



ASP.NET Distributed Data Applications

Written and tested for final release of .NET v1.0

ASP.NET 分布式 数据应用程序高级编程

Alex Homer Dave Sussman 著 魏胤红 任东胜 等译



清华大学出版社
<http://www.tup.com.cn>
<http://www.tup.tsinghua.edu.cn>



ASP.NET 分布式数据 应用程序高级编程

Alex Homer

著

Dave Sussman

魏胤红 任东胜 等译

清华大学出版社

(京)新登字 158 号

北京市版权局著作权合同登记号: 01-2002-3201

内 容 简 介

本书主要介绍如何使用 ASP.NET 创建 Internet 和 Intranet 中的分布式应用程序,书中提供了各种使用数据驱动 Web 应用程序,以及在各个层次高效利用数据的观念和方法。贯穿全书开发了一个完整的示例应用程序,根据该应用程序涉及到的各种理论和实际情况清晰地划分了本书的章节。用户可以在自己的应用程序中重用本书的实用技术和代码片断。本书的代码清单主要使用 Visual Basic .NET 编写,下载代码中同时包含 VB.NET 和 C#两种版本的代码示例。

本书特别适合于 ASP.NET 分布式应用程序开发人员阅读,对于 Web 站点构建和管理人员,也有一定的参考价值。

Alex Homer,Dave Sussman: ASP. NET Distributed Data Applications

EISBN: 1-861004-92-3

Copyright© 2001 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 分布式数据应用程序高级编程/(美)霍墨,(美)苏斯曼著;魏胤红 任东胜等译.

—北京:清华大学出版社,2002

书名原名: ASP. NET Distributed Data Applications

ISBN 7-302-05960-8

I .A... II.①霍...②苏...③魏...④任...III.主页制作—程序设计 IV.TP393.092

中国版本图书馆 CIP 数据核字(2002) 第 078771 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.tsinghua.edu.cn>

责任编辑: 陈宗斌

封面设计: 康博

版式设计: 康博

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: 37.5 字数: 959 千字

版 次: 2002 年 11 月第 1 版 2002 年 11 月第 1 次印刷

书 号: ISBN 7-302-05960-8/TP·3547

印 数: 0001~4000

定 价: 75.00 元

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。

前 言

本书的写作背景

ASP.NET 是在以前各种版本的 ASP 基础之上的巨大飞跃, 它的目标之一就是成为纯 HTML 输出, 以便获得最佳的跨浏览器的兼容性。服务器端的事件架构促进了这一方法的产生, 但是人们在最初的兴奋之中经常会忘记胖客户机的存在, 它们在许多地方处理着数据。分布式数据驱动应用程序并不是新技术, 但是 .NET 的出现使得它在可行性范围和开发的简易性方面都取得了进步。

本书将会从不同的角度研究数据管理和数据应用程序:

- 理解新的 .NET 数据管理体系, 该体系同时针对于关系数据和 XML 数据。
- 掌握 .NET 涉及的不同技术, 以及它们与现实世界需求之间的关系。
- 研究 .NET 数据管理类的应用程序架构和设计内涵。
- 设计利用新的数据管理技术的、分布式数据驱动的 ASP.NET 应用程序。
- 研究胖客户机体系, 以及使用 .NET 为它带来的影响。
- 演示利用多种不同类型的客户机设备有多么容易, 假如用户对它们都拥有熟练的使用经验的话。

本书在许多方面都是《ASP.NET 1.0 高级编程》(清华大学出版社)一书的延伸, 扩展了该书介绍的访问和管理数据的基本技术。在《ASP.NET 1.0 高级编程》中我们首先概述了数据访问技术, 然后比较深入地演示了 .NET 引入的新的数据访问类和技术。但是, 我们只能介绍一小部分应用环境, 没有顾及应用程序的设计问题, 而是关注了针对关系数据(ADO.NET)和 XML 数据访问的数据管理的可行性范围。

