td>
        <!--布置按钮，按下时调用 ValidateBtn_Click-->
        <ASP:Button
            text="验证"

```

```

        OnClick="ValidateBtn_Click"
        runat=server
    />
</td>
<td>
</td>
</tr> </table></form>
</body></html>

```

运行结果如图 5 - 52 所示。



图 5 - 52 RegularExpressionValidator Web 控件举例运行结果

5.23 Repeater Web 控 件

Repeater Web 控件用来显示数据库中的数据。其使用语法如下：

```

<asp:Repeater id="Repeater1" runat=server
    DataSource="% databindingexpression %">
>
    <template name="Header">
        Header template HTML
    </template>
    <template name="Item">
        Item template HTML
    </template >
    <template name="AlternatingItem">
        Alternating item template HTML
    </template >

```

```

<template name="Separator">
    Separator template HTML
</template >
<template name="Footer">
    Footer template HTML
</template >

```

```
<asp:Repeater>
```

Repeater Web 控件举例一：使用 VB 语言编写的程序 ch05/test5_43vb.aspx 的源代码如下：

```

<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<script Language="VB" runat="server">
Function CreateDataView(DB as string,table as string) As ICollection
    dim connstr as string
    connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
    dim cmd as ADODatasetCommand
    cmd=new ADODatasetCommand("Select top 4 id,name,sex,college,class from "&
        table,Connstr )
    dim ds as DataSet = new DataSet()
    cmd.FillDataSet(ds,table)
    CreateDataView= new DataView(ds.Tables(table))
End Function
Sub Page_Load(sender As Object, e As EventArgs)
    Repeater1.DataSource = CreateDataView( "sample.mdb", "new" )
    Repeater1.DataBind()
End Sub
</script>
<Html>
<Body BgColor="White">
<H3>Repeater 控件举例一<HR>
</H3>
<asp:Repeater
    id="Repeater1"
    runat="server">
<template name="ItemTemplate">
    <B><%# Container.DataItem("name") %>
    (<%# Container.DataItem("id") %>)
    </B>
    <blockquote>

```

```

        性别:<%# Container.DataItem("sex") %>,
        学院:<%# Container.DataItem("college") %>,
        班级:<%# Container.DataItem("class") %>
    </blockquote>
</template>
</asp:Repeater>
<hr>
</Body>
</Html>

```

运行结果如图 5 - 53 所示。

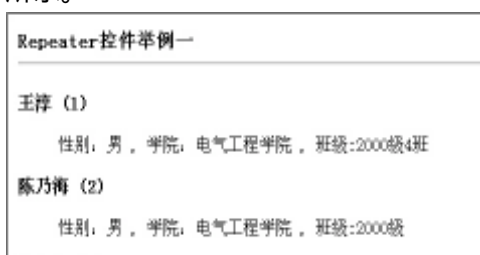


图 5 - 53 Repeater Web 举例一运行结果

举例二：使用 VB 语言编写的程序 ch05/test5_44vb.aspx 的源代码如下：

```

<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<script Language="VB" runat="server">
Function CreateDataView(DB as string,table as string) As ICollection
    dim connstr as string
    connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
    dim cmd as ADODatasetCommand
    cmd=new ADODatasetCommand("Select top 4 id,name,sex from "& table,Connstr )
    dim ds as DataSet = new DataSet( )
    cmd.FillDataSet(ds,table)
    CreateDataView= new DataView(ds.Tables(table))
End Function
Sub Page_Load(sender As Object, e As EventArgs)
    Repeater1.DataSource = CreateDataView( "sample.mdb", "new" )
    Repeater1.DataBind( )
End Sub
</script>
<Html>
<Body BgColor="White">
<H3>Repeate 举例二<HR>

```

```

</H3>

    <asp:Repeater
        id="Repeater1"
        runat="server"
    >
<!--显示数据头部-->
    <template name="HeaderTemplate">
        <Table Border="0" BgColor="Blue">
            <Tr>
                <Td>
                    <Font Color="White" Size="-1">—— — — —人才表格—— — — —</Font>
                </Td>
            </Tr>
        </Table>
        <p>
    </template>

    <template name="ItemTemplate">
<B><%# Container.DataItem("name") %><%# Container.DataItem("id") %>
    </B>
    <blockquote>
        姓名：<%# Container.DataItem("name") %>,
        性别：<%# Container.DataItem("sex") %>
    </blockquote>
    </template>
<!--在数据的底部显示的内容-->
    <template name="FooterTemplate">
        <Table Border="0" BgColor="Blue">
            <Tr>
                <Td>
                    <Font Color="White" Size="-1">—— — — — —表格底部—— — — — —</Font>
                </Td>
            </Tr>
        </Table>
    </template>
</asp:Repeater>
<hr>
</Body>
</Html>

```

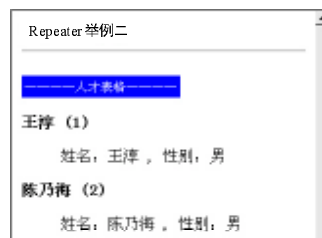


图 5 - 54 Repeater Web 控件举例二运行结果

程序运行结果如图 5 - 54 所示。

5.24 Table、TableRow 和 TableCell 控件

这三个控件是用来显示表格的。Table 控件的使用语法如下：

```
<asp:Table id="Table1" runat="server"
    BackImageUrl="url"
    CellSpacing="cellspacing"
    CellPadding="cellpadding"
    GridLines="None|Horizontal|Vertical|Both"
    HorizontalAlign="Center|Justify|Left|NotSet|Right"
/>
<asp:TableRow>
    <asp:TableCell>
        Cell text
    </asp:TableCell>
</asp:TableRow>
</asp:Table>
```

TableRow 控件的使用语法如下：

```
<asp:TableRow id="TableRow1" runat="server"
    HorizontalAlign="Center|Justify|Left|NotSet|Right"
    VerticalAlign="Bottom|Middle|NotSet|Top"
/>
<asp:TableCell>
    Cell text
</asp:TableCell>
</asp:TableRow>
```

TableCell 控件的使用语法如下：

```
<asp:TableCell id="TableCell1" runat="server"
    ColumnSpan="colcount"
    RowSpan="rowcount"
    HorizontalAlign="Center|Justify|Left|NotSet|Right"
    VerticalAlign="Bottom|Middle|NotSet|Top"
    Wrap="True|False"
/>
    Cell text
</asp:TableCell>
```

使用 VB 语言编写的程序 ch05/test5_45vb.aspx 的源代码如下：

```
<html>
<head>
```

```

<script language="VB" runat="server">
    Sub Page_Load(sender As Object, e As EventArgs)
        Dim numrows As Integer
        Dim numcells As Integer
        Dim i As Integer
        Dim j As Integer
        Dim r As TableRow
        Dim c As TableCell
        '建立表格的行和列
        numrows = CInt(DropDown1.SelectedItem.Value)
        numcells = CInt(DropDown2.SelectedItem.Value)
        For j = 0 To numrows-1
            r = new TableRow()
            For i = 0 To numcells-1
                c = new TableCell()
                c.Controls.Add(new LiteralControl("行" & j & ", 列" & i))
                r.Cells.Add(c)
            Next i
            Table1.Rows.Add(r)
        Next j
    End Sub
</script>
</head>
<body>
    <h3><font face="Verdana">表格举例</font>
</h3>
    <form runat=server>
        <asp:Table id="Table1"
            Font-Name="Verdana"
            Font-Size="8pt"
            CellPadding=5
            CellSpacing=0
            BorderColor="black"
            BorderWidth="1"
            Gridlines="Both"
            runat="server"
        />
    <p>
        表格的行:

```

```

<asp:DropDownList
    id=DropDown1
    runat="server"
    >
    <asp:ListItem Value="1">1</asp:ListItem>
    <asp:ListItem Value="2">2</asp:ListItem>
    <asp:ListItem Value="3">3</asp:ListItem>
    <asp:ListItem Value="4">4</asp:ListItem>
</asp:DropDownList>
<br>
表格的列：
<asp:DropDownList
    id=DropDown2
    runat="server"
    >
    <asp:ListItem Value="1">1</asp:ListItem>
    <asp:ListItem Value="2">2</asp:ListItem>
    <asp:ListItem Value="3">3</asp:ListItem>
    <asp:ListItem Value="4">4</asp:ListItem>
</asp:DropDownList>
<p>
<asp:button
    Text="建立表格"
    runat=server
    />
</form>
</body>
</html>

```

程序运行结果如图 5 - 55 所示。

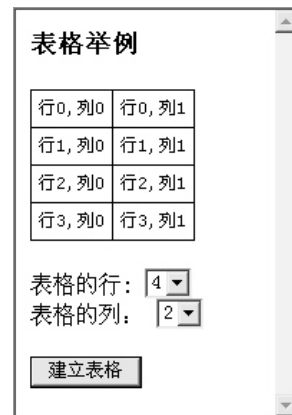


图 5 - 55 Table、TableRow and TableCell 控件举例运行结果

5.25 TextBox Web 控件

TextBox Web 控件用来建立一个文本框。其使用语法如下：

```

<asp:TextBox id=value runat="SERVER"
    AutoPostBack="True|False"
    Columes="characters"
    MaxLength="characters"
    Rows="rows"

```

```

Text="text"
TextMode="Single | Multiline | Password"
Wrap="True|False"
OnTextChanged="OnTextChangedMethod"
>
</asp:TextBox>

```

AutoPostBack 属性如果为 True，则只要文本框的内容更改完毕，就会送到服务器端执行。

MaxLength 属性用于设定最多可以输入的字符的个数。

Rows 属性用于确定行数。

Text 属性用于设定初始值。

TextMode 属性用于设定是文本框、多行文本框或密码等形式。

OnTextChanged 属性用于设定、改变文本框里面的内容时所调用的程序事件。

5.26 ValidationSummary Web 控件

ValidationSummary Web 控件的作用是列出所有未通过检验规则的控件，同时给出有关错误信息。下面的例子就是让客户填写个人信息，只要有错误或者不符合条件，都会给出提示。

```

<%@ Page clienttarget=downlevel %>
<html>
<head>
    <script language="VB" runat="server">
        Sub ListFormat_SelectedIndexChanged(sender As Object, e As EventArgs)
            '以客户的选择来提示信息的显示方式
            valSum.DisplayMode = ListFormat.SelectedIndex
        End Sub
    </script>
</head>
<body>
    <h3><font face="Verdana">ValidationSummary 举例</font></h3>
    <p>
        <form runat="server">
            <table cellpadding=10>
                <tr>
                    <td>
                        <table bgcolor="#eeeeee" cellpadding=10>
                            <tr>

```

```
<td colspan=3>
<font face=Verdana size=2><b>个人信息</b> </font>
</td> </tr>
<tr>
<td align=right>
<font face=Verdana size=2>学号 : </font>
</td>
<td>
<ASP:TextBox id=TextBox1 runat=server />
</td>
<td align=middle rowspan=1>
<!-- 布置 RequiredFieldValidator 使得学号为必须填写项目 -->
<asp:RequiredFieldValidator id="RequiredFieldValidator2"
    ControlToValidate="TextBox1"
    ErrorMessage="必须输入学号的！"
    Display="Static"
    Width="100%"
    runat=server>
    *
</asp:RequiredFieldValidator>
</td> </tr>
<tr>
<td align=right>
<font face=Verdana size=2>性别 : </font>
</td>
<td>
<ASP:RadioButtonList
    id=RadioButtonList1
    RepeatLayout="Flow"
    runat=server>
<asp:ListItem>男</asp:ListItem>
<asp:ListItem>女</asp:ListItem>
</ASP:RadioButtonList>
</td>
<td>
<!-- 布置 RequiredFieldValidator 控件，验证输入的数据 -->
<asp:RequiredFieldValidator
    id="RequiredFieldValidator1"
    ControlToValidate="RadioButtonList1"
```

```

        ErrorMessage="你是男生还是女生？"
        Display="Static"
        InitialValue="" Width="100%" runat=server>
        *

</asp:RequiredFieldValidator>
</td>
</tr>
<tr>
    <td align=right>
        <font face=Verdana size=2>你是哪个学院的：</font>
    </td>
    <td>
        <!--布置下拉选单让客户选择是哪个学院的-->
        <ASP:DropDownList id=DropDownList1 runat=server>
            <asp:ListItem></asp:ListItem>
            <asp:ListItem>土木工程学院</asp:ListItem>
            <asp:ListItem>机械工程学院</asp:ListItem>
            <asp:ListItem>电气工程学院</asp:ListItem>
            <asp:ListItem>计算机学院</asp:ListItem>
            <asp:ListItem>交通运输学院</asp:ListItem>
            <asp:ListItem>管理学院</asp:ListItem>
            <asp:ListItem>物理系</asp:ListItem>
            <asp:ListItem>外语系</asp:ListItem>
            <asp:ListItem>环境系</asp:ListItem>
            <asp:ListItem>成人教育学院</asp:ListItem>
            <asp:ListItem>力学系</asp:ListItem>
            <asp:ListItem>建筑系</asp:ListItem>
        </ASP:DropDownList>
    </td>
    <td>
        <!--布置 RequiredFieldValidator 控件，判断是否进行选择-->
        <asp:RequiredFieldValidator id="RequiredFieldValidator3"
            ControlToValidate="DropDownList1"
            ErrorMessage="你是哪个院系的？？？"
            Display="Static"
            InitialValue=""
            Width="100%"
            runat=server>
            *

```

