PROGRAMMER TO PROGRAMMER™

Beginning C# XML

Essential XML Skills for C# Programmers

C# XML 入门经典

——C# 编程人员必备的 XML 技能

(美) Stewart Fraser 著
Steven Livingstone
毛尧飞 崔 伟 译

清华大学出版社

C# XML 入门经典

—— C#编程人员必备的 XML 技能

(美) Stewart Fraser
Steven Livingstone 著

毛尧飞 崔 伟 译

清华大学出版社

北 京

内 容 简 介

XML 技术是近年来最热门的话题，并且已经广泛应用于编程领域中。

本书主要讲述了 XML 技术在 C#中的应用，同时还介绍了 XML 的相关技术，如 XPath、XSLT 和 XML Schema 等。主要内容包括：XML 语法和格式良好的 XML，XML 命名空间，利用 DTD 和 XML Schema 进行 XML 验证，使用 SOAP 和 Web 服务，以及运用 ADO.NET 进行数据库访问等。本书还用两个案例分析来展示 XML 的具体应用。

本书适合于有一定的 C#编程经验但又想了解 XML 知识的开发人员。

Beginning C# XML：Essential XML Skills for C# Programmers
Stewart Fraser, Steven Livingstone
EISBN：1-86100-628-4

北京市版权局著作权合同登记号　图字：01-2002-5385

图书在版编目(CIP)数据

C# XML 入门经典—— C#编程人员必备的 XML 技能/ (美) 弗雷泽(Stewart Fraser), (美) 利文斯敦(Steven Living stone) 著；毛尧飞，崔伟译. — 北京：清华大学出版社，2003
书名原文：Beginning C# XML：Essential XML Skills for C# Programmers
ISBN 7-302-07467-4

I. C…　II. ①弗… ②利… ③毛… ④崔… III.①C 语言—程序设计②可扩展标记语言，XML—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 095606 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：北京牛山世兴印刷厂
装 订 者：三河市李旗庄少明装订厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：39.25　**字数**：1004 千字
版　　次：2003 年 11 月第 1 版　2003 年 11 月第 1 次印刷
书　　号：ISBN 7-302-07467-4/TP・5513
印　　数：1～4000
定　　价：78.00 元

前　言

欢迎阅读本书。近年来，可扩展标记语言(XML)已经成为应用程序开发中使用最多的专业术语。如今，Microsoft 已经将 XML 内置到其.NET Framework 核心中。本书主要是向希望了解 XML 用途的开发人员讲授 XML 知识(和相关的技术，例如 XPath、XSLT 和 XML Schema)。为了引导读者学习基本的 XML 技巧，我们假定您已经掌握了一些基本的编程技巧(这里指 C#中一些基本的编程知识)，有运用 Visual Studio .NET 的经验，可以将这些技术运用到新技术中。

在本书中，读者不仅可以很好地理解 XML 的概念，知道如何和何时在 C#中使用它，还可以掌握如何使用 XML 构建在单个台式电脑、单个 Web 服务器上运行的应用程序或分布式、多平台的 Web 服务，使用以前的技术难以实现这些功能。

为了加深您对核心概念的理解，本书列举了许多单个示例和两个案例分析。这些例子都是精心选择的，以演示 XML 的功能，帮助您在开始利用这一技术时理解其基本概念。

首先看一下基于 XML 的不同方法如何用于地址簿应用程序的开发。在本书中，项目开发的复杂度会随着读者知识的增加而加大。然后，我们用一章来讲述在实现基于 Web 的新闻门户站点的过程中 XML 和 SQL Server 数据库的运用。

0.1　本书主要内容

- XML 语法和格式良好的 XML
- 使用 XML 命名空间
- 利用 XSLT 转换 XML
- 使用 DTD 和 XML Schema 进行 XML 验证
- 使用 SOAP 和 Web 服务
- 使用 ADO.NET 为应用程序添加数据库访问功能
- 使用 SQLXML 为 SQL Server 2000 启用 XML 支持

第 1 章：在 C#中使用 XML 的原因

本章将讨论一些基本知识。什么是 XML？它为何如此重要？为何说 C#是操纵它的理想语言？更重要的是，可以用 XML 做什么？本章将介绍这些内容，阐述 XML 适合于.NET Framework 和 C#的地方，以及学习使用它的原因。另外，还将介绍第一个非常简单的应用程序。

