

ASP.NET

经典案例设计与实现

郭玉峰 师成林 编著
康祥顺 审校



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

ASP.NET经典案例设计与实现

郭玉峰 师成林 编著

康祥顺 审校

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书主要内容是介绍如何利用微软公司最新的ASP.NET 2.0技术开发网络系统的。ASP.NET 2.0是微软公司继ASP.NET 1.x平台后推出的更优秀的网络系统开发技术软件，它不仅吸收了ASP.NET 1.x平台在过去几年中积累的丰富经验和建议，更引入了很多具有前瞻性的新技术。本书的特点是以实例来验证理论，在本书的任何一个章节中，都配有丰富翔实的案例，并以截图和代码来说明问题。书中内容的知识点也是按照学习的一般步骤：由简单到复杂，由肤浅到深入。本书第1章介绍了基础知识，从第2章到第5章，主要以小案例的形式介绍了ASP.NET 2.0中基本控件的用法。从第6章到第8章则用相对关联的案例介绍了ASP.NET 2.0的高级应用，比如数据库的操作、文件的操作、图形处理技术和网络应用开发技术。后面的几章引入了大型的应用系统开发案例，不仅体现在对前面知识点的回顾和巩固，还引入了项目架构设计的知识。因此，读者学习本书，不仅仅是学习ASP.NET 2.0的编程开发技巧！

本书的实战案例是建立在SQL Server 2005数据库环境基础上的，并利用ASP.NET实现对这些数据库的操作，二者结合比较紧密并且彼此也是近乎完美的相互支持，一般项目中都采用这种组合方式，因此，学习本书不仅是对ASP.NET 2.0技术的学习，也是对SQL Server 2005的学习。

本书适合于学习ASP.NET 2.0技术的各层次读者，如果你是初学者，在本书中可以找到丰富的实战技巧，如果你有ASP.NET 2.0技术基础，那么在本书中同样可以学习到如何将ASP.NET 2.0技术有效完善地应用到系统设计的经验和技巧。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

ASP.NET经典案例设计与实现/郭玉峰，师成林编著. —北京：电子工业出版社，2007.7

ISBN 978-7-121-04537-0

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

中国版本图书馆CIP数据核字（2007）第078833号

责任编辑：徐云鹏

特约编辑：底 波

印 刷：北京天竺颖华印刷厂

装 订：三河市金马印装有限公司

出版发行：电子工业出版社

北京市海淀区万寿路173信箱 邮编：100036

北京市海淀区翠微东里甲2号 邮编：100036

开 本：787×1092 1/16 印张：32 字数：820千字

印 次：2007年7月第1次印刷

定 价：49.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至zltts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

前 言

同当年（1985年11月）微软羞涩地推出那个丑陋的Windows 1.0一样，微软在7年前向全世界推出了它的.NET框架。正如当年人们无法预料Windows的走向一样，当年人们用怀疑的目光看着微软的.NET技术。而时至今日，如同惊叹微软的Windows系统一样，我们惊叹微软的.NET技术平台。在这里，本书主要叙述的主角是.NET技术平台下的一个重要的组成部分——ASP.NET 2.0。

基于.NET框架的ASP.NET技术，使开发新一代的网络信息服务平台更加的方便与快捷。并且，在微软随后发布的产品中，几乎全是围绕.NET所展开的。.NET计划正在全世界如火如荼地进行着，基于.NET框架下的电子商务、电子政务等Web应用服务平台也越来越多地出现在人们面前，它的优越性渐渐地显露出来，而ASP.NET编程技术也越来越被程序员们所喜爱。如果说ASP.NET 1.x是蹒跚学步的话，那么ASP.NET 2.0就已经实现了慢跑，而在不久的将来，ASP.NET 3.0或是ASP.NET 4.0，就可以展翅飞翔了。

