



Beginning Web Programming using
VB.NET & Visual Studio .NET

Web 编程入门经典

—— VB.NET 编程篇

Craig Bowes
Daniel Cazzulino
郑云 王军

等著
等译



清华大学出版社

Web 编程入门经典

—— VB.NET 编程篇

Craig Bowes

Daniel Cazzulino

等著

郑云 王军

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：01-2002-3196

内 容 简 介

本书主要介绍了在.NET中进行Web编程的各种技术。首先介绍了进行Web编程、创建Web窗体以及利用ADO.NET和Visual Studio .NET组件获取和显示数据。接下来阐述了在ASP.NET应用程序中进行调试和错误处理、XML在Web服务中的作用、安全有效地设置Web服务器。最后讲解了应用程序的性能和可伸缩性。书中给出了大量示例可以帮助读者快速掌握所介绍的知识。

本书适用于具有用Visual Basic .NET开发桌面应用程序的经验,现在想要开发Web应用程序的.NET开发人员。

EISBN: 1-86100-736-1

Craig Bowes, Daniel Cazzulino et al

Beginning Web Programming using VB.NET & Visual Studio .NET

Copyright© 2002 by Wrox Press Ltd.

Authorized translation from the 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。

图书在版编目(CIP)数据

Web编程入门经典——VB.NET编程篇/(美)鲍思等著;郑云等译. —北京:清华大学出版社,2003

书名原文:Beginning Web Programming using VB.NET & Visual Studio .NET

ISBN 7-302-06322-2

I. W... II. ①鲍...②郑... III. ①因特网—程序设计②BASIC语言—程序设计 IV.TP393.4

中国版本图书馆CIP数据核字(2003)第009045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>

责任编辑:徐燕萍

印 刷 者:北京昌平环球印刷厂

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:25 字数:640千字

版 次:2003年3月第1版 2003年3月第1次印刷

书 号:ISBN 7-302-06322-2/TP·4770

印 数:0001~4000

定 价:52.00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

在.NET 出现以前,要进行 Web 编程,必须使用脚本语言(主要是 VBScript)添加服务器端代码来创建 ASP 页面。VB6 开发人员要么在编写 ASP 开发人员所使用的 COM 组件时受到限制,要么就只能开发 Windows 软件。有了.NET 后,Visual Basic 开发人员可以很容易地将他们的技能转换到 Web 上,因为现在开发 Web 应用程序与开发 Windows 应用程序非常相似。

.NET Framework 中的类库是结构化的,因此不管我们开发的是桌面应用程序还是 Web 应用程序,所使用的方法都是一样的。ASP.NET,这项.NET Web 开发技术实际上只不过是.NET Framework 中的一些类,与 Windows 窗体中的类是完全一样的。从这个角度来说,Web 开发并不能说是一个很大的飞跃。

不过我们在开发 Web 应用程序的时候还是需要注意一些主要的区别。我们不再探讨在各个机器上运行的应用程序,而是讨论驻留在中心服务器上的应用程序,该应用程序可以每小时被成千上万的客户进行访问。这其中的问题有很多,本书就对性能和可伸缩性方面进行更多的探讨,以确保能更多地增加终端用户的经验。

解决 Web 开发中所包含的问题是学习如何开发 Web 应用程序的关键,如果我们了解了这些问题,就可以知道该如何构建自己的应用程序。

本书读者对象

如果你是这样一个开发人员:具有在 Visual Basic .NET 中开发桌面应用程序的经验,现在想要转移到开发 Web 应用程序的行列中,那么本书就适合于你使用。本书假定你已经了解 Visual Basic 语法和编码技术,并了解如何用这些技术创建带有各种功能的 Web 应用程序。我们将通过书中的各个范例来重点显示出 Web 应用程序和桌面应用程序之间的不同之处和相似之处,并讲解了编写 Web 应用程序的基础知识。你甚至不需要 HTML 知识,因为 Visual Studio .NET IDE 可以让你利用在编写 Windows 窗体应用程序时所使用的拖放操作来创建应用程序。