第 2 章：XML 概述

第 1 章概述了 XML。本章将更为详细地介绍 XML 的概念、它的组成以及在创建 XML 文档时必须遵循的规则和定义。

第 3 章：在.NET 中使用 XML

了解了 XML 的基本构成部分和概念后，接下来讨论 XML 是如何适应.NET Framework 的，以及.NET 对 XML 开发提供的支持；然后论述.NET Framework 是如何在其核心部分使用 XML 的。在本章的最后，将使用本书前面学到的知识研究一个电话簿案例。后面的几章将继续构建这个电话簿应用程序。

第 4 章：在.NET 中读取 XML

本章介绍如何在.NET Framework 中使用 C#和 XML，以及如何结合两者创建强大的应用程序。讨论如何使用.NET 和 C#读取 XML，并介绍如何遍历 XML 文档，读取不同的节点类型。另外还要论述如何读取二进制数据和较大的 XML 文档，以及如何在 XML 文档中进行一些验证。

第 5 章：在.NET 中编写 XML

本章将讨论如何使用一些.NET 类定义的功能，以编程的方式在 C#代码中写入 XML 数据。.NET 开发人员很容易以最少的代码访问那些用于写元素、属性等的非常有用的方法，本章将对此作详细介绍。

第 6 章：在.NET 中实现 DOM

本章将介绍如何在.NET Framework 中实现文档对象模型(DOM)。第 3 章介绍了 DOM 的原理；这里将作更为详细的介绍。在本章中，读者将学习 DOM 的概念、它与流模型的区别，以及如何使用 XmlNode 和 XmlDocument 类。接着在案例分析中使用 DOM。

第 7 章：XPath 和.NET

本章将扩展对在.NET 中使用 DOM 的理解，讨论如何在.NET 中使用 XPath，在 XML 文档中进行复杂的导航和过滤操作。还将论述使用 XML 在.NET 中直接串行化(和反串行化)类的基本知识。

第 8 章：XSLT

本章将介绍 XSL 转换语言(XSLT)，以及.NET Framework 对它的支持。XSLT 用于接收 XML 文档，以另一种格式输出它，例如 HTML 页面或另一个 XML 文档(但并不限于这些格式)。这种转换取决于 XML 文档的内容，通过匹配文档中的元素和属性进行转换(这些元素和属性又可以由命名空间来限定)。

第 9 章：在.NET 中使用 XSLT

本章应用第 8 章中的 XSLT 知识，讨论如何使用.NET 类实现它。

第 10 章：XML 模式——背景知识、语言和一般用法

本章介绍 XML 模式语言，它是近年来 XML 家族中一个重要的新成员，将从根本上取代文档类型定义(DTD)。它可以为 XML 文档定义允许的结构和内容，在使用 XSLT 转换它之前验证 XML 文档是否为指定的结构。模式可用的范围很广，例如文档的简单验证、对商业应用程序的 XML 文档验证和 Web 服务等。

第 11 章：XML 模式和.NET

第 10 章讨论了 W3C 定义的 XML 模式语言的理论和实践部分。本章将把 XML 模式的知识扩展到.NET 中，讨论创建模式文档的编程技术，以及如何使用.NET 类根据这些文档验证 XML 实例。

第 12 章：XML 在 ADO.NET 中的用法

ADO.NET 是用于.NET 应用程序的新的数据访问模型，在许多方面取代和改进了传统的 ADO 技术。最重大的改进之处就是它与 XML 的紧密集成，这也是本章介绍的重点。我们将简单地介绍 ADO.NET 模型，通过一些示例来演示如何使用它，然后介绍如何协同使用 XML 和 ADO.NET。

第 13 章：Web 服务和 Remoting

本章将介绍 Web 服务，以及由.NET Framework 提供的基本技术的更多应用内容。本章将介绍下列内容：Web 服务、全局 XML 体系结构、简单对象访问协议(SOAP)、Web 服务描述语言(WSDL)、Microsoft 的 DISCO 发现 XML 文档、统一描述、发现和集成服务(UDDI)以及 Remoting 和 XML 配置文件。显然，这些主题的范围比较广，但 XML 是这些技术的整个底层框架的基础。在企业中，这些技术都进行了很好的定义和利用，所以 XML 的使用将更为必要，而了解其用途也将非常重要。