我们还是再次温习一下.NET的概念吧。.NET框架是个开放的结构体系，所有的代码都被编译成中间语言，并且受框架的托管。ASP.NET严格基于.NET框架，并为开发Web应用程序提供了一个革命性的编程模型。一方面放弃了以往ASP的解释型运行机制，在ASP.NET中，所有的业务逻辑代码和前台的HTML标记代码是分离的，并且所有的业务逻辑代码都是先编译后执行的；另一方面，ASP.NET提供了丰富的类库，在类库中包括了丰富的控件和基本结构，Web程序员可以像使用Visual Basic一样快速高效地搭建Web应用程序。ASP.NET从根本上解决了编程语言的问题，程序员基本上都可以找到自己所熟悉、所喜爱的一种语言来编写Web应用程序，并且在同一个项目中，不同语言所编写的类库可以彼此互用；抛弃了VBS脚本语言，采用Visual Basic.NET作为默认的开发语言；为.NET平台开发了一种全新的专用语言——C#。ASP.NET通过改良的ADO.NET来实现对数据库的操作，ADO.NET技术采用了离线式的数据操作，提高了分散式应用程序的效率与扩展性。另一方面，ADO.NET与XML的无缝结合，使得ASP.NET操作数据的跨平台性大大增强。

本书的内容共分为12章，为了便于学习，书中内容的知识点和技术难度是逐步增加和深入的。叙述的方式是理论讲解加案例证明，其中以案例为主，理论为辅，这样做的目的是让读者在阅读案例的同时，更加理解知识在案例中的运用效果和技巧。

其中，第1章是入门基础，简要介绍了ASP.NET 2.0的相关情况，并以一个小实例来展开本书内容。第2章是介绍ASP.NET 2.0中基本控件的应用，如文本控件、选择控件、列表控件等，更为重要的是介绍了ASP.NET 2.0中那些新特性的控件，如母板页控件、导航控件等，第3章的主要内容是介绍验证控件的用法，使用这些验证控件，我们在设计Web页面的时候就可以省去很多的精力来验证用户输入数据的合法性。第4章主

要介绍了ASP.NET数据库绑定控件。数据库绑定控件是复杂的Web系统中必不可少的控件，因为丰富的数据信息必须通过它们显示才能更加高效。ASP.NET 2.0中的主要数据库绑定控件包括Repeater控件、DataList控件和GridView控件。其中，GridView控件是由ASP.NET 1.x中的DataGrid控件升级而来的，相比后者，GridView控件拥有更好的使用方法和更丰富的操作属性。

在第5章至第8章中，主要介绍了ASP.NET 2.0的高级应用，如第5章中的ADO.NET数据库技术的应用。在一般情况下，要实现对数据库的操作要利用Connection对象、Command对象或DataAdapter对象来实现对数据的连接与操作。第6章主要介绍了文件操作技术，并用实例验证了如何利用System.IO中的File类实现对文件目录和文件的读写操作。第7章和第8章主要介绍了ASP.NET 2.0中图形处理技术和网络编程技术的应用。

基于对前面几章知识的巩固和对系统开发中系统架构设计知识的引入，在第9章至第12章的内容中，安排了几个经典的系统开发案例，分别是文章管理系统、企业网站系统、网络求职招聘系统和绩效考核系统。在这些章节的内容中，主要突出讲解了对系统架构设计的把握上。其中有些实例比较完整，比如文章管理系统，网络求职招聘系统等。有些案例由于项目本身比较大，代码和知识点比较多，所以只选了系统中几个主要的模块来展开叙述，如企业网站系统和绩效考核系统等。但这不仅不会影响对项目的理解，反而会给读者更多的思考空间，让读者在现有模块的基础上，构思如何完善这个系统，从而达到举一反三的效果。

本书内容翔实，案例丰富，并配有大量的界面截图和源代码，无论是ASP.NET技术的初学者，还是有一定基础的爱好者，都可以在本书中得到帮助。

本书由郭玉峰、师成林编著，康祥顺审校。在本书的编写过程中，为确保内容的正确性，参阅了很多资料，并且得到了众多网站和资深Web程序员的支持和帮助，他们是刘东鸿、李欣、保春艳、张庆、崔竞、张强、张音；在最后的统稿过程中，得到了两位作者的家人杨媛和刁伟伟的理解和支持，使两位主创作者能够全身心投入到案例的构思和写作中。同时感谢电子工业出版社和美迪亚电子信息有限公司的各位老师，谢谢你们帮助和指导。

