

“With this book, you’ll learn how to use your C# skills to develop a feature-rich web application.

We’ll adopt a practical approach, and take advantage of the powerful features integrated into ASP.NET and Visual Studio .NET.

Along the way, you’ll develop a real understanding of the challenges of web programming - so that you’re ready to face the growing market for interactive and attractive web applications.”



Beginning C# Web Applications with Visual Studio .NET

C# Web 应用程序 入门经典

(阿根廷) Daniel Cazzulino 等著
侯 彧 王晓娜 译



清华大学出版社

C# Web 应用程序入门经典

(阿根廷) Daniel Cazzulino 等著

侯 戭 王晓娜 译

清华大学出版社

北 京

内 容 简 介

.NET 的诞生打破了以前泾渭分明的编程领域之间的界限,给 Web 开发带来了极大的变化。

本书探讨了 Web 应用程序开发中所涉及的主要问题,以及与桌面应用程序开发之间的重大差异。本书将重点放在实践上,围绕一个高度交互的 Web 应用程序实例一步一步讲述了开发、调试、部署的每一个阶段。其中前两章讲述了 Web 环境和 Web 开发的基础知识,在随后的 11 章中从理论和实践两个方面讲解了 Web 应用程序的创建、测试、调试和部署等环节,如服务器控件、数据访问、数据绑定、ASP.NET 状态管理、XML 与 Web 开发、Web 应用程序中的 Web 服务、ASP.NET 身份验证、授权和安全、调试和异常处理、利用缓存改进性能以及在 .NET 中发布 Web 应用程序等。

本书适用于熟悉 C# 语言、使用 C# 语言进行过桌面应用程序开发,或使用 ASP、PHP 技术进行过 Web 开发,并且希望学习使用 .NET 和 C# 语言从事 Web 开发的程序员。

EISBN: 1-86100-732-9

Beginning C# Web Applications with Visual Studio .NET

Daniel Cazzulino et al.

Copyright©2002 by Wrox Press.

Original English language edition published by Wrox Press Ltd.

All Rights Reserved.

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。

北京市版权局著作权合同登记号 图字:01-2002-6518

图书在版编目(CIP)数据

C# Web 应用程序入门经典/(阿根廷)卡祖利诺等著;侯戇,王晓娜译.—北京:清华大学出版社,2003

书名原文:Beginning C# Web Applications with Visual Studio .NET

ISBN 7-302-06899-2

I. C… II. ①卡…②侯…③王… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 058526 号

出 版 者:清华大学出版社

地 址:北京清华大学学研大厦

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

邮 编:100084

社 总 机:010-62770175

客户服务:010-62776969

组稿编辑:曹康

文稿编辑:李阳

封面设计:康博

版式设计:康博

印 刷 者:北京鑫海金澳胶印有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:32.5 字数:831 千字

版 次:2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

书 号:ISBN 7-302-06899-2/TP·5109

印 数:1~4000

定 价:65.00 元

出版者的话

随着国际互联网的快速崛起和迅猛发展，计算机之间的互联需求越来越迫切，而目前计算机硬件设备的不兼容性严重束缚了互联网的发展，引发了新一轮的跨平台软件的开发浪潮。软件商纷纷推出新的战略规划和解决方案，Microsoft 提出的 .NET 战略就是其中的经典之作。

在经历这场浪潮的洗涤和考验过程中，全球的软件开发人员都迫切需要了解新的软件技术和开发思路。为了满足国内 IT 从业人员的需要，清华大学出版社从 Wrox 出版公司引进了若干套编程系列丛书，“入门经典”系列是其中不可或缺的入门之作。作为世界著名的编程技术图书的出版公司，Wrox 推出的这套“入门经典”系列丛书主要面向编程的初学者、需要了解 .NET 策略的程序员，以及需要迅速掌握多门编程工具的程序员。该丛书依旧秉承了 Wrox 公司“由程序员为程序员而著 (Programmer to Programmer)”的创作理念，每本书均由世界顶级的编程高手执笔。他们站在资深程序员的高度，循序渐进地为初学者讲述了 .NET 的理念和构架、编程基本思想、编程语言基础、程序的控制和调试、Windows 应用程序的开发、对象编程技术、数据库访问技术、Web 程序开发和 .NET 构架应用等最新的软件开发知识，同时辅以大量操作性强的程序为示例，为读者提供了清晰的编程思路和宝贵的编程经验。