```

        </asp:RequiredFieldValidator>
    </td>
    <td>
    </tr>
    <tr>
    <td>
    </td>
    <td>
        <ASP:Button id=Button1 text="验证一下了" runat=server />
    </td>
    <td>
    </td> </tr>
    </table>
</td>
<td valign=top>
    <table cellpadding=20>
    <tr><td>
        <!--布置 ValidationSummary 输出验证结果提示信息-->
        <asp:ValidationSummary ID="valSum" runat="server"
            HeaderText="你必须输入下面的内容："
            Font-Name="verdana"
            Font-Size="12"
        />
    </td>          </tr>
    </table>
    </td> </tr>
</table>
<font face="verdana" size="-1">选择提示信息的显示方式</font>
    <asp:DropDownList
        id="ListFormat"
        AutoPostBack=true
        OnSelectedIndexChanged="ListFormat_SelectedIndexChanged"
        runat=server
    >
    <asp:ListItem>List</asp:ListItem>
    <asp:ListItem selected>Bulleted List</asp:ListItem>
    <asp:ListItem>Single Paragraph</asp:ListItem>
</asp:DropDownList>
</form></body></html>

```

运行结果如图 5 - 56 所示。

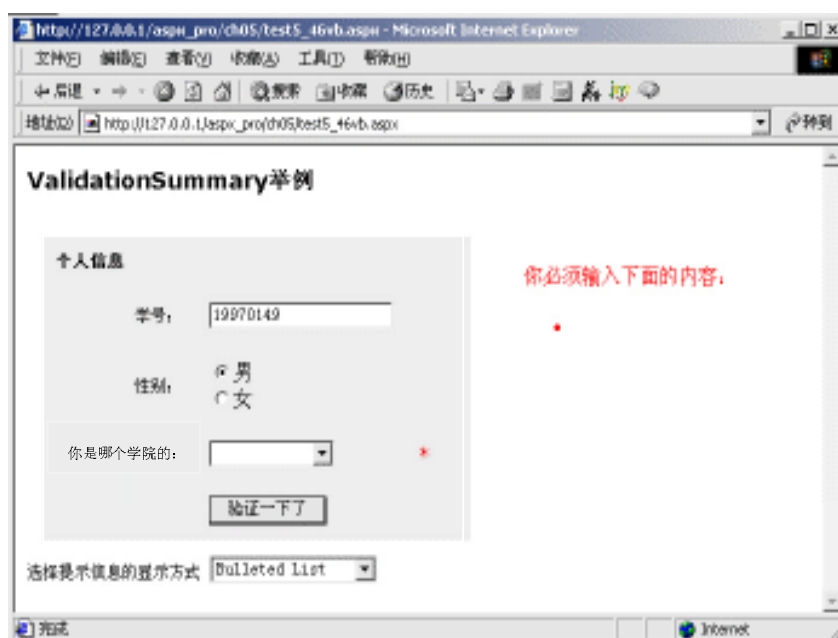


图 5 - 56 ValidationSummary Web 控件运行结果

5.27 本章小结

本章着重讲述了 ASP.NET 的内置服务器控件——Web 控件 ,由于 ASP.NET 提供了强大的数据库访问功能 ,所以对涉及到访问数据库方面的控件作了详细的讲解。同时也提供了丰富的实例 ,供学习借鉴。

第 6 章

ASP.Net 数据库存取技术

主要内容：

- ！ 使用 DataReader 对象
- ！ 使用 DataSet 对象
- ！ 使用 Command 对象

本章主要介绍基于 ASP.Net 的数据库存取技术。在第 5 章中介绍了使用数据控件来访问数据库，数据控件主要有 DataGrid 控件、DataList 控件以及 Repeater 控件。除此之外，本章将介绍有关 ADO+ 的数据库存取对象，它包括 DataReader 对象、DataSet 对象和 Command 对象。

6.1 ADO+ 对象

在 ASP 中，主要使用 ADO 对象来访问数据库，主要有 Connection、Command、Recordset 对象。在 ADO+ 中，主要有 Connection、Command、DataReader、DataSetCommand 及 DataSet 对象。Connection 用来链接数据库，Command 用来执行 SQL 语句，DataReader 读取数据库，DataSetCommand 执行 SQL 语句同时打开数据表格，DataSet 存取数据库。

6.1.1 在 ASP.Net 中访问数据库

在 ASP 中，一般使用下面的语法来链接数据库：

```
<%  
set conn=server.createobject("adodb.connection")  
set rec=server.createobject("adodb.recordset")  
%>
```

在 ASP.Net 中，使用 ADO+ 对象的 Namespace，然后声明 ADO+ 对象，用法如下：

```
<%@ Import Namespace="System.Data" %>  
<%@ Import Namespace="System.Data.ADO" %>  
<script language="VB" runat="server">  
Dim conn As ADOConnection
```

```
Dim dtsCmdobj As ADODataSetCommand
Dim ds As DataSet
Dim strSQL As String
dim connstr_str As String

</script>
```

建立 Connection 对象，打开数据库链接，执行 SQL 语句，有以下两种方法可供选择使用：

第一种方法：

```
dim sql1 as string
"Select id,name,sex,college,class from new"
dim conn as ADOConnection
connstr="DBQ=" & Server.MapPath("sample.mdb")&"&_
Driver={Microsoft Access Driver (*.mdb)};"
conn = new ADOConnection(Connstr)
conn.open( )
dim cmd as ADOCommand
cmd = new ADOCommand(sql1,conn)
cmd.execute( )
```

第二种方法：

```
dim sql1 as string
connstr="DBQ=" & Server.MapPath("sample.mdb")&"&_
Driver={Microsoft Access Driver (*.mdb)};"
dim cmd as ADOCommand
cmd = new ADOCommand(sql1,conn)
cmd.ActiveConnection.open( )
cmd.execute( )
```

6.1.2 DataSetCommand 对象与 Command 对象

DataSetCommand 是建立在 Command 对象基础之上的对象。其功能更加强大，使用方法较简单，其例子如下：

```
connstr="DBQ=" & Server.MapPath("sample.mdb")&"&_
Driver={Microsoft Access Driver (*.mdb)};"
dim cmd as ADODataSetCommand
cmd=new ADODataSetCommand("Select id,name,sex,college,class from new",Connstr )
```

通过执行 SQL 语句和连接数据的语句就可以访问数据库，比使用 Command 对象简练得多。

6.2 DataReader 对象

DataReader 对象和 ADO 中的 Recordset 对象相似。

例如，使用 VB 语言编写的读取程序 ch06/test6_01vb.aspx 的源代码如下：

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<Html>
<Body BgColor="White">
<H3>DataReader 读取数据<HR></H3>
<%
    Dim Cmd As ADOCommand
    Dim Reader As ADODataReader
    Dim ConnStr, SQL As String
    ConnStr = "Provider=Microsoft.Jet.OLEDB.4.0" & _
        ";Data Source=" & Server.MapPath("Sample.mdb")
    SQL = "Select top 5 * From new"
    Cmd = New ADOCommand( SQL, ConnStr )
    Cmd.ActiveConnection.Open()
    Cmd.Execute(Reader)
    While Reader.Read()
        Response.Write( "<UL>" )
        For I = 0 To Reader.FieldCount - 4
            Response.Write( "<LI>" & Reader.GetName(I) )
            Response.Write( " = " & Reader.Item(I) )
        Next
        Response.Write( "</UL>" )
    End While
%>
<Hr>
</Body>
</Html>
```

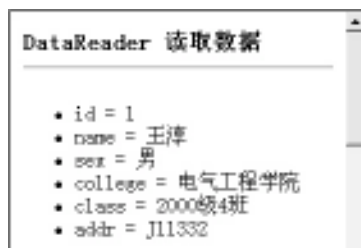


图 6 - 1 DataReader 对象举例一运行结果

运行结果如图 6 - 1 所示。

Read 方法用于读取一个记录。如果读取成功，就会将读取的记录放到 DataReader 对象中，然后将游标下移，返回值为 True；如果读取数据时失败，返回值为 False。在上面的例子中，使用下面的语法显示数据表格中的数据，直到读完所有的数据为止。

```
While Reader.Read()
    Response.Write( "<UL>" )
```

```

For I = 0 To Reader.FieldCount - 4
    Response.Write( "<LI>" & Reader.GetName(I) )
    Response.Write( " = " & Reader.Item(I) )
Next
Response.Write( "</UL>" )
End While

```

FieldCount 属性为访问数据表格的字段数。NEW 表有如图 6 - 2 所示的 9 个字段。

GetName 方法用来读取每一字段的名称，以 NEW 表为例：

```

Reader.GetName(0)    返回：id
Reader.GetName(1)    返回：name
Reader.GetName(2)    返回：sex
Reader.GetName(3)    返回：college
Reader.GetName(4)    返回：class
Reader.GetName(5)    返回：addr

```

Item 属性用来读取数据表格的某一个字段的内容，有以下两种格式：

```

DataReader.Item( 字段序号 )
DataReader.Item( 字段名称 )

```

使用 DataReader 读取数据表格举例二 其使用 VB 语言编写的程序 ch06/test6_02vb.aspx 的源代码如下：

```

<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>

<Html>
<Body BgColor="White">
<H3>DataReader 读取数据<HR></H3>
<%
    Dim Cmd      As ADOCommand
    Dim Reader As ADODataReader
    Dim ConnStr, SQL As String

    ConnStr = "Provider=Microsoft.Jet.OLEDB.4.0" & _
               ";Data Source=" & Server.MapPath("Sample.mdb")
    SQL = "Select top 5 * From new"
    Cmd = New ADOCommand( SQL, ConnStr )
    Cmd.ActiveConnection.Open()
    Cmd.Execute(Reader)

```

new: 表	
字段名称	数据类型
id	数字
name	文本
sex	文本
college	文本
class	文本
addr	文本
telephone	文本
intro	备注
is_select	数字

图 6 - 2 NEW 表的字段设计

```
Dim Data(Reader.FieldCount)
Dim Names(Reader.FieldCount)

'读取数据
For I = 0 To Reader.FieldCount - 1
    Names(I) = Reader.GetName(I)
Next

Response.Write( "此刻 HasMoreRows 的属性为： " & Reader.HasMoreRows )

While Reader.Read()
    Response.Write( "<UL>" )
    Reader.GetValues(Data)
    For I = 0 To Data.Length - 1
        Response.Write( "<LI>" & Names(I) & " = " & Data(I) )
    Next
    Response.Write( "</UL>" )
End While

Response.Write( "此刻 HasMoreRows 的属性为：" & Reader.HasMoreRows )
%>

<Hr>
</Body></Html>
```

运行结果如图 6 - 3 所示。



图 6 - 3 DataReader 对象举例二运行结果

DataReader 其他的属性和方法如表 6 - 1 所示。

表 6 - 1 DataReader 其他的属性和方法

属性和方法	说 明
HasMoreRows	判断 DataReader 是否有数据没有读取
GetDataTypeName	读取数据表格中某一字段的数据类型
GetOrdinal	读取数据表格中某一字段的编号
GetValue	读取数据表格中某一字段的内容
IsNull	判断数据表格中某一字段是否为空
GetValues	读取所有字段的内容

6.3 DataSet 对象

上一节中的 DataReader 对象有点像 ADO 中的 Recordset 对象,但在功能上不如 DataGrid 对象。DataReader 对象和 Command 对象结合使用,而 DataSet 对象和 DataSetCommand 对象结合使用。DataReader 是从一个字段开始读取数据的,而 DataSet 对象可以直接读取整个数据表格,显然 DataSet 对象的功能比较强大。

```
dim connstr as string
'链接数据库
connstr="DBQ=" & Server.MapPath("sample.mdb")&"&";Driver={Microsoft Access Driver (*.mdb)};"
dim cmd as ADODatasetCommand
cmd=new ADODatasetCommand("Select id,name,sex,college,class from new order by id DESC",Connstr)
dim ds as DataSet = new DataSet()
cmd.FillDataSet(ds, "new")
```

从上面的示例中可以看到,示例是用 ADODatasetCommand 来执行 SQL 命令的,然后用 FillDataSet 方法把整个表格的数据存放在 DataSet 对象中。因此可以使用一个 ADODatasetCommand 来执行几个 SQL 命令,在页面中同时显示不同的几个数据表格。

使用 DataSet 对象读取数据表格的名字,其用 VB 语言编写的程序 ch06/test6_03vb.aspx 的源代码如下:

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>

<Html>
<Body BgColor="White">
<H3>DataSet 使用多个 DataTable <HR></H3>
<Blockquote>
<%
    Dim ConnStr, SQL As String
    Dim Cmd As ADODatasetCommand
    Dim ds As DataSet = new DataSet()
```

```
ConnStr = "Provider=Microsoft.Jet.OLEDB.4.0" & _  
        ";Data Source=" & Server.MapPath("Sample.mdb")
```

执行第一个 SQL 命令，产生第一个 DataTable