本书主要内容

通过对本书的学习,相信你到本书每一章结束时都能掌握在.NET 中编写 Web 应用程序时所需要的不同方面。下面是本书中各章的内容简介:

第 1 章——“Web 编程和 ASP.NET 技术”,本章主要介绍了 Web 中的一些概念,服务器端的 Web 应用程序如何进行工作,如何配合使用 ASP.NET。我们将通过一个简单的 ASP.NET 应用程序来了解一些 ASP.NET 应用程序的核心功能。我们还要安装 MSDE, Microsoft SQL Server 桌面引擎,其中包含了我们在本书后面还要用到的范例数据库。



第 2 章——“Web 窗体”，我们首先讨论一些理论上的方法，再讨论如何利用 Visual Studio .NET 来创建基本的 ASP.NET Web 窗体，从而让你自己创建第一个 ASP.NET 应用程序。最后还要看一下如何处理 Web 窗体和页面的生命周期。

第 3 章——“用户界面和服务器控件”，本章介绍了 Web 窗体的核心部分，服务器控件。我们可以像使用 Windows 窗体中的控件一样将这些控件拖放到 Web 窗体上，还可以给窗体添加代码以便与这些控件进行交互。本章还讲解了如何使用应用程序中的控件，如何创建用户控件以封装应用程序中的常见元素，以及动态生成内容。

第 4 章——“ADO.NET”，从某种意义上来说，几乎所有的应用程序都依赖保存在数据存储器中的数据，而不管该存储器是完整的数据库还是一个简单的格式化文本文件。本章介绍了从 Web 应用程序中访问数据时所需要的所有技术，并展示了将它们显示在应用程序中的方法，从而可以让我们编辑、创建和删除数据。

第 5 章——“数据绑定”，访问数据非常重要，但我们还需要能够以一种用户友好的方式将数据显示到页面上。通过将数据库中的数据与 Web 窗体上的控件进行绑定，可以很容易地显示我们所要处理的数据。本章还会讨论如何对 Web 窗体应用模板以改变数据绑定控件的外观。

第 6 章——“Web 应用程序中的调试和错误处理”，调试和错误处理是编程过程中最无味和最让人高兴不起来的事情。没有人喜欢犯错误，而改正错误也是件不让人多么愉快的事。不过，彻底的测试和调试是开发周期中不能被忽视的两个关键步骤。让我们高兴的是，Visual Studio .NET 像对 Windows 编程一样为 Web 编程提供了相当友好的工具。我们将会看到如何利用这些工具来调试应用程序，以及以用户几乎难以察觉的方式处理错误。

第 7 章——“ASP.NET 应用程序、会话和状态”，Web 是一种无状态的媒介，在默认状态下，我们没有什么机制来跟踪由哪个用户给哪个具体的用户发送了哪一条或者说是什信息。显而易见，当我们的用户在页面间导航时，对我们的应用程序来说保留部分信息是很有用的一一比如说他们是谁、他们在哪些字段中填写了信息等等——但是，为了保留这些信息，我们需要在一些全局范围内存储信息，并把它标记为属于这个用户还是另一个用户。在本章中，我们将看一下如何在应用程序级和会话级维护状态，以及如何在代码中使用全局事件。

第 8 章——“XML 和 Web 开发”，XML 现在变得越来越重要，而应用日益广泛的 Internet 连接也变得越来越常见。运行在各个机器上的应用程序可以对相关的任务进行协同工作，因此必须能以被人普遍理解的标准格式来交换信息。XML 主要就是根据这种需要而设计出来的。在本章中，我们主要了解 XML 是如何与 .NET Framework 相互配合的以及如何在 Web 应用程序中使用 XML。

第 9 章——“Web 应用程序中的 Web 服务”，对于任何希望利用 Internet 上的特定标准和协议来使用应用程序功能的人来说，Web 服务是非常强大的。这实际上意味着我们可以在代码中使用 Web 服务，就好像它是我们系统中的一个组件，即使它实际上是位于 Internet 上的。本章讲述了如何使用现有的服务、以及如何创建我们自己的 Web 服务。