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和获取相关的技术支持。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。

前言

.NET 的发布使以前泾渭分明的编程领域之间的界限变得模糊起来,这给开发人员带来了巨大的影响。利用 .NET Framework 类库封装的众多功能,可以使某些迥然不同的任务获得通用的编程接口。

这种变化尤其令人注目的一个领域是 Web 开发。在 .NET 之前,“Microsoft 式”的 Web 应用程序编程指的都是 ASP。那时,ASP 是令人激动的新技术,容易访问——但它也是基于脚本的,效率不高,并且 IDE 是不连接的。Microsoft 在 .NET Framework 中总结了它在 ASP 中吸取的教训。现在,有了 ASP.NET (.NET Web 开发技术),我们能够利用开发 Windows 桌面应用程序时所使用的语言来创建高效、交互的 Internet 应用程序。

在 Visual Studio .NET 中,Microsoft 把这个概念又推前了一步;不仅代码看起来相似,连图形用户界面也相似。Visual C# .NET 令人熟悉的基于窗体的界面既能用于开发桌面程序,也能用于开发 Web 应用程序。如果愿意,您无需编写 HTML 代码,就可以创建一个 Web 应用程序,并且可以利用 Visual Studio .NET 为所有领域的程序员提供的所有测试和调试工具。

.NET Framework 中类库的结构使得我们不管是开发桌面应用程序,还是 Web 应用程序,都可以使用相同的方法。ASP.NET 在 .NET Framework 中其实就是一系列的类,这和 Windows Forms 类是完全相同的。从这个角度看,从桌面应用程序开发到 Web 开发不应该算是多大的一个飞跃。

但是,当我们转向 Web 开发时的确存在一些重要的不同之处需要加以考虑。我们所谈的不再是在单个机器上安装和运行的应用程序;而是把应用程序驻留在一个中央服务器上,每小时可能都有数百或数千个远程客户端访问。现在所涉及的问题和以往大不相同,需要对性能和可伸缩性予以更多的关注,这样才能保证最终用户尽可能有流畅的体验。

本书将致力于研究在 Web 开发中所涉及的种种问题,以及这些问题和我们从前在开发桌面应用程序时遇到的问题有何不同。在澄清这些原则之后,我们将着手创建应用程序;从第 3 章开始,将使用 C#、ASP.NET 和 Visual Studio .NET 创建一个功能强大、高度交互的 Web 应用程序,即 Friends Reunion。

我们把重点放在实践上,通过实践来学习。对书中的每个示例都一步一步地详加描述,对于 Friends Reunion 应用程序开发、调试和部署的每个阶段,我们都将进行讲述和解释。这个应用程序的源代码可以从 <http://www.wrox.com> 下载。

本书读者

本书主要针对对 C# 语言有一定经验的开发人员(这种经验是指实际使用过这种语言,或只是看过讲述 C# 语言的书籍)。这些开发人员可以分为两类:

- 只有少许 Web 开发的经验,或者根本没有,其 C# 的经验大部分是从桌面应用程序的环境

境中获得,并希望运用 C#的知识在.NET 中进行 Web 开发的读者。

- 利用 ASP、PHP 等技术从事过一些 Web 开发,且希望学习使用.NET 和他们喜欢的 C# 语言进行 Web 开发的读者。

本书不要求您具备 Web 环境中的编程经验,但需要您熟悉 C# 编程语言。以前熟悉 Visual Studio .NET 集成开发环境会有助于您的学习,但这不是基本要求。同样,我们不需要您具备 HTML、XML、数据库,或书中提到的其他技术的经验——虽然了解一点儿背景知识也无妨。

