

万水软件项目应用与实例开发丛书

COMPANION  
WEB SITE!

# ASP.NET

# 专业项目实例开发

Microsoft ASP.NET Professional Projects

(修订版)

[美] Hersh Bhasin 著  
高健 英宇 等译



中国水利水电出版社  
www.waterpub.com.cn

万水软件项目应用与实例开发丛书

# ASP.NET 专业项目实例开发 (修订版)

[美] Hersh Bhasin 著

高健英宇等译

中国水利水电出版社

## 内 容 提 要

本书共分为三部分, 通过理论和实例应用, 详细讲述了 ASP.NET 技术。

第一部分是 ASP.NET 编程环境。这一部分从 ASP.NET 基础知识开始, 详细介绍了 ASP.NET Web 表单和控件、在 .NET 框架中使用 ADO.NET、数据绑定、输入验证、用户控件、自定义控件、业务对象、使用 ASP.NET 的 Web 服务、ASP.NET 应用程序、缓存技术、跟踪、安全性, 这些是应用 ASP.NET 的基础。

第二部分是项目设计实例。讲述的实际应用案例包括个人财务管理系统设计、库存管理系统设计、GenEditAdd 控件的应用、在 Visual Studio.NET 中创建 Web 服务。这些实例可以更清楚、更直观地教你使用 ASP.NET。

第三部分是附录。附录 A 是本书实例数据库的安装说明。附录 B 是 HailStorm。

本书适合于 ASP.NET 的初学者, 尤其适合于正在用 ASP.NET 做项目的开发人员。

本书配有源代码, 读者可以从中国水利水电出版社网站上免费下载, 网址:  
<http://www.waterpub.com.cn/softdown/>。

Authorized translation from English Language Edition published by Premier Press, Inc. Original Copyright © 2002, Premier Press, Microsoft ASP.NET Professional Projects. Translation by China WaterPower Press, 2002.

北京市版权局著作权合同登记号: 图字 01-2002-1913

## 图书在版编目(CIP)数据

ASP.NET 专业项目实例开发 / (美) 巴辛 (Bhasin, H.)  
著. 高健等译. —2 版 (修订本). —北京: 中国水利  
水电出版社, 2007

(万水软件项目应用与实例开发丛书)

书名原文: Microsoft ASP.NET Professional Projects

ISBN 978-7-5084-4245-7

I. A... II. ①巴... ②高... III. 主页制作—程序  
设计 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2006) 第 143327 号

|         |  |
|---------|--|
| 书 名     | ASP.NET 专业项目实例开发 (修订版)   |
| 作 者     | [美] Hersh Bhasin 著   |
| 译 者     | 高 健 英 宇 等译   |
| 出 版 发 行 | 中国水利水电出版社 (北京市三里河路 6 号 100044)<br>网址: <a href="http://www.waterpub.com.cn">www.waterpub.com.cn</a><br>E-mail: <a href="mailto:mchannel@263.net">mchannel@263.net</a> (万水)<br><a href="mailto:sales@waterpub.com.cn">sales@waterpub.com.cn</a><br>电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水) |
| 经 售     | 全国各地新华书店和相关出版物销售网点   |
| 排 版     | 北京万水电子信息有限公司   |
| 印 刷     | 北京蓝空印刷厂  |
| 规 格     | 787mm×1092mm 16 开本 30.25 印张 750 千字   |
| 版 次     | 2007 年 1 月第 1 版 2007 年 1 月第 1 次印刷  |
| 印 数     | 0001—4000 册  |
| 定 价     | 48.00 元  |

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

## 修订版序

万水软件项目应用与实例开发丛书由中国水利水电出版社万水公司于 2002 年引进,原书由全球 IT 培训 15 强之一的 NIIT 资深开发人员编写。该丛书一经出版就因其创新性的内容安排、实用的案例开发、全面的知识讲解受到了国内广大读者的喜爱和欢迎。但在读者的使用过程中发现该丛书翻译过程中存在一些不尽如人意的地方,并且随着时间的推移有些内容需要进行合理的删减,所以在该丛书再版之际我们组织有多年开发经验的人员对丛书的部分内容作了修订,希望这套实用的项目开发丛书能更好地为广大读者服务。

修订版将上一版中的图表和程序注释翻译为中文,更正了第一版中对某些术语的翻译,并对第一版中的章节进行了调整和合并,增加了对 XML 和 Web 服务的介绍,增加了一个更有价值的项目实例。

修订后本书分为三部分,第一部分为 ASP.NET 编程环境,逐步深入介绍 ASP.NET 编程的基础知识;第二部分为项目设计实例,结合第一部分的内容,介绍了五个案例;第三部分为附录,是本书实例数据库的安装说明及 HailStorm。

本书既适用于 ASP.NET 初学者,也适用于有一定 ASP.NET 基础但缺乏项目经验的读者。对于前者,可从第一部分开始学习;对于后者,可直接从第二部分开始学习。相信本书对基础知识深入浅出的阐述以及丰富详实的项目实例会对读者有所帮助。

本书的修订工作主要由丁昊凯、陈德颢完成,参与修订的还有许靖雯、童剑、黄卓、黄浩、李强、郝文博、王振江、杨昭昭、林晓珊等。

读者如果有什么意见和建议可以与我们的联系,联系方式为: [xinyuanxuan@263.net](mailto:xinyuanxuan@263.net)。

修订者

2006 年 10 月

## 译者序

微软作为全球软件业最大的公司，在.NET 技术上投入了大量的人力物力，把公司未来的战略重心放在了.NET 上，而从目前看来，.NET 技术无疑代表了未来 Internet 技术的发展方向。

作为 ASP 的下一代版本，ASP.NET 并非从 ASP 3.0 自然演化而来，在许多方面，ASP.NET 与 ASP 有着本质的不同。ASP.NET 完全基于模块与组件，具有更好的可扩展性与可定制性，数据处理方面更是引入了许多激动人心的新技术，正是这些具有革命意义的新特性，让 ASP.NET 远远超越了 ASP，同时也提供给 Web 开发人员更好的灵活性，有效地缩短了 Web 应用程序的开发周期。