第 10 章——“Web 服务器的角色”，只要我们在 Visual Studio .NET 中创建了一个新的 Web 项目，那么就会自动创建和应用许多文件和设置，包括在 Web 服务器上设置我们的 Web 应用程序。本章主要讲解 Web 服务器的工作方式，我们如何在除了默认设置之外对它进行配置。还了解了在 Web 服务器上如何对用户进行身份验证，如何根据用户的身份限制用户对应用程序

的访问权限。

第 11 章——“ASP.NET 身份验证、授权和安全”，应用程序中的安全角色与限制用户访问某些资源或者执行某些操作的需求有关。例如，Web 应用程序可以提供只能由被验证过的用户所使用的管理工具或者提供对注册用户有所限制的某些信息。本章将看一下在 ASP.NET 中用来进行身份验证和授权应用程序用户的可用工具。

第 12 章——“性能和可伸缩性”，本章的主题是性能和可伸缩性之间的合作。不过，需要注意的是，虽然我们经常将这两个主题放在一起讨论，但实际上在定义上两者之间还是有很大差异的。本章将对这两个术语的真正含义、它们之间的区别和两者之间的相互合作进行简洁的说明。另外，我们还要给你一些提示和技巧，帮助你掌握性能并设计可伸缩的 Web 应用程序。

第 13 章——“在 .NET 中发布 Web 应用程序”，在安装 .NET 客户程序、Web 服务和用户控件等时有许多技术和方法。本书最后一章中，我们将讨论 Web 应用程序和 Web 服务的体系结构，当它们出现在客户的机器上时，我们还可以看一下如何在 Visual Studio .NET 中创建和定制 Web Setup 项目。本章将帮助你很好地理解如何获得你已经为客户创建好并送到客户手中的 Web 应用程序，而且要尽可能地不打扰客户。

使用本书的必备条件

为了运行本书中的范例，你必须在 Windows 2000 或 XP 机器上安装 IIS，还要随 Visual Basic .NET 标准版或更高版本一起安装 .NET Framework。安装在 Windows 2000 和 XP 专业版上的 IIS 是个可选组件，必须在安装了操作系统之后才能进行安装。如果有必要的话，你还可以通过 Windows 控制面板中的 Add/Remove Programs applet 来安装 IIS。为了让 ASP.NET 应用程序工作正常，应该在安装 Visual Studio .NET 之前安装 IIS。

用户支持

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。请确保反馈信息提到本书的书名。

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，可以方便地定位所需要的书目。然后，单击 Code 列中的 Download 超链接，或者单击本书的详细页面中的 Download Code 超链接，就可以下载相应的范例代码。

当您单击下载本书中的代码时，将会看到带有以下 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)，就可以使用您原来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，请把信息反馈给我们，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本书页面。然后，单击 **Book 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。

E-mail 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，要求在邮件的主题中带上本书的书名和 ISBN(国际标准图书编号)的后 4 位数字。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。

我们将不返回您的无用邮件，因为我们仅仅需要有用的详细资料，以便节约您和我们的时间。当您发送电子邮件信息后，它将得到下面一系列支持：

- 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。他们备有常见问题的文件，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，一些深层次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)，他们会将问题交给本书的作者。我们一般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。他们会将答复通过电子邮件发送给用户和编辑，进而使所有的读者受益。

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，特将讨论内容加入到 P2P 站点的邮件列表中，而且我们独特的系统除了提供一对一的邮件支持系统外，还通过邮件列表、论坛、新闻组等联系方式使您实现与程序员的联系。如果您向 P2P 发送一个问题，请放心它一定会被登录邮件列表的 Wrox 公司作者和其他相关专家看到。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。
- (2) 从左边的主菜单栏选择适当的类别。
- (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密码。
- (5) 回复您收到的确认邮件。
- (6) 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参数。

本系统提供最好支持的原因

您可以加入整个邮件列表，也可以接收每周的邮件摘要。如果您没有时间和工具来接收邮件列表，可以直接查找我们的在线文档。独特的 Lyris 系统可以将一些没有用的垃圾邮件删除，并保护您的电子邮件地址不被侵扰。当存在加入和离开列表、以及任何有关列表的其他常见问题时，请将邮件发送到 listsupport@p2p.wrox.com。