本书主要内容

本书前两章是介绍性的。这两章将讲述基础知识,具备了这些基础知识,我们才能从第 3 章开始创建 Friends Reunion Web 应用程序。

从第 3 章到第 13 章,每一章都将讨论用 C# 和 ASP.NET 开发 Web 应用程序的一个不同侧面。每一章中,我们从总体上和 Friends Reunion 应用程序的具体环境两个方面入手进行讨论。在这 11 章的篇幅中,将创建、测试、调试和部署一个功能强大、交互式的 Web 应用程序——对每一个步骤都会进行详细的研究。

以下是本书各章的主要内容:

- 第 1 章对 Web 环境进行介绍。讨论了 Web 应用程序和桌面应用程序之间的异同,并解释当用户用浏览器请求 Web 页面时后台所发生的操作。本章旨在使您对影响 Web 应用程序设计方式的问题有一个清晰的了解,并为书中后面的内容作好铺垫。我们还对 Web 服务器进行设置,并列举了几个简单的例子来开始我们的讨论。
- 第 2 章用 Visual Studio .NET 创建几个基本的 ASP.NET Web 窗体。通过逐步创建第一个 ASP.NET 应用程序,我们将对 Web 窗体的处理方式,以及页面的生命周期有所了解。
- 第 3 章专门讲述服务器控件。服务器控件在任何 Web 窗体中都是核心组件——在.NET 中,服务器控件的开发在动态、交互的 Web 站点的开发中居于中心地位。在 Visual Studio .NET 中,我们可以把服务器控件拖放到 Web 窗体中,其方式和在 Windows 桌面应用程序中插入 Windows Forms 控件完全相同;我们还可以用以前的方式,在窗体中添加代码,以便与服务器控件进行交互。
- 第 4 章和第 5 章集中介绍数据。大多数交互的 Web 应用程序都依赖于某种形式的数据的存在和操纵,也依赖于这些数据的存储(或者是存储在功能全面的数据库中,或者是以其他格式存储)。第 4 章中使用 ADO.NET 来访问和操纵数据。第 5 章讲述如何使用数据绑定技术以用户友好的方式在页面中显示数据;本章还将探讨在 Web 窗体中如何应用模板来改变数据绑定控件的外观。
- 第 6 章讨论应用程序、会话和状态。从本质上讲,Web 是一个无状态的介质——当用户请求 Web 页面时,Web 服务器就提交这个 Web 页面,然后就将这个用户忘记了。如果能让 Web 应用程序在用户进行多个页面请求时(如在用户浏览电子商务站点,在购物篮中添加商品时)能够识别用户,那么需要了解可以使用哪些不同的技术在不同页面中,或在一次会话中,或在一个应用程序中保持状态。

