

“My book will arm you with the techniques and knowledge you need to build attractive, fully featured ASP.NET user interfaces that are intuitive to end users no matter what client they are using.”



Professional ASP.NET Web Forms Techniques

ASP.NET Web Forms 高级编程

Alex Homer 著
窦蓉金冶 等译



清华大学出版社

ASP.NET Web Forms 高级编程

Alex Homer 著

窦蓉 金冶 等译

清华大学出版社

北 京

北京市版权局著作权合同登记号：01-2002-5390

内 容 简 介

本书以虚拟的 Wrox Car Company 站点以及一系列相关网页为示例，详细讲述如何利用各种 ASP.NET Web Forms 技术和相关的 .NET Framework 技术，创建一个可方便选择、配置和订购汽车的站点。与此同时，本书还考虑到各种复杂的情况，实现了与多种浏览器的兼容，并为存在某些功能缺陷的用户提供了解决方案，从而使您的 Web 应用程序尽善尽美。

本书适合于对 ASP.NET Web Forms 和 .NET Framework 有一定了解的读者。

Alex Homer : Professional ASP.NET Web Forms Techniques

EISBN: 1-86100-786-8

Copyright© 2002 by Wrox Press Ltd.

Authorized translation from the English language edition published by Wrox Press Ltd.

All rights reserved. For sale in the People's Republic of China only.

Chinese simplified language edition published by Tsinghua University Press.

本书中文简体字版由英国乐思出版公司授权清华大学出版社出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目(CIP)数据

ASP.NET Web Forms 高级编程/(美)霍姆著；窦蓉等译. —北京：清华大学出版社，2003

书名原文：Professional ASP.NET Web Forms Techniques

ISBN 7-302-06635-3

I. A... II. ①霍...②窦... III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字(2003)第 036164 号

出 版 者：清华大学出版社(北京清华大学学研大厦，邮编 100084)

[http:// www. tup. com. cn](http://www.tup.com.cn)

责任编辑：于平

印 刷 者：北京市清华园胶印厂

发 行 者：新华书店总店北京发行所

开 本：787×1092 1/16 印张：31.75 字数：812 千字

版 次：2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷

书 号：ISBN 7-302-06635-3/TP·4963

印 数：0001~5000

定 价：58.00 元

前 言

利用 ASP.NET 可以方便地为 Web 应用程序创建交互式的、直观的用户界面和有吸引力的、激动人心的网页。通过将基于服务器的回送体系结构和 .NET Framework 中提供的大量服务器控件相结合,开发人员可以像利用 VB、Delphi 和 C++ 等语言开发传统可执行程序的事件驱动方式一样,快速地为网站和 Web 应用程序构建基于浏览器的用户界面。

ASP.NET 是一种全新的方法,它与以前版本的 ASP 截然不同。很难全面地掌握其基本原理,以及基于服务器的事件的工作方式(根据用户在页面中的操作来生成相应的事件)。然而,创建 ASP.NET 页面,尤其是使用那些可用的或处于开发过程中的工具来创建页面,将比使用传统的 ASP 方法更加简单,并且可以减少错误的发生。

然而,在用户交互过程中完全使用基于服务器的方案也有其自身的缺点。在一个长时间的等待过程中,或者在网络低速连接时,每当页面的内容需要改变时,就需要连接服务器,这将会干扰应用程序的工作,或者降低网站预期的响应灵敏度。

ASP.NET 的设计思路是,生成一种基于服务器的机制,通过向服务器回送信息来生成和更新页面;但是仍然有部分服务器控件(如常见的 Validation(验证)控件)通过 JScript 脚本实现客户端的交互性能,并将其发送到客户端运行。开发人员完全可以将该功能添加到其他控件中,事实上,这种功能目前已经成为大多数网站和 Web 应用程序的一种标准。

本书主要内容

根据本书的书名和前言,读者完全可以推测出本书的目标是探讨如何开发由 ASP.NET 驱动的用户界面。具体而言,将讨论和研究主要用于创建网站和网页的各种技巧和方法。这些方法具有如下特性:

- 直观、易于使用
- 富有吸引力,功能全面
- 在合适的地方提供了交互性能
- 基于真实场景
- 完成一组预定义的设计要求
- 具有娱乐性,令人难忘,令用户满意

