

# Visual Basic .NET

## XML Web 服务

### 开发人员指南

创建和部署具有产品级质量的 .NET XML Web 服务

涵盖工业 XML Web 服务标准、SOAP、WSDL 和 UDDI

比较数据密集型 XML Web 服务在内联网和 Internet 上的性能

Roger Jennings 著  
丛丽晖 肖刚 译



清华大学出版社  
<http://www.tup.com.cn>

Mc  
Graw  
Hill

# Visual Basic .NET XML Web 服务开发人员指南

Roger Jennings 著

丛丽晖 肖刚 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: 01-2002-4659

### 内 容 简 介

XML Web 服务是当前最重要的 Web 开发技术,它不仅具有各种常规用途,而且还可以应用到大型企业应用集成和 Web 商业项目中。.NET 为 XML Web 服务提供了强大的支持,利用其中的 Visual Basic.NET 可以非常简便快捷地创建和部署企业级 XML Web 服务。

本书从实际出发,全面介绍 XML Web 服务的技术内涵和针对 Visual Basic 的开发方法,具体内容包括 XML Web 服务的快速开发方法、XML Web 服务体系结构和核心工业标准、如何将 VB6 组件转换为 ASP.NET Web 服务、复杂 XML Web 服务的性能优势、SOAP 消息格式之间的转换、利用 SOAP 头验证和授权用户、利用 XSD 模式验证 SOAP 请求和响应文档等。

本书适合有经验的 Visual Basic 6.0 和 Visual Basic.NET 开发人员使用,同时也可作为其他 .NET 开发人员的参考书目。

Roger Jennings: Visual Basic.NET XML Web Services Developer's Guide

EISBN: 0-07-222369-3

Copyright© 2002 by McGraw-Hill, Inc.

Authorized translation from the English language edition published by McGraw-Hill, Inc.

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美国麦格劳-希尔出版集团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的任何部分。

版权所有,翻印必究。

本书封面贴有 McGraw-Hill 激光防伪标签,无标签者不得销售。

### 图书在版编目(CIP)数据

Visual Basic .NET XML Web 服务开发人员指南/(美)罗格,(美)简宁斯著;丛丽晖 肖刚 译.

—北京:清华大学出版社,2002

原书书名: Visual Basic.NET XML Web Services Developer's Guide

ISBN 7-302-05989-6

I. V... II. ①罗...②简...③丛...④肖... III. 计算机网络—BASIC 语言—程序设计 IV. TP393

中国版本图书馆 CIP 数据核字(2002) 第 079137 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: 于平

印 刷 者: 国防工业出版社印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 25.75 字数: 659 千字

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

书 号: ISBN 7-302-05989-6/TP·3571

印 数: 0001~4000

定 价: 53.00 元

# 前 言

本书全面深入地介绍了如何利用 Visual Basic .NET 设计和编写 XML Web 服务以及使用应用程序，另外，在前面几章中还涉及到一些 Visual Basic 6.0 代码。在本书中您将找不到一行 C# 代码。本书并不是一本 XML 的初级读物，其中的章节也没有专门从头开始介绍 XML 基础、编写 XSL 转换或者设计 XSD 模式。要充分讲解这些主题内容，需要比本书还要多的篇幅。本书主要教您如何创建、测试和部署实用的、具有产品质量的、数据密集的 XML Web 服务。在第 1 章结尾处的 Hello World 服务是默认的 ASP.NET 示例，它仅仅是本书中一个普通的示例。

## 注意：

本书使用了 consumer 或 consuming 应用程序或项目之类的术语，而没有使用客户程序这样的术语，是为了避免推断出 XML Web 服务坚持客户/服务器(client/server)模型的结论。尽管本书中的大部分示例服务都同 Windows 或 Web 窗体客户程序进行通信，但其中多数具有产品质量的 XML Web 服务仍有可能在服务器之间通信，且许多服务将调用补充的 Web 服务。

Visual Basic 代码的专用性并不打算再次引起编程语言之间的战争或者以各种方式贬低 C#。C# 是一门新语言，它打算吸引 C++ 和 Java 编程人员转移到 Visual Studio .NET 的托管代码或公共语言运行库(CLR)。Visual Basic 是成熟的语言，目前全世界有数百万编程人员使用 Visual Basic 6.0、VBScript 和 VBA 创建企业级应用程序和组件。C# 能够添加 XML 样式的组件和编写非安全代码，Visual Basic .NET 不支持这两个功能。除此之外，几乎所有能够利用 C# 完成的工作都可以利用 Visual Basic .NET 来处理。省略 C# 示例允许本书涉及更广泛的 XML Web 服务主题。

