第一图书网, tushu007.com

<<ASP.NET 3.5 SP1高级编程>>

图书基本信息

书名：<<ASP.NET 3.5 SP1高级编程>>

13位ISBN编号：9787302215486

10位ISBN编号：7302215480

出版时间：2010-1

出版时间：清华大学出版社

作者：(美) 伊文詹 (Bill Evjen) (美) 汉森姆 (

页数：1507

译者：姜奇平

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

第一图书网, tushu007.com

前言

ASP.NET 3.5是一种建立Web解决方案的令人惊异的技术。
早在ASP.NET 1.0版本于2000年发布时，许多人就认为它在Web应用程序开发方面迈出了具有革命性的一步。
而后来的ASP.NET 2.0更激动人心、更富有革命性，ASP.NET 3.5 Service Pack 1(SP1)则继续朝着这个方向前进，为在Web上建立应用程序提供了目前最佳的框架。
ASP.NET 3.5 SP1建立在已发布的ASP.NET 1.0基础之上，但它主要关注的是开发人员的效率。
本书介绍ASP.NET的所有内容，除了论述新主题外，还列举了一些有关这些新技术的例子。
0.1 简史　　在各个公司考虑为Internet开发应用程序之前，应用程序的开发主要集中在桌面应用程序上。
这些胖客户端应用程序适用于所有场合：家用计算、游戏、办公等。
这种应用程序模型的流行可谓势不可挡。
在这个过程中，Microsoft开发胖客户端应用程序使用的是其主要产品Visual Basic(VB)。
Visual Basic不仅是一种编程语言，它还与便于开发胖客户应用程序的IDE有密切的关系。
在Visual Basic模型中，开发人员可以把控件拖放到窗体上，设置这些控件的属性，给它们提供代码来处理控件的事件。
例如，终端用户单击Visual Basic窗体上的一个按钮时，窗体的隐藏代码就会处理该事件。
在20世纪90年代中期，Internet开始崭露头角。
Microsoft未能将Visual Basic模型转向基于Internet应用程序的开发。
Internet的确有强大的功能，此时胖客户端应用程序模型面临的问题也开始显露出来。
基于Internet的应用程序创建了每个人都能访问的一个应用程序实例。
拥有应用程序的一个实例，意味着在对应用程序进行升级或打补丁时，对这个实例的修改会立即展现给通过浏览器访问该应用程序的每个用户。
为了进入Web应用程序行业，Microsoft开发了Active Server Pages (ASP)。
ASP是开发Web页面的一种快捷方式。
ASP页面由一个页面组成，其中包含了标记和语言的混合。
ASP的强大之处在于，在将页面发送给终端用户的Web浏览器之前，可以在页面上包含在Web服务器上执行的VBScript或JScript代码指令。
这是创建动态Web页面的一种简单方式，动态Web页面是根据开发人员指定的参数进行定制的。
ASP在尖括号和百分号[% %]之间使用脚本来控制服务器端的行为。
开发人员可以先从一组静态的HTML开始建立ASP页面。
页面需要的动态元素用脚本语言(如VBScript或JScript)来定义。
当用户使用浏览器从服务器上请求页面时，asp.dll (这是一个ISAPI应用程序，它在脚本语言和Web服务器之间架起了一座桥梁)就提取页面，根据脚本中指定的编程逻辑定义页面中的动态部分。
定义页面中的所有动态部分后，所得到的结果就是一个HTML页面，该页面被输出到请求客户机的浏览器上。
在开发Web应用程序模型的过程中，静态HTML中混合了越来越多的语言，以帮助处理输出页面的操作方式和外观。
随着时间的推移，ASP页面上将出现非常多的语言、脚本和纯文本，开发人员开始把使用这些特性的页面称为spaghetti code(意大利细面条式代码)。
例如，页面上可能使用了HTML、VBScript、JavaScript、层叠样式表、T-SQL等。
在这种情况下，页面是很难管理的。
ASP进一步演化并推出了新版本。
ASP 2.0和3.0开始流行，因为这些技术非常便于Web页面的创建。
它们的日益流行是因为它们出现在20世纪90年代后期，那时诞生了.com。
在这个阶段，人们开发了许多Web页面和门户，ASP是一种业界领先的技术，个人和公司都使用该技

术建立Web页面。
甚至到现在，仍可以在Internet上找到许多.asp页面——包括Microsoft的一些Web页面。
但在1998年后期，Active Server Pages发布其最后一个版本时，Microsoft雇员Marc Anders 和 Scott Guthrie有了另一个想法。
他们称之为XSP(这个缩写词没有什么特别的含义)——这是以面向对象的方式创建Web应用程序的方法，而不是使用ASP 3.0的过程式方法来创建。
他们把这个想法告诉Microsoft中许多不同的团体，并得到了广泛的认可。
2000年夏，Microsoft的Professional Developers Conference发布了其测试版ASP+。
与会者都非常渴望使用它。
该技术发布时(与.NET Framework 1.0的最终版本一起发布)，被重新命名为ASP.NET——加上.NET标记是因为在那时Microsoft的大多数新产品都加上了这个标记。
在引入.NET之前，传统ASP提供的模型和Visual Basic中开发的模型大相径庭，很少有VB开发人员能开发Web应用程序，Web开发人员也不能开发VB的胖客户应用程序。
这是一条很大的鸿沟，ASP.NET则为此搭建了一座桥梁。
ASP.NET把Visual Basic样式的事件模型引入到Web应用程序的开发中，为无状态的HTTP提供了迫切需要的状态管理技术。
其模型非常类似于早期的Visual Basic模型，因为开发人员可以把控件拖放到设计界面或窗体上，处理控件的属性，处理控件的代码，甚至基于控件的生存周期指定一些事件。
ASP.NET综合了这两个模型的优点，如本书后面所述。
读者一定很希望使用ASP.NET 3.5 SP1这个最新版本，看看这种新技术能给自己带来什么。
下面就讨论ASP.NET的目标，看看它有什么新内容。
0.2 ASP.NET的目标　　ASP.NET 3.5是该产品的另一个重要版本，建立在.NET Framework 2.0的核心功能之上，并带有额外的类和功能。
Framework的这个版本在Microsoft内部的代码名称是Orcas，读者可能听说过其他人把ASP.NET的这个版本称为ASP.NET Orcas。
ASP.NET 3.5继续使ASP.NET开发人员成为Web领域中最高效的开发人员。
本书也重点介绍ASP.NET 3.5 SP1版本中的ASP.NET 3.5和.NET Framework 3.5的新功能。
Microsoft小组刚开始开发ASP.NET 2.0时，就设定了要实现的目标。
这些目标集中于开发人员的效率、管理、性能和可伸缩性。
0.2.1 开发人员的效率　　ASP.NET 3.5的主要目标是效率。
ASP.NET 1.x的发布就已经达到了很高的效率，但效率还能进一步提高吗？
ASP.NET开发小组的一个目标是去除ASP.NET中原来必需的大量繁琐的编码，使常见的ASP.NET任务更容易完成。
开发人员的高效率将在本书中体现出来。
在介绍这些功能之前，首先看看以前的ASP.NET 1.0技术，以便与ASP.NET 3.5进行比较。
程序清单0-1使用ASP.NET 1.0在Web页面上建立了一个表，并可以对所提供的数据进行简单的分页。