由于作者水平有限，时间仓促，书中难免存在疏漏或错误之处，敬请读者批评指正。

为方便读者阅读，若需要本书配套资料，请登录“华信教育资源网”（<http://www.hxedu.com.cn>），在“资源下载”频道的“图书资源”栏目下载。

目 录

第1章 ASP.NET入门基础	1
1.1 走进ASP.NET的世界	1
1.2 从ASP.NET 1.x到ASP.NET 2.0	2
1.3 ASP.NET 2.0新鲜特性体验	2
1.4 本章小结	3
第2章 ASP.NET 2.0精彩控件的应用	4
2.1 常用控件	4
2.2 布局控件	11
2.3 本章小结	22
第3章 验证控件的应用	23
3.1 认识验证控件	23
3.2 常用验证控件的应用	23
3.3 本章小结	35
第4章 ASP.NET数据绑定控件的应用	36
4.1 ASP.NET数据操作语法基础	36
4.2 Repeater控件的应用	43
4.3 DataList控件的应用	46
4.4 GridView控件的应用	55
4.5 本章小结	69
第5章 ADO.NET数据库技术的应用	70
5.1 Command类的操作数据库的应用	70
5.2 DataAdapter类操作数据库的应用	72
5.3 ADO.NET存储过程的应用	74
5.4 DataReader类读取数据的应用	75
5.5 DataSet类保存数据的应用	77
5.6 综合实例——留言系统的设计	78
5.7 本章小结	104

第6章 文件操作技术的应用与实现	105
6.1 目录操作的实现	105
6.2 文件的操作实现	112
6.3 本章小结	119
第7章 ASP.NET图形处理技术实现	120
7.1 GDI+技术入门	120
7.2 生成一个简单的图片	121
7.3 图片型输入验证码的实现	122
7.4 生成业务数据的统计图的实现	125
7.5 简易图形特效处理技术的实现	129
7.6 本章小结	136
第8章 ASP.NET网络技术的实现	137
8.1 网页抓取技术的实现	137
8.2 邮件发送系统的实现	141
8.3 Web Service技术的实现	150
8.4 本章小结	167
第9章 文章管理系统	168
9.1 系统分析与设计	168
9.2 数据库的设计	170
9.3 数据库操作类的实现	172
9.4 系统设计与编码	174
9.5 本章小结	204
第10章 企业网站系统	205
10.1 开发背景分析	205
10.2 系统功能分析	205
10.3 系统架构设计	206
10.4 数据库的设计与实现	206
10.5 公用辅助层的设计	223
10.6 数据层的设计与实现	225
10.7 业务逻辑层的设计与实现	256
10.8 表现层的设计与实现	257
10.9 本章小结	287

第11章 网络求职招聘系统	288
11.1 开发背景分析	288
11.2 系统功能分析	288
11.3 数据的设计	289
11.4 系统编码	308
11.5 本章小结	387
 第12章 绩效考核系统	 388
12.1 系统分析	388
12.2 数据库设计	389
12.3 实体层的实现	407
12.4 数据层的实现	413
12.5 逻辑层的实现	424
12.6 UI层的实现	443
12.7 本章小结	502



第 1 章

ASP.NET入门基础



1.1 走进ASP.NET的世界

ASP (Active Server Page) 是微软公司研发的一种交互式网页编程技术, 从1996年发布的ASP 1.0开始, 它从实验室走向实际应用中, 但是并没有被人们所追宠。1998年, 微软发布了ASP 2.0。在2000年, 微软公司发布了它的革命性的服务器系统Windows 2000, 在该系统中集成了IIS 5.0, 并捆绑了ASP 3.0。由于在ASP中提供了一系列的Web应用程序组件来执行高级功能(如用ADO对象来实现对数据库的操作), 再加上微软系统的稳定支持, 因此, Windows 2000+ASP 3.0便成了当时最流行的WWW服务器模式, ASP也因此在全球风靡起来。ASP的编程语言为VBScript和JavaScript, 运行机制是解释型的。ASP页面文件的后缀名为.asp, 当客户机提交访问的时候, Web服务器就找到该页面, 并交给解释引擎对ASP页面执行一次解释, 并把结果发送给客户机。在当时, 这种技术是具有先进性的。但是随着WWW服务的广泛应用, 越来越多的Web应用程序应用到WWW服务上, 解释型的ASP技术在处理大型Web程序和频繁访问的时候, 给服务器带来瞬间几何级系统开销, 因此, ASP的改进就显得很必要了。

