

10

TP313.4  
D12

# .NET XML高级编程

Dinar Dalvi Joe Gray 等著  
英宇 林琪 费广正 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: 01-2002-2478

## 内 容 简 介

XML 是描述和传送数据的一项成熟技术, 该技术在软件开发的各个领域已经产生了巨大的影响。当 Microsoft 推出 .NET Framework 时, 就广泛地使用了 XML。没有其他的技術會像 XML 一样与 .NET 结合得如此紧密, 这不仅体现在开发人员的工作上, 也体现在整个基础架构中。

本书的目的是尽可能地以最有效的方式为您提供足够的信息, 让您学会如何在 .NET Framework 中使用 XML。本书使用 C# 和 Visual Basic .NET 两种语言, 解释了所有与 XML 相关的 .NET Framework 类库的用法细节, 包括操作、验证、转换和串行化 XML 数据等。本书还介绍了开发人员如何在 .NET Framework 中使用 XML 的强大功能(例如, 如何使用 ADO.NET 和 ASP.NET 的 XML 新功能)。如书中所述, .NET Framework 自身就在配置文件、元数据和 C# 的 XML 代码文档机制中使用了 XML。

本书适合从事 .NET 开发的中高级程序员。阅读本书, 您需要具备 C# 或 Visual Basic .NET、XML 以及 XML 相关技术(如 XSLT、XPath 和 XML Schemas)的基础知识。

Dinar Dalvi Joe Gray et al: Professional XML for .NET Developers

EISBN: 1-861005-31-8

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清华大学出版社和英国乐思出版公司合作出版。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。

书 名: .NET XML 高级编程

作 者: Dinar Dalvi Joe Gray 等著 英宇 林琪 费广正 等译

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.tsinghua.edu.cn>

责任编辑: 李阳

印 刷 者: 北京牛山世兴印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 43.75 字数: 1119 千字

版 次: 2002 年 6 月第 1 版 2002 年 6 月第 1 次印刷

书 号: ISBN 7-302-05522-X/TP·3249

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](http://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](http://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

本书的内容将涵盖两大技术的交集，即 .NET 和 XML。

XML 作为一项热点话题已经有一段时间了。业界对这项由 W3C 推荐，允许在未知平台下进行数据通信和信息存储的技术的巨大接受力是惊人的。从使用 XSLT 转换语言在不同浏览器中显示数据，到在 Web 服务间通过 SOAP 进行消息传输，XML 的应用无处不在。

作为 Microsoft 一个革命性且引以为荣的新视点，.NET 具备以下一些特性：允许以语言无关性的方式编写应用程序，可以在不同语言间共享代码，支持自我描述类，以及在一定程度上实现了对编程代码建立自身文档。.NET 的设计思路就是明确用于方便地进行 Web 服务应用程序的开发。

为了获得这种令人激动的编程新环境，Microsoft 在 .NET 中广泛地应用了 XML 技术。事实上，迄今为止没有其他任何一项技术能够将 .NET 和 XML 技术如此紧密地结合在一起。XML 作为一种通用数据格式，被用于从配置文件到元数据、从 RPC 到对象串行化的所有应用场合。程序员以前可以使用的通过 MSXML 分析器实现的所有 XML 功能现在均被封装在 System.Xml 命名空间中，其性能也得到了提升，同时还增加了一些令人振奋的新功能。与此相关的是 XML 对 ADO.NET 所具备的功能也提供了新的支持。例如，程序员现在已经能够以层次化(XML)和关系型(数据库)这两种方式同时访问和更新数据了。

## 本书读者对象

本书是为那些已经开始进行 .NET 应用程序开发以及那些已经熟悉 XML 的中级或高级程序员编写的。

尽管在本书的第 1 章中向读者介绍了大量的 .NET 概念，但本书并非旨在让开发人员通过它来了解 .NET 技术，毕竟介绍该内容的相关书籍和文章已经很多了。在本书中，我们将转而直接讲解在 .NET Framework 下使用 XML 的核心内容。为了更好地理解本书，读者应具备或正在学习 C# 或 VB.NET 的初步知识。本书中的示例将以这两种语言给出，其中有些将在书的内容中给出，而另外一些代码则提供相应的站点供读者下载。