我们所创建的页面大部分是一个完整应用程序中的一部分内容(对客户购买模式进行了一些优化),这些应用程序的确构成了网站或 Web 应用程序的基础。现在,Web 应用程序已经成为 WWW(World Wide Web,万维网)的一个基本组成成分。本书将创建一个 Web 站点,在该站点中,用户可以从著名的 Wrox 汽车公司(Wrox Car Company)选择、定制和购买一辆新车。

为了创建该网站,首先将着眼于需求分析和设计的步骤(尽管不是很详尽),然后,设计和



勾勒网站的概貌。然而，尽管这是一种自然的开始创建站点的方式，但这一步通常被简化，甚至被省略，从而使得网站的开发工作向着一个偶然的、不可预见的方向发展。当然，这样做并非完全错误，但这样的确增加了开发和维护的难度。即使采用最简单的“启动计划”也可以减少开发和维护周期，同时还可以避免许多由“没有计划”而引发的常见故障。

一旦完成了基本设计，接下来就要学习组成一个完整网站所需的各种资源的开发。当然，网站中经常会出现重复，因此，为了使本书尽可能地发挥其作用，我们冒昧地尝试并涵盖尽可能多的有用主题。本书将介绍以下内容：

- 一个简化整个网站浏览的中央主页(Home 页面)
- 显示后台数据库所提供的可用产品的数据
- 由服务器端代码或客户端代码(或两者同时)驱动的交互内容
- 在运行时创建图像，以及可以为站点赋予生命的动画
- 识别用户并提供可选的个性化的“登录”特性
- 在线订单工具，可以提供订单跟踪、利用电子邮件发送状态报告等功能
- 目前可供使用或正在开发的一些 ASP.NET 开发工具

此外，我们并没有过多地关心后台数据处理的详细细节，而仅仅关心网站运行所需的内容。在清华大学出版社引进并出版的《ASP.NET 分布式数据应用程序高级编程》书中，讨论了后台的整个过程、n 层体系结构的实现，和组件化设计。然而，该书尽可能地解决客户端程序可能出现的各种复杂情况，以便更好地关注数据处理问题。

由于本书篇幅所限，因此本书将重点转向了用户界面，并使用非常简单的数据访问技术。当然，这并不意味着我们没有“恰当地”做这件事，而仅仅表示您没有看到所有 .NET 数据访问类的工作原理的详细讨论，或者对捕获和报告并发错误的长篇介绍。

本书中除了一些演示特定原理或展示语法的简单、抽象的小程序段外，所列出的所有 ASP.NET 代码都可以在 VB.NET 和 C# 中使用。书中所有代码几乎全部采用 VB.NET 编写，但尽量避免任何与特定语言相关的特性。因此，如果愿意采用另一种语言，任何有一定能力的开发人员都可以将这些代码转换成其他语言的代码。由于 VB.NET 和 C# 实现指定特性的方法有着显著的区别，因此，本书中也列出了 C# 版程序。此外，客户端代码采用 JavaScript/JScript 编写。

本书中的源代码可以从 <http://www.wrox.com/> 网站下载。此外，还可以在服务器上在线运行许多示例。详情请参阅 <http://www.daveandal.com/books/7868/>。

注意：

如果读者使用 ASP 2.0 或 3.0，并且已经阅读过 Wrox 的“ASP Webmaster”系列书籍，读者将发现，在某种程度上，本书在所述专题的选择和讲述专题的方式上与“ASP Webmaster”系列书籍类似。但是，本书介绍的是 ASP.NET，因此其范围和基础实现有很多不同之处。

本书读者对象

显然，ASP.NET 和相应的 .NET Framework 类库要比在 ASP 3.0 或者更早版本中所使用的技术(或者，事实上也包括 PHP 和 JSP 等其他环境)复杂得多。因此，每一本介绍 ASP.NET 的书

籍都不可能既包含关于 ASP.NET 的基础介绍或简短教程，又同时讲述更高级的内容。而后者将是本书的核心内容。

这里需要提醒您注意，从您的观点看这是一件好事，因为您并不想为已经熟悉的主题的内容浪费时间和金钱。这就意味着，如果您想成功使用本书，那么您需要熟悉 ASP.NET，并在一定程度上熟悉 .NET Framework 类。这里假设您已经安装，并正在使用 ASP.NET 创建页面，当然也可能使用文本编辑器、Visual Studio .NET，或其他开发工具来创建页面。