2001年, 微软公司推出了ASP.NET(当时被命名为ASP+), 从命名上看, 可以说ASP.NET是ASP 3.0的升级, 实际上ASP.NET是一种全新的交互式网页编程技术, 是网站和XML Web服务的产物, 也是微软公司新的应用开发平台——.NET框架中的核心要素。如果说微软公司的.NET计划是编程技术的一种革命, 那么, ASP.NET则无疑是ASP的一种革命, ASP.NET技术把面向对象的编程技术引入到Web编程中, 这使得在编写Web应用程序的时候, 就像编写Windows应用程序那样简便快捷。

首先从运行机制上来说ASP.NET的改变。ASP.NET被设计成首编译方式。在页面代码成功调试完毕后, 可以进行人工编译, 也可以在系统得到第一次访问请求时自动编译源代码, 编译后生成了一个基类, 以后每次访问都直接调用该基类。单纯从页面运行效率上讲, 编译式的运行机制肯定要高于解释式的。

其次在编码方式上, ASP.NET采用了代码分离的技术。在以往ASP编程的时候, Web程序的UI代码(HTML布局代码)是和程序逻辑代码(VBScript或者JavaScript)混编在一起的, 所带来的弊端就是美工人员不得不熟悉后台代码, 程序人员不得不注意美工的需要, 除此之外的弊端就是整个页面代码层次、结构混乱不堪, 特别是在调试阶段, 对整个程序没有很深入了解的人很难快速地推断、找出错误之处。ASP.NET的代码分离技术则杜绝了这种现象。在ASP.NET中, 所有的逻辑代码是被封装在<Script></Script>代码块中的, 程序员负责编写逻辑

代码，而UI部分可以由控件和HTML标记组成，并由美工人员负责处理。这样的好处是不仅使得程序员和美工人员的工作量大大减少，提高程序的编制效率，而且缩短了程序开发周期（这大概是每个项目经理都希望的事情吧）。

再则从支持语言上来看，ASP.NET放弃了对VBScript的支持，新的支持语言包括Visual Basic.NET（有的书中称为Visual Basic 7.0）、C#（读C Sharp）、J#等。其中，C#是微软推出的一种专门用于ASP.NET新的编程语言，它具有Visual Basic的编程高效性和C++语言的功能强大性，编程语法规则和C/C++很具有相似性。因此，ASP.NET从一诞生，就有了深厚的“群众基础”。



1.2 从ASP.NET 1.x到ASP.NET 2.0

ASP.NET 1.x的推出给全世界的程序员们带来无比的欣喜，随着ASP.NET平台影响范围的扩大，人们在使用过程中，发现ASP.NET 1.x还需要进行很多的改进。经过6年的技术沉淀，随着ASP.NET 2.0的推出，微软带给程序员们的不仅是又一次的欣喜，而且ASP.NET 2.0向程序员们提供了前所未有的新特性。从ASP.NET 1.x到ASP.NET 2.0不仅仅是一个技术的简单的升级，而且ASP.NET 2.0增加了大量的、实用的新特性，比如控件数量和控件类别的增加、页面框架、服务等，比如实现了WebForm跨页面提交的功能等。

首先是ASP.NET 2.0中控件的改变。在ASP.NET 2.0中，提供了更多更实用的控件，并且对有些控件进行了升级。比如新增了控件站点导航控件；比如把TreeView控件包括在ASP.NET 2.0的基本控件之内而不需要再重新部署TreeView控件；比如升级ASP.NET 1.x中DataGrid控件为ASP.NET 2.0中的GridView控件，并扩展了GridView控件的属性和方法；比如增加了登录控件系列，利用这些控件可以轻松实现登录验证、创建新用户、显示登录状态、显示登录用户名、更新和重置密码等。

其次是简化了aspx页面的设计和维护，增加了皮肤文件（Skin），利用皮肤文件可以完成对整个系统页面样式的统一，最为方便的是可以在皮肤文件中直接对Web控件进行样式定义，也可以利用css文件对aspx页面的其他样式进行控制。另外，还增加了母版页，利用母版页可以完成对系统子页面样式的统一，同时也降低了对维护页面的工作量。