与此同时，在使用本书时，读者还需要参考许多介绍 XML 技术的书籍和文章。我们假设读者已经具备了 XML、命名空间和 XSLT 的一般知识以及有关 XML 模式的基础知识。

请参见本书附录 D 获取更多关于 C#、VB .NET 和 XML 技术的资料。

## 本书涵盖的内容

本书将讲解如何在 .NET Framework 中使用 XML，以及程序员如何才能完全利用在 .NET Framework 中使用 XML 的强大功能。本书将实现以下两大目的：

- 让读者理解在 .NET Framework 库中可用于处理 XML 的各种类，例如实现读、写、验证以及 DOM 处理等。
- 解释 .NET Framework 自身是如何使用 XML 的。其中包括基于 XML 的配置文件，在远



程和 Web 服务中使用 XML，以及在处理数据库时，允许使用特别功能的 ADO.NET 中的 XML 集成。

以下是详细的章节和附录划分。

## 章节划分

### 第 1 章 .NET Framework 概述

在本章中，我们将给出一个概述或提醒(根据读者已有的相关知识而定)，向读者讲解 .NET Framework 的内容。

我们将介绍 .NET Framework、公共语言运行时(Common Language Runtime, CLR)和公共语言规范(Common Language Specification, CLS)。将提出以下问题：“什么是程序集？”，接着讨论应用程序域和 Framework 安全问题，再其后，在讨论 ASP.NET 以及可提供的众多控件(包括 HTML、Web、验证和移动控件)之前，我们将研究应用部署和 .NET 语言：C# 和 VB.NET。最后，还将了解用于进行数据库访问的新技术 ADO.NET。

### 第 2 章 .NET Framework 中的 XML

本章我们将带着读者轻松地领略 .NET Framework 中的所有 XML 集成功能。在本书其后的章节中我们将详细讨论该功能。

Microsoft 在 .NET 中引入了几个新的 XML 应用程序，并已经完成了一些创新的工作，改进了 XML API 的内核。我们从讨论在配置文件中 使用 XML 入手，介绍启动、运行以及远程设置等内容。在此将明白 XML 是如何使用 ADO.NET 提供的新功能解决 ADO 存在的问题的。在介绍包括 XmlReader 和 XmlWriter 抽象类在内的类的 System.Xml 命名空间之前，我们将介绍 ASP.NET 并研究其基于流和基于 DOM 的 XML 类。此外还将简单地介绍 .NET 中的 DOM、XPath、XSLT 和 MSXML 等概念，以及相应的 C# XML 代码文档特性。最后，我们将讨论 Framework 对产生和使用 XML 文档的支持。

### 第 3 章 XML 读写

从本章开始(3~7 章)我们将开始详细介绍 System.Xml 命名空间中包含的各项功能。

特别地，这里将介绍 .NET Framework 提供的快速、仅前向的读、写 XML 文档的机制，即 XmlReader 和 XmlWriter 抽象类。我们将深入分析由这两个抽象类派生的 XmlTextReader 和 XmlTextWriter 类。最后，还将继续讨论节点顺序、空白字符、实体、命名空间处理以及其他的命名空间支持。

### 第 4 章 XML 的 DOM 导航

在本章中，我们将介绍 .NET Framework 中由类的 System.Xml 命名空间所提供的 DOM 功能。在此将讨论如何从 URL、文件或内存字符串中打开文档，以及在将它们串行化到 XML 字符串之前如何从这些文档中搜索和访问这些文档的内容。还将介绍 XmlDocument 对象与 XmlReader 和 XmlWriter 抽象类之间的区别，并决定在何处使用哪个更为恰当。最后，我们将创建一个由 XmlDocument 类派生的类，借以向读者展示在 .NET Framework 的类中添加功能是何等容易。

### 第 5 章 XML 的 XSL 转换

.NET Framework 对 XSLT 和 XPath 提供了稳健的支持。在本章中，我们将介绍 .NET Framework 中用于 XSL 转换的技术，即 System.Xml.Xsl 和 System.Xml.XPath 命名空间，以及