```
SQL = "Select CID,name,college from neta"
```

```
Cmd = New ADODatasetCommand( SQL, ConnStr )
```

```
Cmd.FillDataSet(ds, "neta")
```

执行第一个 SQL 命令，产生第二个 DataTable

```
SQL = "Select name,college,class from neta_station"
```

```
Cmd = New ADODatasetCommand( SQL, ConnStr )
```

```
Cmd.FillDataSet(ds, "neta_station")
```

执行第一个 SQL 命令，产生第三个 DataTable

```
SQL = "Select name,college,class from new"
```

```
Cmd = New ADODatasetCommand( SQL, ConnStr )
```

```
Cmd.FillDataSet(ds, "new")
```

```
For I = 0 To ds.Tables.Count - 1
```

```
    Response.Write( ds.Tables(I).TableName & "<P>" )
```

```
Next
```

```
%>
```

```
</Blockquote><Hr>
```

```
</Body>
```

```
</Html>
```

运行结果如图 6 - 4 所示。



图 6 - 4 读取数据表格的名字

上面的程序通过 ADODatasetCommand 对象，分别执行了三个 SQL 语句，产生了 neta、neta_station 和 new 三个 DataTable 对象，构成了一个 Tables 集合对象隶属于 DataSet 对象。

DataTable 对象的使用如此例中用 VB 语言编写的程序 ch06/test6_04vb.aspx，其源代码如下：

```

<%@ Page Language="VB" CODEPAGE = "936" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<script language="VB" runat="server">
    Sub subbtn_OnClick(Sender As Object, E As EventArgs)
        Dim connobj As ADOConnection
        Dim dtsCmdobj As ADODatasetCommand
        Dim dtSobj As DataSet
        Dim str_sql As String
        dim conn_str As String
        str_sql = "select name,sex,college,class from new"
        conn_str= "Provider=Microsoft.Jet.OLEDB.4.0" & _
            ";Data Source=" & Server.MapPath("sample.mdb")
        connobj = New ADOConnection(conn_str)
        dtsCmdobj = New ADODatasetCommand(str_sql, connobj)
        dtSobj = New DataSet()
        dtsCmdobj.FillDataSet(dtsobj, "new")
        dtgrid.DataSource = dtsobj.Tables("new").DefaultView
        dtgrid.DataBind()
    End Sub

    Sub subbtn1_OnClick(Sender As Object, E As EventArgs)
        Dim connobj As ADOConnection
        Dim dtsCmdobj As ADODatasetCommand
        Dim dtSobj As DataSet
        Dim str_sql As String
        dim conn_str As String
        str_sql = "select name,sex,college,class from new where is_select<>1"
        conn_str= "Provider=Microsoft.Jet.OLEDB.4.0" & _
            ";Data Source=" & Server.MapPath("sample.mdb")
        connobj = New ADOConnection(conn_str)
        dtsCmdobj = New ADODatasetCommand(str_sql, connobj)
        dtSobj = New DataSet()
        dtsCmdobj.FillDataSet(dtsobj, "new")
        dtgrid.DataSource = dtsobj.Tables("new").DefaultView
        dtgrid.DataBind( )
    End Sub

    Sub subbtn2_OnClick(Sender As Object, E As EventArgs)
        Dim connobj As ADOConnection
        Dim dtsCmdobj As ADODatasetCommand

```

```
Dim dtSobj As DataSet
Dim str_sql As String
dim conn_str As String
str_sql = "select name,sex,college,class from new where is_select=3"
conn_str= "Provider=Microsoft.Jet.OLEDB.4.0" & _
        ";Data Source=" & Server.MapPath("sample.mdb")
connobj = New ADOConnection(conn_str)
dtsCmdobj = New ADODatasetCommand(str_sql, connobj)
dtSobj = New DataSet()
dtsCmdobj.FillDataSet(dtsobj, "new")
dtgrid.DataSource = dtsobj.Tables("new").DefaultView
dtgrid.DataBind()

End Sub
</script>
<html>
<body>
<meta
http-equiv="Content-Type" content="text/html; charset=gb2312">
<form action="" method="post" runat="server">
<asp:Button
type="submit"
id="subbtn"
text="总的报名情况"
OnClick="subbtn_OnClick"
runat="server"
/>
<asp:Button
type="submit"
id="subbtn1"
text="参见面试的人员"
OnClick="subbtn1_OnClick"
runat="server"
/>
<asp:Button
type="submit"
id="subbtn2"
text="最终推荐的人员"
OnClick="subbtn2_OnClick"
runat="server"
```

```

/>
</form>
<asp:DataGrid
id="dtgrid"
HeaderStyle-Font-Bold="True"
ToolTip="data Grid provided with asp+"
runat="server"
maintainviewstate="false"
/>
</body>
</html>

```

运行结果如图 6 - 5 所示。



name	sex	college	class
王津	男	电气工程学院	2000级4班
阳柯	女	电气工程学院	99
陈东南	男	土木工程学院	2000
裴锦里	男	经济管理学院	99
钟巍	男	计算机与通信工程学院	99

图 6 - 5 DataTable 对象举例运行结果

在上面的例子中，都是使用 DataTable 对象，如

```

Dim connobj As ADOConnection
Dim dtsCmdobj As ADODatasetCommand
Dim dtSobj As DataSet
Dim str_sql As String
dim conn_str As String
str_sql = "select name,sex,college,class from new where is_select<>1"
conn_str= "Provider=Microsoft.Jet.OLEDB.4.0" & _
        ";Data Source=" & Server.MapPath("sample.mdb")
connobj = New ADOConnection(conn_str)
dtsCmdobj = New ADODatasetCommand(str_sql, connobj)
dtSobj = New DataSet( )
dtsCmdobj.FillDataSet(dtsobj, "new")
dtgrid.DataSource = dtsobj.Tables("new").DefaultView
dtgrid.DataBind( )

```

其中 dtgrid.DataSource = dtsobj.Tables("new").DefaultView，就是将 new 的 Datatable 对象的 DefaultView 属性赋值给 DataGrid 的 DataSource 属性，然后由 DataGrid 的 DataBind() 属性来绑定数据。

DataTable 对象的 Columns 属性，用以显示字段标题相关信息。属性意义如表 6-2 所示。

表 6-2 DataTable 对象的 Columns 属性

属性	意义
Caption	字段的标题，如[name]、[sex]、[college]等
AllowNull	该字段是否允许为空
AutoIncrement	该字段是否为[自动增长]
DataType	该字段的类型
ReadOnly	该字段是否为[只读]类型

Rows 属性用来读取数据表格每一行的数据，例如使用 VB 语言编写的程序 ch06/test6_05vb.aspx，其源代码如下：

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<Html>
<Body BgColor="White">
<H3>DataTable 的 Rows 属性<HR></H3>
<%
    Dim ConnStr, SQL As String
    Dim Cmd As ADODatasetCommand
    Dim ds As DataSet = new DataSet()
    Dim dt As DataTable
    ConnStr = "Provider=Microsoft.Jet.OLEDB.4.0" & _
        ";Data Source=" & Server.MapPath("Sample.mdb")
    SQL = "Select top 10 name,sex,college,class from new"
    Cmd = New ADODatasetCommand(SQL, ConnStr)
    Cmd.FillDataSet(ds, "new")
    dt = ds.Tables("new")
    Dim R, C As Integer
    For R = 0 To dt.Rows.Count - 1
        Response.Write( "<UL>" )
        For C = 0 To dt.Columns.Count - 1
            Response.Write( "<LI>" & dt.Columns(C).Caption )
            Response.Write( " = " & dt.Rows(R).Item(C) )
        Next
        Response.Write( "</UL>" )
    Next
%>
<Hr>
```

```
</Body>
```

```
</Html>
```

运行结果如图 6 - 6 所示。

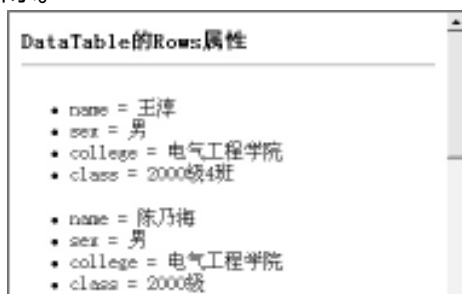


图 6 - 6 Rows 属性用来读取数据举例运行结果

DateSet 和 DataTable 对象可以非常随意地读取数据表格某一行某一列的数据，而 DataReader 对象只能按照顺序读取数据。

6.4 DataView 对象

DataView 对象用来返回数据视图。曾经使用过的函数如下：

```
Function CreateDataView(DB as string,table as string) As ICollection
    dim connstr as string
    connstr="DBQ=" & Server.MapPath(Db)&";&_
    Driver={Microsoft Access Driver (*.mdb)};"
    dim cmd as ADODatasetCommand
    cmd=new ADODatasetCommand("Select id,name,sex,college,class from "& table,
        Connstr )
    dim ds as DataSet = new DataSet()
    cmd.FillDataSet(ds,table)
    CreateDataView= new DataView(ds.Tables(table))
End Function
```

CreateDataView 函数 ,会根据参数打开数据库中的数据表格 ,然后通过 CreateDataView=new DataView(ds.Tables(table))返回数据视图。

建立 DataView 对象之后，然后使用 DataGrid 控件绑定数据，来显示数据表格。

使用 DataView 对象的 Sort 属性来排列数据 ,在第 5 章的 ch05/test5_21vb.aspx 程序中就是通过下面的语句来实现排序的：

```
Sub MyDataGrid_Sort(sender As Object, e As DataGridSortCommandEventArgs)
    SortField = e.SortField
    BindGrid
End Sub
Sub BindGrid()
```

```
MyDataGrid.DataSource = CreateDataView("sample.mdb","new")
```

```
MyDataGrid.DataBind
```

```
End Sub
```

运行结果如图 6 - 7 所示，当按下 college 排序时，会按照汉字的拼音排序。



图 6 - 7 使用 DataView 对象的 Sort 属性来排列数据举例运行结果

按照上面的程序运行结果总是顺序排列数据，如果想反向排序，DataView 对象的 Sort 属性必须加上“ DESC ”。

例如，使用 VB 语言编写的程序 ch06/test6_06vb.aspx 的源代码如下：

```
<%@ Page language="VB" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=gb2312">
<script language="VB" runat="server">
    Dim SortField As String
    '建立数据视图 CreateDataView
    Function CreateDataView(DB as string,table as string) As ICollection
    dim connstr as string
    '链接数据库
    connstr="DBQ=" & Server.MapPath(Db)&";Driver={Microsoft Access Driver (*.mdb)};"
    dim cmd as ADODatasetCommand
    cmd=new ADODatasetCommand("Select id,name,sex,college,class from "& table &" order by id
        DESC",Connstr )
    dim ds as DataSet = new DataSet()
    cmd.FillDataSet(ds,table)
    Dim dv as DataView
    dv=new DataView(ds.Tables(table))
```

在数据视图里面设计排序

```
dv.Sort = SortField
```

```
CreateDataView=dv
```

```
End Function
```

```
Sub Page_Load(sender As Object, e As EventArgs)
```

```
    If Not IsPostBack Then
```

```
        If SortField = "" Then
```

```
            SortField = "sex"
```

```
        End If
```

```
        BindGrid
```

```
    End If
```

```
End Sub
```

'排序时调用的函数 MyDataGrid_Sort

```
Sub MyDataGrid_Sort(sender As Object, e As DataGridSortCommandEventArgs)
```

```
    SortField = e.SortField & " DESC"
```

```
    BindGrid
```

```
End Sub
```

'绑定数据

```
Sub BindGrid()
```

```
    CreateDataView("sample.mdb","new")
```

```
    MyDataGrid.DataBind
```

```
End Sub
```

```
</script>
```

```
<title>
```

在 DataGrid 设置排序

```
</title>
```

```
<body>
```

```
<h3>
```

```
<font face="Verdana">
```

在 DataGrid 设置排序

```
</font>
```

```
</h3>
```

```
<!--布置 DataGrid 控件，显示数据表格-->
```

```
<form runat=server>
```

```
<ASP:DataGrid
```

```
    id="MyDataGrid"
```

```
    runat="server"
```

```
    AllowSorting="true"
```

```
<!--排序调用 MyDataGrid_Sort 命令-->
```

```

OnSortCommand="MyDataGrid_Sort"
BorderColor="black"
BorderWidth="1"
CellPadding="3"
Font-Name="Verdana"
Font-Size="8pt"
HeaderStyle-BackColor="#ccccff"
HeaderStyle-ForeColor="black"
/>
</form>
</body>
</html>

```

在上面的例子中只需对 `SortField = e.SortField & " DESC"` 稍加改动，就可以反向排序。运行结果如图 6-8 所示，排序的结果是从大到小。

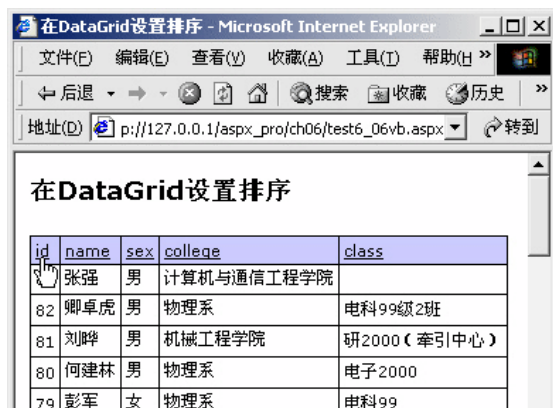


图 6-8 使用 DataView 对象的 Sort 属性来反向排列数据举例运行结果

使用 DataView 对象的 RowFilter 属性实现数据筛选，使得客户端能够方便地找到自己感兴趣的数据。

6.5 Command 对象

如果想对数据库的数据进行浏览、插入、修改、删除等操作，可以使用 Command 对象来执行 SQL 命令，实现对数据的操作。

例如，使用 VB 语言编写的程序 `ch06/test6_07vb.aspx`，可以查看详细的数据，同时还可以更改数据。其源代码如下：

```

<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<script Language="VB" runat="server">