保守地说，在所有产品的 XML Web 服务中，至少有 95% 将同一个或多个关系数据库建立连接。那些仿效四函数(four-function)计算器、转换度量单位、或者返回每日报价(quote-of-the-day)的一般服务，并不表示商业产品。有两个服务在得到一个 ISBN 后检索其他站点，并返回书目报价，但它们也不是商业产品。XML Web 服务的早期采用者(adopter)将是一些企图减少花费并扩展企业应用程序集成(EIA)项目的生存周期的组织。而其中的 EIA 中间件并不简单。

W3C 标准的 XML、XSD 和 XSL/T，以及提供工业标准的 SOAP 1.1+ 和 WSDL 1.1+ 的采用允许我们颠倒今天的 Interop Babel 之塔，最小化 EIA 咨询费用并将 EAI 中间件的生存期扩展到项目连接的软件的下一个修订版之后。B2B 和 B2C(应用范围很小)供应链应用程序将引领 XML Web 服务移植的第 2 阶段。几乎所有 EAI、B2B 和 B2C 中间件都在一端或两端有一个数据库。

除了一个样例 XML Web 服务外，本书中的所有 Web 服务都连接一个或多个 SQL Server 或 MSDE 2000 数据库，这些数据库所拥有的基本表尺寸范围大约都在 35000 到 200000 行之间，并包含实际的或模拟的实用数据。示例应用程序允许我们将某些数据库的尺寸增加到可用磁盘空间的最大限度。通过使用大中型数据库，我们可以对比利用各种编程方法创建的 XML Web 服务的性能，例如，利用 ADO.NET 替代 MDAC 2.7 COM 组件。本书中的第 2 章到第 10 章都



强调本地和远程 XML Web 服务的性能测试。

在您的测试计算机上创建的样例 XML Web 服务有许多在 OakLeaf Systems XML Web 服务演示站点(<http://www.oakleaf.ws>)上运行的产品副本。当前, OakLeaf 站点有一个速度为 1538/727-kbps 的 ADSL 同 Internet 相连, 这样就有足够的带宽模拟利用 T-1 线路(拥有适度的通信负载)连接的站点。远程 XML Web 服务的可用性允许您利用基于 Internet 和内部网的服务论证和比较您的 SOAP consumer 应用程序的性能。为了精确地判定远程和本地 XML Web 服务方法的执行时间, 所有样例 ASP.NET consuming 应用程序都包括详细的跟踪页面。

#### 注意:

西萨摩亚(Western Samoa)就是现在的萨摩亚群岛, 拥有一个 .ws TLD, 但是很可能有许多商业 XML Web 服务提供商将采纳在 .com 领域的 .ws。

#### 本书读者对象

本书主要针对有经验的 Visual Basic 6.0 开发人员, 但并不是指 Visual Basic .NET 专家。如果您有编写和部署 Visual Basic ActiveX 组件的经验, 将有助于学习本书, 但这些经验并不是必须的。本书假设您不熟悉 VBScript、ECMAScript 或者 ASP 编程。本书的第 6 章将展示如何把传统的 Windows 窗体 XML Web 服务的付费用户代码移植到 ASP.NET Web Forms(Web 窗体)中。如果您没有使用过 ASP.NET, 您将很惊讶 Visual Studio .NET 使我们利用 ASP.NET Web Forms 仿效传统的 Windows 窗体变得如此容易。

第 3、4 和 5 章将向您展示如何利用 Visual Basic 6.0 组件使用 Microsoft SOAP Toolkit 2.0+, 然后将组件移植到 Visual Basic .NET 和 ASP.NET Web 服务项目中。Visual Studio .NET 的 Visual Basic .NET Upgrade Wizard(升级向导)能够很好地完成从优秀的 Visual Basic 6.0 代码到 Visual Basic .NET 的迁移任务。许多 Visual Basic 专家(例如 Desaware 公司的 Dan Appleman)都坚持认为, 将具有产品质量的 Visual Basic 6.0 应用程序和组件升级到 Visual Basic .NET 几乎没有什么经济上的因素。第 4 章将论证, 在您将 XML Web 服务从带有 SOAP 包装器的 ActiveX 组件转换到 ASP.NET 时, 服务的性能会有一个相当大的提高。第 5 章说明了一个更好的改进, 那就是利用 ADO.NET 替代基于 COM 的 ADODB 对象。