XslTransform 类。.NET Framework 完全支持 W3C 定义的 XSLT 和 XPath 规范,但也对这些规范提供了许多有用的扩展,这些扩展功能增强了.NET 应用程序中样式表的可用性。我们将研究使用嵌入式脚本<msxsl:script> 进行 XML 文档转换以及如何扩展带扩充对象的样式表。此外还将讨论 XPathDocument 对象。在本章结束之前,我们还将介绍如何调整 XSL 转换以获得更佳性能,以及巧妙的流操作等,最后还将讨论其他的 XPath 函数。

## 第 6 章 验证 XML

本章将介绍 XML 验证语法的不同选项:DTD、XDR 模式和 XSD 模式。接着将讲解在 Visual Studio .NET 中创建一个 XSD 模式的不同途径:使用 XML Designer 进行创建、从数据库中创建、使用 XSD Generator 创建、从 XML 文档中进行创建、从 XDR 模式中创建或者从程序集中创建。我们将讨论模式对象,看看怎样才能将 XML 文档链接到 DTD、XDR 模式和 XSD 模式上,以及怎样才能使用 System.Xml 命名空间中的 XmlValidatingReader 对象实现验证。在此将研究利用 XmlSchemaCollection 在内存中建立一个模式缓存,以优化性能,并处理 XML 文档中未限定或已经由命名空间限定的内容。

## 第 7 章 XML 串行化

在本章中,我们将介绍使用 System.Xml.Serialization 命名空间中的 XmlSerializer 类实现将对象串行化成 XML 数据,更明确地说就是创建串行器,然后对复杂对象、属性、枚举值、数组和复合对象进行串行化和并行化。还将讨论继承的串行化和并行化,且其后将介绍 XML 元素、XML 属性和文本内容的格式化。在本章结束之前,我们将介绍使用带串行化的 XSD 模式,以及将 XSD 类型映射到公共语言运行时类型。最后还将研究对复杂对象图进行串行化。

## 第 8 章 MSXML

在第 8 章之前,我们已经讲解了.NET Framework 库中 System.Xml 命名空间提供的大多数类。因此,在本章中将讲解 MSXML 中提供的在.NET 之前的 Microsoft XML API。在对一些 MSXML 对象与.NET 中的对象进行比较之前,我们将从不同的角度来认识这一点。首先介绍 MSXML 的 XmlDocument 和 FreeThreadedXmlDocument,然后对 MSXML 的 XmlDocument 方法和.NET 的 XmlDocument 方法进行比较。我们将演示如何在.NET System.Xml 命名空间中使用实现类似于 MSXML XmlHttp 对象功能的类。类似地,还将对 MSXML 的 XslTemplate(在 JavaScript 中使用)和.NET 的 XslTransform 进行比较。另外还将对 SAX(尽管可从 MSXML 中获得,但并非可以直接从.NET 中获得)和 DOM 以及 XmlReader 进行比较。最后,我们将讨论在需要使用 MSXML 时,可以利用.NET 代码中的运行时可调用包装器(Runtime Callable Wrapper)。

## 第 9 章 扩展 XmlReader 和 XmlWriter

我们曾在第 3 章中首次详细介绍过 XmlReader 和 XmlWriter,在本章将研究如何扩展 XmlReader 和 XmlWriter 抽象类,使其与模式中指定的语法能够通信。明确地说,就是使用扩展类和运行时可调用包装器,启动一个在 XML 文档、Microsoft Visio 图表和 Microsoft Word 文档的表格之间进行转换的应用程序。

## 第 10 章 ADO.NET

在本章我们将结束对类的 System.Xml 命名空间的讨论,转而介绍在.NET 中如何使用 XML 的更广泛的知识。

在本章中,我们将介绍在 ADO.NET 中 XML 所起的作用,并对 ADO.NET 和 ADO 进行比



较。本章中将讨论新的 DataSet 类的性能，包含读和写 XML、程序化访问或修改其 XML 表示等。这里将着重指出如何实现 DataSet 与 XmlDocument(由 XmlDocument 类派生)同步，并说明为什么需要这样做。我们还将讨论强类型化的 DataSet 的优越性。最后我们将简略介绍如何访问由 SQL Server 2000 所提供的一些功能。