此外，还假设您对使用数据库驱动网站有一定的了解。我们将解释书中使用的数据访问代码，代码的工作原理，并讨论在适当时机可以采用的替代方法。但是，本书中并不讲述建立和管理数据库的基本技巧。

注意事项

- 本书不是针对任何特定技术领域的资料性参考书
- 本书不是针对任何技术的学术论文
- 本书并不特别讲述任何与 ASP.NET 相关的硬件性能或软件配置
- 本书并不特别针对 SQL 编程、服务器优化，或者数据访问理论
- 本书的读者对象并不是系统或数据库管理员，而是针对开发人员

主要特性

- 本书提供了可达到使用目标但不会提高复杂性的实用方法
- 本书是有关 ASP.NET 的众多有用、功能强大的面向界面特性的指南
- 本书探讨了如何设计并构建交互式网站
- 本书简要介绍了众多的辅助技术，如绘图和电子邮件功能
- 本书展示了如何利用 ASP.NET 方便快捷地创建大型网站
- 本书资料丰富、实用，且行文风趣

学习本书的条件

当您的经验与知识水平满足了上一节的基本要求后，您只需再具备以下条件：

- 具备一个开发环境。至少必须具备您喜欢的文本编辑器，可以是 Visual Studio .NET、Microsoft ASP.NET Web Matrix，或者其他任何合适的第三方开发工具；
- 在可以访问的服务器端已经安装和配置了 IIS(Internet Information Server, Internet 信息服务器)，和 .NET Framework。如果选择使用了 Matrix，那么还可以利用它内置的 Web 服务器。我们还推荐您安装包含各种参考资料的 .NET SDK 的完全版本，这样即使没有使用 Visual Studio .NET，也可以方便地查询相关的条目。
- 安装配置数据库，如 Microsoft SQL Server 或 MSDE，当然还可以使用其他类型的 SQL 数据库系统——甚至可使用桌面数据库(如 Microsoft Access)——只需确保已提供可以通过 ASP.NET 访问的 .NET Data Provider。

为了开发和试验，可以在同一台机器上运行数据库、IIS、.NET，以及用户选择的开发环境。这样可以简化调试过程。



用户支持和反馈

我们一贯重视读者的意见，并想知道每位读者对本书的看法，包括读者喜欢和不喜欢的内容，以及读者希望我们下一次完善的地方。您可以通过发送电子邮件(地址为 feedback@wrox.com)来向我们反馈意见。请确保在反馈信息中提到本书的 ISBN 和书名。

源代码和更新

本书所有代码都可以从 Wrox 公司网站(<http://www.wrox.com>)下载。当访问该站点时，通过 Search 工具或书名列表，可以方便地定位需要的书目。然后，单击本书详细信息页面中的 Download Code 链接，就可以下载所有的源代码。

从我们的站点上下载的可用文件都是使用 WinZip 压缩过的文档。把附件保存到本地硬盘上的文件夹中后，需要使用一个解压缩程序(例如 WinZip 或 PKUnzip)来解压缩文件。在解压缩文件时，通常将代码解压缩到每一章所在的文件夹中。在解压缩的过程中，应确保解压缩程序已经选中 Extract to(或对等选项)选项列表下的 Use folder names。

也可以在作者自己的网站联机运行本书中的代码，该站点的地址为：
<http://www.alanddave.com/books/7868>。

勘误表

我们已经尽最大努力确保本书中的文本和代码没有错误，但是错误仍然在所难免。如果您发现本书存在错误，例如拼写错误或不正确的代码段，请反馈信息给我们，我们将不胜感激。勘误表的发送可以节约其他读者学习本书的时间，而且能够帮助我们提供更高质量的信息。请将您的反馈信息以电子邮件的形式发送到 support@wrox.com，它们将被检查，如果正确，将被粘贴到本书的勘误页面上，或者在本书的后续版本中使用。

要在我们的站点上找到勘误表，请访问 <http://www.wrox.com/>，并通过 Search 工具或者书名列表轻松定位本书页面。然后单击 Book Errata 链接即可，该链接位于本书的详细页面中。在这个页面中，您可以看到所有已经由编辑检查并提交的勘误内容。通过单击 Submit Errata 链接，您也可以通知我们您已经发现的勘误内容。