本书共分为三部分内容，从 ASP.NET 的介绍到项目设计实例。通过理论和实例应用，详细讲述了 ASP.NET 技术。第一部分是 ASP.NET 编程环境。这一部分从 ASP.NET 基础知识开始，详细介绍了 ASP.NET Web 表单和控件、在 .NET Framework 中使用 ADO.NET、数据绑定、输入验证、用户控件、自定义控件、业务对象、使用 ASP.NET 的 Web 服务、ASP.NET 应用程序、缓存技术、跟踪、安全性，这些是应用 ASP.NET 的基础。第二部分是项目设计实例。讲述的实际应用案例包括个人财务管理系统设计、库存管理系统设计、GenEditAdd 控件的应用、在 Visual Studio.NET 中创建 Web 服务。这些实例可以更清楚、更直观地教你使用 ASP.NET。第三部分是附录。附录 A 是本书实例数据库的安装说明。附录 B 是 HailStorm。

从本书内容不难看出作者精心设计出来的本书特点：

- 由浅入深地介绍领先的 ASP.NET 技术。
- 理论与实践相结合的典范，大量的项目实例让你感受信手拈来的成功喜悦。
- 详细的数据库访问技术，这是 Web 技术的关键所在。
- 详细的项目实例带你由初级读者向高级应用开发者迈进。

如果你只是一名初学者，通过本书基础知识的学习将很快学会 ASP.NET，如果你已经有了一定的 ASP.NET 知识，或者你本身不是一位 ADO.NET 的开发人员，那么本书丰富而详尽的实例会给你的工作带来很大的帮助。

本书由高健、英宇、董健、郭志博翻译，朱云华审校。参与本书翻译工作的还有：陈贵敏、李长春、谢君英、易重英、王翌、牛韬、姚先勤。由于时间仓促及译者水平有限，书中难免有译误之处，欢迎广大读者提出意见，我们接收读者意见的邮箱为：yingyu@263.net。

译者

2002 年 6 月

# 目 录

修订版序

译者序

## 第一部分 ASP.NET 编程环境

|   |    |                          |    |
|---|----|--------------------------|----|
| <b>第 1 章 ASP.NET 概述</b> .....           | 1  | 3.6.2 将筛选器应用到视图          | 42 |
| 1.1 安装 .NET Framework SDK               | 5  | 3.6.3 DataTable 行和列集合的读取 | 44 |
| 1.2 安装本书的实例数据库                          | 6  | 3.7 DataReader           | 46 |
| <b>第 2 章 ASP.NET Web 窗体和控件概述</b> .....  | 7  | 3.8 数据关系                 | 48 |
| 2.1 基本技术                                | 7  | 3.9 XML                  | 49 |
| 2.2 状态管理                                | 7  | 3.9.1 XML 简介             | 49 |
| 2.3 页面事件                                | 8  | 3.9.2 XML 基本语法           | 50 |
| 2.4 代码分离                                | 9  | 3.9.3 在 ASP.NET 中使用 XML  | 55 |
| 2.5 服务器控件                               | 12 | 3.10 小结                  | 62 |
| 2.5.1 HTML 控件                           | 13 | <b>第 4 章 数据绑定</b> .....  | 63 |
| 2.5.2 Web 控件                            | 16 | 4.1 绑定控件                 | 63 |
| 2.5.3 内部控件                              | 17 | 4.2 DataRepeater         | 65 |
| 2.5.4 增强控件                              | 24 | 4.3 DataGrid             | 67 |
| 2.6 小结                                  | 27 | 4.3.1 基本 DataGrid        | 68 |
| <b>第 3 章 在 .NET 框架中使用 ADO.NET</b> ..... | 29 | 4.3.2 带样式的主表             | 68 |
| 3.1 DataSet                             | 29 | 4.3.3 可编辑的主窗体            | 70 |
| 3.2 Managed Provider (数据提供程序)           | 30 | 4.3.4 排序和分页              | 76 |
| 3.2.1 命名空间                              | 32 | 4.4 DataGrid 的分页         | 78 |
| 3.2.2 Connection 对象                     | 32 | 4.5 DataList             | 81 |
| 3.2.3 DataAdapter 对象                    | 33 | 4.6 XML 数据绑定             | 85 |
| 3.3 Command 对象                          | 34 | 4.7 实现一个主-从关系            | 86 |
| 3.3.1 CommandType                       | 34 | 4.8 小结                   | 90 |
| 3.3.2 执行命令                              | 35 | <b>第 5 章 输入验证</b> .....  | 91 |
| 3.4 Command 对象的操作查询                     | 36 | 5.1 两端验证                 | 91 |
| 3.5 存储过程                                | 39 | 5.2 验证控件                 | 92 |
| 3.5.1 使用参数集                             | 39 | 5.2.1 必填域                | 94 |
| 3.5.2 通过 Execute 关键字调用                  |    | 5.2.2 比较验证               | 94 |
| 存储过程                                    | 41 | 5.2.3 范围验证               | 95 |
| 3.6 DataView                            | 42 | 5.2.4 正则表达式验证            | 96 |
| 3.6.1 默认视图                              | 42 | 5.2.5 自定义验证              | 97 |