本书假定读者掌握了一些基本的 ASP.NET 知识和 .NET Framework 支持的常规的数据管理技术(尽管在书中我们还是会对一些重要的技术提供一些背景知识介绍)。但是, 本书的最大特点就是全书都致力于介绍如何实际创建实用的、交互式的、高效的分布式应用程序。

本书具有的特点

- 本书将会介绍创建处理数据并在 Internet(或者其他 HTTP 网络, 例如本地的 Intranet)工作的, ASP.NET 应用程序的各个方面的内容。
- 提供实用的方法来创建基于运行 ASP.NET 的服务器的任务明确的组件、Web 页面和 Web 应用程序。
- 演示 ASP.NET 能够完成的任务, 无须专门的技术、长时间的开发和复杂的第三方组件。
- 致力于 N 层结构的设计及其编码方式, 使用 SQL Server 作为数据源, 使用简单的 Web 服务器硬件。



- 贯穿全书开发了一个完整的示例应用程序，根据该应用程序涉及到的各种理论和实际情况清晰地划分了本书的章节。

- 本书的实用技术和代码片断可以在用户的应用程序中重用。

- 本书的代码清单主要使用 Visual Basic .NET 编写，下载代码中同时有两个版本的代码示例：VB.NET 版本和 C# 版本。

- 本书内容实用，叙述引人入胜……

本书没有的内容

- 本书不是学术式文章和理论著作。本书使用直截了当的语言介绍应用程序的工作原理，只在必要的时候才使用技术术语，并且避免任何离题的讨论。

- 本书不是一本参考手册，书中介绍的技术是根据应用程序的需要安排的，而不是按照字母表的顺序介绍的。

- 本书不是介绍硬件性能、服务器优化或者软件安全性的书籍。我们会尝试说明如何在设计应用程序时从用户的系统获得最佳的性能，但是我们将不会深入讨论这些领域。

- 本书不是介绍 SQL 编程、数据存储理论和复杂的后台系统的书籍。书中使用的技术能够容易地转换到任何 ASP.NET 访问的后台系统。

- 本书不是为系统管理员和数据库管理员编写的书籍。它是介绍编程的书籍，读者是熟悉 ASP.NET 的开发人员。

- 本书不是侦探小说……

本书涉及的内容

本书主要介绍如何使用 ASP.NET 创建 Internet 和 Intranet 使用的应用程序，并讨论了胖客户机带给应用程序设计和熟练的用户使用经验的可行性。人们通常对于新的 .NET 数据管理系统和页面处理模块如何适应整个分布式应用程序架构，如何改变这一架构，以及如何提供激动人心的新的机遇等问题的认识存在分歧。我们将会花费一些时间研究整体的设计问题，看看如何使用不同的方式来解决这些问题。

我们会在贯穿全书中提供各种使用数据驱动 Web 应用程序，以及各个层次高效利用数据的观念和方法。Web 应用程序通常会从把商务规则封装到组件中获益，.NET Framework 提供了创建各种类型的组件的方法。本书将会介绍如何创建和使用它们。

本书不会专门介绍 Windows 窗体或可执行应用程序，或者它们通常运行的客户机/服务器环境。但是，为了在基于 Web 的分布式环境中良好运行，客户机可以从使用 Windows 窗体或者其他客户端可执行程序获益，这些程序将会访问与基于 ASP.NET 的应用程序连接的服务器。本书也将会介绍这些内容。

目前已经有许多可用的 ASP.NET 应用程序和组件的开发环境，还有一些正在开发。主要的工具可能就是 Visual Studio .NET。从传统意义上讲，ASP 页面创建在文本编辑器或者基于文本的工具中，由于本书主要介绍编程而不是创建网页的工具，我们预先使用一种主要的文本编辑器来开发 ASP.NET 页面。但是，书中会在适当的地方介绍 Visual Studio .NET 创建组件和 Windows 窗体应用程序的功能。