### 第 11 章 ASP.NET

在本章中，我们将深入分析什么是 ASP.NET，它与 ASP 之间的兼容性，以及 ASP.NET 与 XML 彼此间的相关性。我们将介绍 Server 端和 Web 控件，以及错误处理和跟踪。还将研究如何使用 XML 控件显示来自文件或 XmlDocument 对象的 XML，以及使用 DataGrid 控件显示复杂的 XML 文档。本章中讨论的其他主题还有：服务器端缓存、身份验证、授权、配置和部署。我们将研究 XML 配置文档 web.config 的位置、结构以及各个部分。最后还将介绍定制错误、HTTP 处理程序和 HTTP 模块。

### 第 12 章 案例研究：使用 ASP.NET 与 ADO.NET

在该案例研究中，我们将创建一个在线 DVD 租借系统，还将演示在无须引入大块复杂代码的前提下如何实现该应用。特别地，我们将分析 ASP.NET 的特征和服务器端控件，如何使用 ADO.NET 打开数据库连接、读取 XML 格式的数据、以及 ADO.NET 如何具备内置的 XML 支持。在了解如何添加新的 Section 对象和编辑行之前，我们将分析设计和其他技术要求，然后考虑使用服务器端 XSL 以创建与浏览器无关的应用程序。

### 第 13 章 Web 服务与 SOAP

在本章中，我们将介绍 .NET Web 服务的新环境，以及它们是如何使用 XML 技术的。在此将演示在 Visual Studio .NET 中如何创建一个简单的 Web 服务，然后讨论 Web 服务中使用的协议，也就是 HTTP GET/POST 和 SOAP。作为一个示例，这里将创建我们自己的“股票报价(Stock Quote)” Web 服务，并介绍代理代码、Web 服务目录、businessEntity 和 businessService 结构、基于 XML 的 Web 服务描述语言(Web Service Description Language, WSDL, )，以及使用 Session 和 Application 对象等。最后，我们将讨论部署和设计 Web 服务时需要考虑的一些事项，以及同步和异步 Web 服务之间的区别。

### 第 14 章 案例研究：使用 Web 服务

在这个案例研究中，我们将创建一个提供日历功能的简单 Web 服务。在此将逐步讲解该 Web 服务的概念设计，包括使用案例及功能需求等，然后就可以创建一个简单的数据库来支持我们的日历应用程序。本章的主要内容是编写 Web 服务及其组成方法的代码。最后，我们将创建一个简单的 ASP.NET 客户端应用程序来使用该 Web 服务。

### 第 15 章 远程化概述

在第 15 章，我们将讨论远程化的基础知识，随后讲解客户端和服务端对象激活问题。在此将介绍如何对远程对象进行实例化和激活，以及用于控制该行为的选项，同时我们还将讨论 .NET 是如何用代表远程对象的本地代理来实现远程化的。在介绍 objRef 类、生存期租约(lifetime lease)、远程化和注册通道及 HTTP 和 TCP 通道之前，我们将说明创建 .NET 远程应用程序的基础知识，包括创建一个简单的客户机和服务器应用程序等。

### 第 16 章 C#代码文档

在最后一章，即第 16 章，我们将介绍 C#中的 XML 代码文档化的优点。将考虑如何在命令行用/doc:filename 选项、编译器警告以及选项/incremental 来编译注释代码。然后我们将继续



讨论在 Visual Studio .NET 中编译已注释的 C# 代码，并且深入介绍了各种编译器提供的元素和属性。本章的下一部分将分析注释的实际使用、其最佳实践，以及用于公共和非公共类型和成员的注释。然后，我们将研究将 XML 转换成更加可读的帮助文件时所使用的转换，随后给出一个示例转换应用程序及相关的 XSLT 样式表。最后将通过讲解其他应考虑的事项来结束本章的内容，例如 Visual Studio .NET 如何根据 XML 文档标记的使用向我们提供基于 Web 的文档。

## 附录

### 附录 A System.Xml

附录 A 中我们将给出对 System.Xml 命名空间的详细说明。作为 XML 类的总体命名空间，它处理了包含对 XML 数据进行访问、导航和维护在内的各项功能。

### 附录 B System.Xml.Xsl 和 System.Xml.XPath

附录 B 中我们将介绍通用的 System.Xml.Xsl 和 System.Xml.XPath 命名空间。这些命名空间将分别处理 XSL 转换和 XML 文档导航问题。