#### 提示:

要了解 Dan 关于“将 Visual Basic 6.0 代码移植到 Visual Basic .NET”的观点, 请看看他在 2002 年 3 月出版的 .NET Magazine 上发表的“Stop! Don't Port That Code”社论([http://www.fawcette.com/dotnetmag/2002\\_03/magazine/departments/guestop/](http://www.fawcette.com/dotnetmag/2002_03/magazine/departments/guestop/))。

#### 本书主要内容

本书分为 12 章和两个附录。为了快速帮助读者学习编写和部署具有产品质量的 XML Web 服务, 其中的章节按照一个典型的 Visual Basic 6.0 开发人员在采用 Visual Studio .NET 时所遵循的连续开发过程。

第 1 章: “XML Web 服务和 .NET Framework 简介”, 它概述了 XML Web 服务在 EIA、B2B 和 B2C 应用程序中的市场前景和所扮演的角色, 并简要说明本书的示例服务和付费用户, 还描述在 XML Web 服务的延伸过程中存在哪些重要的未来趋势。

第 2 章：“掌握 SOAP 和 WSDL”，它解释了您所需要的一些基础知识，以便了解当前的 SOAP 1.1 和 WSDL 1.1 实际工业标准，它们两个形成 XML Web 服务的基础。本章还包括一个简单的 XML Web 服务示例，该示例基于一个升级的 Visual Basic 6.0 ActiveX 组件和 Northwind SQL Server 2000 样本数据库。

第 3 章：“使用 Microsoft Soap Toolkit 2.0”，它说明如何应用 SOAP 包装器生成用于 ActiveX 组件的 WSDL 文件，其中 ActiveX 组件充当 XML Web 服务且仍保持 COM/DCOM 中间层的能力。本章还介绍如何使用 Toolkit 的 Trace Utility(跟踪实用程序)检查使用 rpc/encoded 格式的 SOAP 请求和响应消息。本章所介绍的技术也适合于 Toolkit 的 3.0 版本。

第 4 章：“升级到 ASP.NET XML Web 服务”，它阐述了用于从 Visual Basic 6.0 组件向 ASP.NET document/literal XML Web 服务迁移的升级、复制和粘贴技术。另外还介绍如何在 Windows 窗体客户程序中添加 Web Reference 及调试 XML Web 服务和它们的付费用户。

第 5 章：“从 ADO 2.5+转移到 ADO.NET”，它描述 ADO.NET 和它的 ADO DB 前身之间的差别，并展示如何使用 SqlConnection 对象来获得重大的性能提升。本章还介绍 ADO.NET 的 DataSet 和 DataTable 对象，并说明创建和使用它们所要求的 Visual Basic.NET 代码。

第 6 章：“将 XML Web 服务测试客户程序转换到 ASP.NET”，它阐述从 Windows 窗体向 Web 窗体的复制和粘贴迁移过程，而其中的窗体在后台还带有适度复杂的 Visual Basic .NET 代码。您可以在窗体上添加 ASP.NET DataGrid 服务器控件，并分析它们对付费用户性能的影响。

第 7 章：“导航 System.Web.Services 和 System.Xml 命名空间”，它全面介绍了支持 ASP.NET XML Web 服务的主要 .NET Framework 命名空间。本章还介绍如何结合以属性为中心的 XML 文档使用 SOAP 头和使用 XmlTextReader 类，其中的 XML 文档由利用 Microsoft 的 SQLXML 3.0 增件创建的 XML Web 服务返回。

第 8 章：“使用 XML Web 服务发送报告”，它着重介绍那些把以元素为中心的 XML 文档作为 SOAP 请求和响应消息来使用的 XML Web 服务。您还可以了解到如何创建从常规的 Transact-SQL 聚集查询中生成 XML 交叉表报告文档的 XML Web 服务。

第 9 章：“利用 XML Web 服务设计表示层”，为了设计出能够与那些图形艺术专家媲美的 Web 页面，本章提出 4 阶段(four-phase)的设计过程。本章的示例展示如何设计一个 ASP.NET 付费用户应用程序，该应用程序在一个完全格式化的表格中显示交叉表文档的值。另外，本章还描述如何创建 Crystal Reports XML Web 服务，目的是以图表的形式在 ASP.NET 付费用户程序中显示交叉表的值，以及如何使用 Office XP Web Services Toolkit 从返回汇总值的 XML Web 服务生成格式化的 Excel 2002 工作表。