书中列举的大多数代码都是使用 Visual Basic .NET 编写的,因为我们觉得这样对大多数开发人员都适合。即使读者喜爱 C#(或者其他语言),也可以方便地读懂 VB.NET 代码。如果我们选择同时提供 VB.NET 和 C#两种语言的代码清单和描述,就会在书中充斥很多不重要的内容,从而减少了对重要功能和技术的介绍。但是,本书的下载示例代码同时提供了 VB.NET 和 C#版本。当然,我们在浏览器中使用的客户机方代码不是 .NET 语言,而是 JavaScript,虽然有一个示例使用了客户端的 VBScript 来演示使用 VBScript 也可以完成工作。

我们还在代码中使用了混合的表示法。读者可以在许多地方看到 Hungarian 表示法,这是为了让读者容易读懂代码,因为这种方法清楚地显示了变量引用的数据类型是什么。在 .NET 中, Microsoft 公司建议我们不要使用这种表示法,但是却没有指出应该使用什么方法。读者如果有建议,可以与作者联系(通过我们的反馈地址 feedback@wrox.com)。如果读者对书中的其他问题有看法,也可以通过该地址与我们联系。

本书的读者对象

本书是为需要深入掌握 ASP.NET 来创建分布式数据驱动的应用程序的开发人员编写的,当他们需要以各种不同的方式显示和处理数据时尤其有用。换句话说,本书是为需要处理创建真正的分布式应用程序任务,而不是简单的读取和更新任务的开发人员编写的。

使用本书需要什么

我们希望读者喜欢使用 ASP.NET,虽然在实际中也会有少数人是迫不得已才使用 ASP.NET。由于本书是介绍 ASP.NET 的,希望读者已经对它熟悉。读者还需要具备一定的数据库使用经验,并且拥有合适的软件和硬件。下面会具体列出对读者的要求。

预备知识和经验

本书假定开发人员已经了解了 .NET,尤其是 ASP.NET,并且有使用以前版本的 ASP 开发 Web 应用程序的经验。但是,有了 Web 应用程序开发的基础知识(可能来自其他环境)和扎实的 ASP.NET 基础就足够了。

同样,掌握了 VB.NET 的编程知识(或者至少大致了解了 VB.NET,因为它与 VBScript 有类似之处)也会很有用。如果读者熟悉 C# 或者其他 .NET 语言,并且喜欢使用该语言,会发现书中介绍的技术也可以方便地使用该语言来实现。下载的示例代码文件中同时提供了 VB.NET 和 C#两种版本。

示例文件可以从 <http://www.wrox.com/> 下载。也可以从 <http://www.daveandal.com/books/4923/> 下载,在这里读者还可以发现书中一些示例的在线演示。

开发工具和运行时环境

读者需要的就是一个文本编辑器和一台运行 .NET Framework 的服务器。但是,为了能够运



行所有的示例，还需要获得和安装 Microsoft Mobile Internet Toolkit (MMIT)。

要想获得 .NET Framework，可以访问 <http://msdn.microsoft.com/net/>。要想获得 MMIT，可以访问 <http://msdn.microsoft.com/vstudio/device/mitdefault.asp>。

读者还需要一台数据库服务器，虽然它可以与 .NET Framework 在同一台物理机器上。本书使用的数据库服务器是 Microsoft SQL Server 2000，读者可以使用任何数据库，只要能够从 ASP.NET 连接上并且在其中可以创建存储过程就行。

我们使用 Microsoft Visual Studio .NET 来创建和编译 .NET 组件，但是 VS.NET 不是必需的，读者也可以从命令行编译类文件。当我们使用内联(inline)格式时，VS.NET 不是我们的 ASP.NET 页面的主开发环境，后台编码格式也不是 VS.NET 要求的。我们的确使用了 VS.NET 来开发 Windows 窗体应用程序，但是下载代码不使用它也可以工作。当然，这意味着读者将无法看到窗体的设计。

有关 Visual Studio .NET 的信息可以从 <http://msdn.microsoft.com/vstudio/> 获得。

为了在不同的环境中测试页面，你可能需要安装多个浏览器。我们在 IE、Opera 和 Netscape Navigator，以及 Nokia 的移动电话模拟器中测试了页面。