第一图书网, tushu007.com

<<ASP.NET 3.5 SP1高级编程>>

内容概要

《ASP.NET 3.5 SP1高级编程(第6版)》是ASP.NET名著的升级版，与上一版相比，本版内容更全面、技术更新，新增了ADO.NET Entity Framework、ADO.NET Dynamic Data和ADO.NET Data Services等内容。

《ASP.NET 3.5 SP1高级编程(第6版)》由世界一流的作者团队编写，全面介绍ASP NET的各种编程技巧，详细讲述ASP NET 3 5 SPl的诸多功能，同时提供C#和VB范例代码。

第一图书网, tushu007.com

作者简介

Bill Evjen，是．NET技术的积极支持者、Microsoft区域主管和MVP。
自从．NET在2000年推出以来，他就积极地介入其中，建立了第一个实际意义上的．NET用户组。
Bill是一位畅销书作者，他的作品包括畅销书《C#高级编程（第6版）》、《Visual Basic 2008高级编程（第5版）》等。

Scott Hanselman，是Microsoft开发部的一位资深项目经理，其主要工作是大力宣传Microsoft在软件开发方面的成就，他还涉足MVP和RD程序领域。
只要有人愿意倾听，他就会谈及计算机以及其他相关主题。

Devin Rader，是Infragistics Web Client团队的一位产品经理，负责领导Infragistics ASP.NET和Silverlight产品的开发，Devin是.NET技术的积极支持者，NET开发团队的重要成员，他经常在用户组上发表文章，还与他人合著了Silverlight1.0等图书。