目 录

第 1 章 Web 编程和 ASP.NET 技术	1
1.1 Web 编程基础	1
1.1.1 Web 应用程序简介	2
1.1.2 当今的 Web 技术	5
1.2 为 Web 开发配置系统	9
1.3 .NET Framework 中的 Web 编程	12
1.3.1 基于控件的体系结构	13
1.3.2 事件驱动的编程模型	13
1.3.3 System.Web 类	14
1.3.4 ASP.NET 和 IIS	14
1.4 Web 应用程序	14
1.4.1 Web 窗体	15
1.4.2 后台编码页	15
1.4.3 应用程序的配置	16
1.5 Microsoft SQL Server 桌面引擎	16
1.5.1 简化的 SQL Server	16
1.5.2 获取和安装 MSDE	17
1.6 小结	20
第 2 章 Web 窗体	21
2.1 Web 窗体与 Windows 窗体	21
2.2 使用 Visual Studio 开发 Web 窗体	26
2.2.1 Toolbox	26
2.2.2 Solution Explorer	27
2.3 Web 窗体的生命周期	30
2.3.1 处理 Web 窗体	30
2.3.2 页面框架初始化	30
2.3.3 用户代码初始化	31
2.3.4 验证	31
2.3.5 事件处理	31
2.3.6 显示	31
2.3.7 内务处理	31
2.3.8 在文件级对 Web 窗体进行处理	36



2.4 小结	37
第 3 章 用户界面和服务器控件	38
3.1 服务器控件	38
3.1.1 HTML 控件	39
3.1.2 HTML 服务器控件	49
3.1.3 Web 服务器控件	53
3.1.4 验证控件	61
3.2 用户控件	66
3.3 定制控件	75
3.4 动态内容	81
3.5 定制模板	83
3.6 小结	86
第 4 章 ADO.NET	87
4.1 ADO.NET	87
4.1.1 ADO.NET 的体系结构	87
4.1.2 ADO.NET 在程序中的应用	91
4.1.3 DataSet 对象	108
4.1.4 Server Explorer	118
4.2 组件	120
4.3 小结	125
第 5 章 数据绑定	126
5.1 数据绑定	126
5.1.1 绑定表达式	127
5.1.2 DataBinder 类	127
5.1.3 向应用程序中添加数据绑定功能	129
5.1.4 绑定到数据集	136
5.2 直观地处理数据	141
5.2.1 数据组件	141
5.2.2 类型化数据集	142
5.3 高级数据绑定	149
5.3.1 分页	151
5.3.2 自由的数据绑定和编辑——数据列表	154
5.3.3 向应用程序中添加数据列表	154
5.4 小结	163
第 6 章 Web 应用程序的调试和错误处理	165
6.1 测试和调试 Web 应用程序时需要考虑的一般事项	165

6.2	标准的 Visual Studio .NET 调试特性	166
6.2.1	断点	166
6.2.2	调试窗口	169
6.2.3	断言和 Output 窗口	171
6.2.4	连接到现有的进程	173
6.3	Try...Catch...Finally	175
6.4	抛出异常	177
6.5	调试客户端脚本	178
6.6	使用 ASP.NET 中的 Trace 指令	180
6.6.1	添加自己的跟踪语句	181
6.6.2	添加站点范围的跟踪信息	184
6.7	Global.asax 和站点范围错误处理	185
6.8	小结	187
第 7 章	ASP.NET 应用程序、会话和状态	188
7.1	“全局”的真正含义	188
7.2	处理应用程序级和会话级事件	189
7.3	应用程序状态	190
7.3.1	应用程序变量	190
7.3.2	应用程序锁定	192
7.3.3	应用程序状态中存储的内容	194
7.4	会话状态	194
7.4.1	会话 ID 和 Cookies	194
7.4.2	Session 变量	196
7.4.3	会话状态存储模式	198
7.4.4	建立状态服务器	198
7.5	使用 ViewState	200
7.6	选择合适的状态模型	203
7.7	小结	204
第 8 章	XML 和 Web 开发	205
8.1	XML 简介	205
8.1.1	XML 的兴起	206
8.1.2	格式良好的 XML	207
8.2	XML 的应用	207
8.2.1	有效的 XML	209
8.2.2	创建可视的 XML 文档	221
8.2.3	在数据视图中创建 XML 文档	223
8.3	在 .NET 中使用 XML	225