### 附录 C .NET 术语表

附录 C 对一些 .NET 术语的解释给出了有益的参考。

### 附录 D 资源

附录 D 给出了本书引用的一些有用的 URL 和文章标题的列表，用于进一步阅读及下载产品或代码。

## 使用本书须知

为运行本书的示例，您必须安装 .NET Framework。可以购买一个 Framework CD-ROM，或者从 Microsoft Web 站点下载。

### 注意：

在我们写这本书的时候，.NET Framework 刚刚发行其 Beta 2 版本，但任何有关修改都将在 Wrox 的 Web 站点 <http://www.wrox.com/> 中发布。

大多数代码可以用您喜欢的文本编辑器编写，并在命令行进行编译和使用 .NET Framework 提供的工具。然而，您可能也想使用 Visual Studio .NET 来创建和编译代码：可以从 <http://msdn.microsoft.com/vstudio/nextgen/beta.asp> 下载。

某些章节需要用到其他软件时，将特别指出。具体地：

- 第 10 章和第 12 章以及第 14 章需要用到 SQL Server 2000。这个软件的 120 天试用期版本可以从 <http://www.microsoft.com/sql/> 下载。

- 第 9 章“定制抽象类”需要用到 Microsoft Word 和 Microsoft Visio。然而，该示例可以运行于只有 Microsoft Word 的情况下，但有许多功能将受到限制。

本书的代码可以从 <http://www.wrox.com/> 下载。更详细的内容在下面的“客户支持”一节中给出。

## 客户支持

我们非常想知道您对于本书的看法：您喜欢它的什么内容、不喜欢它的什么内容，以及您



认为我们可以如何在下次做得更好。您可以通过发一封 E-mail 给 [feedback@wrox.com](mailto:feedback@wrox.com) 发表您的意见。请注意别忘了在意见中提及本书的书名。

### 如何下载本书的示例代码

访问 Wrox 的站点：<http://www.wrox.com/>，简单地通过该站点所提供的 Search 功能便可定位到该书名，或者通过使用书名列表找到它。在书籍的详细页(detail page)的 Code 栏中单击 Download 或直接单击 Download Code 即可。

站点中用于下载的文件都利用 WinZip 进行了压缩。当您在硬盘的文件夹中保存了这些附件后，必须用解压缩程序如 WinZip 或 PKUnzip 等对其进行解压缩。解压缩这些文件时，代码通常将解压缩到相应的章节文件夹中。开始解压缩之前，请确保您的软件(WinZip、PKUnzip 等)已经设置为使用的文件夹名。

### 勘误表

我们尽了最大努力确保在本书的内容和代码中不出现错误。然而，世上无完人，出现错误也是难免的。如果您在我们的书中发现一个错误，例如拼写的错误或一段错误的代码，我们非常高兴能得到您的反馈。您若能向勘误表中发送这个错误内容，将节省其他读者大量的时间，当然，您也将帮助我们提供更高质量的信息。您需要做的就是简单地发送一封 E-mail 到 [support@wrox.com](mailto:support@wrox.com)。您的信息将被检查，如果是正确的，我们将把它发送到勘误表中，或者在该书的后续版本中使用。

为在 Web 站点中找到某书的勘误表，请到 <http://www.wrox.com/>下，简单地通过我们的 Advanced Search 功能或从标题列表中定位到该书的位置。在 Book Errata 链接上单击，它位于该书详细页的封面图形下方。

### E-mail 支持

如果您想直接向详细了解本书的一个专家询问有关本书的问题，可以将 E-mail 发送到 [support@wrox.com](mailto:support@wrox.com)，在 E-mail 的主题中要包含本书的书名和 ISBN 的后 4 位数字。一个典型的 E-mail 应包含以下内容：

- 在主题部分包括书名，ISBN 的后 4 位数字和页号。
- 在 E-mail 的消息体中包含您的姓名、联系信息以及所提的问题。

我们不会向您发送垃圾邮件。我们需要你的详细信息以节省彼此的时间。您在发送 E-mail 消息时，将经过以下一系列支持：

- 客户支持——您的消息将递送给我们的客户支持员工，他们将是第 1 个阅读这条消息的人员。他们有最常问到问题的文件，并将立即回答任何关于本书的一般性问题或与 Web 站点相关的一些问题。