Netscape Navigator 可以从 <http://home.netscape.com/computing/download/index.html> 获得。Opera 可以从 <http://www.opera.com/> 获得。Nokia MIT 可以从 <http://forum.nokia.com/> 获得。

一些声明

阅读本书时，读者会遇到一个有趣的新的缩略词 FEGURM。它与使用 ADO.NET DataSet 对象执行数据源更新有关，在第 9 章将会看到。尽管我们花费了几小时试图想出一个合适的记忆方法，最终想到的仍然是 FEGURM(Feeling Extremely Good Using Reliable Methods)。如果读者有更好的建议，请与我们联系，提供最好建议的读者将会免费获得我们的一本书。

客户支持

我们一贯重视读者的反馈意见，希望知道读者对本书的看法：批评、赞扬和建议都欢迎。电子邮件地址是 feedback@wrox.com。

如何下载本书的示例代码

访问 Wrox 站点 <http://www.wrox.com/>，只需从 Search 栏找到书名，或者使用书名列表。单击 Code 栏的 Download，或者在本书的详细页面中单击 Download Code。

站点中可供下载的文件使用 WinZip 格式压缩。读者在硬盘的文件夹中保存附件后，需要使用 WinZip 或者 PKUnzip 解压缩。解压缩文件后，代码通常解压缩到相应章的文件夹中。开始解压缩进程时，要确保 WinZip 或 PKUnzip 设置的文件夹名正确。

勘误表

尽管我们努力确保本书的文本和代码准确无误,但是仍旧会有挂一漏万之处。读者若发现书中的错误,如拼写错误或者代码运行错误,请把勘误表发往电子邮件地址 support@wrox.com。

在 <http://www.wrox.com/> 站点可以找到勘误表,通过 **Advanced Search** 或者书名列表找到书名,单击 **Book Errata** 链接,它在本书对应的详细页面的图像下面。

电子邮件支持

如果读者希望就书中的问题直接向知晓详情的专家询问,可以发电子邮件到 support@wrox.com,电子邮件中要带上书名和 ISBN 书号的后 4 位数字。通常电子邮件中要包括如下内容:

- 书名和 ISBN 书号的后 4 位数字,主要问题所在的页码。
- 读者的姓名、联系方法,以及主要问题。

我们不会向读者寄送邮件,这样可以节省双方的时间。对于读者的电子邮件一般会经过以下处理流程:

- 客户支持:读者的电子邮件首先寄送给我们的客户支持人员,他们能够回答询问频率最高的一些问题,并能立即解答书中或者 Web 站点的常见问题。

- 编辑:较难的问题会送给负责本书的技术编辑,他们熟悉编程语言和软件产品,能够回答相关的细节性技术问题。

- 作者:最后,对于编辑无法解答的问题,将转发给作者。尽管我们不愿意干扰作者的写作工作,仍旧会向他们转发一些具体的问题。全体 Wrox 作者都会对他们的著作提供技术支持。他们会向客户支持人员或者编辑发送电子邮件,使得所有读者都能受益。

Wrox 的技术支持工作只对直接与书中内容相关的问题负责,对于超出正常支持范围的问题,可以通过 <http://p2p.wrox.com/forum> 站点上的讨论列表获得支持。

p2p.wrox.com

作者之间的讨论将加入到 P2P 邮件列表中。我们独一无二的系统通过邮件列表、论坛和新闻组提供了程序员到程序员(programmer to programmer)之间的联系,这是一种一对一的电子邮件支持系统。读者向 P2P 发送电子邮件查询时,会被许多 Wrox 作者和当时在邮件列表中的业界专家看到。读者可以从 p2p.wrox.com 站点找到许多不同的邮件列表,它们不仅对阅读本书有用,对读者开发实际的应用程序也大有帮助。

订阅邮件列表遵循以下步骤:

- (1) 进入站点 <http://p2p.wrox.com/>。
- (2) 从左侧的菜单栏中选择合适的类别。
- (3) 单击希望加入的邮件列表。
- (4) 遵循相关指令进行订阅,并输入你的电子邮件地址和密码。
- (5) 确认希望接受电子邮件。
- (6) 使用订阅管理器加入更多的邮件列表,并设置电子邮件优选项。