第 14 章：案例分析：一个简单的新闻门户网站

该案例分析演示了可以使用本书介绍的哪些技术来创建一个简单的新闻门户网站。我们使用绑定到 DataSet 的可编辑 DataGrid，简化了远程内容提供者的内容注册，还演示了在与 XmlDataDocument 同步时，如何使用 XPath 查询 DataSet。

本章演示了如何使用 XSD 模式验证内容，确保正确地结构化所提交的 XML/HTML 文件，以便用于我们的系统中。最后介绍了如何从 SQL Server 2000 数据库的查询中直接返回 XML，可以使用 XSLT 将它直接转换到浏览器中，这可以转换成任何浏览器类型，轻松改变站点的布局。

0.2 本书读者对象

本书主要帮助开发人员了解利用 XML 能实现什么功能。不过，本书不是针对初学者的，它假定读者具备下列知识：

- 一些 C#的知识——本书并不需要特别高级的 C#功能，但读者需要熟悉 C#的基本语法。
- 知道如何运用 Visual Studio .NET。

0.3 使用本书的条件

本书中的示例要用 Visual Studio .NET 专业版或标准版和 SQL Server 2000 或 MSDE 来运行，操作系统需要是 Windows 2000 或 Windows XP 专业版。

示例的完整源代码可从 http://www.wrox.com 站点上下载。

0.4 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。

0.4.1 源代码和更新

在学习本书中的示例时，您也许倾向于手动输入所有代码。许多读者都愿意这样做，主要是因为这是一种能够帮助我们熟练掌握所需编码技巧的好途径。不过，无论您是否希望手动输入所有示例代码，我们都将在 Wrox.com 站点上放置本书示例的所有源代码，以满足那些希望得到源代码的读者的要求。

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信息页面中的 Download Code 超链接，就可以下载相应的示例代码。

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，应确保解压缩程序已经选中 Extract to(或对等选项)选项并使用原有文件夹名作为解压目标文件名。

即使您喜欢手动输入示例代码，但仍然可以使用我们的源代码来检验可获得的结果——如果您想自己可能存在输入错误时，示例源代码可以帮助您验证错误，得到正确结果。如果您不喜欢手动输入示例代码，那就需要从我们的站点下载源代码。总之，源代码有利于您更新和调试示例程序。

0.4.2 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 超链接即可，该链接位于本书的详细信息页面中。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。也可以通过单击 Submit Errata 链接，通知我们您已经发现的勘误内容。

0.4.3 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的问题，可以发送电子邮件到 support@wrox.com，要求在邮件的主题栏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（本书是 6284）和问题所在的页码。
- 正文部分应包括读者的名字、联系信息和问题。

我们将不处理无用邮件，因为我们仅仅需要有用的详细资料，以便可节约您和我们的时间。当您发送一个电子邮件信息时，它将经过下面一系列支持：

- 客户支持：首先，您的信息将被递送到我们的客户支持人员手中，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)，他们将求助于本书的作者。我们将尽量使作者免受干扰，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。

0.4.4　p2p.wrox.com 站点

为了便于作者和其他人讨论，请加入到 P2P 站点的邮件列表中，除了一对一的邮件支持系统外，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、新闻组等其他服务相联系。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发现。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，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。

目　　录

第1章　在C#中使用XML的原因

Web 自出现以来，已在经历几个不同的阶段后走向成熟。不到 10 年以前，Web 页面只能显示静态的文本和图形。第一个重要的改进是将 Web 页面与后端数据库集成起来，通过基于 Web 的用户界面显示并处理 Web 页面的内容。随着 Internet Explorer 4 的引入，首次出现了 Web 服务，同时也带来了不好的名声。尽管在最初推(Push)技术上的第一次尝试有许多未完善之处，例如服务器广播“通道”，但可以只传递客户机想要的信息，而不是大量数据。最近的、也是最重要的阶段就是客户机可以在需要时找到它们自己的 Web 服务，并集成它们，而不是依赖推技术。这可以用于下载软件和服务包的最新版本，或在级别突然降低时查看自己的库存量和广播警报。