技术支持

如果您希望直接询问本书中的问题，可以发送电子邮件到 support@wrox.com。一封典型的电子邮件应包括下面的内容：

- 在主题栏中必须有本书的书名、ISBN 的后 4 位数字(本书的对应号码为：7868)，和问题所在的页码。
- 正文部分应包括读者的姓名、联系方式，和问题。

我们将不返回无用邮件，因为仅仅需要有用的详细资料，以便可节约您和我们的时间。当您发送一条电子邮件消息时，它将经过下面一系列的支持：

- 客户支持——首先，您的信息将被递送到客户支持人员手中，并由他们阅读。对于一些被频繁提到的问题将被归档，并将立即回答有关本书或者 Web 站点的任何常见问题。
- 编辑支持——接着，一些有深度的问题将被送到对本书负责的技术编辑手中，他们在程序设计语言或者特定的产品上有着丰富的经验，能够回答相关主题的详细技术问题。问题一旦得到解决，编辑会及时将勘误表发送到我们的 Web 站点上。
- 作者支持——最后，如果编辑不能回答您的问题(这种情况很少发生)，他们将请求本书的作者。我们将尽量保护作者免受干扰，以便不影响其写作。然而，我们也非常高兴转寄给他们一些特殊的问题。所有 Wrox 公司的作者都为他们的书提供技术支持。作为回应，他们将发送电子邮件给用户和编辑，进而使所有的读者受益。

注意：

Wrox 公司的支持过程仅仅对那些与我们出版的书目内容直接相关的问题提供支持，对于超出常规书目支持的问题，您可以从 <http://p2p.wrox.com/> 论坛中的公共列表中获得支持信息。

p2p.wrox.com 站点

为了便于作者和其他人讨论，特将编程人员加入到 P2P 站点的邮件列表中，而且我们独特的系统将 **programmer to programmer™**(由程序员为程序员而著)的编程理念与邮件列表、论坛、新闻组，以及所有其他服务内容(一对一的邮件支持系统除外)相联系。如果您向 P2P 发送一个问题，应该相信它一定会被登录邮件列表的 Wrox 公司作者和其他相关专家所检查到。无论您是在阅读本书，还是在开发自己的应用程序，都可以在 p2p.wrox.com 站点中找到许多对自己有所帮助的邮件列表。

按照下面的步骤可以预订一个邮件列表：

- (1) 登录 <http://p2p.wrox.com/> 站点，并从左边的菜单栏选择一个适当的类别。
- (2) 单击您希望加入的邮件列表。
- (3) 按照说明订阅并填写自己的邮件地址和密码。
- (4) 回复您收到的确认邮件。
- (5) 使用预订管理程序加入更多的邮件列表并设置自己的邮件首选项。

出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft, IBM, Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

1-17565/18

目 录

第 1 章 开始	1
1.1 为什么使用 ASP.NET	1
1.1.1 对术语的注释	2
1.1.2 ASP.NET 基础	2
1.2 设计交互式 Web 站点	18
1.2.1 站点总体设计	19
1.2.2 Home 页面	22
1.2.3 汽车型号详情(Model Details)页面	23
1.2.4 比较特性页面	28
1.2.5 订单页面	28
1.2.6 其他“外围”页面	29
1.3 小结	29
第 2 章 浏览器与 ASP.NET 服务器控件的兼容性	31
2.1 内置智能	31
2.2 测试应用程序	33
2.3 查看测试页面	35
2.3.1 服务器控件 Hyperlink	37
2.3.2 Navigator 理解 Style 属性	40
2.3.3 Calendar 控件	40
2.4 指定高层和低层客户端	42
2.5 列表控件格式设置和模板的应用	44
2.5.1 DataList 控件的输出	46
2.5.2 为 DataList 控件添加模板	48
2.5.3 更复杂的 DataGrid 控件	49
2.5.4 在 DataGrid 控件中指定列信息	51
2.6 处理缺少客户端脚本支持的情况	53
2.6.1 示例: Amaya 中的 LinkButton 控件	55
2.6.2 解决“没有脚本”的问题	56
2.6.3 ASP.NET 的 Validation 控件	58
2.7 ASP.NET 的 Panel 控件	60
2.8 与输出缓存功能相关的注意事项	63
2.8.1 通过浏览器版本缓存	64