还有就是ASP.NET 2.0中的服务与APIS，这部分主要包括成员资格管理、角色管理、用户配置、数据缓存、配置与管理工具等。利用这些功能，可以快速实现用户和角色管理、个性化用户配置、数据缓存、应用程序可视化配置等任务。以成员资格和角色管理功能为例，在ASP.NET 1.x时代，实现这两项功能是比较繁琐和困难的；而在ASP.NET 2.0中，不仅提供了默认数据库表，而且只需调用几个简单的方法，就能够轻松地实现成员和角色管理功能。



1.3 ASP.NET 2.0新鲜特性体验

在上面的章节中，我们只是简要介绍了ASP.NET 2.0中的新特性，让读者先心里有谱。当然，理论不能与实践脱节，下面我们就利用登录控件来演示一下ASP.NET 2.0的新特性。

Visual Studio 2005的工具箱中默认加载了丰富的控件，并按类别放置，图1.1显示了登

录类控件在工具箱中的位置。

在工具箱中，我们选择CreateUserWizard控件，然后拖到页面Default.aspx的设计视图中，然后运行程序。在我们没有完全填写那些必须填写的内容项而单击创建用户的情况下，页面就会自动在必填项后面出现红色的“*”来提示用户，并且阻止了页面的提交动作。别忘了，在以前这个功能需要编写JavaScript客户端脚本或者利用其他的验证控件来实现，而在ASP.NET 2.0中，我们没有编写任何代码，只是拖拉一下控件就完成了，运行结果如图1.2所示。

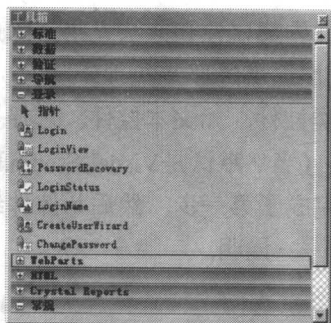


图1.1 登录控件

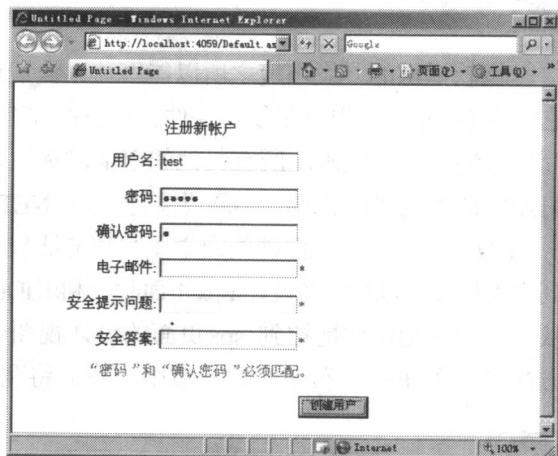


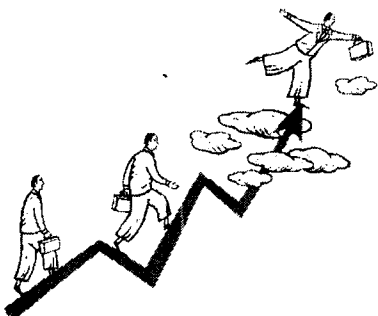
图1.2 登录控件的运行结果

经过上面的实例，我们可以明显地感觉到，ASP.NET 2.0新特性的功能控件是多么的实用。在本书以后的章节中，将会按照不同的知识类别或控件，并且以丰富的实例形式来展开内容。



1.4 本章小结

本章简要介绍了ASP.NET 2.0的一些基本概念和ASP.NET 2.0的新特性，由于这些不是本书的重点，这里只是做了简要的介绍。在本书以后的章节中，通过对实例的设计与实现，并对应丰富的源代码来具体展示ASP.NET 2.0的这些新特性。



第2章

ASP.NET 2.0精彩控件的应用

微软提供的开发环境之所以能够让人快速地学会编程、成倍地提高开发效率，是因为在它提供的IDE中，提供了很多控件。这些控件有丰富的属性和方法，我们只需要简要地设置，简单地编写几句代码，所有其他的事情，都被封装在控件内部属性和方法之中。因此，在学习ASP.NET技术的开始，就要先熟悉一下.NET平台为我们提供的那些控件。

本章按章节和控件种类列举了常用的比较重要的几类控件，如文本控件、选择类控件、列表类控件，等等。在设计aspx页面时，利用IDE开发工具（当然默认是Visual Studio.NET），从工具箱中把控件拖放到aspx页面的设计视图中，这就完成了第一步，然后为控件编写事件方法或设置属性。在下面的章节中，我们将利用实例来一一说明。