8.3.1	读取和验证 XML	225
8.3.2	接收上传的文件	230
8.3.3	验证来自 Web 应用程序的 XML	232
8.3.4	处理上传的 XML 数据	235
8.3.5	查询 XML 文档: XPath	240
8.3.6	动态构建 XPath 表达式	250
8.3.7	XmlDocument 和 XmlReader	252
8.4	小结	254
第 9 章	Web 应用程序中的 Web 服务	255
9.1	访问第三方 Web 服务	255
9.2	创建 XML Web 服务	261
9.2.1	选择最好的设计	261
9.2.2	Web 服务中的 Web 服务	264
9.3	使用 Web 服务	269
9.4	SOAP 消息	272
9.4.1	SOAP 消息类型	272
9.4.2	SOAP 消息格式	273
9.5	验证和捕获错误	275
9.6	处理 Web 服务错误	276
9.7	Web 服务的有效性	277
9.7.1	减少传输量	277
9.7.2	缓存结果	278
9.8	Web 服务是无状态的	279
9.9	异步通信	279
9.10	小结	282
第 10 章	Web 服务器的角色	283
10.1	Web 服务器的工作方式	283
10.2	Internet Information Services (IIS)	284
10.2.1	IIS 用作 Web 服务器	284
10.2.2	配置服务器范围的设置	290
10.2.3	在 IIS 中配置 ASP.NET 应用程序	295
10.3	向下锁定 IIS	297
10.4	ASP.NET 和 IIS	297
10.5	小结	302
第 11 章	ASP.NET 身份验证、授权和安全	303
11.1	概述	303

11.2	安全的基础结构	304
11.2.1	基本术语	304
11.2.2	身份验证和授权	304
11.2.3	基于角色的安全	306
11.2.4	负责人和身份	306
11.2.5	处理和初始化	307
11.2.6	应用程序安全配置	310
11.3	Windows 身份验证	312
11.4	Passport 身份验证	313
11.5	Forms 身份验证	314
11.6	定制身份验证和基于角色的安全	319
11.7	小结	327
第 12 章	性能和可伸缩性	328
12.1	实际应用中的可伸缩性	328
12.2	评测性能	329
12.3	WAS 工具	332
12.4	应力测试的目的	337
12.5	进行缓存以改进性能	338
12.5.1	缓存类型	338
12.5.2	输出缓存	339
12.5.3	分段缓存	343
12.5.4	数据缓存	345
12.6	小结	347
第 13 章	在 .NET 中发布 Web 应用程序	348
13.1	部署应用程序	348
13.2	多层部署	349
13.3	ASP.NET Web 服务的体系结构	349
13.4	ASP.NET Web 应用程序的体系结构	353
13.5	Visual Studio .NET Web 安装项目	355
13.6	使用安装项目	363
13.6.1	修改安装用户界面	364
13.6.2	启动条件	369
13.6.3	自定义动作	371
13.7	合并模块	375
13.8	小结	376

第1章 Web编程和ASP.NET技术

使用 Visual Basic .NET、Visual Studio .NET 和 ASP.NET 创建 Web 应用程序并不比创建 Windows 应用程序困难。的确如此, Microsoft 创建的 .NET Framework 类库, 有两个重要意义。第一, 通过将各种不同的功能放置在同一个庞大的库中, 您可以非常自主地为 Windows 开发选择编程语言, 库的功能可以被 Visual Basic、C#、C++等(这些您已经知道的)语言来使用。第二, 特别是创建了库后, 使得 Microsoft 对库中包含的功能进行了条理化, 从而程序员能够以统一的和可预知的方式来使用这些功能——这就是我们所要说明的重点。