实现这一阶段的两大独立而关键的技术是 Microsoft 的.NET Framework 和 W3C 标准 Extensible Markup Language(可扩展标记语言)，即常说的 XML。.NET Framework 帮助在平台之间移植代码，使应用程序与以不同语言编写的组件顺利地结合使用，就像它们用同一种语言编写的一样。它也模糊了在自己的本地机器上运行的应用程序和可以通过 Web 访问的应用程序在编程方面的界线。XML 也有一组类似的目的，因为它是一个基于文本的标准，以结构化格式表示数据。它可以用于任何平台，由不同语言编写的不同应用程序和组件使用。由于它完全是用文本编写的，非常适合于基于 Web 的应用程序。这两者在 Microsoft 的未来 Web 开发计划中扮演着重要角色。

本章不准备详细讨论这些领域，也不深入讨论太多的新术语；只是介绍一些基本的知识。例如，什么是 XML？为何它如此重要？为何说 C#是处理它的理想工具？最重要的是，可以用 XML 做什么？本章将介绍这些内容，并了解 XML 适合于.NET Framework 和 C#之处，以及学习使用它的原因。

1.1　使用 XML 的原因

读者在阅读本书时，已经知道为何要使用 C#。.NET Framework 的最大特点就是所选择的语言对底层操作并无影响，因此也可把本书内容看作讲述 C#或 C++如何与 XML 一起使用，因为 Common Language Runtime(公共语言运行库)在底层确保这些语言在本质上是一样的。

然而，这里需要对 XML 作一些说明。XML 是一种类似于 HTML 的标记语言，本书假定读者有一些 HTML 的基本知识(如果没有，建议读者在阅读本书前先对它了解一下)。HTML 这种语言显示了可以在浏览器上查看的内容和数据，但没有对它所显示的数据作任何说明。考虑下面这个示例，这是一个 HTML 文档，它描述了本书的作者。

```
<html>
  <head>
    <title>Contacts</title>
```

```
</head>
<body>
  <b>Steven Livingstone</b> lives at 123 Anystreet, Anytown in WA. His postal
 code is 12345 and he lives in the USA.<br>
  <br>
  <b>Address of Stewart Fraser: </b>
  <br>123 Anotherstreet
  <br>Anothertown
  <br>NY
  <br>12997
  <br>USA.
 </body>
</html>
```

HMTL 标记向浏览器描述了文档将如何显示，但没有如下指示：

- 数据由两个不同的部分组成，即姓名和地址
- 列出了两个作者的信息
- 一个记录存储在一行中，而另一个记录存储在 6 行中

事实上，HTML 并不做这些工作；它只是以 Internet Explorer 或 Netscape Navigator 可以显示的方式来格式化数据，如图 1-1 所示。

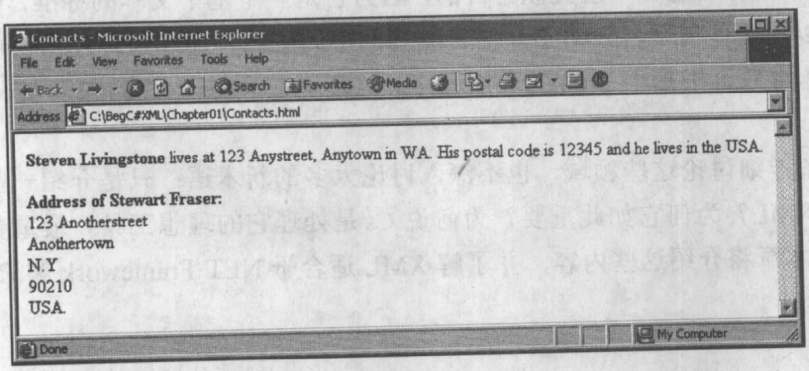

图　1-1

同样，HTML 也不能只显示 Web 页面的一部分、或过滤出页面上某些不相关的信息、或将类似的信息组合在一起。它每次都是显示整个 Web 页面。

而 XML 并不包含浏览器用于显示它的任何内容，因此在 XML 页面中，只包含标记和保存在其中的数据，不能显示图形、表或框架。XML 对数据采用了一个更明确的方法，准确声明数据表示的意思。下面的示例名为 Contacts.xml，它是前面 HTML 文档的对应 XML 文档：

```
<?xml version="1.0"?>

<contacts>
  <contact>
    <firstname>Steven</firstname>
    <lastname>Livingstone</lastname>
```