2.1 常用控件

2.1.1 文本类控件的应用

文本类控件常用的为TextBox控件和Label控件。TextBox控件主要用于用户输入数据，Label控件主要用于显示数据。这两个控件都有很多样式属性，以供根据不同的需要配置不同的外观。

TextBox的一个重要的属性就是“TextMode”，三个选择属性为：

- (1) SingleLine: 单行数据文本，并可以结合“MaxLength”属性控制输入内容的长度；
- (2) Password: 密码输入框，并可以结合“MaxLength”属性控制输入内容的长度；
- (3) MultiLine: 多行文本输入区域，并结合“Conlumnns”控制有效显示的行数。

Label控件是一个标签控件，可以显示输入文字信息，在HTML代码中翻译为标记代码。

下面通过实例来学习这两个控件的用法。

实例页面TextControl.aspx的HTML代码如下。

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="TextControl.aspx.cs" Inherits="Text-
Control" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>文本类控件</title>
</head>
<body>
```

```

<form id="form1" runat="server">
<div>
    &nbsp;  <br />
    <br />
    <table border="1" cellpadding="0" cellspacing="0" style="width: 100%">
        <tr>
            <td style="width: 39px">
                UserName</td>
            <td style="width: 100px">
                <asp:TextBox ID="txt_Name" TextMode="singleLine" runat="server"></asp:TextBox></td>
        </tr>
        <tr>
            <td style="width: 39px">
                PassWord</td>
            <td style="width: 100px">
                <asp:TextBox ID="txt_Pwd" runat="server" TextMode="Password"
Width="153px"></asp:TextBox></td>
        </tr>
        <tr>
            <td style="width: 39px">
                Information</td>
            <td style="width: 100px">
                <asp:TextBox ID="txt_info" runat="server" TextMode="MultiLine"></asp:
:TextBox></td>
        </tr>
        <tr>
            <td style="width: 39px; height: 21px;">
                </td>
            <td style="width: 100px; height: 21px;">
                <asp:Button ID="btn_Submit" runat="server" Text="确认" OnClick="btn_Submit_
Click" /></td>
        </tr>
        <tr>
            <td style="width: 39px">
                Response</td>
            <td style="width: 100px">
                <asp:Label ID="lb_ResInfo" runat="server" Text="Label"></asp:Label></td>
        </tr>
    </table>
</div>
</form>
</body>
</html>

```

TextControl.aspx的btn_Submit的事件代码如下。

```

protected void btn_Submit_Click(object sender, EventArgs e)
{
    string str=""

```

```

str+="您的名字: "+txt_Name.Text.Trim()+"<br/>";
str+="您的密码: "+txt_Pwd.Text.Trim()+"<br/>";
str+="您的信息: "+txt_info.Text;
this.lb_ResInfo.Text=str;
}

```

运行结果如图2.1所示。

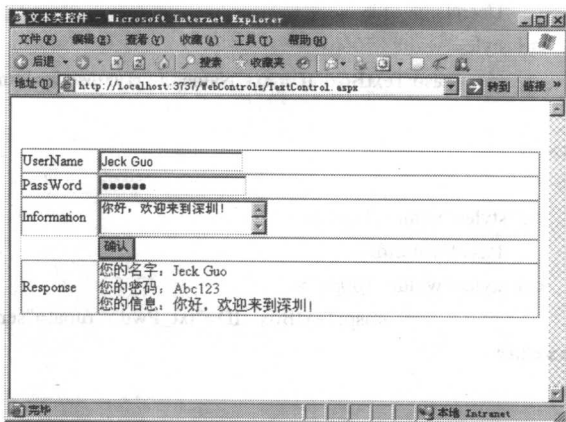


图2.1 TextControl.aspx的运行结果

2.1.2 选择类控件的应用

常用的选择类控件是CheckBox控件和RadioButton控件。CheckBox控件是复选框，一个组内允许多个CheckBox控件出现选中状态，而RadioButton控件则是互斥的，即在同一组内，只允许一个RadioButton控件出现选中状态，在使用RadioButton控件时，通过指定GroupName属性，可以让多个RadioButton控件在不同的域内起作用。

另外，还有两个供用户选择的控件CheckBoxList控件和RadioButtonList控件。CheckBoxList控件是CheckBox控件的组合，通过CheckBoxList控件更容易获取选中的项；RadioButtonList控件是组合了RadioButton控件，这样就不需要指定每个RadioButton空间的GroupName属性了。

下面通过一个实例来演示这四个控件的用法。

实例页面ChooseControl.aspx的HTML代码如下。

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ChooseControl.aspx.cs" Inherits=
"ChooseControl" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>选择控件</title>
</head>
<body>
<form id="form1" runat="server">
<div>
&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; <table border="1" cellpadding="0" cellspacing="0"
style="width: 100%">