第 10 章：“应用高级 XML Web 服务技术”，它介绍如何生成和利用 XSD 模式来利用 XmlValidatingReader 类验证 XML 请求和响应文档以及它们的元素数据类型。本章还介绍如何将 ASP.NET 的 document/literal SOAP 格式修改为 rpc/encoded 格式，且串行化复杂数据类型为以元素为中心的 XML 文档；以及介绍如何使用异步 XML Web 服务和 SOAP 扩展。

第 11 章：“用 UDDI 为公共 Web 服务做广告”，它简要介绍 Microsoft public UDDI 1.0 注册库并展示如何注册自己的 XML Web 服务，以致 Visual Studio .NET 的 Add Web References 对话框将找到它们。本章还包括 Microsoft C# 示例应用程序的 Visual Basic .NET 端口，用于程序化查找 UDDI 1.0 注册库。



第 12 章：“与第三方 Web 服务交互操作”，它说明了新生的 Web Services Interoperability (WS-I) 组织在促进 XML Web 服务互操作性(基于标准一致的配置文件)方面所扮演的角色。除了讨论 Microsoft 服务的互操作性问题，本章还讨论利用 SOAP 工具创建的服务的互操作性问题。

附录 A：“安装样本数据库”，它包含下载、连接和配置 6 个 SQL Server/MSDE 2000 样本数据库的详细指令，这些数据库支持您在学习本书时创建和修改的大部分 XML Web 服务。本附录还描述如何安装和配置可选的 CFRSQL 数据库，以支持 OakLeaf Code of Federal Regulations XML Web 服务的本地版本和使用的页面。

附录 B：“用 CFRClient 程序扩展 CFRSQL 数据库”，它解释如何使用一个允许填充 CFRSQL 数据库的本地副本的应用程序的 Visual Basic 6.0 和 .NET 版本。如果您没有下载附录 A 的 CFRSQL 样本数据库，也不影响阅读该附录。

### 示例代码下载

本书中的示例代码都可以在 McGraw-Hill/Osborne 的下载页面(<http://www.osborne.com/downloads/downloads.shtml>)中下载。将这个页面滚动到本书的 Source Code(源代码)记录项，该记录项在项目符号列表的第 2 级中有下面所示的链接：

- VBWSReqd.zip 以一个简单的存档文件包含了本书所要求的所有示例代码。VBWSReqd.zip 在驱动器的根目录下创建一组 VBWS## 和 VBWS\_OCE 文件夹(##是拥有两个阿拉伯数字的章号)。该文件的大小约为 18MB，解压缩之后约为 70MB。

### 提示：

许多 Visual Basic .NET 示例被硬编码到 D:\drive 目录，因此可以将 VBWSReqd.zip 文件和单独每章的示例文件解压缩到 D:\逻辑驱动器中，以使您的文件更加容易使用。

- VBWS##.zip 包含单独一章的示例代码的专用文档。您必须创建专用\VBWS##文件夹，该压缩文件将要扩展到这个文件夹。在下载列表中的记录项显示.zip 文件和扩展文件的大小。扩展的示例代码文件的大小在 8 到 11 671KB 之间。

- VBWS\_OCE.zip 包括 6 个数据库，以支持 OakLeaf Consumer Electronics (OCE) 示例 XML Web 服务和付费用户。您必须下载这个 10.8MB 的 .zip 文件(解压缩后大小为 42.3MB)，以获得有益于本书的重要内行编码。因为您将数据库连接到 SQL Server 和 MSDE 2000，所以扩展.mdf 的.ldf 文件的目录并不重要。附录 A 中有 SQL Server/MSDE 2000 的安装指令。

- VBWS\_CFR.zip 提供了 OakLeaf Code of Federal Regulations 数据库的“启动器(starter)”版本。如果下载并安装了这个数据库，要确保从 U.S. Government Printing Office(行政出版署)的 Electronic CFR (e-CFR) 测试站点下载 VBWSAB.zip 文件来填充这个数据库。

- VBWSAB.zip 包含填充 CFRSQL 数据库的 CFRClient Windows 窗体应用程序的 Visual Basic 6.0 和 .NET 版本。解压缩这个文件的目标子文件夹的名称和位置并不重要。这个 .zip 文件只有 850kB，解压缩之后为 3.2MB。CFRClient 项目要求大量的 Visual Basic 代码，并要求两个编写和测试方法。如果使用高速(T-1 或速度相当的连接)Internet 连接，下载和传输 U.S. Code of Federal Regulations 的所有部分内容需要大约 10 天的时间。