- 第 7 章和第 8 章集中讲述 XML——在 Internet 连接日益普及的今天，XML 已经成为一个重要话题。我们在第 7 章将了解标记的概念，以及标记和数据驱动的应用程序的广泛联系，还将用 XML 模式创建自己的 XML 语言。第 8 章探讨如何利用 XML 模式在应用程序中添加数据传输功能，这利用了 XML 的一个特性：对于跨 Internet 的传输，XML 可以说是一个完美的工具。
- 第 9 章将暂时离开 Web 站点的话题，介绍另一种不同的 Web 应用程序——Web 服务。Web 服务使我们能够把应用程序的功能提供给其他客户应用程序——这些客户应用程序使用标准和协议通过 Internet 请求我们的 Web 服务。这也意味着我们也可以在代码中利用别人的 Web 服务，就像利用我们本地系统的组件一样——即使只能通过 Internet 访问它们。本章将讨论如何创建 Web 服务，以及如何利用其他开发人员创建的 Web 服务。
- 第 10 章介绍 ASP.NET 的身份验证、授权和安全性。应用程序需要安全性是因为需要限制某个用户(或应用程序)对特定资源的访问，或它们执行某些操作的能力。例如，我们可能想在 Web 应用程序中包括管理工具，并且除了授权的用户外，我们不希望其他人访问这些管理页面。本章介绍了 ASP.NET 提供的用来对应用程序的用户进行身份验证和授权的工具。
- 第 11 章研究两个不同但彼此相关的话题：调试和异常处理。在我们理解了可能会出现各种不同的程序故障后，调试工作也会变得非常轻松，有了 Visual Studio .NET 和 .NET Framework 提供的众多调试工具和技术，调试工作会更轻松。所以本章的前半部分研究这个话题。后半部分使用 .NET 的异常处理机制来处理某些可能会在运行时发生的错误，并防止应用程序因此而崩溃。
- 第 12 章的两个话题也是各不相同，但彼此相关：性能和缓存。我们将理解什么是“高性能”，将讲述几个分析应用程序在现实的环境中性能表现的技术。我们对应用程序进行压力测试，来了解它的性能如何，在这个过程中将分析可能产生瓶颈的地方，以及如何消除这些瓶颈。接下来讲解和缓存有关的问题，并采用不同的缓存技术来减少应用程序的处理工作，并因此优化服务器资源的使用。
- 最后，第 13 章，学习如何部署应用程序。Visual Studio .NET 提供了几个简单易用的工具，可以用这些工具来创建自己的部署向导。本章演示了如何准备部署应用程序的工作(Web 站点、数据库等)，我们把所有的工作都封装到了一个很容易使用的安装向导中。

本书有两个附录。附录 A 概述了 Friends Reunion Web 应用程序的结构和功能。附录 B 包含 IIS 5.x Web 服务器和 MSDE 数据引擎的详细设置信息，还介绍了 Microsoft 新推出的 Web 服务器软件 IIS 6.0。

使用本书的前提条件

下面列出了运行本书代码所需的系统配置：

- 合适的操作系统。可以是 Windows 2000 Professional、Server 或 Advanced Server Edition(截止到本书为止，最新的服务包是 SP2)，或 Windows XP Professional Edition。

- Internet Information Server (IIS) 5.0 或 5.1。上述的操作系统都打包了这个服务器。在本书写作时还没有发布 IIS 6.0，但它也可以用作 Web 服务器软件。
- .NET Framework SDK。
- Visual Studio .NET (或 Visual C# .NET) Standard Edition 或更高版本。
- SQL Server Desktop Engine(也称为 MSDE)，或 Microsoft SQL Server。

提示：

.NET Framework SDK、Visual Studio .NET 和 Visual C# .NET 中都提供了 MSDE。安装 MSDE 只需执行 InstMSDE.exe 文件(位于\Program Files\Microsoft.NET\FrameworkSDK\Samples\Setup\MSDE 文件夹下)。MSDE 提供的管理工具和 SQL Server 提供的管理工具不同，但有几个命令行实用程序在文件 \Program Files\Microsoft.NET\FrameworkSDK\Samples\Setup\html\ConfigDetails.htm 中进行了详细描述。另外要注意，Visual Studio .NET 提供了一个很有用的接口，通过该接口能访问任何 SQL Server 或 MSDE 数据库。

注意，Windows XP Home Edition 没有提供 IIS，也不能运行 IIS。如果在此操作系统上进行 ASP.NET Web 开发，可以考虑使用 ASP.NET Web Matrix 工具，该工具能从 <http://www.asp.net> 免费下载。它提供了有限的 ASP.NET Web 服务器功能，但使用该版本，您将不能在 Visual Studio .NET 中运行 Web 项目。

本书假设您至少熟悉在 Visual C# .NET 环境中的桌面应用程序编程。在学习本书的过程中，我们将详细讲述其他技术(如 ASP.NET、ADO.NET、XML 和 SQL)，但以前对这些技术有所了解将会对您有所帮助。