2.8.2 通过客户端目标实现缓存	65
2.9 小结	66
第3章 基础知识	68
3.1 获得示例文件	68
3.2 Wrox Car Company 站点的客户端的技术需求	69
3.2.1 支持或需要的标准	70
3.2.2 ASP.NET 会话	71
3.2.3 客户端脚本	75
3.2.4 在 Wrox Car Company 站点中的用户和用户 ID	75
3.3 检查是否支持会话和脚本	76
3.3.1 检测客户端的功能和用户 ID	77
3.3.2 default.aspx 页面中的代码	78
3.3.3 default.aspx 页面中的 HTML 和内容	79
3.3.4 no-script.htm 页面	81
3.3.5 home.aspx 页面中的检测代码	83
3.3.6 no-sessions.htm 页面	84
3.4 跨浏览器兼容性问题	87
3.5 解决浏览器兼容性问题	89
3.6 使所有访问者可以访问站点	102
3.7 小结	104
第4章 Home 页面	106
4.1 页面整体结构	106
4.2 为页面设置样式	112
4.3 提供可访问性导航辅助功能	117
4.3.1 实现“Skip To”链接	118
4.3.2 记住那些“title”和“alt”属性	121
4.4 指向其他页面的链接	122
4.5 “新闻和特别服务”部分	125
4.5.1 “新闻”列表的数据访问代码	125
4.5.2 选择合适的 ASP.NET List 控件	129
4.5.3 “新闻”列表中的静态部分	129
4.5.4 “新闻”列表使用的 ASP.NET Repeater 控件	130
4.6 汽车型号列表	133
4.6.1 选择合适的列表控件	133
4.6.2 更改已启用脚本的客户端的行为	134
4.6.3 “Car Models”列表的数据访问代码	134
4.6.4 “Car Models”列表的 DataGrid 控件	136

4.7 小结	145
第 5 章 旗标、页脚和电子邮件	147
5.1 关于 ASP.NET 用户控件	147
5.1.1 可重用代码和内容	147
5.1.2 创建用户控件	148
5.1.3 在 ASP.NET 页面中使用用户控件	151
5.2 动画页面旗标	152
5.2.1 创建动画旗标控件	153
5.2.2 使用动画旗标控件	167
5.3 页脚部分	168
5.3.1 页脚中的动态超链接	169
5.3.2 创建页脚控件	169
5.3.3 使用页脚控件	173
5.4 在邮件列表中添加用户	173
5.4.1 提交电子邮件地址的控件	174
5.4.2 处理电子邮件地址的代码	181
5.5 小结	188
第 6 章 辅助页面	189
6.1 辅助页面的结构	189
6.1.1 总体布局	189
6.1.2 “Terms & Conditions” 和 “Contact” 页面	190
6.1.3 MainLink 用户控件	194
6.2 静态或动态内容的选择	197
6.2.1 使静态的内容动起来	197
6.2.2 使动态的内容静态	198
6.3 “Read News Item” 页面	201
6.3.1 在页面中插入动态内容	202
6.3.2 “Read News” 页面的 HTML 和代码	203
6.4 Compare Models 页面	208
6.4.1 复杂图像的访问性问题	209
6.4.2 构建 “Compare Models” 页面	213
6.4.3 测试可访问性	234
6.5 小结	237
第 7 章 创建图形和图表	238
7.1 动态生成的图形和图像	238
7.1.1 性能和高速缓存	239



7.1.2 在 Web 页面中使用动态图像	240
7.2 .NET Framework 绘图类	243
7.3 创建一个简单饼图	245
7.3.1 提供数据源	246
7.3.2 饼图示例页面	248
7.3.3 在图像中创建和使用颜色	252
7.3.4 通用饼图例程	261
7.4 汽车比较图表	268
7.4.1 为图表获得值和数据	268
7.4.2 绘制适当的图表	271
7.4.3 绘制直线图表	272
7.4.4 绘制条形图	279
7.5 小结	287
第 8 章 交互式 Web Forms	288
8.1 满足设计要求	288
8.1.1 初步设计	288
8.1.2 支持无脚本功能的客户端	294
8.1.3 支持仅文本的客户端	294
8.1.4 支持页面阅读器(Page-Reader)客户端	295
8.1.5 保存报价并关闭窗口	296
8.2 Model Details 页面纲要	297
8.2.1 打开 Model Details 页面	297
8.2.2 使用回送结构	297
8.2.3 框架操作	300
8.2.4 实现“tabbed 对话框”	300
8.2.5 处理没有客户端脚本支持的情况	302
8.2.6 保存报价和订单	305
8.3 Model Details 页面的结构	306
8.3.1 导入和注册 IE Web 控件	306
8.3.2 页面的 HTML 部分	307
8.4 小结	319
第 9 章 交互式的型号细节页面	321
9.1 用于 Model Details 页面的数据	321
9.1.1 汽车细节数据库表	321
9.1.2 用于 Model Details 页面的存储过程	323
9.1.3 金融 PMT 值表	325
9.1.4 获取和存储页面数据	332