|              |                              |            |               |  |            |
|--------------|------------------------------|------------|---------------|--|------------|
| 5.3          | ValidationSummary 控件.....    | 98         | 8.7           | 小结 .....                                   | 153        |
| 5.4          | IsValid 属性 .....             | 98         | <b>第 9 章</b>  | <b>使用 ASP.NET 的 Web 服务 .....</b>           | <b>154</b> |
| 5.5          | 禁用客户端验证.....                 | 99         | 9.1           | Web 服务基础 .....                             | 155        |
| 5.6          | 小结 .....                     | 99         | 9.1.1         | 什么是 Web 服务 .....                           | 155        |
| <b>第 6 章</b> | <b>用户控件 .....</b>            | <b>100</b> | 9.1.2         | 为什么使用 Web 服务 .....                         | 156        |
| 6.1          | 创建简单的用户控件.....               | 100        | 9.1.3         | 了解.NET 中实现 Web 服务的<br>方法 .....             | 156        |
| 6.2          | 属性显现 .....                   | 101        | 9.2           | 编写一个简单的 Web 服务.....                        | 158        |
| 6.3          | 使用用户控件为网站设计导航系统 .....        | 102        | 9.2.1         | 测试服务 .....                                 | 158        |
| 6.4          | 小结 .....                     | 106        | 9.2.2         | WSDL 规范.....                               | 159        |
| <b>第 7 章</b> | <b>自定义控件 .....</b>           | <b>107</b> | 9.2.3         | 通过 HTTP Get 调用 Web 服务 .....                | 161        |
| 7.1          | 用 Visual Basic 创建一个简单的控件 ... | 107        | 9.2.4         | 通过 HTTP Post 调用 Web 服务 .....               | 162        |
| 7.1.1        | 第 1 步: 创建一个类文件.....          | 107        | 9.2.5         | 通过 SOAP 调用 Web 服务 .....                    | 163        |
| 7.1.2        | 第 2 步: 创建 DLL.....           | 108        | 9.3           | 使用 Visual Studio 创建 Web 服务 .....           | 166        |
| 7.1.3        | 第 3 步: 创建 Web 窗体 .....       | 109        | 9.4           | 从 Web 窗体中调用 Web 服务.....                    | 170        |
| 7.2          | 用 C# 创建一个简单的控件 .....         | 109        | 9.5           | 使用 WebService Behavior 进行函数<br>调用.....     | 173        |
| 7.2.1        | 第 1 步: 创建一个类文件.....          | 109        | 9.6           | 如何部署 Web 服务.....                           | 177        |
| 7.2.2        | 第 2 步: 创建 DLL 文件.....        | 110        | 9.7           | 小结 .....                                   | 178        |
| 7.2.3        | 第 3 步: 创建 Web 窗体 .....       | 111        | <b>第 10 章</b> | <b>ASP.NET 应用程序.....</b>                   | <b>179</b> |
| 7.3          | 通用编辑/添加自定义控件.....            | 111        | 10.1          | 创建虚拟目录.....                                | 179        |
| 7.4          | 配置文件 .....                   | 114        | 10.1.1        | 把现有文件夹更改为虚拟目录 .....                        | 179        |
| 7.5          | 创建控件 .....                   | 116        | 10.1.2        | 在 Windows 2000 个人 Web<br>服务器上创建新的虚拟目录..... | 180        |
| 7.5.1        | 第 1 步: 编辑模式 .....            | 116        | 10.2          | Global.asax 文件.....                        | 181        |
| 7.5.2        | 第 2 步: 添加模式 .....            | 118        | 10.3          | Global.asax 和应用程序状态 .....                  | 185        |
| 7.5.3        | 第 3 步: 开始创建 .....            | 119        | 10.4          | Session 状态 .....                           | 187        |
| 7.5.4        | 第 4 步: GenEditAdd 自定义控件 ...  | 128        | 10.4.1        | in-Process 模式.....                         | 188        |
| 7.6          | 使用 GenEditAdd 自定义控件.....     | 136        | 10.4.2        | out-of-Process 模式 .....                    | 189        |
| 7.7          | 小结 .....                     | 139        | 10.4.3        | SQL Server 模式.....                         | 190        |
| <b>第 8 章</b> | <b>业务对象 .....</b>            | <b>140</b> | 10.4.4        | cookieless 状态.....                         | 191        |
| 8.1          | Bin 目录.....                  | 140        | 10.5          | 配置文件.....                                  | 192        |
| 8.2          | 命名空间和配件.....                 | 140        | 10.6          | 小结 .....                                   | 198        |
| 8.3          | 一个简单的 Visual Basic 业务对象..... | 140        | <b>第 11 章</b> | <b>缓存技术.....</b>                           | <b>199</b> |
| 8.4          | 一个简单的 C# 组件.....             | 142        | 11.1          | 输出缓存.....                                  | 199        |
| 8.5          | 区分 Web 窗体和组件服务.....          | 144        | 11.2          | 页面数据缓存.....                                | 201        |
| 8.6          | 数据库类 .....                   | 144        | 11.3          | 文件和键值依赖.....                               | 202        |
| 8.6.1        | Visual Basic.NET 数据库类.....   | 144        | 11.4          | 小结.....                                    | 205        |
| 8.6.2        | 编译数据库类 .....                 | 147        | <b>第 12 章</b> | <b>跟踪.....</b>                             | <b>206</b> |
| 8.6.3        | 测试数据库类 .....                 | 148        | 12.1          | 页面级跟踪.....                                 | 207        |
| 8.6.4        | C# 的数据库类.....                | 150        |               |  |            |
| 8.6.5        | 编译 C# 类.....                 | 152        |               |  |            |
| 8.6.6        | 测试 C# 类.....                 | 152        |               |  |            |

|                        |            |                              |     |
|------------------------|------------|------------------------------|-----|
| 12.2 应用程序级跟踪.....      | 207        | 13.2 Passport 验证提供者.....     | 217 |
| 12.3 取消跟踪.....         | 209        | 13.3 基于 Windows 的验证.....     | 218 |
| 12.4 小结.....           | 209        | 13.3.1 启用基本安全验证.....         | 218 |
| <b>第 13 章 安全性.....</b> | <b>210</b> | 13.3.2 设置许可权.....            | 219 |
| 13.1 基于窗体的验证.....      | 211        | 13.3.3 编辑 web.config 文件..... | 219 |
| 13.1.1 一个简单的例子.....    | 211        | 13.4 小结.....                 | 220 |
| 13.1.2 使用数据库来存储密码..... | 216        |                              |     |