用户支持

我们一贯重视您的意见，并想知道您对本书的看法，包括您喜欢和不喜欢的内容，以及您希望我们在下一次完善的地方。您可以通过发送电子邮件(地址为 feedback@wrox.com)向我们反馈意见。请确保在反馈信息中提到本书的书名。

如何下载本书的示例代码

当您访问 Wrox 公司站点(地址为 <http://www.wrox.com/>)时，通过 Search 工具或书名列表，可以方便地定位到需要的书目。然后，单击 Code 列中的 Download 超链接，或者单击本书的详情页面上的 Download Code 超链接，就可以下载相应的示例代码。

当您下载本书中的代码时，将会看的带有以下 3 个选项的 Web 页面：

- 如果您是 Wrox Developer Community 的成员(即如果您已经在 ASPToday、C#Today 或者 Wroxbase 上注册)，就可以使用一贯的用户名和密码进行登录以下载代码。
- 如果不是它们的成员，则会向您询问是否愿意注册为会员，以便可以免费下载代码。此外，也可以从 Wrox Press 下载免费的文章。注册为会员后，可以得到本书升级版本和新版本的有关信息。
- 第三个选项是完全绕过注册过程，直接下载代码。

对于本书而言,不注册也能下载代码,但是,如果您愿意注册后下载代码,您的注册信息不会泄漏给第三方。关于这方面的详细条款和条件,可以通过单击下载页面上的相关链接来查看。

从我们的站点上下载的文件都是使用 WinZip 压缩过的文档。保存文件到本地磁盘上的文件夹中后,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(WinZip、PKUnzip, 其他)被设置为使用原有的文件夹名。

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。您的反馈信息将被检查,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 Advanced Search 或者书名列表轻松定位到本书页面。然后,单击 Errata 超链接即可,该链接位于本书的详细页面中的封面图解下面。

技术支持

如果您希望直接向详细了解本书的专家咨询本书中问题,可以发送电子邮件到 support@wrox.com。一个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 位数字和问题所在的页码。
- 邮件正文中应包括读者的名字、联系信息和问题。

我们将不给您发送垃圾邮件。我们需要您的详细资料,以便节约双方的时间。当您发送电子邮件信息时,要经过下面一系列支持:

- 用户支持:首先,您的信息将被递送到我们的用户支持人员手中,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。
- 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/> 论坛的公共列表中获得支持信息。

p2p.wrox.com 站点

为了与作者和其他程序员讨论，请加入 P2P 邮件列表。除了一对一的电子邮件支持系统之外，我们独特的系统还通过邮件列表、论坛和新闻组提供由程序员到程序员的支持。Wrox 的作者、编辑和其他业界的专家也包括在我们的邮件列表中。

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帮助的邮件列表。和本书特别相关的是站点的 .NET 类别中的 aspx 和 aspx_professional 列表。

按照下面的步骤可以订阅一个邮件列表：

- (1) 登录 <http://p2p.wrox.com/> 站点。
- (2) 从左边的主菜单栏中选择一个适当的类别。
- (3) 单击希望加入的邮件列表。
- (4) 按照说明订阅并填写自己的邮件地址和口令。
- (5) 回复您收到的确认邮件。
- (6) 使用订阅管理程序加入更多的邮件列表并设置自己的电子邮件参数。