9.2	Model Details 页面中的代码	334
9.3	“Model Details” 选项卡页面详解	340
9.3.1	颜色选择部分	340
9.3.2	引擎选择部分	345
9.3.3	“可选配件” 选择部分	348
9.3.4	“标准特性” (Details)部分	350
9.3.5	金融服务选择部分	358
9.4	小结	361
第 10 章	汽车价格和报价	363
10.1	计算汽车的价格	363
10.2	计算金融条款	371
10.2.1	ShowFinanceTerms 例程	372
10.2.2	ClearFinanceTerms 例程	379
10.3	将配置保存为报价	380
10.3.1	已保存的报价和订单表	380
10.3.2	更新事务中的数据	381
10.3.3	保存报价	382
10.4	未引用脚本的页面	390
10.4.1	避免使用 TabStrip 和 MultiPage 控件	391
10.4.2	处理缺少 AutoPostBack 的情况	393
10.4.3	替换 TreeView 控件	394
10.4.4	在缺少 setOptionButtons 函数的情况下执行管理功能	397
10.4.5	在缺少 mainWindowPage 函数的情况下执行管理功能	397
10.4.6	在缺少 Close 按钮的情况下执行管理功能	399
10.5	小结	399
第 11 章	登录和注册页面	401
11.1	Web 站点用户和身份标识	401
11.1.1	身份验证	402
11.1.2	授权	403
11.1.3	角色扮演(Impersonation)	404
11.1.4	ASP.NET 身份验证方法	405
11.2	Wrox Cars 的登录和注册过程	412
11.2.1	根文件夹 web.config 文件	413
11.2.2	“secure” 子文件夹的 web.config 文件	413
11.2.3	针对缺少 Cookie 的客户端的身份验证	414
11.2.4	逐步处理	416
11.2.5	为什么使用匿名用户 ID	418



11.2.6	为什么使用综合的登录和注册页面	419
11.3	登录和注册页面	419
11.3.1	HTML 页面声明	419
11.3.2	客户端脚本函数	428
11.3.3	ASP.NET 服务器端代码	430
11.4	小结	446
第 12 章	报价和订单	447
12.1	将列表控件与关联的数据进行绑定	447
12.1.1	绑定到单独的数据源	447
12.1.2	绑定嵌套的列表控件	448
12.1.3	绑定到层次化数据	458
12.2	构建 My Cars 页面	460
12.2.1	My Cars 页面的 HTML 代码	461
12.2.2	My Cars 页面的代码	469
12.3	小结	486

第1章 开 始

俗话说,“不打破鸡蛋,就做不出煎蛋卷”。的确如此。如果您与本书作者一样学习同一学校的烹饪技术(将鸡蛋从冰箱中取出,然后放在微波炉中),那么,可能自己就会错过将鸡蛋打碎的亲身体验。但是,有的人却不得不全心全意做这件事情。当然,除非从饲养可以直接生成煎蛋卷的小鸡开始。

但是这些体验对创建一个网站或者 Web 应用程序又有什么关系呢?正如您从本书的书名和前言中可以猜想到的,本书的主要内容是使用 ASP.NET 作为驱动力,为应用程序设计并创建界面。利用前面所说的烹饪术语,在整个 ASP.NET 应用程序的设计过程中的几乎每一个阶段,我们都需要问一个简单问题:能否不使用服务器控件而直接创建一个 ASP.NET Web 应用程序呢?