本系统为什么能提供最佳支持

读者可以加入邮件列表，并每周接收一次。如果读者没有时间，或者不方便接收邮件列表，可以搜索我们的在线档案室。垃圾邮件已经被删除，读者的电子邮件地址由独一无二的 Lyris 系统保护。任何加入或退出邮件列表订阅的请求，或者其他有关邮件列表的请求，都会发送到 listsupport@p2p.wrox.com。

目 录

第 1 章 分布式应用程序	1
1.1 什么是分布式数据应用程序	1
1.1.1 我们的定义	2
1.1.2 为什么需要分布	2
1.2 如何在 Internet 上进行分布式工作	3
1.2.1 Internet 协议	4
1.2.2 Internet 客户机	4
1.3 分布式应用程序的元素	5
1.3.1 客户机设备和客户机应用程序	7
1.3.2 服务器端数据存储	8
1.3.3 处理和表示信息	10
1.3.4 数据访问技术概述	13
1.3.5 关于“关系”数据访问	14
1.3.6 ADO.NET 数据访问类	15
1.4 深入理解 System.Xml	22
1.4.1 表示数据和信息	22
1.4.2 System.Xml 类	26
1.5 数据访问和 N 层模型	28
1.5.1 数据的用途是什么	29
1.5.2 一些 N 层数据访问情形	31
1.6 小结	36
第 2 章 组件和数据访问	38
2.1 不同类型的组件	38
2.1.1 .NET 组件	38
2.1.2 COM 和 COM+ 组件	41
2.1.3 用户控件	42
2.1.4 包含文件	43
2.1.5 服务器控件	44
2.2 访问与展示数据	44
2.2.1 向胖客户机传递数据	44
2.2.2 为低端客户机管理数据	44
2.2.3 显示 XML 数据	45



2.3	DataSet 对象	45
2.3.1	DataSet 对象的结构	45
2.3.2	何时使用 DataSet	48
2.4	关系数据访问组件示例	48
2.4.1	建立示例	48
2.4.2	访问及返回关系数据	49
2.5	小结	76
第 3 章	访问 XML 文档	78
3.1	XML 数据访问组件示例	78
3.2	小结	103
第 4 章	应用程序管道化	105
4.1	客户机端技术概论	105
4.2	远程传输数据到客户机	110
4.2.1	将关系数据远程传送到客户机	110
4.2.2	远程传送 XML 数据到客户机	112
4.2.3	在客户机上运行 .NET Framework	114
4.3	使用 ASP.NET 会话	125
4.3.1	ASP.NET 中的会话配置	125
4.3.2	检测会话支持	127
4.4	客户机检测示例	128
4.4.1	检测过程概述	129
4.4.2	检测客户机类型	129
4.4.3	default.aspx 页面	132
4.4.4	检测客户机端脚本支持	134
4.4.5	设备特有的主页	136
4.4.6	浏览结果	139
4.5	小结	143
第 5 章	处理低端客户机	144
5.1	多客户机订单应用程序	144
5.2	检测客户机类型	146
5.3	数据访问层	149
5.4	HTML 版本的订单应用程序	150
5.4.1	工作流程	150
5.4.2	搜索顾客	151
5.4.3	显示顾客列表	153
5.4.4	显示订单列表	159