当然, 任何事情都不是一蹴而就的(如果是的话, 本书也就不会需要介绍这么多内容)——要想成为一名高效的 Web 应用程序开发人员, 您还需要学习许多新知识。但是, 正如您在学习过程中将要看到的那样, 开发 Web 应用程序的过程与您创建 Windows 应用程序的过程基本上大同小异。

在整本书的讲述过程中, 我们将讨论您所熟悉的有关使用 Visual Basic .NET 和 Visual Studio .NET 编程的内容, 并向您展示如何将它们和 ASP.NET 结合起来创建 Web 站点。在本章中, 我们首先揭去 Web 编程世界的神秘面纱, 接着说明 Web 应用程序是如何工作的, 进而介绍 .NET Web 应用程序。具体将介绍如下所示的内容:

- 什么是 Web 应用程序以及客户机和服务器的角色
- Windows 应用程序和 Web 应用程序的不同之处
- 当前所使用的 Web 开发技术、如何与服务器通信进行页面请求, 以及如何将响应发送给客户
- .NET Web 应用程序以及 Web 窗体开发与 Windows 窗体开发的相似之处
- 什么是 ASP.NET, ASP.NET 使用的 .NET Framework 中的类以及 ASP.NET 如何与 Web 服务器一起工作
- .NET Web 应用程序的组成部分, 以及我们如何封装功能

本章的目的是让您理解大量的 Web 应用程序幕后的理论知识, 并熟悉与 .NET Web 开发有关的主要问题。我们将介绍在整本书中要使用的术语, 并讨论您已熟悉的 Web 应用程序在幕后实际上是如何工作的——以及如何使用 .NET 将它们装配为一个整体。首先让我们概括介绍一下什么是 Web 应用程序。

1.1 Web编程基础

随着全球的人们对于 Internet 的兴趣日益升温, 世界各地的开发人员都创建了 Web 站点, 通过电子邮件或对等网络应用程序来共享信息, 传递文件等等。全世界有这么多的计算机能够



访问同样的信息，并且我们能够发送电子邮件，使这些邮件在几秒钟内便能到达世界另一端，实现所有这一切的各种技术感觉上可能会使人有点畏缩。但是，现在让我们首先介绍一些您所熟悉的概念，然后以此为基础来进行构建。

1.1.1 Web 应用程序简介

我们已经几次提到了“Web 应用程序”，但是如果该术语听起来不容易理解的话，让我们一起来看一个现实生活中的 Web 应用程序例子：eBay，一个在线拍卖的 Web 站点。

注意：

该范例是为了解释当您请求一个页时，实际上所发生的情况，以及在请求页之后您所看到的可视化输出与创建该输出的代码之间的联系。一开始看起来可能很简单，但是很快我们将了解实质内容。

我们如何使用 Web 应用程序呢？首先，打开一个浏览器窗口，接着在浏览器的地址栏中输入站点地址(<http://www.eBay.com>)，然后单击 Go 按钮或者按下 Enter 键。连接到 Internet 上的速度可能有所不同，但最终您会看到如图 1-1 所示的页面。