## 第二部分 项目设计实例

|                                     |            |                                |            |
|-------------------------------------|------------|--------------------------------|------------|
| <b>项目 1 个人财务管理.....</b>             | <b>221</b> | 18.3 在账目图表单中加入 Web 服务.....     | 273        |
| 项目 1 概述.....                        | 221        | 18.4 在事务窗体中加入 Web 服务.....      | 276        |
| <b>第 14 章 个人财务管理程序的设计.....</b>      | <b>222</b> | 18.5 在试算表中加入 Web 服务.....       | 281        |
| 14.1 组.....                         | 222        | 项目 1 小结.....                   | 285        |
| 14.2 Groups 表.....                  | 222        | <b>项目 2 库存管理系统.....</b>        | <b>286</b> |
| 14.3 masters 表.....                 | 224        | 项目 2 概述.....                   | 286        |
| 14.4 tr_header 表.....               | 225        | <b>第 19 章 库存管理系统的设计.....</b>   | <b>287</b> |
| 14.5 transactions 表.....            | 225        | 19.1 库存主表.....                 | 287        |
| 14.6 TblSelection 表.....            | 225        | 19.2 事务标题表.....                | 288        |
| <b>第 15 章 账目图表.....</b>             | <b>227</b> | 19.3 库存明细表.....                | 288        |
| 15.1 插入和更新主表记录.....                 | 227        | 19.4 所支持的组件.....               | 289        |
| 15.2 主 Web 窗体.....                  | 229        | <b>第 20 章 库存管理主表.....</b>      | <b>291</b> |
| 15.2.1 更新逻辑.....                    | 230        | 20.1 存储过程 p_stock_masters..... | 291        |
| 15.2.2 添加记录.....                    | 231        | 20.2 库存主 Web 窗体.....           | 292        |
| 15.2.3 删除模式.....                    | 233        | 20.3 添加记录.....                 | 298        |
| 15.2.4 RunSql 函数.....               | 233        | 20.4 删除模式.....                 | 299        |
| <b>第 16 章 事务.....</b>               | <b>241</b> | 20.5 RunSql 函数.....            | 300        |
| 16.1 插入和更新事务.....                   | 241        | 20.6 排序.....                   | 300        |
| 16.2 更新 Masters 表中的 closing 字段..... | 244        | 20.7 在 DataGrid 中分页.....       | 302        |
| 16.3 关于触发器的讨论.....                  | 246        | <b>第 21 章 库存变动.....</b>        | <b>309</b> |
| 16.4 事务维护.....                      | 247        | 21.1 插入和更新事务.....              | 309        |
| 16.4.1 选择窗体.....                    | 247        | 21.2 stock_detail 表的触发器.....   | 311        |
| 16.4.2 事务窗体.....                    | 249        | 21.3 库存事务.....                 | 313        |
| <b>第 17 章 试算表报表.....</b>            | <b>261</b> | 21.4 库存事务窗体.....               | 313        |
| <b>第 18 章 项目 1 的补充.....</b>         | <b>265</b> | 21.4.1 ReBind 函数.....          | 314        |
| 18.1 创建通用数据库 Web 服务.....            | 265        | 21.4.2 添加模式.....               | 315        |
| 18.2 设计导航系统.....                    | 270        | 21.4.3 更新模式.....               | 316        |
| 18.2.1 导航链接.....                    | 271        | 21.4.4 函数 RunSql.....          | 317        |
| 18.2.2 用户控件.....                    | 272        | 21.4.5 删除模式.....               | 317        |
| 18.2.3 使用控件.....                    | 273        | <b>第 22 章 库存余额报表.....</b>      | <b>325</b> |



|                                     |     |                                |     |
|-------------------------------------|-----|--------------------------------|-----|
| 项目 2 小结 .....                       | 328 | 26.3.3 将列转换为模板列 .....          | 380 |
| 项目 3 GenEditAdd 控件 .....            | 329 | 26.4 添加面板 .....                | 382 |
| 项目 3 概述 .....                       | 329 | 26.5 指定 DataGrid 命令方法 .....    | 383 |
| 第 23 章 使用 GenEditAdd 控件 .....       | 331 | 26.6 方法 .....                  | 383 |
| 23.1 编译控件 .....                     | 331 | 第 27 章 使用 Visual Studio.NET 创建 |     |
| 23.2 配置文件 .....                     | 331 | Web 服务 .....                   | 388 |
| 23.3 把 GenEditAdd 连到 DataGrid ..... | 332 | 27.1 建立通用数据库 Web 服务 .....      | 388 |
| 23.4 一个例子 .....                     | 334 | 27.2 为 Web 服务写脚本 .....         | 389 |
| 第 24 章 扩展 GenEditAdd 控件 .....       | 337 | 27.3 从 Web 窗体调用 Web 服务 .....   | 393 |
| 24.1 下拉列表栏 .....                    | 337 | 项目 4 小结 .....                  | 401 |
| 24.2 必填域 .....                      | 344 | 项目 5 网上销售系统 .....              | 402 |
| 24.3 可编辑的字段 .....                   | 345 | 项目 5 概述 .....                  | 402 |
| 24.4 字段名 .....                      | 346 | 第 28 章 数据库设计与实现 .....          | 403 |
| 24.5 存储过程名 .....                    | 347 | 28.1 需求分析 .....                | 403 |
| 24.6 完整代码列表 .....                   | 348 | 28.2 数据库的逻辑设计 .....            | 403 |
| 项目 3 小结 .....                       | 360 | 第 29 章 通用处理类功能设计 .....         | 407 |
| 项目 4 Visual Studio.NET .....        | 361 | 29.1 BasePage 类 .....          | 407 |
| 项目 4 概述 .....                       | 361 | 29.2 Action 类 .....            | 411 |
| 第 25 章 使用强类型数据集显示数据库                |     | 29.3 Ac 类功能设计 .....            | 413 |
| 数据 .....                            | 362 | 第 30 章 数据访问层 .....             | 418 |
| 25.1 创建 C# Web 应用程序 .....           | 362 | 30.1 数据访问层基类 .....             | 418 |
| 25.2 浏览应用程序文件夹 .....                | 363 | 30.2 数据访问层具体应用 .....           | 420 |
| 25.3 浏览生成的文件 .....                  | 363 | 第 31 章 用户表示层 .....             | 421 |
| 25.4 浏览项目属性 .....                   | 364 | 31.1 DataGrid 前台页功能代码 .....    | 421 |
| 25.5 数据库交互 .....                    | 365 | 31.2 DataGridOperate 类 .....   | 424 |
| 25.6 浏览为 SqlConnection 和            |     | 31.3 小结 .....                  | 437 |
| SqlDataAdapter 生成的代码 .....          | 369 | 第 32 章 页面设计 .....              | 438 |
| 第 26 章 用 Visual Studio.NET 编写 CRUD  |     | 32.1 系统菜单导航页 .....             | 438 |
| 应用程序 .....                          | 374 | 32.2 系统登录页面 Default.aspx ..... | 442 |
| 26.1 创建新的 C# ASP.NET Web 应用         |     | 32.3 权限管理模块 .....              | 446 |
| 程序 .....                            | 374 | 32.4 用户资料 .....                | 448 |
| 26.2 Data 组件 .....                  | 375 | 32.5 用户管理界面 .....              | 452 |
| 26.3 DataGrid .....                 | 376 | 32.6 单据操作 .....                | 463 |
| 26.3.1 选择 DataGrid 列 .....          | 377 | 32.7 页面注册 .....                | 467 |
| 26.3.2 添加 Add、Edit 和 Delete .....   | 378 | 项目 5 小结 .....                  | 469 |