第一图书网, tushu007.com

书籍目录

第1章 应用程序和页面框架 11.1 应用程序的位置选项 11.1.1 内置的Web服务器 11.1.2 IIS 31.1.3 FTP 31.1.4 Web站点需要FrontPage Extensions 41.2 ASP.NET页面结构选项 41.2.1 内联编码 61.2.2 隐藏代码模型 81.3 ASP.NET 3.5的Page指令 101.3.1 @Page指令 111.3.2 @Master指令 131.3.3 @Control指令 141.3.4 @Import指令 151.3.5 @Implements指令 161.3.6 @Register指令 171.3.7 @Assembly指令 171.3.8 @PreviousPageType指令 171.3.9 @MasterType指令 181.3.10 @OutputCache指令 18.1.3.11 @Reference指令 181.4 ASP.NET页面的事件 191.5 处理回送 201.6 跨页面的传送 211.7 ASP.NET应用程序文件夹 271.7.1 \App_Code文件夹 271.7.2 \App_Data文件夹 311.7.3 \App_Themes文件夹 321.7.4 \App_GlobalResources文件夹 321.7.5 \App_LocalResources文件夹 321.7.6 \App_WebReferences文件夹 321.7.7 \App_Browsers文件夹 321.8 编译 331.9 Build Provider 361.9.1 使用内置的Build Provider 371.9.2 使用自己的Build Provider 371.10 Global.asax 421.11 通过Visual Studio 2008使用类 461.12 小结 50第2章 ASP.NET服务器控件和客户端脚本 512.1 ASP.NET服务器控件 512.1.1 服务器控件的类型 522.1.2 用服务器控件构建页面 532.1.3 处理服务器控件的事件 542.2 给服务器控件应用样式 572.2.1 控件的常见属性 572.2.2 使用层叠样式表改变样式 582.3 HTML服务器控件 612.3.1 HtmlControl基类 632.3.2 HtmlContainerControl类 642.3.3 所有的HTML类 642.3.4 使用HtmlGenericControl类 652.4 通过JavaScript处理页面和服务器控件 662.4.1 使用Page.ClientScript.Register ClientScriptBlock 672.4.2 使用Page.ClientScript. RegisterStartupScript 692.4.3 使用Page.ClientScript.RegisterClientScriptInclude 712.5 客户端回调 712.5.1 比较典型的回送和回调 712.5.2 使用回调功能：一种简单的方法 732.5.3 使用带一个参数的回调功能 772.5.4 使用回调功能：一个复杂的例子 802.6 小结 86第3章 ASP.NET Web服务器控件 873.1 Web服务器控件概述 873.2 Label服务器控件 883.3 Literal服务器控件 903.4 TextBox服务器控件 903.4.1 使用Focus()方法 913.4.2 使用AutoPostBack 923.4.3 使用AutoCompleteType 933.5 Button服务器控件 943.5.1 CausesValidation属性 943.5.2 CommandName属性 953.5.3 使用客户端JavaScript的按钮 963.6 LinkButton服务器控件 973.7 ImageButton服务器控件 983.8 HyperLink服务器控件 993.9 DropDownList服务器控件 1003.10 从集合中可视化地删除数据项 1023.11 ListBox服务器控件 1043.11.1 允许用户选择多项 1043.11.2 使用ListBox控件的例子 1053.11.3 给集合添加项 1073.12 CheckBox服务器控件 1073.12.1 如何确定复选框是否被选中 1083.12.2 给复选框赋值 1093.12.3 排列复选框的文本 1093.13 CheckBoxList服务器控件 1093.14 RadioButton服务器控件 1123.15 RadioButtonList服务器控件 1143.16 Image服务器控件 1153.17 Table服务器控件 1163.18 Calendar服务器控件 1193.18.1 从Calendar控件中选择一个日期 1193.18.2 从Calendar控件中选择要输出的日期格式 1203.18.3 选择日期、星期或月份 1213.18.4 使用日期范围 1213.18.5 修改日历的样式和操作方式 1233.19 AdRotator服务器控件 1263.20 Xml服务器控件 1273.21 Panel服务器控件 1283.22 PlaceHolder服务器控件 1303.23 BulletedList服务器控件 1313.24 HiddenField服务器控件 1353.25 FileUpload服务器控件 1373.25.1 使用FileUpload控件上传文件 1373.25.2 给上传文件指定正确的ASP.NET权限 1403.25.3 理解文件大小的限制 1413.25.4 从同一个页面上传多个文件 1423.25.5 把上传的文件放在Stream对象中 1453.25.6 把文件内容从Stream对象移动到Byte数组中 1453.26 MultiView和View服务器控件 1463.27 Wizard服务器控件 1493.27.1 定制边栏导航 1513.27.2 AllowReturn属性 1513.27.3 使用StepType属性 1513.27.4 给Wizard控件添加标题 1523.27.5 使用向导的导航系统 1533.27.6 使用Wizard控件的事件 1533.27.7 使用Wizard控件显示窗体元素 1553.28 ImageMap服务器控件 1593.29 小结 161第4章 验证服务器控件 1624.1 有效性验证 1624.2 客户端和服务器端的验证 1634.3 ASP.NET验证服务器控件 1634.3.1 验证原因 1654.3.2 RequiredFieldValidator服务器控件 1654.3.3 CompareValidator服务器控件 1704.3.4 RangeValidator服务器控件 1724.3.5 RegularExpressionValidator服务器控件 1764.3.6 CustomValidator服务器控件 1774.3.7 ValidationSummary服务器控件 1824.4 关闭客户端验证功能 1844.5 为错误通知使用图像和声音 1854.6 使用验证组 1864.7 小结 190第5章 使用Master页面 1915.1 需要Master页面的原因 1915.2 Master页面基础 1935.3 编写Master页面 1945.4 编写内容页面 1975.4.1 混合页面类型和语言 2005.4.2 指定要使用的Master页面 2015.4.3 使用页面标题 2025.4.4 使用Master页面中的控件和属性 2035.5 在Master页面中指定默认内容 2105.6 以编程方式将Master页面赋予内容页面 2125.7 Master页面的嵌套 2125.8 容器特定的Master页面 2165.9 事件的触发顺序 2175.10 高速缓存Master页面 2175.11 ASP.NET AJAX和Master页面 2185.12 小结 220第6章 主题和Skin 2216.1 使用ASP.NET中的主题