# 目 录

<b>第 1 章 XML Web 服务和 .NET Framework 简介</b> .....	<b>1</b>
1.1 认识 Web 服务的实质 .....	2
1.2 Web 服务体系结构 .....	3
1.2.1 普通 Web 服务 .....	3
1.2.2 XML Web 服务 .....	5
1.2.3 XML Web 服务示例 .....	7
1.2.4 XML Web 服务提供商的角色 .....	18
1.2.5 XML Web 服务的缺陷 .....	18
1.3 核心 XML Web 服务“标准” .....	19
1.3.1 简单对象访问协议(SOAP) .....	20
1.3.2 Web 服务描述语言(WSDL) .....	25
1.3.3 XML 模式定义(XSD)语言 .....	28
1.3.4 通用描述、发现和集成(UDDI) .....	30
1.3.5 W3C 安全标准: XML 签名、XML 加密和 XML 密钥管理规范(XKMS) .....	31
1.3.6 Microsoft 全球 XML Web 服务体系结构的扩展 .....	31
1.3.7 其他被提议的 XML Web 服务语言 .....	32
1.4 .NET 中的 XML Web 服务支持 .....	33
1.4.1 System.Web.Services 命名空间 .....	34
1.4.2 ASP.NET Web 服务项目 .....	34
1.4.3 ASP.NET 页面和基于 COM 的 XML Web 服务 .....	36
<b>第 2 章 掌握 SOAP 和 WSDL</b> .....	<b>40</b>
2.1 比较 SOAP/WSDL 和 DCOM 的实现方法 .....	40
2.1.1 使用 DCOM 的 Visual Basic 6.0 ActiveX DLL .....	41
2.1.2 使用 SOAP Toolkit 2.0 的 Visual Basic 6.0 ActiveX DLL .....	42
2.1.3 使用 .NET Client Web 引用的 ASP.NET Web 服务 .....	43
2.2 交换 SOAP 消息 .....	44
2.3 利用 RPC 样式的消息调用方法 .....	48
2.3.1 发送 RPC 请求 .....	49
2.3.2 处理 RPC 响应消息 .....	49
2.4 处理 SOAP 故障 .....	52
2.5 研究 WSDL 文件 .....	53
2.5.1 分析 GetOrder.wsdl 文件 .....	56





2.5.2	评审 SOAP Toolkit 自动生成的 WSMML 文件	58
2.6	使用 Header 元素扩充 SOAP 消息	59
2.7	预览 GetNwindOrder ASP.NET Web 服务	60
2.7.1	完成并测试 Web 服务	64
2.7.2	分析 ASP.NET 的 WSDL 文档	66
<b>第 3 章</b>	<b>使用 Microsoft SOAP Toolkit 2.0</b>	<b>71</b>
3.1	分析一个典型的 B2C/B2B 项目的工作流	72
3.1.1	订单处理	72
3.1.2	运送及开发货单	78
3.2	使用 OCE_Orders 项目的测试客户程序	91
3.2.1	OakLeaf 客户电子订单测试程序	91
3.2.2	继续订单处理	92
3.3	安装 Microsoft SOAP Toolkit 2.0 并运行示例代码	93
3.3.1	指派一个 IIS 虚拟目录和服务名	93
3.3.2	运行 Visual Basic Calc RPC 应用程序	94
3.3.3	分析 SOAP 客户代码	95
3.4	为 Web 服务创建 ActiveX DLL	96
3.5	利用 SOAP Toolkit 2.0 Wizard 使 ActiveX 组件受到 SOAP 支持	99
3.5.1	为 OCE_OrdersAX 服务添加虚拟目录	99
3.5.2	使用 Wizard 创建 WSDL 文件	99
3.5.3	修复 Wizard 创建的 WSDL 文件	102
3.6	用 SoapClient 对象调用 XML Web 服务	105
3.6.1	后期绑定 SoapClient 对象	105
3.6.2	处理运行时错误	106
3.6.3	用 OCE 测试客户程序验证服务	107
3.7	使用 Trace Utility 调试消息	108
3.8	将 XML Web 服务部署到一个产品服务器	110
<b>第 4 章</b>	<b>升级到 ASP.NET XML Web 服务</b>	<b>111</b>
4.1	升级过程综述	112
4.2	将 XML Web 服务客户程序升级到 Visual Basic .NET 中	113
4.2.1	下载本章的示例文件	113
4.2.2	从后期绑定向前期绑定迁移	113
4.2.3	使用 Upgrade Wizard 创建 OCE_TestClient	114
4.2.4	纠正构建和运行时错误	115
4.2.5	使程序集名称和命名空间相一致	116
4.3	将 ActiveX DLL 升级为 Visual Basic .NET 组件	117
4.3.1	创建包含升级 ActiveX DLL 的解决方案	117