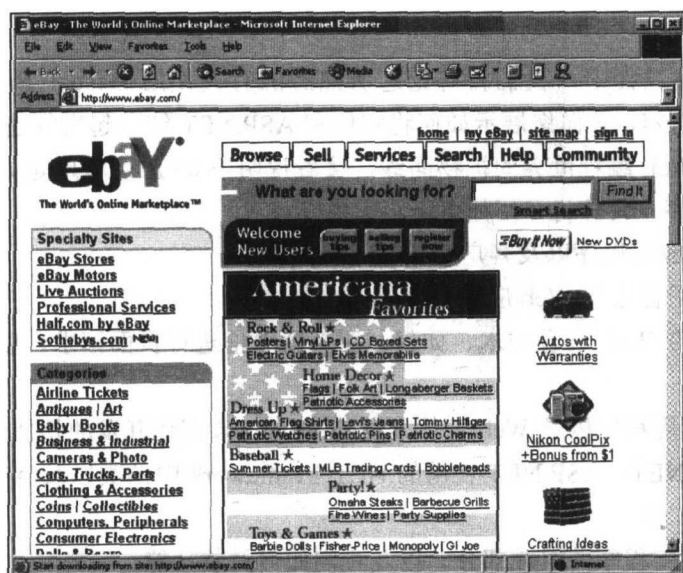


图 1-1

这是 eBay Web 站点的主页。在该页上，可以看到能够通向一些您可能感兴趣的站点区域(例如我们可以更新用户配置信息的地方，以及提供我们可能感兴趣交易的功能)的链接。单击其中的一个链接，就会被定位到不同的页面，该页面可能显示与一组特定项有关的信息，或者提供一个“登录”区域，我们可以在上面修改一些 eBay 已经存储的与我们有关的信息。

现在让我们来介绍一下这个表面上看似简单的过程是如何进行的。

1. 服务器的作用

我们可以从自己的计算机上“浏览”eBay 站点，但是很明显我们正在查阅的所有信息并非

来自于自己的计算机。大家看到的文本、图像，以及保存 eBay 站点上有关销售项的详细资料的数据库都位于服务器中(事实上，为了满足客户对站点的访问量，像 eBay 这样的站点很可能要使用到大量的服务器。换句话说就是，站点上的数据和代码都保存在“服务器上”)。

注意：

在本章中，当我们提到“服务器”的时候，是指单台 Web 服务器计算机，或者是协同工作用来共同分摊负载和提高性能的多台服务器计算机。

服务器端代码

当您在浏览器窗口浏览 Web 页时，您所看到的大部分内容——链接和图像等——是服务器传送给您的 HTML 代码的浏览器表示。如果您选择 eBay 站点上的一个类别——例如，乐器——您就会看到一系列要出售的乐器。但是如果一个项已经售出应该怎么办？站点如何让余下每个项的报价信息保持更新状态呢？答案来自于服务器端代码。

我们在编写服务器端代码时，可以指定当有人查看乐器的时候，应该搜索数据库中的相关项。这些项是由用户指定的参数来进行排序，可以从数据库中动态获得，因此我们每次请求浏览该页的时候就会得到几乎相同的输出。如果我们在一个小时之后再次请求同样的页，那么先前列出的一些项可能已经售出，还有一些新数据项可能已经被添加。但服务器端的代码保持不变，它会向数据库发出同一个查询，但是结果——返回到浏览器中的 HTML 代码——会根据我们所浏览的内容以及浏览的时间而变化。

2. 客户机的作用

通常，谈论客户机时，我们指在计算机上运行的 Web 浏览器——例如，Internet Explorer、Netscape、Mozilla、Opera 等。一般而言，这些浏览器只是用来显示发送给它们的 HTML 代码。它们不与服务器上的数据库通信，因此不能访问我们正在讨论的个人细节，或者有关特定项的细节。在大部分时间，客户机只是去做被要求做的事情。

3. Web 应用程序和 Windows 应用程序之间的区别

根据目前我们所介绍的全部内容，我们可以描绘出一个 Web 应用程序如何工作的流程图。当我们打开浏览器来查看 Web 应用程序时，我们就是在使用一个工具，该工具根本就不知道它正在试图访问的应用程序。无论是否使用了服务器端代码，或者是否发送了纯 HTML 文件，无需任何安装就可以查看 Web 应用程序。这就是 Web 的模式。

将 Web 应用程序和典型的 Windows 应用程序的工作情况相对比，在 Windows 应用程序中，绝大多数用来控制用户浏览内容的代码都必须安装在客户端，而服务器——如果有服务器的话——只不过往往用于数据存储。

现在，在不同的情况下这两种解决方案都有优点，但是我们的着重点在于理解如何能够使用相似的开发技术来创建这些似乎相差甚远的应用程序类型。区别这两种应用程序的技巧(如果您想这样称呼)就是要认识到：当我们在 Visual Studio .NET 中使用 Visual Basic .NET 开发 Windows 应用程序时，我们是在创建组成该应用程序界面的 Windows 窗体。而当我们在 Visual Studio .NET 中使用 Visual Basic .NET 开发 ASP.NET 应用程序时，我们创建的 Web 窗体详细说