```

```

Function CreateDataView( Db As String, Table As String ) As DataView
    Dim Provider, ConnStr, SQL As String
    '建立连接
    Provider = "Microsoft.Jet.OLEDB.4.0;"
    ConnStr = "Provider=" & Provider & _
        "Data Source=" & Server.MapPath( Db )
    '建立 ADODatasetCommand 对象
    Dim Cmd As ADODatasetCommand
    Cmd = New ADODatasetCommand( "Select name,id,class,tel,addr from " & Table, ConnStr )
    Dim ds As DataSet = new DataSet()
    Cmd.FillDataSet(ds, Table)
    '建立数据视图
    CreateDataView = new DataView(ds.Tables(Table))
End Function

'页面被浏览时，所调用的事件程序 Page_Load
Sub Page_Load(sender As Object, e As EventArgs)
    UpdateString.Value = ""
    '使用 CreateDataView 函数建立数据源
    DataList1.DataSource = CreateDataView( "Sample.mdb", "neta_station" )
    DataList1.DataBind()
End Sub

'根据客户操作，分别执行相应的程序事件
Sub DataList_ItemCommand(sender As Object, e As DataListCommandEvent-Args)
    Select Case e.CommandSource.Text
        Case "详细情况"
            DataList1.SelectedIndex = e.Item.ItemIndex
        Case "资料修改"
            DataList1.EditItemIndex = e.Item.ItemIndex
        Case "返回"
            DataList1.SelectedIndex = -1
        Case "取消"
            DataList1.EditItemIndex = -1
        Case "更新"
            Dim ConnStr, SQL As String

            If Len(UpdateString.Value) <> 0 Then
                Dim Number As Label = e.Item.FindControl("Number")
                '生产更改数据库的 SQL 语句
                SQL = "Update neta_station Set " & UpdateString.Value & _

```

```

" Where id = " & Number.Text
ConnStr = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
"Data Source=" & Server.MapPath( "Sample.mdb" )
'建立 ADOCommand 对象，更改数据库
Dim Cmd As ADOCommand = New ADOCommand( SQL, ConnStr )
    Cmd.ActiveConnection.Open( )
    Cmd.Execute( )
End If
DataList1.EditItemIndex = -1
End Select
'使用函数 CreateDataView，建立数据源
DataList1.DataSource = CreateDataView( "Sample.mdb", "neta_station" )
    DataList1.DataBind( )
End Sub
'记录所更改的数据字段，并且生成 UpdateString.
Sub Text_Changed(sender As Object, e As EventArgs)
    Dim Field As String

    Select Case sender.Id
        Case "class"
            Field = "class"
        Case "tel"
            Field = "tel"
        Case "addr"
            Field = "addr"
    End Select

    If Len(UpdateString.Value) = 0 Then
        UpdateString.Value = Field & " = " & sender.Text & ""
    Else
        UpdateString.Value &= ", " & Field & " = " & sender.Text & ""
    End If
End Sub
</script>
<Html>
<Body BgColor="White">
<H3>可以编辑数据的 DataList<HR></H3>
<!--布置 DataList 来显示数据-->
<Form runat="server">

```

```
<asp:DataList
    id="DataList1"
    runat="server"
    Border="1"
    BorderColor="Black"
    CellPadding="2"
    CellSpacing="0"
    HeaderStyle-BackColor="#aaaadd"
    AlternatingItemStyle-BackColor="lightgray"
    SelectedItemStyle-BackColor="yellow"
    OnItemCommand="DataList_ItemCommand" >
```

```
<template name="HeaderTemplate">
```

```
姓名( 序号 )
```

```
</template>
```

```
<!--使用模板来显示命令行-->
```

```
<template name="ItemTemplate">
```

```
<%# Container.DataItem("name")%>
```

```
(<%# Container.DataItem("id")%>)
```

```
<asp:LinkButton
```

```
    id="Detail"
```

```
    runat="server"
```

```
    Text="详细情况" />
```

```
<asp:LinkButton
```

```
    id="Edit"
```

```
    runat="server"
```

```
    Text="资料修改" />
```

```
</template>
```

```
<template name="SelectedItemTemplate">
```

```
姓名: <%# Container.DataItem("name")%><BR>
```

```
序号: <%# Container.DataItem("id")%><BR>
```

```
班级: <%# Container.DataItem("class")%><BR>
```

```
电话: <%# Container.DataItem("tel")%><BR>
```

```
住址: <%# Container.DataItem("addr")%><BR>
```

```
<asp:LinkButton
```

```
    id="Title"
```

```
    runat="server"
```

```
Text="返回"
/>
</template>

<template name="EditItemTemplate">
姓名: <asp:Label
    id="Name"
    runat="server"
    Text='<%# Container.DataItem("name")%>' />
    <BR>
序号: <asp:Label
    id="Number"
    runat="server"
    Text='<%# Container.DataItem("id")%>' />
    <BR>
班级: <asp:TextBox
    id="class"
    runat="server"
    OnTextChanged="Text_Changed"
    Text='<%# Container.DataItem("class")%>' />
    <BR>
电话: <asp:TextBox
    id="tel"
    runat="server"
    OnTextChanged="Text_Changed"
    Text='<%# Container.DataItem("tel")%>' />
    <BR>
住址: <asp:TextBox
    id="addr"
    runat="server"
    OnTextChanged="Text_Changed"
    Text='<%# Container.DataItem("addr")%>' />
    <BR>
<asp:LinkButton
    id="Update"
    runat="server"
    Text="更新" />
<asp:LinkButton id="Cancel" runat="server" Text="取消" />
```

```

</template>
</asp:DataList>
<Input Type="Hidden" id="UpdateString" runat="server">
</Form>
<hr>
<asp:Label id="Label1" runat="server" />
</Body>
</Html>

```

运行结果如图 6 - 9 所示。

在上面的例子中，布置了一个<Input Type="Hidden" id="UpdateString" runat="server">控件，用来记录更改字段的 SQL 语句中的 Update 命令中的更新字符串。例如上面把班级、电话、住址更改之后，UpdateString.Value 变成：

```
class = ' 电子 2000 - 40 'tel = ' 028-7600767-2 'addr = ' 普诚斋 11332-2 '
```

UpdateString.Value 在 Page_Load 事件中首先设定为空：

```

Sub Page_Load(sender As Object, e As EventArgs)
    UpdateString.Value = ""

```

Text_Changed 事件程序，在例子中使用如下：

```

Sub Text_Changed(sender As Object, e As EventArgs)
    Dim Field As String
    '首先判断哪一个字段被修改
    Select Case sender.Id
        Case "class"
            Field = "class"
        Case "tel"
            Field = "tel"
        Case "addr"
            Field = "addr"
    End Select
    '将字段名称和修改后的字段值连接成 UpdateString
    If Len(UpdateString.Value) = 0 Then
        UpdateString.Value = Field & " = " & sender.Text & ""
    Else
        UpdateString.Value &= ", " & Field & " = " & sender.Text & ""
    End If
End Sub

```

在更改的过程中，只要文本框里的数据发生改变，就会调用 Text_Changed 事件，最后形成 UpdateString。比如：只是班级被改变，则 UpdateString.Value 变成

```
class = ' 电子 2000 - 40 '
```



图 6 - 9 可以编辑数据的 DataList

如果三个都改变，则 UpdateString.Value 变成

```
class = ' 电子 2000 - 40 ',tel = '028-7600767-2',addr = ' 誉诚斋 11332-2 '
```

按下更新时所执行的程序如下：

```
Dim ConnStr, SQL As String
If Len(UpdateString.Value) <> 0 Then
Dim Number As Label = e.Item.FindControl("Number")
SQL = "Update neta_station Set " & UpdateString.Value & _
    " Where id = " & Number.Text
ConnStr = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=" & Server.MapPath( "Sample.mdb" )
Dim Cmd As ADOCommand = New ADOCommand( SQL, ConnStr )
    Cmd.ActiveConnection.Open()
    Cmd.Execute()
End If
```

首先利用 e.Item.FindControl 找到 EditItemTemplate 中的 Number 控件，执行上面的语句后，Number 变量就等于 EditItemTemplate 中的 Number 控件的值了。

6.6 XML 与 DataSet 对象

本文将介绍如何使用“可扩展的标记语言(XML)”作为数据的标准格式。它概述了 XML 是什么、它产生的原因以及为什么它对表示和交换数据来说是极有价值和技术。

Web 能够使任何地方的任何人之间相互通讯。广泛接受的标准（这对完全使用 Web 的潜力是至关重要的）允许 Web 在多种交互操作的技术层上通讯。一个重要层是可视化显示和用户界面，例如 HTML、GIF 和 JavaScript（TM）之类的现有标准。这些标准允许创建一次页面，并且在不同时间向众多接收者显示。

尽管可视化和用户界面标准是必要层，但它们并不足以表示和管理数据。当前，Internet 只是文本和图片的访问媒体，并没有智能搜索、数据交换、自适应表示和个人化的标准。Internet 必须突破现行信息访问和显示标准的限制；Internet 必须设置信息理解标准（表示数据的通用方式），以使用户端能够更好地搜索、移动、显示和处理上下文中的信息。由于 HTML 是一种描述如何表示 Web 页的格式，所以 HTML 并不能完成以上处理；HTML 并不表示数据。例如，HTML 不能执行如下操作：

- 向医生提供将处方发送到药剂师的标准方式。
- 允许医药实验室以所有接收者均能分析的格式发表统计信息。
- 以所有接受者均能解码和处理的表格描述电子支付。
- 提供搜索法律图书馆中有关某一主题的所有诉讼文档的标准方式。
- 指定以何种方式传送公司目录中的信息，以便销售人员可以脱机工作、向客户显示目录、接受定单并以标准格式上载这些定单。

简而言之，尽管 HTML 提供了用于显示的丰富工具，但 HTML 并没有提供任何基于标准的管理数据的方式。

正如数年前用于显示的 HTML 标准扩展了 Internet 一样，数据显示标准亦将扩展 Internet。数据标准将是商业交易、公布个人喜爱的配置文件、自动协作和数据共享的工具。并将以此格式编写医疗记录、制药研究数据、半导体部件图以及采购定单。这将开创众多新用途，这些新用途均基于在 Web 上到处移动结构数据的标准表示，正如当前移动 HTML 页一样容易。数据标准是 XML 和 XML 扩展名。

6.6.1 什么是 XML

XML 是一种元标注语言，该语言提供一种描述结构数据的格式。这有助于更精确地声明内容，方便跨越多种平台的搜索结果。此外，XML 将起用新一代的基于 Web 的数据查看和处理应用程序。

1. 数据的结构表示

XML 提供可以广泛执行并易于配置的数据的结构表示。XML 是万维网协会(W3C)定义的用于优化 Web 上的传输的 SGML 的子集，XML 确保结构数据是一致的，并独立于应用程序或供应商。由此可产生互用性，并启动新一代的商业和电子商务 Web 应用程序。

XML 提供可以编码各种情况(从简单至复杂) 的内容、语义和架构的数据标准，可被用于标注以下对象：

- 普通文档。
- 诸如约会记录或采购定单之类的结构记录。
- 具有数据和方法的对象，如 Java 对象或 ActiveX 控件的持续型表单。
- 数据记录，如查询的结果集。
- 有关 Web 站点的元内容，如“频道定义格式(CDF)”。
- 图形表示，如应用程序的用户界面。
- 标准架构实体和类型。
- Web 上的信息与用户之间的所有链接。

一旦该数据在客户机桌面上，就可以在多个视图中处理、编辑和表示它，而无需返回到服务器。由于较低的计算量和带宽负载，所以目前的服务器更易于升级。此外，由于以 XML 格式交换数据，所以可以轻而易举地合并来自不同来源的数据。

XML 对 Internet 和大型企业的 Intranet 环境是颇有价值的。这是由于 XML 提供使用灵活、开放、基于标准格式的互用性，并具备访问遗留数据库和将数据传输到 Web 客户机的新方式。可以更快地生成应用程序(应用程序更易于维护)，并且可以在结构数据上轻易提供多种视图。

2. XML 文档

XML 是一种基于文本的格式，在许多方面类似于 HTML，后者是专为存储和传输数据而设计的。XML 源是由 XML 元素组成的，每个 XML 元素包括一个开始标记<title>，一个结束标记</title>以及两个标记之间的信息(称为内容)。就像 HTML 一样，XML 文档保存利用标记注释的文本。然而，与 HTML 不同的是，XML 允许无限的标记集，各标记集并不

表示如何显示，而是表示其含义。例如，可以将 XML 元素标记为价格、订单编号或名称。由文档的作者确定使用何种数据以及哪种标记名称最合适。

XML 文档的创建很容易。如果用户熟悉 HTML，那么可以很快地学会采用 XML 格式进行编写。在下面示例中，利用 XML 描述天气预报，可以用 XML 的扩展名保存该文件，如 Weather.xml。

```
<weather-report>
  <date>1998 年 3 月 25 日</date>
  <time>08:00</time>
  <area>
    <city>西雅图</city>
    <state>华盛顿</state>
    <region>西海岸</region>
    <country>美国</country>
  </area>
  <measurements>
    <skies>局部多云</skies>
    <temperature>46</temperature>
    <wind>
      <direction>西南</direction>
      <windspeed>6</windspeed>
    </wind>
    <h-index>51</h-index>
    <humidity>87</humidity>
    <visibility>10</visibility>
    <uv-index>1</uv-index>
  </measurements>
</weather-report>
```