4.3.2	在组件中消除 ETC	118
4.3.3	改变对升级组件的客户引用	120
4.4	在引入的组件代码中创建 ASP.NET Web 服务	121
4.4.1	添加 ASP.NET XML Web 服务并复制方法代码	122
4.4.2	检查 .asmx 文件的 WSDL 文档	124
4.4.3	添加 XML Web 服务描述	129
4.4.4	用帮助页面测试 XML Web 服务	131
4.4.5	将 ASP.NET 项目文件移动或复制到一个新的位置	132
4.5	将 Visual Basic .NET 客户应用程序迁移到 Web Reference	133
4.5.1	为本地 ASP.NET 服务添加 Web 引用	134
4.5.2	为远程 XML Web 服务建立 Web 引用	136
4.5.3	用 Web 引用对象替换 SoapClient	137
4.6	调试本地及远程 ASP.NET XML Web 服务	140
4.6.1	远程 ASP.NET 服务调试	141
4.6.2	关闭对远程 XML Web 服务的调试	142
4.7	处理与 SOAP 相关的错误	142
4.8	利用 SoapClient 对象使用 ASP.NET XML Web 服务	143
4.9	评估升级的 .NET 客户程序和 XML Web 服务的性能	143
<b>第 5 章</b>	<b>从 ADO 2.5+转移到 ADO.NET</b>	<b>146</b>
5.1	从 ADO 视点观察 ADO.NET	146
5.1.1	托管提供者	147
5.1.2	数据命令和数据读取器	148
5.1.3	DataAdapter、DataSet 和 DataTable 对象	148
5.2	使用 SqlConnection 对象	149
5.2.1	用 SqlConnection 和 SqlCommand 对象替换 ADODB.Connection	149
5.2.2	执行 SqlCommands	150
5.2.3	托管事务处理	150
5.2.4	使用 SqlDataReader 对象	151
5.3	使用并测试 ADO.NET SqlConnection、SqlCommand 以及 SqlTransaction 对象	153
5.3.1	下载并安装本章的示例代码	154
5.3.2	将 ADODB.Connection 对象替换为 SqlConnection	155
5.3.3	用 SqlCommand 代替 ADODB.Command.Execute 方法和标量 Recordset	156
5.3.4	用 Connection 对象传递 Transaction 对象	158
5.3.5	从 Firehose 游标移动到 SqlDataReader 对象	160
5.3.6	在相同的 SqlConnection 上调用 ExecuteReader 和 ExecuteNonQuery 或者 ExecuteScalar	162
5.4	用 DataTable 对象替代 Recordset 对象	164
5.4.1	用 SQL DataAdapter 填充 DataTable 对象	164