- 编辑——更深层次的查询将转寄给负责本书的技术编辑。他们具有编程语言或特定产品的经验，能够回答有关该主题的相关详细技术问题。

- 作者——最后在一个不大可能发生的情况下，技术编辑也无法回答您的问题，他们将把您的请求转寄给本书的作者。我们试图尽量避免作者从他们的写作中分心，然而，我们也非常乐意将一些特定的请求转寄给他们。所有 Wrox 作者都将帮助支持他们写作的书。他们将他们的

回答发送给客户和编辑，然后所有读者将从中受益。

Wrox 支持的处理只对与我们出版的书籍直接相关的问题提供支持。对所有本书范围之外的问题将通过我们的 <http://p2p.wrox.com> 论坛提供支持。

### **p2p.wrox.com**

为参与作者与读者的讨论请加入我们的 P2P 邮件列表。我们的独特系统提供建立在邮件列表、论坛和新闻组之上的程序员到程序员(programmer to programmer™)的联系，所有都附加在一对一的 E-mail 支持系统中。如果您向 P2P 发送了一个请求，可以确信这个请求被许多 Wrox 作者和其他一些在邮件列表中的勤奋的专家进行了检查。在 [p2p.wrox.com](http://p2p.wrox.com) 中，您将发现许多不同的列表将提供帮助，不仅在读这本书时，而且在开发自己的应用程序时您将同样从中受益。特别适合本书的邮件列表有 xml, ado\_dotnet, aspx, dotnet\_webservices, c\_sharp 以及 vb\_dotnet 列表等。

若想订阅一个邮件列表，按以下步骤进行即可：

- (1) 访问 <http://p2p.wrox.com/>。
- (2) 从左边的菜单栏中选择适当的类别。
- (3) 单击您想加入的邮件列表。
- (4) 按其指示填写 E-mail 地址和密码。
- (5) 回复您收到的确认 E-mail。
- (6) 使用订阅管理器加入更多列表并设置常用的 E-mail。

### **为什么说该系统提供了最好的支持**

当您加入一个邮件列表后，可以有选择性地接收所张贴的个别 E-mail，以及接收一个每日消息摘要，或者根本不接收任何 E-mail，而只通过 Web 站点进行阅读或答复。您也可以搜索我们的在线档案。垃圾邮件将被删除，因此只会有相关邮件寄送给您。关于加入或退出列表服务的查询，以及其他类似的一般性查询，请将 E-mail 发送到 [listsupport@p2p.wrox.com](mailto:listsupport@p2p.wrox.com)。

# 目 录

<b>第 1 章 .NET Framework 概述</b> .....	<b>1</b>
1.1 .NET Framework.....	1
1.1.1 公共语言运行时.....	2
1.1.2 Framework 类.....	17
1.2 .NET 语言.....	18
1.2.1 C#简介.....	18
1.2.2 VB .NET 综览.....	21
1.3 ASP.NET.....	23
1.3.1 HTML 控件.....	23
1.3.2 Web 控件.....	24
1.3.3 移动控件.....	26
1.3.4 Web 服务.....	27
1.4 global.asax.....	29
1.4.1 应用程序指令.....	29
1.4.2 服务器端对象标记.....	30
1.4.3 服务器端的 include 指令.....	30
1.5 ADO.NET.....	30
1.5.1 数据访问.....	30
1.5.2 ADO 与 ADO.NET.....	30
1.5.3 XML 和 ADO.NET.....	31
1.6 小结.....	32
<b>第 2 章 .NET Framework 中的 XML</b> .....	<b>33</b>
2.1 从.INI 文件到现在的注册表.....	33
2.1.1 使用 XML 建立配置文件的好处.....	33
2.1.2 配置文件的类型.....	34
2.1.3 配置文件的 XML 结构.....	35
2.2 XML 代码文档.....	42
2.3 ADO.NET, 数据和 XML.....	45
2.3.1 ADO 的局限性.....	45
2.3.2 进入 ADO.NET.....	45
2.4 ASP.NET 和 XML.....	46
2.5 Web 服务.....	46