## 第三部分 附录

|                    |     |                      |     |
|--------------------|-----|----------------------|-----|
| 附录 A 安装样例数据库 ..... | 470 | 附录 B HailStorm ..... | 473 |
|--------------------|-----|----------------------|-----|

# 第一部分 ASP.NET 编程环境

## 第 1 章 ASP.NET 概述

从 ASP 到 ASP.NET 可以说是一场重大的革命，与 ASP 相关的数据访问服务 ADO 在 ASP.NET 中称为 ADO.NET。ASP 受到很多限制——它是非结构化语言，所以它的代码和表示逻辑混杂在一起，这使得应用程序难以理解和维护。由于 ASP 的这个限制，代码分离几乎是不可能的。你无法把表示逻辑移交给 Web 设计者、把代码移交给开发人员并让他们同时来开发这个应用程序。和基于 Windows 的应用程序开发不一样，ASP 没有一个内部的部件或编程模型，ASP 开发者不得不把标记语言、脚本环境和服务器平台结合到一起来进行工作。另外，ASP 的支持工具也是很有限的，虽然在 Visual InterDev 中引入了 Visual Basic 界面可以给窗体加上或去掉一些部件如文本框和标签，但这是非常笨拙的，而且会给窗体中加入非常多的不必要的代码，这就使得很多开发人员不愿意使用这个产品。

作为 ASP 的数据访问部件，ADO 设计为一种专为那些基于客户/服务器的应用程序的数据访问需要服务的视图。但是，在 Web 编程时要遵循不同的规则。一个客户/服务器应用程序没有必要优化数据库连接，一个典型的数据库操作会打开一个数据库连接并保持其开放状态，直到 ADO 记录集的循环操作完成。但是在基于 Web 的环境中，数据库连接是很昂贵的。所以 Web 编程需要一种不用连接就能操作数据的方法，这样就产生了远程数据服务（Remote Data Services, RDS）。随着 XML 的出现，请求/响应模式成为今天的标准范例。为了和这个基于消息的通信系统保持一致，RDS 增加了对 HTTP 的支持，这就允许在中间层进行事务处理逻辑调用。XML 遵循多种不同的分级数据模型（XMLDOM），这和微软的数据访问技术（Microsoft Data Access Components, MDAC）所遵循的模型是相关的。要想利用 XML 数据，就必须在 MSXML 和 MDAC 之间做出选择。ADO.NET 解决了这种尴尬的局面，它在底层就提供了对 XML 的支持，这和对数据库数据进行操作是很相似的，所以也就没有了在 MDAC 和 MSXML 之间做选择的问题。

作为 ASP.NET 的基本构造块之一，Web 窗体将在第 2 章“ASP.NET Web 窗体和控件概述”中进行讨论。在那儿会提出代码分离（Code Behind）的概念，Code Behind 是一个把表示逻辑和脚本写入不同文件的过程。使用代码分离是为了消除混乱的东西和 spaghetti 代码（spaghetti 代码是指那些与表达逻辑混杂在一起的脚本代码），而传统的 ASP 好像鼓励编写这种代码。ASP.NET 提供了和 Visual Basic 类似的基于服务器的、事件驱动的编程模型，使得可以利用一些所见即所得（WYSIWYG, What You See Is What You Get）工具，如 Visual Studio。

ASP.NET 引入了两套控件：HTML 控件和 Web 控件，统称为“服务器控件”。这些控件弥补了 HTML 的不足，此外还提供了交互过程中保留状态、检测浏览器类型（在这一点上也

是弥补了 HTML 的不足)以及作为构造块为组合控件服务的方法。这些控件驻留在服务器上,然后向浏览器输出简单的 HTML。因为所有的浏览器都能理解 HTML,所以 ASP.NET 也就克服了在不同浏览器上的兼容性问题。

ASP.NET 中的 HTML 控件(如文本框、表单、按钮等)还是我们用得最多的那些普通 HTML 控件,只是现在添加了一个新的“runat=server”属性。这些控件的主要用途是提供一个到 ASP.NET 的快速迁移路径,这样通过添加“runat=server”就能把许多 HTML 控件转换成 ASP.NET 控件。但是 Web 控件是高度抽象和有效的。有 4 类 Web 控件分别为内部控件、增强控件、List Bound 控件和验证控件。内部控件是传统 HTML 控件的 ASP.NET 版本,包括文本框、按钮和下拉列表等。在 ASP.NET 中,这些控件有特殊的前缀以区别于普通的 HTML 控件,而且它们也有“runat=server”属性,这样用下面的语句就能创建一个文本框:

```
<asp:TextBox id="text1" runat="server"/>
```

Web 控件的目的是简化控件的命名方法,功能重复的控件被简化成单一的控件。这些控件都包含一些属性如前景色 (ForeColor)、字体 (Font)、背景色 (BackColor)、启用 (Enabled) 等。开发人员只需记住属性设置的名称就能将其应用在所有控件上。

增强控件由 Calendar 和 AdRotator 组成,Calendar 用来为一些低级浏览器输出 HTML (这些浏览器不支持 DHTML) 而为一些高级浏览器输出 DHTML,AdRotator 用来显示循环的广告。

List Bound 控件是第 4 章“数据绑定”的主题,它又包括 3 个控件:DataGrid、DataList 和 DataRepeater。这些控件自动地把数据库数据显示成列表和数据表格,开发人员可以将大量的模板应用到这些控件,以便完成更高级的用户定制。DataGrid 甚至可以用来进行数据编辑。

第 5 章“输入验证”中讨论了验证控件,编写验证代码的各种操作自动进行。它又包括 5 个验证控件和一个验证摘要控件,即:RequiredFieldValidator、RegularExpressionValidator、CompareValidator、RangeValidator、CustomValidator 和 ValidationSummary。顾名思义,每个控件的作用是显而易见的。例如,RequiredFieldValidator 控件的作用是不允许用户的必填域是空白的。同样,RangeValidator 控件是用来校验用户的输入是否在指定的范围内。在 ASP.NET Web 窗体中包含验证是一项很简单的工作,你需要做的只是把输入文本框和相应的验证控件联系起来。