各标记表示各数据项的含义（是否为<日期>元素、<区域>元素等等），而并不描述显示数据的次序和形式。此数据的所有接收者均可将该文档解码，以便用于自己的特定目的。例如，某人可以利用该信息制定当天的计划，而气象研究员可以将其作为西雅图的历史记录。

3. 可扩展的功能

用户可以在 XML 中定义无限的标记集。虽然可以使用 HTML 标记以粗体或斜体的方式显示文字，但 XML 可提供一個用于标记结构数据的框架。XML 元素可以将其关联数据声明为零售价格、营业税、书名、降雨量或其他任何需要的数据元素。随着整个组织以及 Internet 上的其他用户采用 XML 标记，将出现搜索和处理数据的相应能力，而不管发现该数据的应用程序如何。一旦定位了数据，就可以在网络上传送并以多种方式在浏览器中显示，或者传送到其他应用程序以供进一步处理和查看。

标记表示一块数据。通常，它对应于某个表中的某个字段。然而，这并不是必要的。标记可以是一个计算列（价格×数量）。没有理由认为一个 XML 文件表示某个表中的所有数据。正像通常看到的那样，XML 将表示包括许多表的查询结果。只要接收应用程序能够搞清 XML 中的数据的意思，数据是从哪里来的以及该数据如何到达 XML 文件中并不重要。数据是从表示和处理中分离出来的。

XML 的优势在于它保持了用户界面和结构数据之间的分离。HTML 指定如何在浏览器中显示数据，而 XML 则定义内容。在 HTML 中，使用标记告诉浏览器以粗体或斜体的方式显示数据；而在 XML 中，只使用标记来描述数据，如城市名、温度和气压等。在 XML 中，使用诸如“扩展样式语言（XSL）”和“层叠样式表（CSS）”之类的样式表，来表示浏览器中的数据。XML 把数据从表示和处理中分离出来，使用户可以通过应用不同的样式表和应用程序显示和处理数据。

把数据从表示中分离出来，能够无缝集成众多来源的数据。可以将用户信息、采购订单、研究结果、账单支付、医疗记录、目录数据以及其他数据转换为中间层上的 XML，以便像 HTML 页显示数据一样很容易地联机交换数据。然后可以在 Web 上将按照 XML 编码的数据传送到桌面。对于大型数据库或文档中存储的遗留信息无需进行更新，并且由于使用了 HTTP 在网络上传送 XML，所以此功能不需要更改。

4. 使 XML 数据自描述

在 XML 中“文档类型定义（DTD）”可以伴随文档，且文档类型定义实质上是定义文档的规则，例如哪些元素可被显示以及元素之间的结构关系。当接收应用程序没有传入数据的内部描述时，DTD 将帮助验证该数据。然而，在 XML 中 DTD 是可选的。

与 DTD 一起发送的数据称为有效 XML。此时，XML 语法分析程序应对照 DTD 中定义的规则校正传入的数据，以确保该数据被正确构造。没有与 DTD 一起发送的数据称为结构良好的 XML。这里，可以使用基于 XML 的文档实例，如以上显示的层次结构气象数据，可用来隐含地描述它自己。

对于有效 XML 和结构良好的 XML，XML 编码的数据是自描述的，这是由于描述标记和数据是混合的。XML 使用开放、灵活的格式，在任何需要交换和传送信息的地方均可使用 XML，这使得 XML 功能更加强大。

例如，可以使用 XML 描述有关 HTML 页的信息，或使用 XML 描述商业规则中包含的数据或电子商务交易中的对象，如发票、采购订单和定货单等。由于 XML 是与 HTML 分离的，所以可以将 XML 添加到 HTML 文档中。W3C 定义了一种格式，根据该格式，可以将基于 XML 的数据或 XML 数据岛封装到 HTML 页中。通过将 XML 数据岛嵌入到 HTML 页中，并使用 XML 中包含的语义信息，就可以根据传送的数据生成多个视图。此外，还可以使用 XML 强制应用程序，如分布式打印、数据库搜索及其他程序。

5. 架构

架构是 XML 文档规则的正式规格，即元素名称。架构表示文档中允许哪些元素以及允许哪些组合。如 XML 数据工作组中定义的新架构语言，符合 W3C 的 XML 数据和文档内容说明（DCD）”建议提供了和 DTD 一样的功能。但是，这些架构是可扩展的，所以开发人员可用附加的信息（诸如那些数据类型、继承和演示文稿规则）来扩展它们。这使得这

些新的架构语言在功能上比 DTD 强大得多。

除了 XML 数据和 DCD 外, Microsoft 和其他公司还建议用 XML 自身来表达 XML 文档架构的词汇。这使 XML 数据能够描述它自己的结构。表达 XML 内部的架构给 XML 格式添加了巨大的能量, 因为它使检查某些数据的软件了解它的结构成为可能, 而不需用任何事先内建数据结构的描述。

使用架构, 作者正好可以定义在文档中允许什么样的元素名, 以及在每个元素中允许什么样的子元素、属性和关系。作者可从其他架构中引入片断, 以及通过继承来扩展类型。于是, 在保留了词汇树结构简单性的同时, 还允许元素间的复杂关系。

作者可以发明他们自己的架构, 或者共享别的作者创建的架构。读者可以检查架构参考以检验已经收到的文档类型是否正确, 可使用架构中的信息来自动检验文档的结构。

想使用 XML 的公司, 需要一个寻找关于其他公司和应用程序支持的架构、文档和公司进程的信息的简单办法。想象一下, 消费者要付出的巨大代价相当于每一家公司都要定义它自己的发布信息的方法。即使是 Web, 建立和维护 Web 站点的费用也超出了某些公司的能力。有无数的公司可以发布它们的信息, 但是缺少如何用安全的、可控的方式来发布这些信息的标准, 这势必导致成千上万种不同的执行方法、导航方式和内容深度。允许这种“混乱”状态泛滥的成本负担将波及消费者。

Microsoft 已经选择了通过创建和管理 <http://www.biztalk.org/>, 以最大限度地克服此问题。该站点将成长为定位、管理、学习和发布使用在成千上万应用程序中的 XML、XSL 和信息模型的大门。架构的全功能联机仓库已经在 1999 年初秋交付使用。

6. 开放式标准

XML 是基于经过验证的、针对 Web 进行优化的技术标准。Microsoft 正在与其他大公司和 W3C 研究小组共同努力, 确保对工作在多系统和多浏览器上的开发人员、作者和用户的互用性和支持, 并不断加强 XML 标准。

XML 是由下列相关标准组成的:

- 可扩展的标记语言(XML) 是一个“建议”, 即 W3C 审批进程的最后阶段。这就是说, 该标准是稳定的, 并且受到 Web 和工具开发人员的热烈欢迎。

- XML 命名空间是一个“建议”, 它描述了命名空间的语法并支持命名空间认可的 XML 分析程序。

- 文档对象模型(DOM) 1 级是一个“建议”, 它提供了有计划地访问脚本中结构化数据的标准, 使开发人员能够与基于 XML 数据的计算机进行相容的交互。

- 可扩展的样式表语言(XSL) 是当前的工作草稿。XSL 由两个模块组成: “XSL 转换语言”和“XSL 格式化对象”。转换语言可用于为显示而转换 XML。因为 XSL 的两个部分是模块, 所以转换语言可独立应用于通用目的的转换, 包括将 XML 转换为格式更好的 HTML。CSS 可用于简单结构的 XML 数据, 但是只能以接收的次序显示信息。

- XML 链接语言(XLL) 和它的伴侣 XML 指针语言(XPointer) 是当前的工作草稿。XLL 是 XML 链接语言, 它提供了 XML 中的链接, 类似于 HTML 中的链接, 然而功能更强。用 XLL, 链接可以是多向的, 而且链接可存在于对象级, 不是仅在页面级。

注意: Internet Explorer 5 不支持 XLL。

XML 结构化架构, 诸如 XML 数据注释和 XML 的文档内容描述(DCD) 中描述的那些是 W3C XML 数据工作组的课题。

7. 从 XML 中受益

XML 给基于 Web 的应用程序带来了强大的功能和灵活性, 它为开发人员和用户提供了许多引人注目的优点:

- 更有意义的搜索。
- 开发灵活的 Web 应用程序。
- 不同来源的数据集成。
- 本地的数据计算和操作。
- 多个数据视图。
- 粒状更新。
- 更有意义的搜索。

数据可使用 XML 来作惟一标记, 潜在地允许客户指定对自己感兴趣的信息进行搜索的标准。不使用 XML, 则需要搜索应用程序以了解每个数据库的架构, 此架构描述了该数据库的构造。事实上这是不可行的, 因为每个数据库描述它的数据的方式不同。而使用 XML 可以智能化地满足客户搜索要求。

8. 灵活的 Web 应用程序的开发

找到数据之后, 可将 XML 交给其他应用程序、对象和中间层服务器作下一步处理, 或者送至桌面, 在浏览器中查看。XML 连同用于显示的 HTML、用于逻辑的脚本以及用于数据的交互和显示的一般对象模型一起, 为灵活的三层 Web 应用程序开发提供了所需的技术。

9. 不同来源的数据集成

搜索多个不兼容的数据库的能力, 至今尚属不可能。XML 启用了容易结合的不同来源的结构化数据。软件代理可用于在中间层服务器上, 集成来自后端数据库和其他应用程序的数据。这数据再递交给客户机或其他服务器, 作下一步的聚合、处理和分布。

XML 的可扩展性和灵活性, 使它能够描述各种类型的应用程序中包含的数据, 这些应用从描述 Web 页面的集合到数据记录, 应有尽有。此外, 由于基于 XML 的数据是自描述的, 不需要内建的传入数据的描述, 便能交换和处理数据。

10. 本地计算和操作

在 XML 格式的数据被递交给客户机之后, 该数据可以使用客户机应用程序执行的计算进行剖析, 并在本地进行编辑和操作。用户不仅仅可以显示数据, 还可使用各种方法操作数据。“XML 文档对象模型(DOM)”还允许使用脚本或其他编程语言操作数据。省略了和服务端之间的数据传输, 便可进行数据计算。将查看数据的用户界面和数据本身分离, 使得从前只能在高端数据库上找到的功能强大的应用程序, 现在即可以使用简单、灵活、开放的格式创建。

11. 多个数据视图

数据被传递到桌面之后, 可使用不同的方法查看。通过使用简单、开放和可扩展的方式描述结构化数据, XML 补充了广泛用于描述用户界面的 HTML。另外, HTML 描述了数

据的外观,而 XML 则描述了数据本身。由于现在显示与数据分离了,使用 XML 定义的数据可以指定多个不同的视图,使数据得以更好地显示。可以根据客户机配置、用户喜好和其他准则决定本地数据的显示方式。CSS 和 XSL 提供了描述特定数据视图的说明性机制。

12. 粒状更新

数据可以使用 XML 粒状更新,每当部分数据被更改后,不需要再发送一个完整的结构化数据集。只是将更改了的元素从服务器发送到客户机上,而且用不着刷新整个用户界面就可以显示更改了的数据。现在,即使视图保持不变,只更改了一个数据项,整个页面也必须重建。这严重限制了服务器的可缩放性。

同样,XML 允许添加其他数据,如预报的最高和最低温度、预期的降雨量和概率(按百分比)。此附加的信息将流入用户的现有视图,而不使用浏览器发送新的视图。如果还需要其他信息(如大气压),不用重建便可以发送。

13. 前景

作为表达结构化数据的行业标准,XML 向组织、软件开发人员、Web 站点和最终用户提供了许多功能。随着重要市场(如高级数据库搜索、网上银行、医药、法律、电子商务和其他领域)的更为高级的市场数据格式被创建,这种功能将进一步增强。当站点不但要查看数据,而且要分配数据时,正是良机。

当前,客户服务正从呼叫中心和物理位置移向 Web 站点,因此将受益于 XML 的强大功能。其次,因为大多数业务应用程序涉及数据和数据库记录的传输,如订货单、发票、客户信息、约会、地图等等,XML 将在 Internet 上执行大量业务应用程序,从而大大改变最终用户的可能发生的事情。此外,Web 站点上的信息,不管是存在文档中,还是在数据库中,都可以使用基于 XML 的面向 Intranet 的词汇来标记。这些词汇,对需要在客户和供应商之间交换信息的中小型公司来说,也很有帮助。

一个重要的、尚待开发的市场,就是便于最终用户建造自己的协作 Web 站点的开发工具,包括从遗留数据库信息和现存的用户界面生成 XML 数据的工具。此外,还应开发描述公文包或其他数据的标准架构,例如,它应该能使用排版、图表和 Excel 等电子表格的其他功能。描述从遗留数据库生成的 XML 的说明性工具和可视工具是非常好的机会。查看 XML 数据的自定义工具可使用 Visual Basic 开发系统、Java 和 C++ 编写。

XML 要求强大的新工具来表示文档中丰富的、复杂的 XML 数据。通过将对用户友好的显示层映射到可动态更改的复杂的层次数据集的顶层,可以实现这一点。XML 数据可能使用的布局,包括折叠分级、PivotTable 动态视图和每个公文包的简单表格。

Web 站点可以提供股票行情、新闻或实时交通数据。通过过滤 Web 广播或智能轮询复制这些站点的服务器树,可以获得这些数据。使用 XML 可以避免信息超载,办法是编写自定义的规则来处理电子邮件中的陈旧的信息。基于 XML 的、用户构造这些规则的工具,以及运行这些工具的服务器和客户机软件,是个极好的巨大市场。“标准的对象模型”能启用这些功能,它们通常被写在脚本中,用于过滤传入的消息、检查存储的消息、创建待发邮件、访问数据库等等。可编写这些代理程序,以便在任何地方自动运行。