5.4.5 显示订单详细记录	167
5.5 订单列表应用程序的移动式版本	171
5.5.1 工作流程	172
5.5.2 从服务器提取数据	173
5.5.3 移动式页面的内容	177
5.6 小结	190
第 6 章 处理胖客户机	192
6.1 使用 XML	192
6.1.1 为什么是 XSL 和 XSLT	192
6.1.2 展示并传递 XML	193
6.2 在 IE5 及以上版本中使用 XML	194
6.2.1 建立中间层	194
6.2.2 选择顾客	200
6.2.3 选择订单	211
6.2.4 查看订单的详细信息	217
6.3 在 IE5 及以上版本中使用 Web 服务	221
6.3.1 了解顾客数据 Web 服务	222
6.3.2 建立中间层	224
6.3.3 IE5 的 WebService 行为	228
6.4 一个超文本应用程序示例	232
6.4.1 运行超文本应用程序示例	233
6.4.2 建立中间层	235
6.4.3 HTA 页面内容	236
6.4.4 选择订单	241
6.5 处理带定界符的文本数据	242
6.5.1 为什么使用带定界符的文本数据	242
6.5.2 建立中间层	242
6.5.3 选择顾客	247
6.5.4 选择并查看订单	253
6.5.5 使用 XML 的客户机端数据绑定	258
6.6 小结	259
第 7 章 远程访问.NET 客户机	260
7.1 远程访问——一个新事物	260
7.2 远程访问对象	261
7.2.1 可远程访问的对象类	261
7.2.2 远程服务器	261
7.2.3 远程客户机	263



7.2.4	IIS 掌管的服务器	265
7.2.5	部署对象	266
7.3	Windows 应用程序	267
7.3.1	设计选择	268
7.3.2	远程访问 Windows 应用程序的数据	269
7.3.3	使用 Web 服务	274
7.4	增量下载	278
7.4.1	创建下载窗体	278
7.4.2	动态加载订单窗体	278
7.4.3	下载高速缓存	281
7.5	部署	282
7.6	安全措施	284
7.7	动态应用程序	291
7.8	小结	292
第 8 章	在低端客户机上更新数据	293
8.1	更新数据存储概览	293
8.1.1	尽可能使处理过程高效率	294
8.1.2	权衡结果并提供反馈	295
8.1.3	通过控制访问来保护数据	295
8.1.4	控制并发数据更新	296
8.2	.NET 中的数据更新技术	297
8.2.1	在更新数据时使用事务处理	303
8.2.2	更新时管理并发操作	306
8.3	HTML 3.2 客户机的数据更新示例	309
8.3.1	应用程序概览	309
8.3.2	显示和编辑订单	314
8.3.3	显示和编辑订单内容	341
8.4	移动设备的数据更新示例	353
8.4.1	应用程序设计	353
8.4.2	查看应用程序	355
8.5	小结	357
第 9 章	更新远程高速缓存数据	359
9.1	架构回顾	359
9.1.1	在客户机上高速缓存更新	360
9.1.2	将更新回送给服务器	361
9.1.3	加载一个 DataSet	362
9.2	执行更新过程	377

9.2.1 提供来自更新过程的反馈	377
9.2.2 更新过程详述	380
9.2.3 一个数据更新过程示例	382
9.3 小结	397
第 10 章 更新数据的组件	398
10.1 UpdateDataSet 组件	398
10.1.1 利用组件技术使跟踪成为可能	399
10.1.2 组件中通用的方法	401
10.1.3 管理插入行的 IDENTITY 列	402
10.1.4 UpdateAllOrderDetails 方法	413
10.1.5 创建参数化的 Command 对象	424
10.2 使用 UpdateDataSet 组件	430
10.3 小结	440
第 11 章 胖客户机更新应用程序	441
11.1 应用程序概述	442
11.2 中间层 ASP.NET 页面	444
11.2.1 从数据访问层中获取数据	444
11.2.2 提供给客户机的数据	445
11.3 在客户机上加载 Diffgram	445
11.3.1 加载订单数据	446
11.3.2 加载承运商数据	447
11.3.3 加载产品数据	448
11.4 编辑.NET DataSet Diffgram	449
11.4.1 浏览更新的 diffgram	450
11.4.2 更新的 diffgram 格式	453
11.5 edit-orders.aspx 页面代码和 HTML	457
11.5.1 创建订单列表	457
11.5.2 向服务器传递更新的 diffgram	458
11.5.3 显示所选择的订单详细记录	461
11.5.4 创建订单行列表	467
11.5.5 讨论无效数据	468
11.5.6 更新订单 diffgram	469
11.6 小结	489
第 12 章 协调更新错误	490
12.1 协调更新错误概述	490
12.2 协调的详细过程	495