为每一个程序员提出的技术问题的答案几乎总是不变的:“它视情况而定”,但总的来说,您可以如此放心地回答:ASP.NET 页面中的很多地方都有这些聪明的小东西的影子。毕竟,它们形成了许多 ASP.NET 页面设计技术所围绕的核心内容,而决不仅仅像是(继续前面的烹饪介绍)一个玉米饼。尽管,将服务器控件与面粉或玉米薄饼相比较或许将内容扯得远了一些。

因此,根据这些大家通常会产生的主题外的疑惑,我们在本章将实际介绍的内容是什么呢?根据标题的提示,本章的目的是为了学习如何开始创建一个基于 ASP.NET 的交互式网站或一个 Web 应用程序,在该网站或 Web 应用程序中为用户提供了一个既有趣又有用的界面。具体包括以下内容:

- 为什么 ASP.NET 为创建交互式页面和网站提供了一个功能强大的环境
- 利用 ASP.NET 简化创建动态页面的基本方法
- 在页面和应用程序中可以使用的 ASP.NET 的内在特性
- 使用客户端和服务端处理的影响和理由
- 如何设计网站才能更好地利用 ASP.NET

本章将从设计方法的最基本的内容——纸和笔开始,到本章的最后,应该能够开始创建一些网页。然而,首先我们要学习的内容是:利用 ASP.NET 提供的新工具和新技术创建网页,这与采用以前的方法相比要简单和快捷。

1.1 为什么使用 ASP.NET

事实上,您在阅读本书时,可能意味着您已经是一位 ASP.NET 开发人员,您只是想了解如何能最有效地使用从其他书籍和刊物中已经学到的各种技术。另外,您也可能正想尝试一下 ASP.NET,来验证在创建高性能动态网站和应用程序时,ASP.NET 是否比目前所使用的开发环境更好。



如果您属于第一类读者,那么接下来的几节内容将在如何充分利用 ASP.NET 设计自己的网站方面提供一些指导。如果您是第二类读者,那么本章将给您如何起步的更有帮助的指导。当然,设计网站的总体方法对了解 ASP.NET 到底能实现什么、到底能如何简化实现过程还是应该有一定的帮助的。

1.1.1 对术语的注释

不久前,人们还是可以很容易地区分网站和基于 Web 的应用程序。网站主要使用静态 HTML 页面来显示信息,而基于 Web 的应用程序则是普通的应用程序,它通常是由在本地机器上运行的可执行文件转换而来,最终可以通过 Internet 执行的应用程序。

然而,现在的发展方向是任何应用程序都应该能在 Web 上运行,这就意味着大多数新应用程序应该采用可以在 HTTP 上运行的远程界面进行设计和创建。与过去创建的网站相比,许多目前创建的网站为了满足显示和收集信息的需要,都提供了动态访问后台系统的能力,从而增强了网站的交互性。

那么,如果网站和基于 Web 的应用程序都真正变成了面向后台数据存储的界面,它们是不是就相同呢?从设计和实现的角度考虑,我们是否应该不再将它们作为两种不同的系统和方法呢?

当然,还是有一些网站不是“交互式”的,他们只是为了显示信息。但是,公正地说,一些基于 Web 的应用程序界面也只是显示信息,换句话说,它们不是真正的“交互式”界面。

是鸟?是飞机?不,是 Wapplisite!

既然这样,我们不再讨论二者的区别。现在,应该不再将它们看作是两种不同的系统,而应该作为完全相同的系统来对待。重要的是它们所展现的交互级别。但是,到底应该如何称呼它们呢?在本书中,通常称作“站点”或者“应用程序”,二者都应该表示相同的内容。

这里还需要强调的一个词是“web”和“application”相结合而形成的“weblication”。但是,这个词并不能覆盖我们同样在处理网站这样一个事实。因此,我们更喜欢的一个新词是“wapplisite”,我们倾向于认为这个词是经久不衰的(在线词典的网站管理员们请注意)。如果在本书后面的内容中看到了这个单词,将其替换成麻烦一些的短语“Web application or Web site”(Web 应用程序或网站),就可以知道我们所要表达的真实含义。除此之外,您还得到了经济上的节省,毕竟,使用这个词语,可以在每页上节省更多有用空间!

1.1.2 ASP.NET 基础

在开始研究网站设计的方法之前,还是值得首先详细探讨一下 ASP.NET 为什么以及如何提供一些因素,而我们需要这些因素使任务尽可能简化和高效。答案围绕着一个基础的变化展开:ASP.NET 允许我们创建将 HTTP 作为 Web 上惟一的传输机制使用的分布式应用程序。

1. 事件驱动的应用程序

采用面向操作系统的“基于窗口”(window-based)界面运行的现代应用程序,就其工作方式而言,应该是采用事件驱动方式,而不是过程化的方式(这里,我们在术语“window-based”中使用的是小写的 w,以表示它指的是系统的窗口,而不是 Microsoft Windows)。