- 5.4.2 通过代码从 DataRowCollection 检索值 .....165
- 5.4.3 用 DataTable 对象替代 GetDistInventory 方法的 rstBrand Recordset .....166
- 5.5 将 XML Web 服务迁移到 ADO.NET ..... 168
  - 5.5.1 OmegaBank .NET 组件和相关的 Web 服务 .....168
  - 5.5.2 分销商.NET 组件和 XML Web 服务 .....175
- 5.6 比较 ADO 和 ADO.NET 组件及服务的性能 ..... 179
- 第 6 章 将 XML Web 服务测试客户程序转换到 ASP.NET ..... 181**
  - 6.1 使用新的 ASP.NET Web 窗体仿真 Windows 窗体 ..... 182
  - 6.2 粘贴和修复 Web 窗体的 Visual Basic.NET 后台代码 ..... 185
  - 6.3 维护 HTTP 请求之间的可变状态 ..... 187
    - 6.3.1 添加所需的 TextBox 控件 .....187
    - 6.3.2 重新编写测试客户程序代码 .....188
  - 6.4 分析与 Web 有关的行为和性能的变化 ..... 189
    - 6.4.1 比较逐步订单处理和添加一个订单的行为 .....189
    - 6.4.2 评估 ASP.NET 的智能导航功能 .....190
    - 6.4.3 将特定浏览器设置为目标 .....190
    - 6.4.4 在页面级跟踪执行情况 .....191
    - 6.4.5 添加 Trace.Write 语句以识别造成瓶颈的问题 .....193
    - 6.4.6 在应用程序级跟踪 .....195
  - 6.5 添加 DataGrid 服务器控件显示产品项和订单信息 ..... 197
    - 6.5.1 在 DataGrid 服务器控件中显示订单产品项 .....197
    - 6.5.2 滚动 DataTable 来显示订单信息 .....200
  - 6.6 测试填充 DataGrids 对客户程序性能的影响 ..... 202
- 第 7 章 导航 System.Web.Services 和 System.Xml 命名空间 ..... 204**
  - 7.1 探讨 System.Web.Services 命名空间 ..... 204
    - 7.1.1 System.Web.Services 类 .....204
    - 7.1.2 System.Web.Services.Protocols 类 .....206
    - 7.1.3 System.Web.Services.Configuration 类 .....208
    - 7.1.4 System.Web.Services.Description 类 .....209
    - 7.1.5 System.Web.Services.Discovery 类 .....209
  - 7.2 添加和处理 SOAP 头 ..... 209
    - 7.2.1 在 XML Web 服务中实现 SOAP 头 .....210
    - 7.2.2 从 WSDL 文档删除 HTTP GET 和 POST 协议 .....212
    - 7.2.3 使用 ASP.NET 客户程序测试初始 SOAP 头的修改 .....212
    - 7.2.4 修改 XML Web 服务以便处理头值 .....214
  - 7.3 遍历 System.Xml 命名空间 ..... 215
    - 7.3.1 System.Xml 成员类 .....215

7.3.2	System.Xml.Serialization 类	216
7.3.3	其他 System.Xml 类	217
7.3.4	SoapFormatter 类	217
7.4	使用 Microsoft SQLXML 3.0 创建 XML Web 服务	218
7.4.1	添加用于 XML Web 服务的存储过程	219
7.4.2	设置虚拟目录	219
7.4.3	运行 SQLXML3 Windows 窗体项目	222
7.4.4	使用 XmlTextReader 处理以属性为中心的 XML 数据文档	223
<b>第 8 章</b>	<b>使用 XML Web 服务发送报告</b>	<b>227</b>
8.1	创建业务智能报告	228
8.1.1	模拟历史订单或发票数据	228
8.1.2	构建 XML BI 报告文档	233
8.2	开发提供数据报告的 XML Web 服务	237
8.2.1	指定 SOAP 头	237
8.2.2	设计 XML 请求文档	238
8.2.3	分析 Windows 窗体测试工具要求	239
8.3	剖析 CTHarness 交叉表报告项目	239
8.3.1	读取请求文档	240
8.3.2	生成汇总查询	241
8.3.3	创建和填充中间交叉表数组	244
8.3.4	生成 XML 响应文档	247
8.4	将测试工具的代码迁移到 XML Web 服务	249
8.4.1	指派 WebMethod 和启用跟踪	250
8.4.2	将 Web Reference 添加到测试工具并测试性能	251
<b>第 9 章</b>	<b>利用 XML Web 服务设计表示层</b>	<b>253</b>
9.1	创建用于 XML Web 服务的产品级 ASP.NET 页面	253
9.1.1	第一阶段: 使用 XML Web 服务的最初页面布局和代码	254
9.1.2	第二阶段: 显示响应文档	255
9.1.3	第三阶段: 添加格式和导航功能	255
9.1.4	第四阶段: 修改最终的页面设计并调整 Netscape 浏览器	256
9.2	将 XML 交叉表报告转换为格式化的表	256
9.2.1	第一阶段: 生成并发送请求文档	257
9.2.2	第二阶段: 使用 XmlTextReader 对象填充表	259
9.2.3	第三阶段: 格式化表和添加属性值	261
9.2.4	第四阶段: 使有效起始日期同日历的边界对齐	264
9.3	深入研究 DHTML 导航	266
9.3.1	客户端导航脚本	267