6.6.2 XML 格式的数据库文件

可以建立如下数据库文件 ch06/info.xml :

```
<NewDataSet>
  <xsd:schema id="NewDataSet"
    targetNamespace=""
    xmlns=""
    xmlns:xsd="http://www.w3.org/1999/XMLSchema"
    xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xsd:element name="登记表">
      <xsd:complexType content="elementOnly">
        <xsd:all>
          <xsd:element name="序号" minOccurs="0" type="xsd:int"/>
          <xsd:element name="姓名" minOccurs="0" type="xsd:string"/>
          <xsd:element name="班级" minOccurs="0" type="xsd:string"/>
          <xsd:element name="电话" minOccurs="0" type="xsd:string"/>
          <xsd:element name="住址" minOccurs="0" type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="NewDataSet"
      msdata:IsDataSet="True">
      <xsd:complexType>
        <xsd:choice maxOccurs="unbounded">
          <xsd:element ref="登记表"/>
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>

  <登记表>
    <序号>1</序号>
    <姓名>机器猫</姓名>
    <班级>机械 2 班</班级>
    <电话>760000</电话>
    <住址>东门一号</住址>
  </登记表>
```

<登记表>

<序号>2</序号>

<姓名>一休</姓名>

<班级>哲学一班</班级>

<电话>1382000000</电话>

<住址>日本东京</住址>

</登记表>

<登记表>

<序号>3</序号>

<姓名>唐老鸭</姓名>

<班级>中文 6 班</班级>

<电话>0139524252121</电话>

<住址>美国</住址>

</登记表>

<登记表>

<序号>4</序号>

<姓名>米老鼠</姓名>

<班级>计算机 7 班</班级>

<电话>013625214521</电话>

<住址>美国</住址>

</登记表>

<登记表>

<序号>5</序号>

<姓名>蜡笔小新</姓名>

<班级>艺术 9 班</班级>

<电话>013512346462</电话>

<住址>日本</住址>

</登记表>

<登记表>

<序号>6</序号>

<姓名>大通毅</姓名>

<班级>足球 1 班</班级>

<电话>013713123156</电话>

<住址>日本</住址>

</登记表>

```

<登记表>
  <序号>7</序号>
  <姓名>大嘴鳄鱼</姓名>
  <班级>生物系</班级>
  <电话>013945213321</电话>
  <住址>北京</住址>
</登记表>

```

```
</NewDataSet>
```

访问 XML 数据库的程序 ch06/test6_08vb.aspx 的源代码如下：

```

<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<script Language="VB" runat="server">
    Dim ds As DataSet = new DataSet()
    在页面被浏览过程中，调用 Page_Load 程序事件
    Sub Page_Load(sender As Object, e As EventArgs)
        '定义文件操作对象 FileStream
        Dim fs As FileStream
        Dim reader As StreamReader
        Dim Path As String
        Path = Server.MapPath( "info.xml" )
        '读取文件
        fs = New FileStream(Path, FileMode.Open, FileAccess.Read)
        reader = New StreamReader(fs, Encoding.Default)
        ds.ReadXml(reader)
        '建立数据源
        Grid1.DataSource = ds.Tables("登记表").DefaultView
        '绑定数据
        Grid1.DataBind()
    End Sub
    Sub ChangePage(sender As Object, e As DataGridPageChangedEventArgs)
        Grid1.DataSource = ds.Tables("登记表").DefaultView
        Grid1.DataBind()
    End Sub
</script>

<Html>
<Body BgColor="White">

```


读取 XML 文件之后，会得到一个 StreamReader 对象，要将此对象传入 DataSet 对象的 ReadXML 方法，就可以读取 XML 格式的数据了。

6.7 本章小结

本章主要讲述了在 ASP.Net 中访问 Web 数据库的初步知识，同时还介绍了目前最流行的 XML 语言，以及如何访问 XML 格式的数据库。

第 7 章

实 例 介 绍

主要内容：

- | 留言簿
- | 客户注册系统
- | 文件上传
- | 发关 E-mail
- | 差错及排除错误
- | ASP.Net 中的 Session 对象
- | 提高执行效率的 Cache

本章通过具体的实例，进一步讲解如何运用 ASP.Net。此外，还讲解了如何查错、提高服务器的执行效率等。

7.1 留 言 簿

使用 SQLServer 为后台数据库编写一个简单留言簿，其程序 ch07/test7_01.aspx 的源代码如下：

```
<%@ Import Namespace="System" %>
<%@ Import Namespace="System.Text" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SQL" %>
<!--显示中文-->
<%@page responseencoding="gb2312" %>
<script language="C#" runat="server">
    void Page_Load(Object sender, EventArgs e) {
        //连接 SQLServer
        SqlConnection myConnection = new SqlConnection
        ("server=127.0.0.1;uid=bible_cjiao;pwd=cjiao;database=singnet");
```

```

        SqlDataReader myreader;
        if (Request.QueryString["action"]=="del") {
            //执行删除
            String delCmd="Delete From GuestBook Where book_id=" + Request.QueryString["book_id"];
            SqlCommand myDelCommand = new SqlCommand(delCmd.ToString(),myConnection);
            myDelCommand.ActiveConnection.Open( );
            myDelCommand.Execute(out myreader);
            Message.InnerHtml = "<B>删除留言！</B>";
            myDelCommand.ActiveConnection.Close();
        }
        if (Request.Form["action"]=="save") {
            //保存留言
            String insertCmd = "Insert Into GuestBook(UserName,Email,OICQ,Post_time,Content)
                Values('" + Request.Form["UserName"] + "','" + Request.Form["Email"] + "','"
                + Request.Form["OICQ"] + "','" + DateTime.Now + "','" + Request.Form["Content"] + "')";
            SqlCommand myInsertCommand = new SqlCommand(insertCmd.ToString(),myConnection);
            myInsertCommand.ActiveConnection.Open( );
            myInsertCommand.Execute(out myreader);
            Message.InnerHtml = "<B>留言成功！</B>";
            myInsertCommand.ActiveConnection.Close( );
        }

        SQLDataSetCommand myCommand = new SQLDataSetCommand("select * from GuestBook
            Order By post_time DESC",myConnection);
        DataSet ds = new DataSet( );
        myCommand.FillDataSet(ds, "UserName");
        MyDataList.DataSource = ds.Tables["UserName"].DefaultView;
        MyDataList.DataBind( );
    }
</script>
<html>
<body>
<title>留言板</title>
<center>
<table>
<div id=Message runat=server></div>
    <p>
        <a href="test7_01.aspx">刷新</a>
    </p>
<form action=test7_01.aspx method=post>

```

```

<!--留言-->
    <tr><td>大名 :
        </td>
        <td><input type=text name=UserName size=10>
        </td>
    </tr>
    <tr><td>您的电子邮件 :
        </td>
        <td><input type=text name=Email size=10>
        </td>
    </tr>
    <tr>
        <td><td>OICQ :
        </td>
        <td><input type=text name=OICQ size=10>
        </td>
    </tr>
    <tr><td valign=top>留言 :
        </td>
        <td><textarea colspan=50 rows=6 name=content></textarea>
        </td>
    </tr>
    <tr>
        <td colspan=2><input type=submit value="提交">
        </td>
    </tr>
    <input type=hidden name=action value=save>
</table>
</form>
<table>
    <ASP:Repeater id="MyDataList" runat="server">
    <!--使用模板显示留言-->
        <template name="itemtemplate">
            <tr>
                <td>大名 :
            </td>
            <td>
                <%=# DataBinder.Eval(Container.DataItem, "UserName") %><a
href="test7_01.aspx?action=del&book_id=<%=# DataBinder.Eval(Container.DataItem, "book_id") %>">删除留言
            </a>

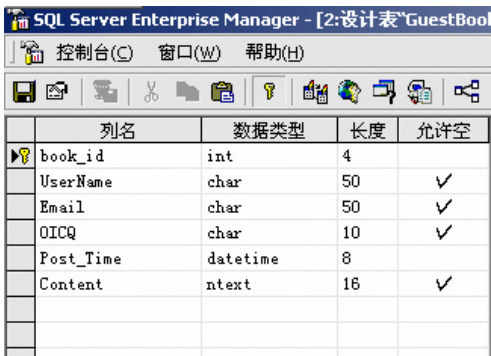
```

```

        </td>
    </tr>
    <tr>
        <td>电子邮件 :
    </td>
    <td><%# DataBinder.Eval(Container.DataItem, "Email") %>
    </td>
    </tr>
    <tr>
        <td>OICQ :
    </td>
    <td><%# DataBinder.Eval(Container.DataItem, "OICQ") %>
    </td>
    </tr>
</tr>
    <tr>
        <td>留言时间 :
    </td>
    <td><%# DataBinder.Eval(Container.DataItem, "Post_Time") %>
    </td>
    </tr>
    <tr>
        <td>留言内容 : </td>
    <td><%# DataBinder.Eval(Container.DataItem, "Content") %></td>
    </tr>
    <tr>
        <td colspan=2 height=10>
    </td>
    </tr>
    <tr>
        <td colspan=2 height=1 bgcolor=black> </td>
    </tr>
</tr>
    <tr> <td colspan=2 height=10></td>
    </tr>
    </template>
</ASP:Repeater>
</table>
</body>
</html>

```

在上面的例子中，在数据库中设计的数据表格 GuestBook 如图 7 - 1 所示。
运行结果如图 7 - 2 所示。



	列名	数据类型	长度	允许空
	book_id	int	4	
	UserName	char	50	✓
	Email	char	50	✓
	QICQ	char	10	✓
	Post_Time	datetime	8	
	Content	ntext	16	✓

图 7 - 1 留言簿数据表格结构



留言成功！

刷新

大名:

您的电子邮件:

OICQ:

留言:

提交

大名: ajiao 删除留言

电子邮件: j3c@263.net

OICQ: 40223047

留言时间: 2001-05-17T20:52:33

留言内容: 我希望明天是一个晴天！

图 7 - 2 留言簿程序运行结果

7.2 客户注册系统

使用 SQLServer 为后台数据库提供通用的用户注册系统，用户提交的资料保存在数据库中。在注册时，对用户提交的数据进行检验，只有通过验证才能够注册成功。

例如，其程序 ch07/test7_02.aspx 的源代码如下：

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SQL" %>
<html>
<head><title>ASP.NET 注册</title>
<Script Language="C#" RunAt=Server>
void btnSubmit_Click(Object Src,EventArgs E)
{
    //如果页面是正确的
    if(Page.IsValid)
    {
        String strSQL;
        strSQL="INSERT INTO User_Info VALUES (@UserName,@PassWord,@Email)";
        //连接数据库
        SqlConnection cnn=new SqlConnection("Server=127.0.0.1;UID=bible_cjiao;PWD=cjiao;
        DataBase=singnet");
        SqlCommand cmm=new SqlCommand(strSQL,cnn);
        cmm.Parameters.Add(new SqlParameter("@UserName",SQLDataType.VarChar,20));
```

```
cmm.Parameters["@UserName"].Value=txtName.Text;
cmm.Parameters.Add(new SqlParameter("@PassWord",SQLDataType.VarChar,20));
cmm.Parameters["@PassWord"].Value=txtPassWord.Text;
cmm.Parameters.Add(new SqlParameter("@Email",SQLDataType.VarChar,60));
cmm.Parameters["@Email"].Value=txtEmail.Text;
cmm.ActiveConnection.Open();
try
{
    cmm.Execute();
    cmm.ActiveConnection.Close();
//页面转到 result.aspx
    Page.Navigate("result.aspx");
}
//捕获错误
catch
{
    lblMsg.Style["color"]="red";
    lblMsg.Text="插入数据出错";
}
}
}
</Script>
</head>
<body>
<h2 align="center">ASP.NET 注册程序</h2>
<ASP:Label ID="lblMsg" RunAt=Server/>
<form RunAt=Server>
    <table bgcolor="#eeeeee" cellpadding="10" align="center">
        <tr>
            <!--填写姓名等注册内容-->
            <td>姓名: </td>
            <td>
                <ASP:TextBox ID="txtName"
                    MaxLength="20"
                    RunAt=Server
                />
            </td>
            <td>
                <!--布置验证规则，使得姓名栏为必须填写-->
            </td>
        </tr>
    </table>
</form>
</body>
</html>
```

```

        <ASP:RequiredFieldValidator ID="rfvName"
            ControlToValidate="txtName"
            Display="Dynamic"
            InitialValue=""
            ErrorMessage="*"
            RunAt=Server>
    </ASP:RequiredFieldValidator>
</td>
</tr>
<tr>
    <td>密码 : </td>
    <td>
        <ASP:TextBox ID="txtPassWord"
            TextMode="PassWord"
            MaxLength="20"
            RunAt=Server/>

    </td>
    <td>
        <!-- 布置验证规则，使得密码栏为必须填写-->
        <ASP:RequiredFieldValidator ID="rfvPassWord"
            ControlToValidate="txtPassWord"
            Display="Dynamic"
            InitialValue=""
            ErrorMessage="*"
            RunAt=Server>
    </ASP:RequiredFieldValidator>
    </td>
</tr>
<tr>
    <td>确认密码 :
    </td>
    <td>
        <ASP:TextBox ID="txtRePassWord"
            TextMode="PassWord"
            MaxLength="20"
            RunAt=Server
        />