```

```

        <tr>
            <td style="width: 100px">
                <asp:CheckBox ID="cb_cb1" runat="server" Text="cb_cb1" />
                <asp:CheckBox ID="cb_cb2" runat="server" Text="cb_cb2" /></td>
            </tr>
            <tr>
                <td style="width: 100px">
                    <asp:CheckBoxList ID="cbl_cbl1" runat="server" RepeatDirection="Horizontal"
                        Width="199px">
                        <asp:ListItem>cb1</asp:ListItem>
                        <asp:ListItem>cb2</asp:ListItem>
                        <asp:ListItem>cb3</asp:ListItem>
                    </asp:CheckBoxList></td>
                </tr>
            <tr>
                <td style="width: 100px">
                    <asp:RadioButtonList ID="rbl_rbl1" runat="server" RepeatDirection="Horizontal">
                        <asp:ListItem>rbl1</asp:ListItem>
                        <asp:ListItem>rbl2</asp:ListItem>
                        <asp:ListItem>rbl3</asp:ListItem>
                    </asp:RadioButtonList></td>
                </tr>
            <tr>
                <td style="width: 100px">
                    <asp:RadioButton ID="rb_rb1" GroupName="Group1" runat="server" Text="rb_rb1" />
                    <asp:RadioButton ID="rb_rb2" GroupName="Group1" runat="server" Text="rb_rb2" /></td>
                </tr>
        </table>
        <asp:Button ID="btn_submit" runat="server" Text="显示选择情况" OnClick="Button1_Click" />
    </div>
</form>
</body>
</html>

```

ChooseControl.aspx中Button1_Click的事件代码如下。

```

protected void Button1_Click(object sender, EventArgs e)
{
    string str = "";
    if (this.cb_cb1.Checked)
        str += "cb_cb1已经选中！<br/>";
    if (this.cb_cb2.Checked)
        str += "cb_cb2已经选中！<br/>";
    string str1 = "";
    foreach (ListItem li in this.cbl_cbl1.Items)
    {
        if (li.Selected)
            str1 += li.Text + " ";
    }
}

```

}



Mode="Multiple"表示允许多选，SelectionMode="Single"表示单选。

ListControl.aspx页面的HTML代码如下。

```

<!--ListControl" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>选择列表控件</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<table border="1" cellpadding="0" cellspacing="0" style="width: 100%; height: 249px;">
<tr>
<td style="width: 40%">

```



```

        <asp:DropDownList ID="ddl_list" runat="server" Width="100%">
    </asp:DropDownList></td>
    <td style="width: 10%">
        <asp:Button ID="btn_Left" runat="server" Text="右移动->" OnClick="btn_Left_Click" />
        <asp:Button ID="btn_Right" runat="server" Text="<<-左移动" OnClick="btn_Right_Click" />
        <asp:Button ID="btn_resite" runat="server" Text="重新生成" OnClick="btn_resite_Click"
Width="87px" />
    </td>
    <td style="width: 40%">
        <asp:ListBox ID="lb_list" runat="server" Width="100%"></asp:ListBox>
    </td>
</tr>
</table>
</div>
</form>
</body>
</html>

```

ListControl.aspx.cs的代码如下。

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
public partial class ListControl : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {
            creatDataTable( );
        }
    }
    //产生模拟数据，并绑定到控件中。
    private void creatDataTable( )
    {
        DataTable dt = new DataTable( );
        DataRow dr;
        DataColumn dc = new DataColumn("txt");
        dt.Columns.Add(dc);
        dc = new DataColumn("value");
        dt.Columns.Add(dc);
        for (int i = 1; i < 11; i++)

```