目 录

第 1 章 环境和体系结构	1
1.1 Web 模式	2
1.1.1 桌面应用程序和 Web 应用程序	2
1.1.2 Web 服务器和 Web 客户端	3
1.2 为 Web 开发配置系统	6
1.3 动态 Web 应用程序	15
1.3.1 动态内容的优缺点	16
1.3.2 客户端处理和服务器端处理	16
1.3.3 保持状态	21
1.3.4 Web 应用程序体系结构	22
1.3.5 ASP.NET Web 应用程序	22
1.4 小结	24
第 2 章 .NET 环境下的 Web 开发	25
2.1 ASP.NET 简介	25
2.2 ASP.NET 的其他内容	42
2.3 小结	47
第 3 章 用户界面和服务器控件	49
3.1 服务器控件	49
3.1.1 HTML 控件	51
3.1.2 HTML 服务器控件	60
3.1.3 Web 服务器控件	65
3.1.4 验证控件	73
3.2 用户控件	79
3.3 自定义控件	89
3.4 动态内容	96
3.5 避免代码重复	99
3.6 小结	102
第 4 章 ADO.NET	103
4.1 ADO.NET 的体系结构	103
4.2 ADO.NET 在程序中的应用	107
4.2.1 将数据添加到数据库	107

4.2.2	从数据库中检索数据	112
4.2.3	更改数据库中的数据	118
4.3	DataSet 对象	126
4.3.1	Data Adapter	127
4.3.2	使用 DataSet 对象	128
4.4	小结	138
第 5 章	数据绑定	139
5.1	组件简介	139
5.2	数据绑定	144
5.2.1	绑定表达式	145
5.2.2	DataBinder 类	145
5.2.3	向 Friends Reunion 应用程序添加数据绑定	146
5.2.4	绑定到数据集	154
5.3	直观地处理数据	160
5.3.1	数据组件	160
5.3.2	类型化数据集	161
5.4	高级数据绑定	167
5.4.1	分页	170
5.4.2	自由格式的数据绑定和编辑——数据列表	172
5.4.3	向应用程序添加数据列表	172
5.5	小结	185
第 6 章	ASP.NET 状态管理	186
6.1	状态存储和作用域	186
6.2	会话状态	187
6.1.1	控制会话状态	199
6.1.2	配置会话状态	203
6.3	应用程序状态	205
6.4	视图状态	212
6.5	暂态	221
6.6	利用 cookie 保存信息	228
6.6.1	利用查询字符串传递数据	234
6.6.2	利用隐藏的窗体字段传递数据	235
6.7	小结	235
第 7 章	标记语言和 XML	236
7.1	标记语言	237
7.1.1	超文本标记语言	238

7.1.2 可扩展标记语言(XML).....	239
7.2 XML 数据交换	243
7.2.1 XML 模式和验证	243
7.2.2 标记语言、模式和验证	246
7.3 小结	260
第 8 章 XML 和 Web 开发	262
8.1 在 Visual Studio .NET 中创建 XML 文档	262
8.1.1 以可视方式创建 XML 文档	263
8.1.2 在数据视图中创建 XML 文档	266
8.2 在 .NET 中以编程方式使用 XML	267
8.2.1 读取和验证 XML	268
8.2.2 接收上传文件	274
8.2.3 从 Web 应用程序验证 XML	276
8.2.4 处理上传的 XML 数据	279
8.2.5 查询 XML 文档: XPath	285
8.2.6 动态创建 XPath 表达式	295
8.2.7 XmlDocument 和 XmlReader 的比较	297
8.3 小结	299
第 9 章 Web 应用程序中的 Web 服务	300
9.1 Web 服务概述	300
9.1.1 Web 服务和服务器/浏览器模式的关系	301
9.1.2 Visual Studio .NET 对 Web 服务的支持	302
9.2 提供 Web 服务	302
9.2.1 测试 Web 服务	306
9.2.2 复杂数据类型	309
9.3 消费 Web 服务	311
9.4 SOAP	316
9.5 Web 服务中的错误处理	318
9.6 Web 服务的效率	321
9.6.1 缓存	321
9.6.2 异步通信	322
9.6.3 改进我们的 Web 服务	323
9.7 第三方的 Web 服务	328
9.8 小结	329