12.3	借助于 Web 服务的胖客户机更新.....	520
12.3.1	问题和解决办法.....	520
12.3.2	中间层 Web 服务组件.....	521
12.3.3	Web 服务客户机应用程序.....	523
12.4	其他的胖客户机更新技术.....	537
12.4.1	采用界定符的胖客户机更新.....	537
12.4.2	超文本应用程序更新.....	538
12.5	小结.....	539
第 13 章	远程.NET 应用程序的数据更新.....	540
13.1	Windows Forms 应用程序.....	540
13.2	新的可编辑的订单表单.....	545
13.2.1	DataGrids 和 TableStyles.....	546
13.2.2	Windows 表单数据绑定.....	546
13.2.3	维护当前行.....	550
13.2.4	到目前为止的代码.....	552
13.3	处理数据更改.....	554
13.3.1	添加行.....	554
13.3.2	删除行.....	554
13.3.3	取消编辑.....	555
13.4	将更改送到数据库中.....	557
13.5	并发冲突的处理.....	559
13.5.1	ErrorProvider 对象.....	560
13.5.2	错误处理表单.....	560
13.5.3	识别错误.....	562
13.5.4	更改视图.....	567
13.5.5	纠正错误.....	570
13.5.6	导航.....	574
13.5.7	被删除行的处理.....	575
13.5.8	保存被纠正的数据.....	577
13.6	实际应用中的应用程序.....	578
13.7	小结.....	579
后话.....		580

第1章 分布式应用程序

如果本书是一本侦探小说，那么本章可能会以如此的情景开场：雨下得好大，Cressey 几乎看不见路的那边从房间脏兮兮的窗帘里透出的微弱的、桔黄色的光。然后就描述场景和介绍相关的人物。继续往下面看，书中就会讲到各个人物之间如何相互影响，从中可以知悉故事中人物的个性以及每个人物在其中担当的角色。很快您就会在脑海中浮现出故事情节的一幅幅画面以及其中涉及到的每个人物，当您沉浸于故事情节中时，画面会变得更加清晰而完整。最后将情节推向高潮，读者将会看到通过所有人物的相互作用，一个令人心潮澎湃的结局就产生了。

诚然，“分布式应用程序”这样的章节不会像小说那样立即就能吸引住读者的目光，并且我们在这里也不想这样做。但是，我们希望在.NET 的世界里营造这样的一个故事，让读者循着这个“故事”的步骤一步一步地达到那个有用的结局——哪怕过程不如侦探小说那般令人激动人心也没关系。

首先我们布置“场景”，让您知道相关的背景知识。再往下将介绍主要的特性，并帮助您熟悉它们各自的显著特点。书中将介绍如何将它们结合在一起，并帮助读者对整个过程中有一个清晰的印象。最后，我们将向读者展示通过它们的相互作用如何产生我们自己的“心潮澎湃”的结果——一个基于.NET 和其他技术的分应用程序。

下面是第1章的内容：

- 分布式应用程序的定义
- 计划、设计和实现一个分布式应用程序的基本要素
- 在实现一个应用程序时可以用到的各种对象和技术
- 这些对象怎样通过相互作用来提供我们所没有的或者需要的功能
- 概述 ADO.NET 关系数据访问类
- 概述 System.Xml 文件访问类
- 怎样将数据访问嵌入到 N 层模块中

我们将以通常的背景知识开场，并尽量避免类似侦探小说中的太多模糊的引用。当然，作者有时仍用一两个拙劣的双关语和一些即席评论，但这只不过是我们叙述过程中的“小丑”而已。

1.1 什么是分布式数据应用程序

分布式应用程序这个术语在以 Web 为中心的世界中极为流行，开发人员利用它来指涉各种各样的应用程序。我们使用“应用程序”这个术语来指涉如此之多的不同的对象有点力不从心。这里使用的是它的一般意义：一个能完成有用事情的程序。我们同样还会以更加具体的和技术的方式使用这个术语。