ADO.NET 是 ADO 的最新体现,这将在第 3 章“在 .NET 框架中使用 ADO.NET”中讨论。在 ADO.NET 中,可以说对 ADO 进行了“大修”(可能用“代码完全重写”更准确),ADO 的基础——记录集已经退役了,像前面说的,记录集只能理解那些正确地处理数据库数据的操作方法。随着 XML 的出现,提出了一种不同种类的、分级的数据模型,记录集就难以胜任了。在 ASP.NET 中引入了一种新的对象 DataSet,它是数据库在内存中的拷贝,由表、列、关系、约束和数据组成。它允许多个表之间相互关联,类似于数据库间的外键关系。你可以在基于它们的关系的表之间进行定位。DataSet 具有很多突出的性质,比如,它可以应用各种数据源,还可以接收来自数据库、XML 文件、代码或者是用户输入的数据。无论在 DataSet 中数据的来源是什么,都可以通过同一套标准的 API 进行操作。DataSet 以 XML 流传输,这样就能在任意两个对象(不仅仅是 COM 对象)之间传输而不会受到防火墙的干涉。而从一个组件向另外一个组件传输分离的 ADO 记录集时,则必须要使用 COM 编程。

用户控件在第 6 章“用户控件”中讨论,它是由服务器端包含文件(include files)演变而来的,包含文件是静态文件。另一方面,用户控件提供了对象模型支持,使得能够对属性

和用户方法进行编程。用户控件的工作原理就像 ASP 的内部控件，可以暴露属性和方法。在第 6 章设计了一个用户控件，它可以根据在 XML 文件中指定的 URL 为一个网站自动建立导航链接。

ASP.NET 有一种很清楚很优雅的方法来创造自定义控件。在第 7 章“自定义控件”中，详细论述了创造自定义控件的过程。同时还会向你展示如何创建一个组件（称之为 GenEditAdd），你可以通过这种方法来扩展 DataGrid 的功能。DataGrid 没有插入记录的功能，使用 GenEditAdd 组件可以自动化插入记录的过程。你可以使用 GenEditAdd 组件代替 DataGrid 提供的编辑功能，这要求你把大量的事件编写成代码。GenEditAdd 组件只要求简单的属性设置并且代码产生是自动化的。

把事务逻辑封装进组件中一直是 Web 和客户/服务器应用程序的很重要的部分。ASP.NET 大大减化了注册组件的过程。如果你过去开发过 COM 对象，一定知道注册组件的痛苦。一个组件必须使用 regsvr32 工具注册，如果修正一下某个组件，为了重新注册这个组件，整个服务器就得停下来，我们称之为 DLL Hell。在 ASP.NET 中，组件可以简单地复制和粘贴在 bin 目录中，而不需要注册更新。在第 8 章“业务对象”中我们来研究这个重要的主题。

Web 服务是 .NET 舞台的主角，也是本书的主要内容，它的众多项目也反映了它的重要性。一个 Web 服务的实现是三方面的结合：一个用 .NET 兼容语言编写的函数、SOAP 和 XML。当你在很多地方都需要重用某个逻辑时，最好的办法是把它用代码作为一个函数写出来。共享一些通用目标的函数集可以结合成一个事务对象。例如，4 个基本的数据库操作是插入、删除、更新和选择，我们可以为每个操作写一个通用的函数，然后把它们打包成一个事务对象，称之为 DataBaseClass，那么这个把几个函数绑在一起的类可以在任何需要使用其功能的对象中初始化和调用。一个 Web 服务就是具有 Web 功能的事务对象，是可以提供 Web 调用的函数集，惟一不同之处是在这个函数之前有一个特殊的标签把它们标记为 Web 服务。一个叫做 SOAP（Simple Object Access Protocol，简单对象访问协议）的标准设定了进行函数调用的机器必须遵循的规则，机器通过发回一个结果集来响应函数调用。请求和响应用 XML 产生，而 XML 文档遵循由 SOAP 标准设定的规则。XML 信息交换使得一个客户应用程序能够调用服务器上的函数，而不管运行的是什么操作系统，是用什么编程语言写的程序，或者它们支持什么组件模型。这一切都是因为 XML 基本上是一个文本文件，所有的机器都能理解，而且 SOAP 使用 HTTP，HTTP 是最普通的 Internet 传输协议，所有的 Web 浏览器都使用它。在第 9 章“使用 ASP.NET 的 Web 服务”中对 Web 服务进行了详细的讨论。

第 10 章“ASP.NET 应用程序”涉及的是 ASP.NET 应用程序。一个 ASP.NET 应用是一个 IIS 虚拟目录和它的子目录。你的所有 Web 应用程序文件都放在这个文件夹中，这个文件夹有一个特殊的子目录 bin，所有编译好的事务对象和 Web 服务都放在该子目录中。当你想要注册一个新组件时，你只需把 DLL 文件复制然后粘贴到这个文件夹中就可以了（与使用 regsvr32 不同）。这个文件夹也包含两个特殊的文件：Web.Config 和 global.asax。Web.config 是一个 XML 文件，你可以用它来配置应用程序的各个方面。例如，可以用它来配置安全、缓存或跟踪服务。global.asax 文件包含应用程序级的编程指令、应用程序和会话级的事件处理程序以及声明那些应用程序的各个部分都能访问的对象。一般来讲，这个文件增强了 ASP 中 global.asa 文件提供的功能。

第 11 章“缓存技术”讲述的缓存技术是把经常访问的网页保持在存储器中的过程。缓存

技术的理论基础是网站的一些内容的创建是很昂贵的,这些内容应该一旦产生就存储在存储器中,并保持一定的时间,直到内容改变了。随后对这些资源的调用不是重新创建,而仅仅是从缓存中重新得到这些资源。这样的内容是典型的保持一定时间不改变的资源,比如购物清单和价格清单。

第 12 章“跟踪”讨论了跟踪。开发者经常会用写大量的 `Response.Write()` 语句代码来调试错误代码。当找到问题所在后,必须清除掉这些调试语句。这种方法很麻烦而且容易出错,因为你偶尔也会和调试语句一起删除不该删除的代码。ASP.NET 引入了一种很好的写这些调试代码的方法,通过增加页面级指令来进行调试(或者在 `Web.config` 文件中配置),然后使用 `Trace.write` 代替 `Response.write` 写调试语句。当调试窗体的时候,就不必再从窗体主体中删除这些语句了。只需简单地禁止跟踪,这些语句就不会在浏览器中显示出来了。