第 10 章	ASP.NET 身份验证、授权和安全	330
10.1	概述	330
10.2	安全的基础结构	331
10.2.1	基本术语	331
10.2.2	处理和初始化	334
10.2.3	应用程序安全配置	337
10.3	Windows 身份验证	340
10.4	Passport 身份验证	341
10.5	Forms 身份验证	341
10.6	定制身份验证和基于角色的安全	347
10.7	小结	355
第 11 章	调试和异常处理	356
11.1	错误的不同类型	356
11.1.1	语法错误	357
11.1.2	语义错误	358
11.1.3	输入错误	359
11.2	调试 Web 应用程序	360
11.2.1	ASP.NET 的跟踪功能	360
11.2.2	.NET 中的跟踪和断言	369
11.2.3	Visual Studio .NET 调试器	373
11.3	异常	387
11.3.1	异常的不同用途	387
11.3.2	异常和结果代码的比较	387
11.3.3	异常和异常处理	388
11.3.4	未处理的异常	396
11.4	小结	400
第 12 章	缓存和性能调整	401
12.1	性能的含义	401
12.2	对应用程序进行性能测试	409
12.3	缓存	418
12.4	ASP.NET 的缓存	421
12.4.1	输出缓存	421
12.4.2	数据缓存	426
12.5	ASP.NET 的其他性能技巧	434
12.5.1	控制 ViewState	434
12.5.2	Response.Redirect 和 Server.Transfer	437
12.5.3	谨慎使用 Web 控件	438

12.5.4 使用会话状态	439
12.5.5 监控缓存 API	440
12.5.6 改进数据库访问	440
12.6 小结	441
第 13 章 在 .NET 中发布 Web 应用程序	442
13.1 .NET 出现之前的部署	442
13.2 .NET 中的部署	443
13.2.1 XCOPY 部署	443
13.2.2 部署项目	443
13.3 手工部署 Web 应用程序	444
13.4 Visual Studio .NET 中的安装项目	448
13.4.1 Web Setup 项目	449
13.4.2 相关性和输出	454
13.5 使用安装项目	456
13.6 高级功能	458
13.6.1 添加自定义文件	458
13.6.2 编辑用户界面	459
13.6.3 生成项目	463
13.6.4 自定义动作	464
13.6.5 配置设置	467
13.7 启动条件	467
13.7.1 Search Target Machine 节点	468
13.7.2 Launch Conditions 节点	468
13.8 小结	470
13.9 Web 应用程序——概述	470
附录 A Friends Reunion 应用程序	472
附录 B IIS 和 MSDE 的配置	475
B.1 IIS 的配置	475
B.1.1 配置服务器端的设置	475
B.1.2 在 IIS 中配置 ASP.NET 应用程序	480
B.2 锁定 IIS	484
B.3 ASP.NET 和 IIS	484
B.4 IIS 6 的简介	488
B.5 Microsoft SQL Server Desktop Engine	491
B.5.1 小型的 SQL Server	491
B.5.2 获得和安装 MSDE	491
B.5.3 在 Server Explorer 中使用 MSDE	494

第1章 环境和体系结构

Windows 桌面应用程序和 Web 应用程序之间存在着很多差异，其中有一个最基本的差异。这个差异是由应用程序自身和它的用户界面之间的相对位置造成的。

- 运行 Windows 桌面应用程序时，用户界面会出现在运行这个应用程序的机器屏幕上。应用程序和它的用户界面之间的消息通过机器的操作系统进行传递。这里只涉及到一台机器，不存在网络。
- 运行 Web 应用程序时，用户界面能够出现在任一台机器的浏览器中。应用程序和它的用户界面之间的消息必须通过网络进行传递。因为通常情况下，Web 应用程序和它的用户界面总是位于两台独立的机器上。

这个体系结构差异会表现在很多方面。如果您以前习惯于编写桌面应用程序，那么在您第一次接触 Web 应用程序时，会遇到很多新问题。可以肯定，Web 应用程序的最大意义在于，最终用户可以不必再和应用程序位于同一台机器上——他们甚至可以不在一个国家！但是，在技术、实践和设计方面依然存在很多需要考虑的问题。例如：