    </td>
    <td>

```

```

<!--布置验证规则，使得验证密码栏为必须填写-->
    <ASP:RequiredFieldValidator ID="rfvRePassWord"
        ControlToValidate="txtRePassWord"
        Display="Dynamic"
        InitialValue=""
        ErrorMessage="*"
        RunAt=Server>
</ASP:RequiredFieldValidator>
<!--验证两次输入的密码是否一致-->
    <ASP:CompareValidator ID="cvRePassWord"
        ControlToValidate="txtRePassWord" ControlToCompare="txtPassWord"
        Display="Dynamic"
        ErrorMessage="密码确认错，请注意字母大小写！"
        Runat=Server>
</ASP:CompareValidator>
</td>
</tr>
<tr>
    <td>电子邮件：</td>
    <td>
        <ASP:TextBox ID="txtEmail" MaxLength="60" RunAt=Server/>
    </td>
    <td>
        <!--布置验证规则，使得 Email 栏为必须填写-->
        <ASP:RequiredFieldValidator ID="rvfEmail"
            ControlToValidate="txtEmail"
            InitialValue=""
            Display="Dynamic"
            ErrorMessage="*"
            RunAt=Server>
        </ASP:RequiredFieldValidator>
        <!--验证电子邮件的合法性-->
        <ASP:RegularExpressionValidator ID="revEmail"
            ControlToValidate="txtEmail"
            Display="Dynamic"
            ErrorMessage="无效电子邮件地址！"
            ValidationExpression="^[\\w-]+@[\\w-]+\\. (com|net|org|edu|mil)$"
            RunAt=Server>
        </ASP:RegularExpressionValidator>
    </td>
</tr>

```

```

        </td>
    </tr>
    <tr>
        <td colspan="3" align="center">
            <ASP:Button ID="btnSubmit"
                OnClick="btnSubmit_Click"
                Text="提交"
                RunAt=Server/>
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

用来显示注册成员列表的程序 ch07/result.aspx 的源代码如下：

```

<%@ Page language="C#" Debug="true"%>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.ADO" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<script language="C#" runat="server">
    //使用 ICollection 来建立数据视图
    ICollection CreateDataView(string Db,string table) {
        string connstr;
        //连接本地 SQLServer 数据库
        connstr="Provider=SQLOLEDB;Data Source=(local);Initial Catalog=singnet;User ID=bible_cjiao;
            Password=cjiao;";
        // 声明 ADODatasetCommand
        ADODatasetCommand cmd;
        cmd=new ADODatasetCommand("Select UserName,Email from "+table,connstr);
        DataSet ds = new DataSet();
        cmd.FillDataSet(ds,table);
        DataView dv = new DataView(ds.Tables[table]);
        //函数返回值 dv
        return dv;
    }
    void Page_Load(Object sender, EventArgs e) {
        MyDataGrid.DataSource = CreateDataView("singnet","User_Info");
    }
}

```

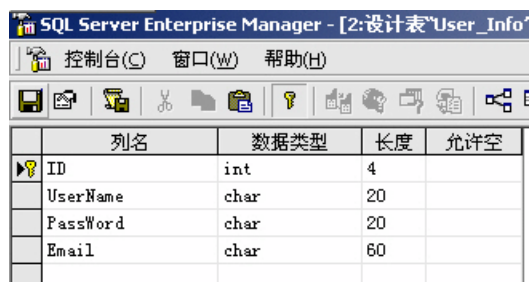
```

        MyDataGrid.DataBind();
    }
</script>
<title>注册用户列表</title><body>
<h3><font face="Verdana">注册用户列表</font></h3>
<form runat=server>
    <!--布置 DataGrid 显示用户列表-->
    <ASP:DataGrid id="MyDataGrid" runat="server"
        BorderColor="black"
        BorderWidth="1"
        GridLines="Both"
        CellPadding="3"
        CellSpacing="0"
        Font-Name="Verdana"
        Font-Size="8pt"
        HeaderStyle-BackColor="#aaaadd"
    />
</form>
</body>
</html>

```

在数据库中建立的数据表格 User_Info，如图 7-3 所示。

运行结果如图 7-4 所示。



列名	数据类型	长度	允许空
ID	int	4	
UserName	char	20	
Password	char	20	
Email	char	60	

图 7-3 数据表格 User_Info 的结构



ASP.NET注册程序

姓名:

密码:

确认密码:

电子邮件:

图 7-4 注册程序运行结果

注册用户列表如图 7-5 所示。

运行以后，在数据库表格 User_Info 中的数据如图 7-6 所示。

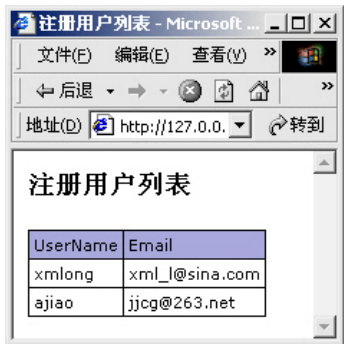


图 7-5 显示注册用户列表

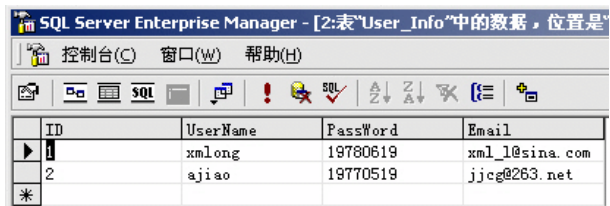


图 7-6 表格 User_Info 中的数据

7.3 文件上传

在 ASP 中，经常使用组件来上传文件到服务器。在 ASP.Net 中，已经内置了文件上传功能，使用 `HtmlInputFile` 控件可轻松上传文件到服务器。其程序 `ch07/test7_03.aspx` 的源代码如下：

```
<script language="VB" runat="server">
    Sub UploadFile(sender As Object, e As EventArgs)
        Dim I As Integer
        Dim FInfo As String
        '利用循环上传多个文件
        For I = 0 To Request.Files.Count - 1
            Dim PostedFile As HttpPostedFile = Request.Files(I)
            If PostedFile.ContentLength > 0 Then
                ' 显示上传成功文件的信息
                FInfo = FInfo & "文件名称: " & PostedFile.FileName & "<br>"
                FInfo = FInfo & "文件大小: " & PostedFile.ContentLength & "<br>"
                FInfo = FInfo & "文件类型: " & PostedFile.ContentType & "<p>"
                ' 使用 Split 取文件名文件
                Dim FileSplit() As String = Split( PostedFile.FileName, "\")
                '取上传的文件名
                Dim FileName As String = FileSplit(FileSplit.Length-1)
                ' 使用 SaveAs 方法将文件保存到当前路径
                PostedFile.SaveAs( Server.MapPath(".") & "\ " & FileName )
            End If
        Next
        FileInfo.InnerHtml = FInfo
    End Sub
```

```

</script>
<Html>
<Body BgColor=White>
<H3>上传多个文件<Hr></H3>
<!--布置 Form 表单-->
<Form Enctype="multipart/form-data" runat="server">
选择上传文件：<br>
<Input Type="File"
        id="FileUp1"
        runat="server"><br>
<Input Type="File"
        id="FileUp2"
        runat="server"><br>
<Input Type="File"
        id="FileUp3"
        runat="server">
<br>
<!--布置按钮控件按下调用 UploadFile 事件程序 -->
<Asp:button id="Upload"
        OnClick="UploadFile"
        Text="上传"
        runat="server"/>

</form><Hr>
<!--显示上传文件信息-->
<Div id="FileInfo" runat="server"/>
</body></html>

```

运行结果如图 7-7 所示。



图 7-7 文件上传程序运行结果

单击“上传”按钮，文件被上传到服务器当前目录中，并显示如图 7-8 所示的文件信息。



图 7-8 显示的上传文件信息

7.4 发送 E-mail

在 ASP.Net 中发送电子邮件已经成为 Web 程序设计的基本对象，通过实例来说明如何发送电子邮件(E-mail)。

```
<%@ Import Namespace="System.Web.Util" %>
<script Language="VB" runat="server">
    Sub Button1_Click(sender As Object, e As EventArgs)
        Dim mail As MailMessage = New MailMessage
        '设定 mail 的属性
        mail.To = mailTo.Value
        mail.From = mailFrom.Value
        mail.Subject = mailSubject.Value
        If Format.SelectedItem.Text = "纯文本格式" Then
            mail.BodyFormat = MailFormat.Text
        Else
            mail.BodyFormat = MailFormat.Html
        End If
        mail.Body = mailBody.Value
        '发送邮件
        SmtpMail.Send(mail)
        '显示发送信息
        Label1.Text = "邮件顺利发送成功！"
    End Sub
</script>
<Html>
<Body BgColor="White">
```

```
<H2>发送电子邮件<Hr></H2>
<Form runat="server">
<Table Border=1>
  <Tr>
    <Td>收件人 : </Td>
    <Td><Input Type="Text" id="mailTo" Size=40 runat="server"/>
    </Td>
  </Tr>
  <Tr>
    <Td>发件人 : </Td>
    <Td><Input Type="Text"
      id="mailFrom"
      Size=40
      runat="server"/>
    </Td>
  </Tr>
  <Tr>
    <Td>标题 : </Td>
    <Td><Input Type="Text"
      id="mailSubject"
      Size=40
      runat="server"/>
    </Td>
  </Tr>
  <Tr>
    <Td>内容 : </Td>
    <Td><TextArea id="mailBody" Rows=5 Cols=40 runat="server">
</TextArea>
    <Br>
    <Asp:RadioButtonList id="Format"
      runat="server"
      RepeatDirection="Horizontal"
      RepeatColumns="2">
      <asp:ListItem Selected>Html 格式</asp:ListItem>
      <asp:ListItem>纯文本格式</asp:ListItem>
    </Asp:RadioButtonList>
    </Td>
  </Tr>
</Table>
```

```
<Input Type="Button"
      Value="发送"
      id="Button1"
      OnServerClick="Button1_Click"
      runat="server"/>

</Form>
<Hr>
<asp:Label id="Label1"
      runat="server"/>

</Body>
</Html>
```

如果想在本地发送邮件，则在 IIS 信息服务里面必须安装 SMTP 服务，并且 SMTP 服务必须处于运行状态，如图 7 - 9 所示。

填写好以后，单击“ 发送 ”按钮，就可以发送电子邮件。如果发送成功，则给以提示。运行结果如图 7 - 10 所示。

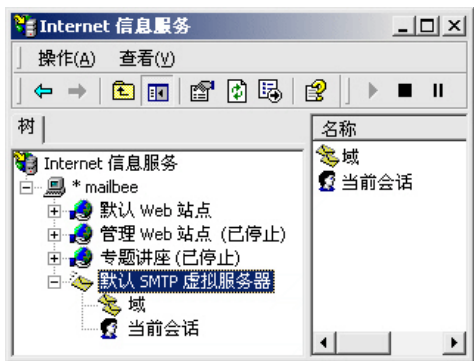


图 7 - 9 Internet 信息服务中的 SMTP 服务



图 7 - 10 发送邮件运行结果程序界面

通过设定如表 7 - 1 所示的 MailMessage 的属性来实现邮件抄送、暗送等等。

表 7 - 1 MailMessage 部分属性说明

属 性	说 明
To	收件人
From	发件人
Cc	抄送
Bcc	暗送
Subject	邮件主题
BodyFormat	信件格式纯文本格式、HTML 格式

7.5 差错及排除错误

在编写程序的时候，错误是不可避免的，在 ASP.NET 中，可以通过如下的方法来检查错误：

- 通过设定 config.web 文件的 customerrors 区段。
- 使用跟踪(Trace)功能。
- 差错工具程序(Debugger)。
- 通过上面的方法可以比较快捷地找出错误，然后来修改。

7.5.1 设定 config.web 区段

在 config.web 中，有如下一段设置：

```
...  
<customerrors mode="RemoteOnly" />  
...
```

其中错误的提示信息如图 7 - 11 所示。

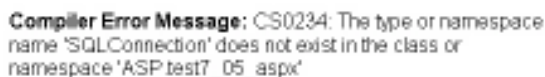


图 7 - 11 编译错误信息

浏览已经出错的网页 ch07/test7_05.aspx 时，会出现如图 7 - 12 所示的结果。



图 7 - 12 网页有错误时出现的错误提示信息

其中在图 7-12 中，错误的第 8 行已经显示出来，根据错误提示，在源程序中缺少声明 Namespace，从而使 SQLConnection 不能使用。

只要将 ch07/test7_05.aspx 的文件头部

```
<%@ Import Namespace="System" %>
<%@ Import Namespace="System.Text" %>
<%@ Import Namespace="System.Data" %>
<!--%@ Import Namespace="System.Data.SQL" %-->
```

更改为

```
<%@ Import Namespace="System" %>
<%@ Import Namespace="System.Text" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SQL" %>
```

就可以顺利地浏览网页了。

7.5.2 使用 Trace 追踪功能

Trace 功能在 config.web 文件中如下：

```
...
<trace
    enabled="false"
    requestlimit="10"
    pageoutput="false"
    tracemode="SortByTime"
/>
...
```