2216.1.1 给单个ASP.NET页面应用主题 2216.1.2 把主题应用于整个应用程序 2236.1.3 删除服务器控件中的主题 2236.1.4 删除Web页面上的主题 2246.1.5 使用Master页面时理解主题 2246.1.6 StyleSheetTheme属性 2256.2 创建自己的主题 2256.2.1 创建正确的文件夹结构 2256.2.2 创建Skin 2266.2.3 在主题中包含CSS文件 2286.2.4 使主题包含图像 2316.3 定义多个Skin选项 2336.4 编程使用主题 2356.4.1 编程指定页面的主题 2356.4.2 编程指定控件的SkinID 2366.5 主题、Skin和定制控件 2366.6 小结 240第7章 ASP.NET 3.5中的数据绑定 2417.1 数据源控件 2417.1.1 SqlDataSource控件 2437.1.2 LINQ数据源控件 2537.1.3 AccessDataSource控件 2577.1.4 XmlDataSource控件 2587.1.5 ObjectDataSource控件 2597.1.6 SiteMapDataSource控件 2637.2 配置数据源控件的高速缓存 2647.3 存储连接信息 2647.4 组合使用绑定的列表控件和数据源控件 2677.4.1 GridView控件 2677.4.2 编辑GridView中的数据行 2817.4.3 删除GridView数据 2877.4.4 DetailsView控件 2897.4.5 使用DetailsView插入、更新和删除数据 2947.4.6 ListView控件 2967.4.7 FormView控件 3037.5 其他数据绑定控件 3077.5.1 DropDownList、ListBox、RadioButtonList和CheckBoxList 3077.5.2 TreeView控件 3077.5.3 Ad Rotator控件 3087.5.4 Menu 3087.6 内联数据绑定语法 3097.6.1 DataBinder语法的变化 3107.6.2 XML数据绑定 3107.7 表达式和表达式构建器 3117.8 小结 316第8章 使用ADO.NET管理数据 3178.1 ADO.NET的基本功能 3178.1.1 常见的ADO.NET任务 3188.1.2 ADO.NET的基本名称空间和类 3228.1.3 使用Connection对象 3238.1.4 使用Command对象 3258.1.5 使用DataReader对象 3268.1.6 使用数据适配器 3288.1.7 使用参数 3318.1.8 DataSet和DataTable 3338.1.9 将Oracle用作ASP.NET 3.5的数据库 3388.2 DataList服务器控件 3408.2.1 可用的模板 3408.2.2 使用ItemTemplate 3418.2.3 使用其他布局模板 3438.2.4 使用多个列 3468.3 ListView服务器控件 3478.3.1 可用的模板 3478.3.2 使用模板 3478.3.3 创建布局模板 3498.3.4 创建ItemTemplate 3508.3.5 创建EditItemTemplate 3518.3.6 创建EmptyItemTemplate 3528.3.7 创建InsertItemTemplate 3528.3.8 结果 3538.4 将Visual Studio用于ADO.NET任务 3548.4.1 建立与数据源的连接 3548.4.2 使用DataSet设计器 3568.4.3 使用DataSet对象CustomerOrders 3608.5 命令的异步执行 3648.5.1 SqlCommand类的异步方法 3648.5.2 IAsyncResult接口 3658.5.3 AsyncCallback 3658.5.4 WaitHandle类 3658.5.5 ADO.NET中异步处理的方式 3668.5.6 取消异步处理 3848.5.7 异步连接 3848.6 小结 384第9章 使用LINQ查询 3859.1 LINQ to Objects 3859.1.1 传统的查询方法 3859.1.2 用LINQ替代传统的查询 3949.1.3 数据分组 4019.1.4 其他LINQ运算符 4029.1.5 LINQ连接 4039.1.6 使用LINQ分页 4049.2 LINQ to XML 4059.3 LINQ to SQL 4109.3.1 通过LINQ插入、更新和删除查询 4179.3.2 扩展LINQ 4219.4 小结 422第10章 使用XML和LINQ to XML 42310.1 XML的基础 42410.1.1 XML InfoSet 42610.1.2 XSD/XML模式定义 42610.1.3 在Visual Studio 2008中编辑XML和XML模式 42810.2 XmlReader和XmlWriter 43010.2.1 使用XDocument替代XmlReader 43310.2.2 使用模式和XmlTextReader 43410.2.3 使用XDocument根据模式来验证 43610.2.4 包含NameTable优化 43710.2.5 从XML中提取.NET CLR类型 43910.2.6 ReadSubtree和XmlSerialization 44110.2.7 通过LINQ to XML从XML中创建CLR对象 44310.2.8 用XmlWriter创建XML 44410.2.9 用LINQ for XML创建XML 44610.2.10 2.0版本中XmlReader和XmlWriter的改进 44910.3 XmlDocument和XPathDocument 44910.3.1 DOM存在的问题 44910.3.2 XPath、XPathDocument和XmlDocument 45010.4 DataSet 45410.4.1 将DataSet保存到XML中 45410.4.2 XmlDataDocument 45510.5 XmlDataSource控件 45710.6 XSLT 46110.6.1 XslCompiledTransform 46210.6.2 XSLT调试 46610.7 数据库和XML 46710.7.1 FOR XML AUTO子句 46710.7.2 SQL Server 2005和XML数据类型 47110.8 小结 476第11章 IIS7开发 47811.1 IIS7的模块化体系结构 47811.1.1 IIS-WebServer 47911.1.2 IIS-WebServerManagementTools 48111.1.3 IIS-FTPPulishingService 48111.2 IIS7的扩展体系结构 48211.3 IIS7和ASP.NET集成的管道 48211.4 建立定制的Web服务器 48311.4.1 更新包的依赖关系 48411.4.2 在Windows Vista上安装IIS7 48511.4.3 在Windows Server 2008上安装IIS7 48511.4.4 命令行安装选项 48611.4.5 Unattended安装选项 48711.4.6 升级 48711.5 Internet Information Services(IIS)Manager 48811.5.1 Application Pools 48911.5.2 Web Sites 49111.5.3 层次结构的配置 49311.5.4 委托 49711.6 把应用程序从IIS6迁移到IIS7 50011.7 小结 502第12章 提供程序模型概述 50312.1 提供程序概述 50312.2 ASP.NET 3.5中的提供程序模型 50512.2.1 设置提供程序，以使用Microsoft SQL Server 7.0、2000、2005或2008 50612.2.2 Membership提供程序 51112.2.3 Role提供程序 51512.2.4 Personalization提供程序 51912.2.5 SiteMap提供程序 52012.2.6 SessionState提供程序 52212.2.7 Web事件提供程序 52412.2.8 配置提供程序 53212.2.9 WebParts提供程序 53412.3 提供程序的配置 53512.4 小结 536第13章 扩展提供程序模型 53713.1 提供程序是较大体系结构中的一层 53713.2 通过基于属性的编程改变提供程序的行为 53813.2.1