2.6	XML 串行化	47
2.7	基于流和基于 DOM 的 XML 类	49
2.7.1	System.Xml 简介	49
2.7.2	.NET 中的 MSXML	55
2.8	小结	56
<b>第 3 章</b>	<b>读写 XML</b>	<b>57</b>
3.1	所用的技术	57
3.2	System.Xml 程序集	57
3.3	读取 XML 文档	57
3.3.1	XmlReader 抽象类	58
3.3.2	XmlReader 与 SAX 的比较	58
3.3.3	XmlTextReader 类	58
3.3.4	XmlNodeReader 类	74
3.3.5	XmlValidatingReader 类	74
3.4	编写 XML 文档	74
3.4.1	XmlWriter 抽象类	74
3.4.2	XmlTextWriter 类	75
3.5	一个更复杂的示例	92
3.5.1	问题	92
3.5.2	解决方案: 封装 Reader 和 Writer	93
3.6	小结	98
<b>第 4 章</b>	<b>XML 的 DOM 导航</b>	<b>99</b>
4.1	所用的技术	99
4.2	System.Xml 程序集	99
4.3	理解 DOM 类	99
4.4	用法准则	101
4.5	打开一个已有文档	101
4.5.1	从 URL 打开文档	101
4.5.2	从文件打开文档	102
4.5.3	从内存字符串打开文档	102
4.6	在文档中导航	103
4.7	在文档中搜索内容	103
4.7.1	GetElementsByTagName()方法	103
4.7.2	GetElementById()方法	106
4.7.3	SelectNodes()方法	108
4.7.4	SelectSingleNode()方法	110
4.8	访问内容	112

4.8.1	元素	112
4.8.2	文本	115
4.8.3	注释	117
4.8.4	处理指令	117
4.8.5	属性	119
4.8.6	实体处理	122
4.8.7	空白字符处理	122
4.8.8	命名空间支持	126
4.8.9	验证	128
4.9	修改文档	128
4.9.1	删除内容	128
4.9.2	修改内容	132
4.9.3	创建新内容	134
4.9.4	插入内容	136
4.9.5	加入属性	136
4.10	从头创建一个文档	138
4.11	一个更复杂的示例	140
4.11.1	一个商务问题	140
4.11.2	解决方案: 扩展 XmlDocument 类	140
4.12	小结	147
<b>第 5 章</b>	<b>XML 的 XSL 转换</b>	<b>148</b>
5.1	所用的技术	148
5.2	System.Xml.Xsl 命名空间	148
5.3	System.Xml.XPath 命名空间	148
5.4	XSLT 处理	149
5.4.1	XslTransform 类	149
5.4.2	用 msxsl:script 编写嵌入式脚本语言	153
5.4.3	向一个转换传递参数	156
5.4.4	用扩展对象扩展样式表	160
5.4.5	调整 XSLT 转换以获得更佳性能	164
5.5	其他 XPath 函数	170
5.5.1	XPathNavigator 抽象类	170
5.5.2	用 XPath 表达式测试节点	171
5.5.3	用 XPath 表达式选择节点	173
5.5.4	预编译 XPath 表达式	176
5.6	小结	178



<b>第 6 章 验证 XML</b> .....	179
6.1 为 XML 文档定义语法.....	179
6.1.1 用 DTD 定义语法.....	181
6.1.2 用 Microsoft XDR 模式定义语法.....	182
6.1.3 用 XSD 模式定义语法.....	183
6.2 在 Visual Studio .NET 中创建 XSD 模式.....	184
6.2.1 用 XML Designer 创建 XSD 模式.....	184
6.2.2 在数据库中创建 XSD 模式.....	187
6.2.3 使用 XSD Generator 创建 XSD 模式.....	191
6.3 使用模式对象模型.....	195
6.3.1 模式对象模型类的层次结构.....	195
6.3.2 使用模式对象模型创建模式.....	197
6.4 在 .NET Framework 中验证 XML.....	208
6.4.1 链接 XML 文档到 DTD 或模式.....	208
6.4.2 执行验证.....	210
6.4.3 为模式建立缓存.....	215
6.5 小结.....	217
<b>第 7 章 串行化 XML</b> .....	218
7.1 串行化一个对象为 XML 数据.....	219
7.1.1 定义一个用于串行化的类.....	219
7.1.2 创建用于串行化的类.....	220
7.1.3 创建一个 XML 串行化器.....	220
7.1.4 串行化对象.....	221
7.1.5 检查 XML 文档中已经串行化的数据.....	222
7.2 并行化 XML 数据到对象.....	223
7.2.1 定义用于并行化的类.....	223
7.2.2 创建用于特殊类型的 XML 串行化器.....	223
7.2.3 并行化对象.....	223
7.2.4 使用并行化对象.....	224
7.3 处理意外的 XML 内容.....	225
7.3.1 处理意外属性.....	226
7.3.2 处理意外元素.....	228
7.3.3 处理一般意外节点.....	229
7.4 串行化和并行化复杂对象.....	231
7.4.1 串行化和并行化属性.....	231
7.4.2 串行化和并行化枚举值.....	233
7.4.3 串行化和并行化数组.....	234