在第 13 章讨论安全性。ASP.NET 通过认证提供者实现认证,这些认证提供者是包含鉴别发出申请的用户的凭证所必需的代码。当前有三种认证可用:Windows 认证、Passport 认证和 Cookie 认证。这三种认证我们都将进行讨论。

项目 1 (第 14~18 章)告诉你如何使用 ASP.NET Web 窗体建立一个具有 Web 功能的个人财务管理器。这个项目包括 4 章的内容。在这个项目中,我带来一个产品,它是一个基于客户/服务器的个人财务账目管理模块,并且为它增加了 Web 功能。个人财务管理器是一个账目管理应用程序,比如 Quicken 或 Microsoft Money,可以用来操作你的银行、现金、信用卡和投资账户。这个项目是要设计一个产品质量账目管理应用程序,并且要利用已存储好的过程和数据库触发器。它包括维护你的账目图表的 Web 窗体、事务细节,甚至能草拟一个试算表报表。

Internet 为传统的应用程序设计方法带来了一些令人兴奋的机会,账目管理应用程序的各种模块不必再用“线”连在一起。使用 ASP.NET 和 Web 服务,我们就能设计出利用 Internet 和 HTTP 发送和接收数据的应用程序。在项目 1 的补充(包括 5 章的内容)中,我们建立了一个可以与任何数据库进行交互访问的通用数据库访问服务。这个 Web 服务接收一个数据库连接并以有效的 SQL 查询为参数,如果查询是一个操作查询(插入、更新或删除),正确的操作就会被执行,如果查询是一个选择查询(返回一个结果集),就返回给调用对象一个 `DataSet`。然后这个 `DataSet` 用来绑定像 `DataGrid` 一样的控件。我通过把它结合到项目 1 开发的个人财务管理器中论证了这个服务。这个项目也证明了在第 6 章“用户控件”中建立的导航用户控件的用途。这个导航控件使用在 XML 文件中定义的 URL 建立了应用程序的站点导航系统。使用与主应用程序分离的导航系统的好处是你可以添加或删除链接(通过修改 XML 文件)而不用改变应用程序的 Web 页面。

在项目 2 (第 19~22 章)中讲解了另外一个应用程序,一个传统的客户/服务器应用程序,我们对它进行了修补,增加了 Web 功能。这是一个库存管理应用程序,这个应用程序利用了我们在项目 1 中补充开发的数据库 Web 服务类,也利用了各种存储过程和触发器。

在项目 3 (第 23~24 章)中增强了自定义控件 `GenEditAdd` (最初在第 7 章开发的)的功能,`GenEditAdd` 控件可以用来插入或更新数据库记录。虽然 `DataGrid` 确实具有编辑数据库记录的能力,但它却不能插入记录,`DataGrid` 的编辑模式是非常麻烦的,因为你不得不把大量的 `DataGrid` 事件写成代码。这个控件的开发是用来提高 `DataGrid` 的有效性,它能与 `DataGrid` 建立联系并一致的方式提供编辑和插入功能。该控件通过设置各种属性来发挥作用,并且代码产

生是自动的。

在项目 4 (第 25~27 章) 中, 讨论了 Visual Studio.NET 的重要特征。第 25 章“使用强类型数据集显示数据库数据”首先概述了 Visual Studio.NET 的重要特征, 聚焦在各种向导、工具和可用的组件上。当然, 也会向你展示如何通过利用 Visual Studio.NET 的拖放特性来使用 DataSet 控件显示数据库信息。在第 26 章“用 Visual Studio.NET 编写 CRUD 应用程序”中, 将告诉你如何添加、删除和更新数据库记录, 也会向你展示如何通过 Visual Studio.NET 的分页和排序来定制 DataGrid。最后, 在第 27 章“使用 Visual Studio.NET 创建 Web 服务”, 将会向你讲述如何通过 Visual Studio.NET 来开发和使用 Web 服务。

在项目 5 (第 28~32 章) 中用 C# 实现了一个基于数据编程的公司小灵通销售信息管理系统。与前面的项目相比, 此系统架构设计得较为灵活, 编写了相应的通用类并采用了简洁的权限模块处理。通过这种方法实现与数据库的交互, 大大减轻了开发复杂度, 有助于系统管理员统一管理。

## 1.1 安装 .NET Framework SDK

.NET SDK 可以从 Microsoft 的下载站点 <http://msdn.microsoft.com/net/> 下载。下载文件相当大, 你应该考虑订购一个 CD, 很便宜, 只需支付很少的运费。

可以得到两种版本, 分别为标准版和高级版。高级版包括一些额外的功能, 如输出缓存、web farm 会话状态、代码访问以及对 4 种以上 CPU 的支持。

安装方法通过运行 setup.exe 文件直接安装。如果有提示的话, 你应该升级 Windows Installer Component, 并为你的 Windows 打上最新的补丁。还应该升级你的 MDAC (Microsoft Data Access Components) 到最新版本, 当前最新版本是 2.7。如果安装器提示 ADO 2.7 没有安装, 不管它, 继续安装。后面会让你选择是否安装 SDK 例子, 这些例子是丰富的信息资源, 应该选择安装。一个很有名的实例 MSDE (Microsoft Data Engine) 会随着这些例子安装进去, 它包含了实例数据库。

### 提示

Microsoft SQL 的限时评测版可以从 <http://www.microsoft.com/sql/evaluation/trial/2000/default.asp> 获得, 也可以只需支付运费来订购它的 CD。

当你的 SQL 服务器开始运行之后, 就可以安装 ASP.NET QuickStart 实例。这些例子是非常好的 ASP.NET 培训资源, 要安装这些例子, 打开 Microsoft NET Framework SDK/Samples and Quick Start Tutorials 链接, 这是随着 SDK 安装添加到你的程序组里的。一旦这些实例安装完成, 就可以通过 <http://localhost/quickstart/default.asp> 进行访问。

在安装完 SDK 之后, 你所需要的只是用一个文本编辑器来编写你的脚本。也可以订购 Visual Studio CD (还是很小的花费), 用它来开发你的脚本。如果已经有了 Visual Studio 的 CD, SDK 框架就在第 2 张 CD 上。