通过SqlMembershipProvider建立简单的密码结构 53813.2.2 通过SqlMembershipProvider建立强类型的密码结构 54113.3 ProviderBase类 54213.4 建立自己的提供程序 54413.4.1 创建CustomProviders应用程序 54413.4.2 构建需要的类框架 54513.4.3 创建XML用户数据存储 54913.4.4 在web.config文件中定义提供程序实例 55013.4.5 没有实现Membership Provider类的方法和属性 55013.4.6 实现MembershipProvider类的方法和属性 55113.4.7 使用XmlMembership Provider进行用户登录 55913.5 扩展预定义的提供程序 56013.5.1 用新的LimitedSqlRoleProvider提供程序限制角色功能 56013.5.2 使用新的LimitedSql RoleProvider提供程序 56413.6 小结 567第14章 站点导航 56814.1 基于XML的站点地图 56814.2 SiteMapPath服务器控件 57014.2.1 PathSeparator属性 57214.2.2 PathDirection 属性 57414.2.3 ParentLevelsDisplayed属性 57414.2.4 ShowToolTips 属性 57514.2.5 SiteMapPath控件的子元素 57514.3 TreeView服务器控件 57614.3.1 标识TreeView控件的内置样式 57914.3.2 TreeView控件的各个部分 58014.3.3 将TreeView控件绑定到XML文件上 58114.3.4 在TreeView中选择多个选项 58314.3.5 在TreeView控件中指定定制的图标 58614.3.6 指定用于连接节点的线条 58714.3.7 编程使用TreeView控件 58914.4 Menu服务器控件 59414.4.1 对Menu控件应用不同的样式 59514.4.2 Menu事件 59914.4.3 把Menu控件绑定到XML文件上 60014.5 SiteMap数据提供程序 60214.5.1 ShowStartingNode属性 60214.5.2 StartFromCurrentNode属性 60314.5.3 StartingNodeOffset属性 60414.5.4 StartingNodeUrl属性 60414.6 SiteMap API 60514.7 URL映射 60714.8 站点地图的本地化 60814.8.1 为本地化构建Web.sitemap文件 60814.8.2 修改Web.config文件 60914.8.3 创建程序集资源(.resx)文件 61014.8.4 测试结果 61114.9 安全补偿 61214.9.1 启动管理员的角色管理功能 61214.9.2 建立管理员的配置段 61314.9.3 启用安全补偿功能 61414.10 嵌套站点地图文件 61614.11 小结 618第15章 个性化 61915.1 个性化模型 61915.2 创建个性化属性 62015.2.1 添加简单的个性化属性 62015.2.2 使用个性化属性 62115.2.3 添加一组个性化属性 62515.2.4 使用分组的个性化属性 62615.2.5 为个性化属性定义类型 62615.2.6 使用定制的类型 62715.2.7 提供默认值 62915.2.8 把个性化属性指定为只读 63015.3 匿名个性化 63015.3.1 支持终端用户的匿名身份 63015.3.2 使用匿名身份事件 63315.3.3 个性化属性的匿名选项 63415.3.4 有关匿名用户配置信息的警告 63415.4 编程访问个性化功能 63515.4.1 迁移匿名用户 63515.4.2 profile的个性化 63615.4.3 确定是否继续自动保存profile值 63715.5 个性化提供程序 63815.5.1 使用SQL Server Express Edition 63915.5.2 使用Microsoft的SQL Server 7.0/2000/2005/2008 64015.5.3 使用多个提供程序 64215.6 管理应用程序的配置信息 64215.6.1 ProfileManager类的属性 64315.6.2 ProfileManager类的方法 64315.6.3 建立ProfileManager.aspx页面 64415.6.4 检验ProfileManager.aspx页面的代码 64715.6.5 运行ProfileManager.aspx页面 64815.7 小结 649第16章 成员和角色管理 65016.1 身份验证 65016.2 授权 65016.3 ASP.NET 3.5的身份验证 65116.3.1 为成员建立Web站点 65116.3.2 添加用户 65316.3.3 请求证书 66616.3.4 处理验证用户 67316.3.5 显示在线用户数 67616.3.6 处理密码 67716.4 ASP.NET 3.5的授权 68116.4.1 使用LoginView服务器控件 68116.4.2 为角色管理建立Web站点 68416.4.3 添加和检索应用程序角色 68716.4.4 删除角色 68916.4.5 给角色添加用户 69016.4.6 获取某个角色的所有用户 69016.4.7 获取包含某个用户的所有角色 69216.4.8 从角色中删除用户 69316.4.9 检查角色中的用户 69316.4.10 角色的高速缓存方式 69416.5 使用Web站点管理工具 69516.6 Membership API的公共方法 69616.7 Roles API的公共方法 69616.8 小结 697第17章 Portal Framework和Web Part 69817.1 Web Part简介 69817.2 建立动态的模块化Web站点 70017.2.1 WebPartManager控件 70017.2.2 使用区域进行布局 70117.2.3 WebPartZone控件 70417.2.4 允许用户改变页面的模式 70617.2.5 修改区域 71517.3 使用Portal Framework中的类 72017.4 创建定制的Web Part 72317.5 连接Web Part 72817.5.1 建立提供程序Web Part 72917.5.2 建立用户Web Part 73217.5.3 连接ASP.NET页面上的Web Part 73417.5.4 连接Web Part时处理Master页面的难点 73617.6 小结 737第18章 用ASP.NET进行HTML和CSS设计 73818.1 警告 73818.2 HTML和CSS概述 73918.2.1 CSS的引入 74018.2.2 创建样式表 74018.2.3 CSS规则 74218.2.4 CSS继承 75018.2.5 元素的布局和定位 75118.2.6 在Visual Studio中使用HTML和CSS 75818.2.7 ASP.NET 2.0 CSS——友好的控件适配器 76418.3 小结 764第19章 ASP.NET AJAX 76519.1 理解对AJAX的需求 76519.1.1 在AJAX出现之前 76619.1.2 AJAX改变了这个过程 76619.2 ASP.NET AJAX和Visual Studio 2008 76919.2.1 客户端技术 76919.2.2 服务器端技术 77019.2.3 用ASP.NET AJAX开发 77019.3 ASP.NET AJAX应用程序 77119.3.1 建立没有AJAX的简单ASP.NET页面 77319.3.2 建立包含AJAX的简单ASP.NET页面 77419.4 ASP.NET AJAX的服务器端控件 78019.4.1 ScriptManager控件 78019.4.2 ScriptManagerProxy控件 78219.4.3 Timer控件 78419.4.4 UpdatePanel控件 78519.4.5 UpdateProgress控件 78919.5 使用多个UpdatePanel控件 79219.6 使用页面历史记录 79519.7