- 消息传递。由于一个正在运行的应用程序必须通过网络和它的用户界面进行通信，所以它们两者之间会有一个传递消息的途径，这就是“网络验证”。
- 处理用户界面。Web 应用程序是如何把要显示的按钮、文本、标签等告知基于浏览器的用户界面，又如何对它们进行安排和设计呢？
- 安全性。如果一个 Web 应用程序应用于一个公共网络，我们就需要防止不受欢迎的用户访问这个应用程序，或者干扰获得授权的用户。
- 多用户。Web 应用程序可以通过一个远程机器得以执行，因此(从原理上讲)它可以同时被两个或多个用户执行。
- 标识和状态。当用户首次访问时，Web 应用程序如何对其进行标识，并且当此用户返回时如何识别这个用户？

如果您正从桌面应用程序开发转向 Web 开发，那么这些只是其中的几个问题，这些问题都由一个简单的事实引起，即一个正在执行的 Web 应用程序和它的用户界面(通常情况下)是物理独立的。

本章会为您解决其中一些问题，其他问题将留待本书的其他章节予以解决。本章将特别关注 Web 环境以及 Web 应用程序的体系结构，因此需要理解应用程序和其用户界面位于不同机器的含义所在。

本章主要介绍以下内容：

- Web 的工作原理——从用户开始请求页面到用户接到响应
- Web 应用程序的含义，以及它是如何组成的

- HTTP 的目的，以及它在浏览器和 Web 应用程序服务器之间进行请求/响应这种交互作用时所起的作用
- 驻留 Web 应用程序时 Web 服务器的作用，以及组织 Web 应用程序时虚拟目录的作用
- 静态内容和动态内容之间的区别
- 客户端代码和服务器端代码如何对动态内容产生不同的影响

首先向您介绍 Web 的工作原理，以及处理 Web 页面请求的方式。

1.1 Web 模式

利用 Web 建立一个应用程序，使其运行在一个中央位置，这样一来，世界上任何角落的用户都可以访问这个程序——只需通过一个浏览器和一个 Internet 连接。从功能的角度考虑，最早的 Web 应用程序并不是真正的“应用程序”，但是它们利用了这个基本概念把文档驻留在一个单独的中央位置，并支持用户远距离访问这些文档。

随着 Internet 在全球掀起热潮，全世界的开发人员都开始创建 Web 应用程序，这些程序的设计在功能上要强大得多。Web 应用程序不再仅仅作为共享文档的中央资源；现在，它们除了仍然作为一个中央资源之外，同时还可以供我们进行交互式使用，用以购买物品、计算税款、以及发送和接收电子邮件，而孩子们则可以体验 Web 这种激动人心的交互式学习方法。

现在，所有先前只出现在桌面应用程序领域内的任务，都可以利用 Web 应用程序来执行。不用再安装任何软件——只需通过 Internet 把浏览器指向那个应用程序即可。

1.1.1 桌面应用程序和 Web 应用程序

如果您曾利用 C# 语言构建过 Windows 桌面应用程序，那么编写一个与之类似的 C# Web 应用程序应该并非难事。您可以在用户界面(UI)中设计与其相似的窗体，并使它以相同的方式响应鼠标单击和按钮按下事件，您还可以用同种方式对此应用程序进行后端处理，如图 1-1 和图 1-2 所示。

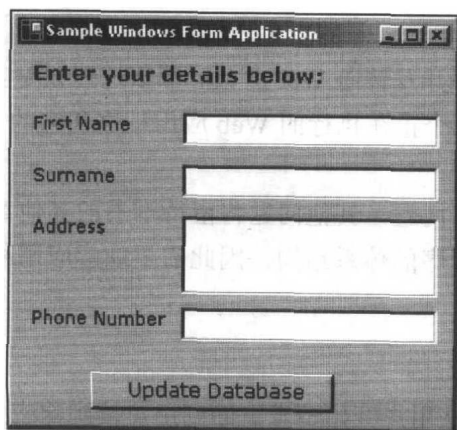


图 1-1