为了在程序执行的同时看到运行状况，在页面的最前面可做如下的标记：

```
<%@ Page Trace="true" %>
```

在浏览页面时，会显示一些程序运行状况的信息，这些都有助于判断程序的运行状况，作为查错的参考。

使用 Trace 功能后，除了正常的显示信息，还显示区段的内容，如表 7-2 所示。

表 7-2 Trace 功能显示区段名称和内容

区 段 名 称	显 示 内 容
Request Details	通过 Request 方式向服务器发送请求的状态
Trace Information	追踪过程的信息
Control Tree	网页使用的控件以及控件之间的关系
Cookies Collection	网页使用的 Cookie
Headers Collection	浏览器的信息
Server Variables	服务器变量

运行结果如图 7-13 所示。



图 7-13 使用 Trace 功能时的运行结果

ASP.NET 的 Trace 功能除了可以显示上述的内容外，还可以人为地显示程序执行过程中在 Trace Information 区段中的数据，调用 Trace.Write 或 Trace.Warn。

例如，其程序 ch07/test7_07.aspx 的源代码如下：

```
<%@ Page Trace="True" %>
```

```
<script language="VB" runat="server">
```

```
Sub UploadFile(sender As Object, e As EventArgs)
```

```
Trace.Warn( "UploadFile", "开始上传 UploadFile()" )
```

```
Dim I As Integer
```

```
Dim FInfo As String
```

```
For I = 0 To Request.Files.Count - 1
```

```
Trace.Warn( "UploadFile", "上传文件 For I = " & I )
```

```
Dim PostedFile As HttpPostedFile = Request.Files(I)
```

```
If PostedFile.ContentLength > 0 Then
```

```
'显示上传文件信息
Trace.Warn( "UploadFile", "文件名=" & PostedFile.FileName )
Trace.Warn( "UploadFile", "文件大小=" & PostedFile.ContentLength )
Trace.Warn( "UploadFile", "文件类型=" & PostedFile.ContentType )
FInfo = FInfo & "文件名: " & PostedFile.FileName & "<br>"
FInfo = FInfo & "文件大小: " & PostedFile.ContentLength & "<br>"
FInfo = FInfo & "文件类型: " & PostedFile.ContentType & "<p>"

' 保存文件
Dim FileSplit() As String = Split( PostedFile.FileName, "\" )
Dim FileName As String = FileSplit(FileSplit.Length-1)
PostedFile.SaveAs( Server.MapPath(".") & "\" & FileName )
End If
Next

FileInfo.InnerHtml = FInfo
Trace.Warn( "UploadFile", "完成 UploadFile()" )
End Sub
</script>

<Html>
<Body BgColor=White>
<H3>Trace 追踪程序执行过程<Hr></H3>

<Form Enctype="multipart/form-data" runat="server">
请选择要上传的文件<br>
<Input Type="File" id="FileUp1" runat="server">
<br>
<Input Type="File" id="FileUp2" runat="server">
<br>
<Input Type="File" id="FileUp3" runat="server">
<br>
<Asp:button id="Upload"
OnClick="UploadFile"
Text="上传"
runat="server"/>

</form><Hr>
<Div id="FileInfo" runat="server"/>
</body>
</html>
```

运行结果如图 7-14 所示。

aspx.page	Begin RaisePostBackEvent	0.801152	0.000000
UploadFile	开始上传 UploadFile()	3.124493	2.323341
UploadFile	上传文件For I = 0	3.414910	0.290418
UploadFile	文件名=D:\cjiao\test.log	3.414910	0.000000
UploadFile	文件大小=253	3.414910	0.000000
UploadFile	文件类型=application/octet-stream	3.414910	0.000000
UploadFile	上传文件For I = 1	3.765414	0.350504
UploadFile	文件名=D:\cjiao\bible_com1.mdb	3.765414	0.000000
UploadFile	文件大小=137216	3.765414	0.000000
UploadFile	文件类型=application/msaccess	3.765414	0.000000
UploadFile	上传文件For I = 2	3.765414	0.000000
UploadFile	文件名=D:\cjiao\new.htm	3.765414	0.000000
UploadFile	文件大小=102941	3.765414	0.000000
UploadFile	文件类型=text/html	3.765414	0.000000
UploadFile	完成UploadFile()	3.775429	0.010014
aspx.page	End RaisePostBackEvent	3.795458	0.020029
aspx.page	Begin PreRender	3.795458	0.000000

图 7-14 使用 Trace.Warn 方法追踪程序运行过程

7.5.3 查错工具(Debugger)

ASP.Net 提供的 Debugger 工具是查错的最佳工具，它可以方便地设定中断点、逐步执行程序、观察变量以及堆栈的情况。

在使用 Debugger 时，服务器的执行效率比正常方式低，因此 ASP.Net 在缺省设置情况下，以正常模式来编译.aspx 文件。为了查错，必须在 config.web 文件中做如下设定：

```
...
<compilation debugmode="true">
...
```

启动查错工具 DbgUrt.exe(ASP.Net 提供的 Debugger 工具在 C:\Program Files\Microsoft.Net\FrameworkSDK\GuiDebug\DbgUrt.exe 中)，运行 DbgUrt.exe，就可以检测.aspx 文件了。选取菜单 Debug/Processes 即可，如图 7-15 所示。

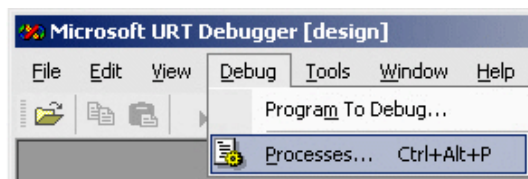


图 7-15 打开进程

这时会出现如图 7-16 所示的对话框。

如果没有看到 xspwp.exe，请先启动浏览器浏览任意的.aspx 网页，单击“ Refresh ”按钮，选取之后单击“ Close ”按钮。然后选择菜单 File/Open/File，如图 7-17 所示。

可以同时选择多个想测试的文件。以 ch07/test7_06.aspx 为例来说明检测程序的步骤。在 DbgUrt.exe 中打开 ch07/test7_06.aspx 文件，运行情况如图 7-18 所示。

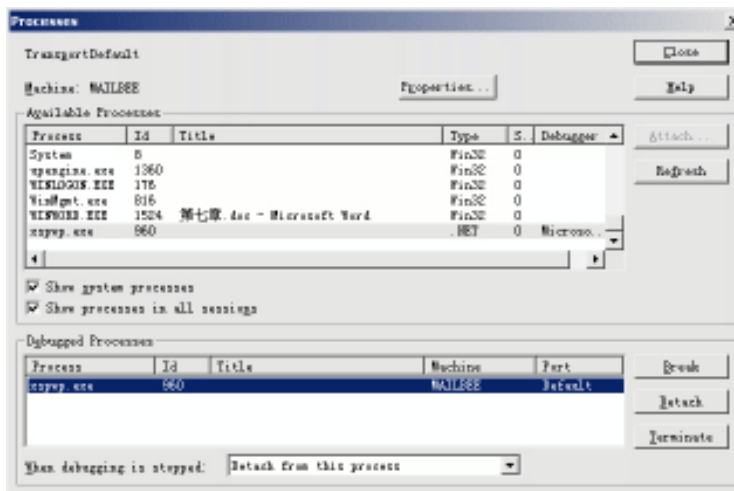


图 7-16 选择 xspwp.exe

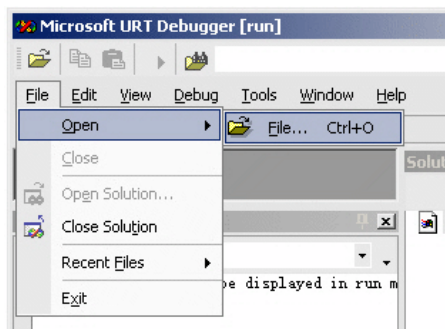


图 7-17 打开要检测的文件

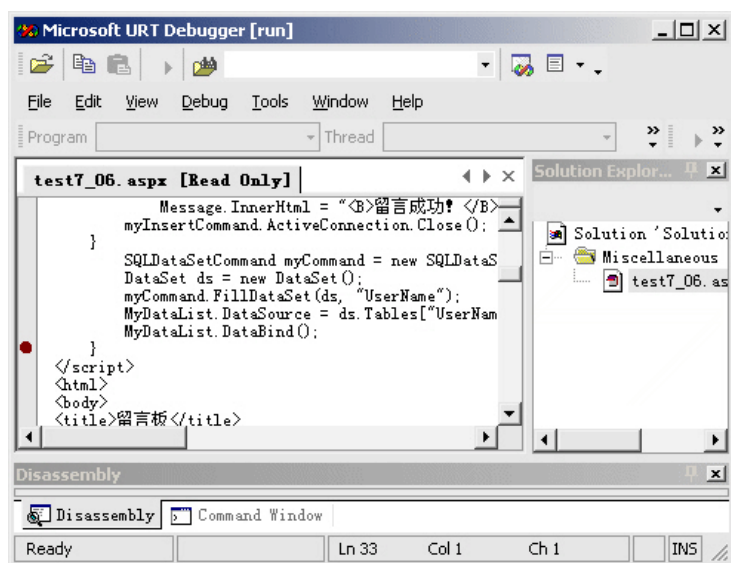


图 7-18 检测 .aspx 文件

设定中断点，将鼠标移动到要检测的中断点，按下 F9 键，换行语句前面出现  符号，表示中断点设置成功。可以在程序中设定多个中断点。取消断点的方法是；在已经设立断点的行按下鼠标，然后按下 F9 键，取消原来设定的断点。

设定好断点之后，使用浏览器浏览页面，则页面就会停在设定断点的位置，Debugger 程序就会变成当前窗口，如图 7-19 所示。

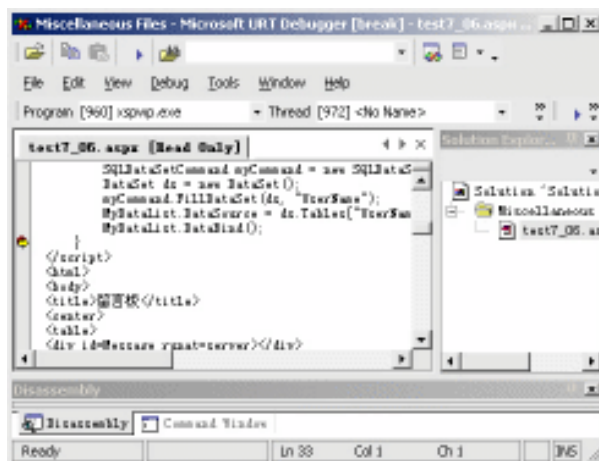


图 7-19 Debugger 使得程序运行到中断点

使用查错工具时，服务器的荷载明显增加，处理速度减慢，此时 Server 会停止所有客户的请求，所以不要在服务器上面进行检测。除了这些查错功能外，利用 Debug 菜单栏，还可以做好多的事情，如图 7-20 所示。

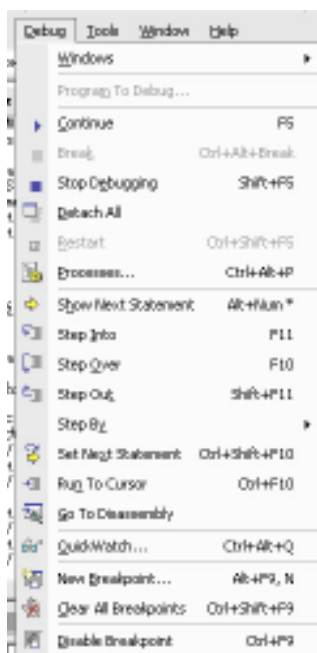


图 7-20 Debug 菜单栏

7.5.4 ASP.Net 中的 Session 对象

在 ASP 中, 如果想使用 Session 对象来记录数据, 则浏览器的 Cookie 功能必须打开。在缺省的情况下, ASP.Net 的 Session 对象依赖于 Cookie, 但是通过 config.web 文件的设置, 可以使得 Session 对象不必依赖 Cookie, 缺省设置如下:

```
<configuration>
...
  <sessionstate
    mode="inproc"
    cookieless="false"
    timeout="20"
    sqlconnectionstring="data source=127.0.0.1;user id=sa;password="
    server="127.0.0.1"
    port="42424"
  />
...
</configuration>
```

只要把下面语句

```
cookieless="false"
```

更改为

```
cookieless="true"
```

就可以使得 ASP.Net 的 Session 对象不依赖 Cookie。

7.5.5 提高执行效率的 Cache

浏览页面时, IIS 首先判断该网页是否含有在服务器端执行的程序, 如果有, 则通过编译以后将其下载到客户端, 否则直接下载到客户端。由于少了服务器端的执行, 因而以 .htm 文件的执行效率高于 .aspx、.asp、.jsp 等需要在服务器端执行的程序网页。

为了提高执行的效率, 对于第一次被浏览的 .aspx 文件, ASP.Net 先将其编译成 MSIL, 并保存。如果同一页面再次被浏览时, 则直接读取 MSIL 加以编译执行, 由此来提高网页的执行效率。

除了上述方法之外, ASP.Net 还提供 Cache(高速缓存) 功能, 也可以提高执行效率。ASP.Net 提供的 Cache 功能分为 Output Cache 和 Data Cache 两种。

Output Cache 的使用方法非常简单, 只要在 .aspx 网页的最前面加上如下语句即可:

```
<%@ OutputCache Duration="秒数" %>
```

Duration 表示在系统中停留的秒数。对于不是经常变动的网页, 使用 OutputCache 可以大大地提高服务器的运行效率。Data Cache 可以将局部数据存储起来。限于篇幅, 在此就不再赘述。

7.6 本章小结

本章中通过实例，介绍了留言本、客户注册系统、文件上传、发送电子邮件以及使用查错工具查错等等。