脚本合并 80119.8 小结 805第20章 ASP.NET AJAX控件工具集 80620.1 下载和安装 80720.1.1 新的Visual Studio模板 80820.1.2 给Visual Studio 2008工具箱添加新控件 80820.2 ASP.NET AJAX控件 81020.2.1 ASP.NET AJAX控件工具集的扩展程序 81220.2.2 ASP.NET AJAX控件工具集中的服务器控件 85720.3 小结 868第21章 安全性 87021.1 身份验证和授权 87021.2 应用身份验证措施 87121.2.1 [authentication]节点 87121.2.2 基于Windows的身份验证 87221.2.3 基于窗体的身份验证 87921.2.4 Passport身份验证 88821.3 验证特定的文件和文件夹 88821.4 编程授权 88921.4.1 使用User.Identity 89021.4.2 使用User.IsInRole() 89121.4.3 用WindowsIdentity显示更多信息 89221.5 身份和模拟 89421.6 通过IIS进行保护 89621.6.1 IP地址和域名限制 89621.6.2 使用文件扩展名 89721.6.3 使用ASP.NET MMC管理单元 89921.6.4 使用IIS 7.0 Manager 90021.7 小结 901第22章 状态管理 90222.1 如何选择 90222.2 ASP.NET中的Session对象 90422.2.1 会话和事件模型 90422.2.2 配置会话状态的管理 90622.2.3 进程中的会话状态 90622.2.4 进程外的会话状态 91422.2.5 SQL支持的会话状态 91822.2.6 用其他提供程序扩展会话状态 92122.2.7 无cookie的会话状态 92222.2.8 选择维护状态的正确方式 92322.3 Application对象 92322.4 查询字符串 92422.5 cookie 92422.6 回送和跨页面的回送 92522.7 隐藏的字段、ViewState和ControlState 92722.8 为短时间状态存储应用HttpContext.Current.Items 93122.9 小结 932第23章 高速缓存 93423.1 高速缓存 93423.1.1 输出高速缓存 93423.1.2 部分页面(UserControl)的高速缓存 93723.1.3 Post-Cache Substitution 93823.1.4 HttpCachePolicy和客户端高速缓存 94023.2 编程高速缓存 94223.2.1 使用Cache对象高速缓存数据 94223.2.2 控制ASP.NET高速缓存 94323.2.3 高速缓存依赖性 94323.3 使用SQL Server高速缓存依赖性 94923.3.1 使数据库支持SQL Server高速缓存禁用功能 95023.3.2 使表支持SQL Server高速缓存禁用功能 95023.3.3 对SQL Server 2000的影响 95023.3.4 查看支持SQL高速缓存禁用功能的表 95123.3.5 使表不支持SQL Server高速缓存禁用功能 95223.3.6 使数据库不支持SQL Server高速缓存禁用功能 95223.3.7 SQL Server 2005高速缓存禁用功能 95323.4 配置ASP.NET应用程序 95423.5 测试SQL Server高速缓存禁用功能 95523.5.1 给页面添加多个表 95723.5.2 给SQL Server高速缓存依赖性和Request对象建立关联 95723.5.3 给SQL Server高速缓存依赖性和Cache对象建立关联 95823.6 小结 961第24章 调试和错误处理技术 96224.1 设计期间的支持 96224.1.1 语法通知 96224.1.2 Immediate和Command窗口 96424.1.3 任务列表 96424.2 跟踪 96524.2.1 System.Diagnostics.Trace 和ASP.NET的Page.Trace 96624.2.2 页面级的跟踪 96624.2.3 应用程序的跟踪 96624.2.4 查看跟踪数据 96624.2.5 在组件中跟踪 96924.2.6 跟踪的传送 97124.2.7 TraceListener 97124.2.8 诊断选项 97524.2.9 Web事件 97624.3 调试 97824.3.1 需要的内容 97824.3.2 IIS和ASP.NET开发服务器 97924.3.3 启动调试会话 98024.3.4 有助于调试的新工具 98324.3.5 客户端的Javascript调试 98624.3.6 SQL存储过程的调试 98824.4 异常和错误处理 98824.4.1 处理页面上的异常 98924.4.2 处理应用程序异常 98924.4.3 HTTP状态码 99024.5 小结 991第25章 文件I/O和流 99225.1 使用驱动器、目录和文件 99225.1.1 DriveInfo类 99325.1.2 Directory和DirectoryInfo类 99625.1.3 File和FileInfo类 100225.1.4 使用路径 100625.1.5 文件和目录的属性、特性和访问控制列表 101025.2 读写文件 101725.2.1 Stream类 101725.2.2 Reader和Writer类 102125.2.3 压缩流 102525.3 处理串行端口 103025.4 网络通信 103125.4.1 WebRequest和WebResponse 103225.4.2 发送邮件 103825.5 小结 1038第26章 用户控件和服务器控件 104026.1 用户控件 104026.1.1 创建用户控件 104026.1.2 与用户控件交互 104326.1.3 动态加载用户控件 104526.2 服务器控件 105026.2.1 WebControl项目的建立 105026.2.2 控件的属性 105526.2.3 控件的显示 105626.2.4 添加标记属性 105926.2.5 给HTML设置样式 106126.2.6 主题和skin 106326.2.7 添加客户端特性 106526.2.8 检测和响应浏览器功能 107326.2.9 使用ViewState 107626.2.10 引发PostBack事件 108026.2.11 处理回送数据 108326.2.12 复合控件 108526.2.13 模板化控件 108826.2.14 在设计期间创建控件 109526.3 小结 1114第27章 模块和处理程序 111527.1 处理HTTP请求 111527.1.1 IIS 5/6和ASP.NET 111527.1.2 IIS 7和ASP.NET 111627.1.3 ASP.NET请求处理 111727.2 HttpModule 111727.2.1 修改HTTP输出 111927.2.2 URL的重写 112227.2.3 IIS 通配符 112627.3 HttpHandler 112727.3.1 一般的处理程序 112727.3.2 在IIS中映射文件扩展名 113127.4 小结 1133第28章 使用业务对象 113428.1 使用ASP.NET 3.5中的业务对象 113428.1.1 创建预编译的.NET业务对象 113428.1.2 在ASP.NET应用程序中使用预编译的业务对象 113728.2 COM交互操作性：在.NET 中使用COM 113828.2.1 运行库可调用包装(RCW) 113828.2.2 在ASP.NET代码中使用COM对象 113928.2.3 错误处理 114428.2.4 用.NET应用程序部署COM组件 114628.3 在非托管代码中使用.NET 114828.3.1 COM可调用包装(CCW) 114828.3.2 在COM对象中使用.NET组件 115028.3.3 早期绑定和后期绑定 115328.3.4 错误处