我曾经谈到用 Visual Studio 进行开发, 在项目 5 中将有一个完整的项目来探索这套开发工具。把 Visual Studio 的讨论留到最后, 因为我想让读者在使用 Visual Studio IDE (它隐藏了代码开发的复杂性) 这样的向导式工具之前, 先熟悉 ASP.NET 的内部组件。我向大家强烈推荐

一个叫 TextPad 的文本编辑器, 这是一个共享软件, 可以从 <http://www.textpad.com> 下载。也可以从相应的站点下载 .NET 的语法定义文件, 这个文件会以不同的颜色显示各种不同的 ASP.NET 关键词。

## 1.2 安装本书的实例数据库

在本书的 Web 站点 <http://www.premierpressbooks.com/downloads.asp> 提供了本书的项目和实例使用的实例数据库。在开始时安装这个数据库是很重要的, 要做到这一点, 在 `\sample\database` 文件夹下找到 `create.sql` 文件, 在 ISQL (查询分析器) 下运行它, 然后会产生大量的表、存储过程和触发器。这个数据库对象产生的细节在附录 A “安装样例数据库” 里有详细的描述。

## 第 2 章 ASP.NET Web 窗体和控件概述

ASP.NET 的窗体设计克服了以前 ASP 页面中的很多缺陷。在 ASP 页面里，由于 HTML 元素和脚本代码不可避免地要互相混杂在一起，这使得最终页面的代码非常混乱，而且这些页面不易于用 WYSIWYG (What You See Is What You Get, 所见即所得) 工具编辑。ASP.NET 改进了 ASP 页面，并对许多功能进行了加强。ASP.NET 提供了类似于 Visual Basic 的基于服务器和基于事件的编程模型，它引入了一种叫做“Code Behind (代码分离)”的技术，这种技术允许开发者把脚本代码和 HTML 标记分开存放在不同的文件里。ASP.NET 引入了两套控件：HTML 控件和 Web 控件，统称为“服务器控件”。这些控件弥补了 HTML 的先天不足，此外还在往返传送中提供了保留浏览器状态的方法，还可以检测浏览器的类型，它们还可用作某些组合控件的内建模块。当用户提交一个窗体或发生引起向服务器发送消息的事件时，就会出现往返传送。例如，用户填写了一个窗体的文本框并点击了“提交”按钮，服务器处理传递来的信息然后把页面发送回客户端显示，窗体的原始状态由 ASP.NET 保持。这就意味着当用户填写了文本框然后向服务器提交了窗体，往返传送完成后，文本框还会保留这个信息。这是个很受欢迎的改变，因为使用传统的 ASP 编程时，用户输入的值一旦发送就没有了，开发者不得不注意状态的保持。

### 2.1 基本技术

创建一个 ASP.NET 窗体，只需把一个文本文件或者 HTML 文件以 .aspx 扩展名保存即可，无需特殊的工具，甚至可以使用记事本来进行编辑。也可以使用 Visual Studio.NET 这样的快速应用程序开发环境 (RAD) 来创建 ASP.NET 窗体，利用它可以直接在窗体上拖放控件。

ASP.NET 还提供了选择性的向下兼容。例如，可以使用标准的 ASP 代码，并且可以把脚本标记和带有 <%> 代码片段的 HTML 元素混合在一起，ASP 和 ASP.NET 应用程序可以一起在 IIS 上运行而不会有什么冲突。但是，使用 Visual Basic Scripting Edition 开发的 ASP 应用程序需要做一些修改才能在 ASP.NET 下使用。

在 ASP.NET 里，脚本块是被编译而不是被解释执行的，因此性能得到了提高。编译代码就是为了把代码指令转变成机器语言，但是在 ASP.NET 中，代码并不是直接编译成机器语言，而是编译成一种叫做 Microsoft Intermediate Language (MSIL 或 IL) 的中间语言，IL 代码由实时编译器 (just-in-time compiler, JIT) 进一步编译成机器语言。JIT 编译器会编译被调用的每一部分，而不是一次性把整个应用程序编译，这就使得启动速度很快。编译代码的结果被存储起来直到退出应用程序，这样当每次调用某部分代码的时候就不必再重新编译了。通过这个过程，可以使 IL 代码的执行速度和执行本地机器代码的速度相媲美。

### 2.2 状态管理

虽然可以使用脚本块，但脚本块不能将自身置于一个清晰的编程环境中。考虑要在 post



back 表单中保持状态, 即接受用户输入的值然后再跳转到该页面。因此必须记录输入值, 这样, 如果用户的输入有误, 它就可以显示用户输入的错误值以使用户纠正。

在 ASP 环境中编写这样一个窗体的代码, 使用 Response 对象来提取输入值并使用 `<%=some variable%>` 代码片段来显示前面已经输入的值。下面是一个例子。

### State.asp

```
-----
<html>
  <form method="post">
    <br>
    <h3> Name: <input name="name" type="text" value="<%=Request.form("name")%>">
    <br> <input type="submit" value="submit">
  </form>
</html>
-----
```

在 ASP.NET 中, 在窗体中使用服务器控件时, 状态管理是自动完成的。如下例:

### State.aspx

```
-----
<html>
  <body style="background-color="beige"; font-family="verdana"; font-size='10pt'">
  <form method="post" runat=server>
    <h3> Name: <asp:textbox id="Name" runat="server"/>
    <asp:button text="Lookup" runat="server"/>
  </form>
</body></html>
-----
```

注意, 窗体是自动记住输入值的。当与服务器状态管理控件结合使用时会有一些缺陷, 只能使用 post 方法, 而且在页面中只能有一个窗体。

## 2.3 页面事件

由于 ASP.NET 的核心是面向对象的, 因此可以用类似 Visual Basic 的方式为各种事件编写代码。当窗体加载时, 就会激发 Page\_Load 事件, 窗体控件就变得可用了; 当用户继续与窗体进行交互时, 又产生其他的事件; 当页面卸载时发生窗体卸载事件。由于这个基于事件的架构, 开发者最终可以在 Web 应用程序上开始使用基于事件的编码技术。如图 2.1 显示了如何使用这些技术接收用户输入的值来完成一个计算。

### Events.aspx

```
-----
<%@ Page Language="vb" %>
<html>
  <head>
    <script Runat="server">
      Sub Calculate(src As Object, e As EventArgs)
        Amount.Text = Cstr(cint(qty.text)*cint(price.text))
      End sub
    </script>
  </head>
</html>
-----
```