7.4.4 串行化和并行化复合对象	236
7.4.5 带继承的串行化和并行化	241
7.5 改进串行化过程	243
7.5.1 为串行化定义简单格式化	244
7.5.2 高级改进问题	249
7.6 在串行化中使用 XSD 模式	258
7.6.1 从类产生 XSD 模式	259
7.6.2 从 XSD 模式产生类	261
7.7 串行化对象图	265
7.8 小结	272
<b>第 8 章 MSXML</b>	<b>273</b>
8.1 什么是 MSXML	273
8.2 MSXML 对象模型	274
8.2.1 DOMDocument 和 FreeThreadedDOMDocument	275
8.2.2 XMLHTTP	280
8.2.3 XSLTemplate	284
8.2.4 SAX	290
8.3 在 .NET 代码中使用 MSXML	290
8.4 MSXML 以后是什么	293
8.5 小结	294
<b>第 9 章 扩展 XmlReader 和 XmlWriter</b>	<b>295</b>
9.1 为什么要扩展 XmlReader 和 XmlWriter	295
9.1.1 为什么要使用一个 XML 接口	295
9.1.2 为什么要使用 XmlReader/XMLWriter 作为接口	296
9.2 工程的范围	296
9.3 设置这个工程	301
9.4 创建 XmlSimpleReader 和 XmlSimpleNode	302
9.5 创建 XmlWord2000Reader	307
9.6 创建 XmlSimpleWriter	313
9.7 创建 XmlWord2000Writer	315
9.8 创建 XmlVisioReader	321
9.9 创建 XmlVisioWriter	325
9.10 小结	330
<b>第 10 章 ADO.NET</b>	<b>331</b>
10.1 XML 在 ADO.NET 中的作用	331
10.2 ADO.NET 与 ADO 的比较	332





10.2.1	两者的相似点 .....	332
10.2.2	两者的不同点 .....	332
10.2.3	非连接数据 .....	334
10.3	托管数据提供者 .....	334
10.3.1	SQL Server .NET 数据提供者 .....	334
10.3.2	OLE DB .NET 数据提供者 .....	335
10.3.3	ODBC .NET 数据提供者 Beta 1 .....	335
10.3.4	公共模型 .....	335
10.4	数据集(DataSets) .....	338
10.4.1	数据集与 ADO Recordset 的比较 .....	339
10.4.2	数据集与 IDataReader 的比较 .....	339
10.4.3	数据集集合 .....	339
10.5	用数据集读和写 XML .....	354
10.5.1	WriteXml 方法 .....	354
10.5.2	WriteXmlSchema 方法 .....	355
10.5.3	GetXml 和 GetXmlSchema 方法 .....	355
10.5.4	ReadXml 方法 .....	359
10.5.5	把模式信息加载到数据集中 .....	366
10.5.6	XML 命名空间 .....	371
10.5.7	强类型化的数据集 .....	372
10.6	XmlDataDocument .....	375
10.6.1	与一个数据集建立同步 .....	375
10.6.2	元素和行转换 .....	383
10.7	从 SQL Server 2000 获取 XML .....	386
10.7.1	使用 FOR XML .....	386
10.7.2	SQLXML 托管类 .....	388
10.8	小结 .....	390
<b>第 11 章</b>	<b>ASP.NET .....</b>	<b>391</b>
11.1	概述 .....	391
11.2	什么是 ASP.NET .....	391
11.2.1	语言无关执行 .....	392
11.2.2	编译执行 .....	392
11.2.3	事件驱动的处理过程和 Web Form .....	392
11.2.4	服务器端控件 .....	392
11.2.5	定制 Web 控件 .....	393
11.2.6	错误处理和跟踪 .....	394
11.2.7	服务器端缓存 .....	394