理 115428.3.5 用COM应用程序部署.NET组件 115628.4 小结 1157第29章 ADO.NET EntityFramework 115829.1 能否使用同一种语言 115829.1.1 概念层和逻辑层 116029.1.2 层之间的映射 116029.2 创建第一个实体数据模型 116029.2.1 使用EDM向导 116129.2.2 使用ADO.NET实体设计器 116329.2.3 使用EDM建立ASP.NET页面 116429.3 理解关系 116729.3.1 一对一和一对多关系 116729.3.2 多对一和多对多关系 117029.4 EDM中的继承 117329.5 使用存储过程 117729.6 EntityDataSource控件 118029.6.1 创建基本页面 118029.6.2 配置数据源控件 118129.7 小结 1185第30章 ASP.NET Dynamic Data 118630.1 使用Visual Studio 2008创建基本应用程序 118630.1.1 默认应用程序中创建的核心文件 118730.1.2 Dynamic Data应用程序 118830.1.3 连接数据库 119330.1.4 在Global.asax文件中注册数据模型 119630.1.5 样式和布局 119830.1.6 应用程序的结果 119830.2 使用动态数据路由 120130.3 控制显示方式 120530.4 小结 1207第31章 建立和使用服务 120831.1 不同系统之间的通信 120831.2 建立一个简单的XML Web服务 121031.2.1 WebService页面指令 121131.2.2 查看Web服务的基类文件 121131.2.3 把定制的数据集显示为SOAP 121231.2.4 XML Web服务接口 121531.3 使用简单的XML Web服务 121731.3.1 添加Web引用 121831.3.2 在客户应用程序中调用Web服务 121931.4 Web服务的传输协议 122131.4.1 HTTP-GET 122331.4.2 HTTP-POST 122531.4.3 SOAP 122631.5 Web方法的重载 122631.6 高速缓存Web服务的响应 122931.7 SOAP标题 123031.7.1 用SOAP标题建立Web服务 123031.7.2 通过SOAP标题使用Web服务 123231.7.3 使用SOAP 1.2请求Web服务 123431.8 异步使用Web服务 123631.9 Windows CommunicationFoundation 123831.9.1 迈向SOA的一大步 123931.9.2 WCF概述 124031.9.3 建立WCF服务 124031.10 建立WCF使用程序 124731.10.1 添加服务引用 124731.10.2 使用数据合同 125031.10.3 名称空间 125531.11 小结 1255第32章 ADO.NET Data Services 125632.1 创建第一个服务 125732.1.1 添加Entity Data Model 125732.1.2 创建服务 125932.2 查询接口 126332.2.1 读取数据表 126432.2.2 从表中读取特定的项 126532.2.3 关系 126832.2.4 展开关系 127032.2.5 结果集中的排序 127332.2.6 在结果集中移动 127432.2.7 过滤内容 127532.3 在ASP.NET中使用ADO.NET Data Services 127632.4 小结 1280第33章 本地化 128133.1 文化和地区 128133.1.1 理解文化类型 128233.1.2 ASP.NET线程 128233.1.3 服务器端的文化声明 128533.1.4 客户端的文化声明 128633.1.5 翻译值和行为 128733.2 ASP.NET 3.5资源文件 129433.2.1 使用本地资源 129533.2.2 使用全局资源 130033.3 资源编辑器 130233.4 小结 1303第34章 配置 130434.1 配置概述 130434.1.1 服务器配置文件 130534.1.2 应用程序配置文件 130834.1.3 配置设置的应用 130834.1.4 检测配置文件的变化 130934.1.5 配置文件的格式 130934.2 公共配置设置 131034.2.1 连接字符串 131034.2.2 配置会话状态 131134.2.3 编译配置 131534.2.4 浏览器的功能 131734.2.5 定制错误 131934.2.6 身份验证 132034.2.7 匿名身份 132234.2.8 授权 132334.2.9 锁定配置设置 132534.2.10 ASP.NET页面配置 132634.2.11 include文件 132734.2.12 配置ASP.NET运行时设置 132834.2.13 配置ASP.NET辅助进程 133034.2.14 存储与应用程序相关的设置 133234.2.15 配置文件的编程 133334.2.16 保护配置设置 133934.2.17 编辑配置文件 134334.3 创建定制段 134434.3.1 使用NameValueFileSectionHandler对象 134534.3.2 使用DictionarySectionHandler对象 134734.3.3 使用SingleTagSectionHandler对象 134834.3.4 使用定制的配置处理程序 134934.4 小结 1351第35章 测试 135235.1 使用事件日志 135235.1.1 读取事件日志 135335.1.2 写入事件日志 135535.2 使用性能计数器 135735.2.1 通过管理工具查看性能计数器 135735.2.2 建立基于浏览器的管理工具 135935.3 应用程序的跟踪 136535.4 健康监控 136535.4.1 健康监控系统的提供程序模型 136635.4.2 健康监控的配置 136735.4.3 通过配置写入事件：运行示例 137435.4.4 把事件路由到SQL Server中 137535.4.5 缓存Web事件 137735.4.6 通过电子邮件发送Web事件 137935.5 小结 1384第36章 管理 138536.1 ASP.NET Web站点管理工具 138536.1.1 Home选项卡 138636.1.2 Security选项卡 138636.1.3 Application选项卡 139636.1.4 Provider选项卡 139936.2 在Vista的IIS上配置ASP.NET 140036.2.1 .NET Compilation 140136.2.2 .NET Globalization 140236.2.3 .NET Profile 140236.2.4 .NET Roles 140236.2.5 .NET Trust Levels 140336.2.6 .NET Users 140336.2.7 Application Settings 140536.2.8 Connection Strings 140536.2.9 Pages and Controls 140636.2.10 Providers 140636.2.11 Session State 140736.2.12 SMTP E-mail 140836.3 小结 1408第37章 打包和部署ASP.NET应用程序 140937.1 部署各个部分 140937.2 部署之前的准备 141037.3 部署Web应用程序的方法 141137.3.1 使用XCOPY 141137.3.2 使用VS Copy Web Site选项 141337.3.3 部署预编译的Web应用程序 141537.3.4 建立安装程序 141737.4 安装程序的选项 142437.4.1 使用部署项目的属性 142437.4.2 File System编辑器 142737.4.3 Registry编辑器 143037.4.4 File Types编辑器 143137.4.5 User Interface编辑器 143137.4.6 Custom Actions编辑器 143337.4.7 Launch Conditions编辑器

第一图书网, tushu007.com

143437.5 小结 1434附录A 迁移ASP.NET旧项目 1435A.1 迁移过程并不难 1435A.1.1 同时运行Framework的多个版本 1435A.1.2 升级ASP.NET应用程序 1435A.2 混合版本——窗体验证 1437A.3 升级——ASP.NET保留的文件夹 1438A.4 ASP.NET 3.5页面是XHTML 1439A.5 在ASP.NET 3.5中没有硬编码的.js文件 1440A.6 在Visual Studio 2008中转换ASP.NET 1.x应用程序 1441A.7 从ASP.NET 2.0迁移到3.5 1445附录B ASP.NET终极工具 1448B.1 调试更简单 1448B.1.1 Firebug 1448B.1.2 YSlow 1449B.1.3 IE WebDeveloper和FirefoxWebDeveloper工具栏 1450B.1.4 Aptana Studio——JavaScript IDE 1452B.1.5 评测器：dotTrace或ANTS 1452B.2 参考资料 1453B.2.1 PositionlsEverything.net、QuirksMode.org和HTMLDog.com 1454B.2.2 Visibone 1454B.2.3 www.asp.net 1454B.3 整理代码 1455B.3.1 来自Devexpress的Refactor! for ASP.NET 1455B.3.2 Code Style Enforcer 1455B.3.3 Packer for .NET——JavaScript最小化器 1456B.4 Visual Studio 插件 1457B.4.1 用于Visual Studio 的ASPX Edit Helper插件 1458B.4.2 Power Toys Pack Installer 1458B.5 扩展ASP.NET 1459B.5.1 ASP.NET AJAX控件工具集 1460B.5.2 Atif Aziz的ELMAH——错误记录模块和处理程序 1460B.5.3 Helicon的ISAPI_Rewrite 1461B.6 一般目的的开发工具 1462B.6.1 Telerik的在线代码转换器 1463B.6.2 WinMerge和Differencing工具 1463B.6.3 Reflector 1464B.6.4 CR_Documentor 1465B.6.5 Process Explorer 1466B.7 小结 1467附录C Silverlight 2和ASP.NET 1468C.1 入门 1468C.2 Silverlight控件 1471C.2.1 Silverlight控件 1472C.2.2 MediaPlayer控件 1478C.3 Silverlight和JavaScript 1479C.3.1 JavaScript API 1479C.3.2 HTML Bridge 1481C.4 小结 1486附录D ASP.NET大于ASP.NET MVC 1487D.1 抽象：Web窗体能做好的工作 1487D.1.1 基本的Web窗体应用程序 1488D.1.2 事件的重要性 1491D.2 漏洞：Web窗体不适合哪些场合 1492D.2.1 ViewState 1493D.2.2 控制尖括号 1493D.2.3 客户ID 1494D.2.4 测试 1494D.3 返回基础：ASP.NET MVC相信… 1494D.3.1 管弦乐作曲和编程 1495D.3.2 Separation of Concerns的含义 1495D.3.3 可维护性 1495D.4 考虑可测试性 1496D.5 对ASP.NET MVC的一般反应 1497D.5.1 这看起来像是1999年的传统ASP 1497D.5.2 谁移动了我的[asp：Cheeserunat=“server”] 1497D.5.3 另一个Web框架 1497D.6 为什么“(ASP.NET] ASP.NET MVC) ==True” 1498D.6.1 在配置之上的约定 1500D.6.2 第三个请求 1503D.6.3 请求的生命周期 1505D.7 小结 1505附录E ASP.NET在线资源 1506E.1 作者的Blog 1506E.2 ASP.NET有影响的Blog 1506E.3 Web站点 1506

第一图书网, tushu007.com

<<ASP.NET 3.5 SP1高级编程>>

编辑推荐

ASP．NET 3．5使Web应用程序的开发有了质的飞跃，它拥有Visual Studio 2008的强大功能，大大改进了C#2008和Visual Basic 2008语言，并集成了强大的LINQ新技术以及用户熟知且喜欢的ASP．NET 2．0 Framework。

《ASP.NET 3.5 SP1高级编程(第6版)》概述了ASP．NET 3．5 SP1的诸多功能，提供了大量的C#和VB范例，论述了ADO．NET Entity Framework、ADO．NET Dynamic Data和ADO．NET Data Services等新内容。

ASP．NET 3．5提供了许多服务器控件，如ListView和非常灵活的GridView，它还改进了AJAX技术，合并了Visual Studio 2008中的JavaScript调试技术。

在《ASP.NET 3.5 SP1高级编程(第6版)》中，世界一流的作者团队介绍了AJAX工具箱中的一些新控件、回退按钮历史功能和脚本合并功能，还阐述了WCF的新功能，包括对DataContractSerializer的修改。

《ASP.NET 3.5 SP1高级编程(第6版)》主要内容 ◆服务器控件的概念以及它在ASP．NET开发中的重要作用 ◆使用主页面功能创建模板化的ASP．NET页面 ◆使用来自企业数据库(包括SQLServer)的数据 ◆调试、打包、部署ASP．NET应用程序，监控其健康状况和性能，处理错误 ◆使用LINQ快速而有条理地检索、更新、删除数据，并将LINQ与现有的技术作比较 ◆针对全世界的用户以多种语言本地化网站 ◆向ASP．NET应用程序添加AJAX功能 ◆新数据访问方法的优点 ◆使用和扩展提供程序模型，以访问数据存储、进程等 ◆Scott Hanselman的ASP．NET终极开发工具集中的免费工具 《ASP.NET 3.5 SP1高级编程(第6版)》读者对象

《ASP.NET 3.5 SP1高级编程(第6版)》适用于希望学习ASP．NET 3．5 SP1的Visual Studio 2008、C#3．0(2008)或Visual Basic 9(2008)程序员和开发人员。

第一图书网, tushu007.com

<<ASP.NET 3.5 SP1高级编程>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com