9.3.2	OpenTOC.aspx 页面的后台 Visual Basic .NET 代码	270
9.3.3	GetTocById 和 GetTocByLevel Web 方法的 Visual Basic .NET 代码	272
9.4	使用 Crystal Reports 创建 XML Web 服务	276
9.4.1	创建 vw2001OrdersByQuarter 视图	276
9.4.2	将交叉表报告添加到 Windows 窗体	277
9.4.3	从 Crystal Report 生成 XML Web 服务	280
9.4.4	设计使用 Crystal Reports 服务的 ASP.NET 页面	281
9.5	利用 Excel 2002 使用 Office XP Web Services Toolkit	282
9.5.1	分析 Excel 2002 示例应用程序	283
9.5.2	修改 ReportsWS 服务	284
9.5.3	在 Excel 工作表中显示 XML Web 服务交叉表报告	285
<b>第 10 章</b>	<b>应用高级 XML Web 服务技术</b>	<b>290</b>
10.1	使用 XSD 模式验证 XML 请求和响应消息	290
10.1.1	使用 Xsd.exe 生成模式	291
10.1.2	消除嵌套关系错误	292
10.1.3	强制请求文档元素	293
10.1.4	编写 XmlValidatingReader 对象	295
10.2	使用 Visual Studio .NET 的 XML Designer	300
10.2.1	Schema 视图	300
10.2.2	Data 视图	302
10.2.3	XML 视图	303
10.3	在 XML 请求文档中指定 XSD 数据类型	303
10.3.1	创建强类型模式	304
10.3.2	修改客户应用程序以处理类型化的模式	306
10.3.3	调整 ReportsWS XML Web 服务	308
10.4	在 XML Web 服务中进行元素数据类型检查	308
10.4.1	将客户模式验证复制到新方法中	309
10.4.2	用产品版本替换第 8 章的 CTReportsWS 服务	310
10.4.3	验证 XML Web 服务有效性过程	311
10.5	用 rpc/encoded 格式替代 document/literal SOAP 消息格式	311
10.5.1	创建 document/encoded Web 服务的副本	312
10.5.2	添加 SoapRpcMethod 属性	313
10.5.3	使用客户程序的付费用户测试复制的 Web 服务	315
10.6	串行化复杂数据类型	315
10.6.1	从现有文档定义类	316
10.6.2	串行化类实例	317
10.6.3	将 XmlRootAttribute 添加到类声明	319

10.6.4	输入请求和响应类	320
10.6.5	使用 Xsd.exe 创建类声明	321
10.6.6	使用 DataSet 的强类型化服务	322
10.7	处理无效的或慢速 Web 服务	323
10.7.1	在同步或异步处理之间决定	323
10.7.2	采用向下的服务战略	324
10.7.3	处理同步超时	324
10.7.4	诱发延迟以模拟慢速或非响应服务	325
10.8	异步使用 XML Web 服务	326
10.9	添加 SOAP 扩展	327
10.9.1	回顾 Visual Studio .NET 的 TraceExtension 示例	328
10.9.2	使 TraceExtension 示例具有可操作性	328
10.9.3	启用消息日志和读取日志文件	332
10.10	高速缓存 XML Web 服务	335
<b>第 11 章</b>	<b>用 UDDI 为公共 Web 服务做广告</b>	<b>336</b>
11.1	浏览 Microsoft UDDI 1.0 注册库站点	337
11.1.1	业务实体页面	337
11.1.2	业务服务页面	343
11.1.3	实例细节和 tModel 页面	346
11.2	在操作员站点注册服务	348
11.2.1	添加一个业务实体	349
11.2.2	指定业务以及地理分类	350
11.2.3	注册服务的 tModel	351
11.2.4	添加业务服务	353
11.3	在 UDDI 注册库中添加 Web 引用	355
11.4	审查 UDDI 规范	357
11.4.1	UDDI 1.0 和 2.0 通用规范	357
11.4.2	UDDI 2.0 中增加的规范	360
11.5	使用 UDDI .Net SDK	360
11.5.1	UDDI .NET SDK 1.76 Beta 版本	361
11.5.2	UDDI .NET SDK 2.0 Beta 1 版本	364
<b>第 12 章</b>	<b>与第三方 Web 服务互操作</b>	<b>365</b>
12.1	使 Web 服务互操作性标准化	366
12.1.1	Web 服务互操作性组织	367
12.1.2	SOAP 创建者组织	368
12.2	查找用于 .NET 互操作性测试的 Web 服务	369
12.2.1	XMethods 站点	369



12.2.2	SalCentral Web 服务中间商	370
12.3	利用 Windows 窗体付费用户测试互操作性	370
12.4	利用标准模式	373
12.4.1	可扩展的业务报表语言	374
12.4.2	电子商务 XML (ebXML)	376
12.4.3	XML.org 的模式注册库	376
12.4.4	BizTalk.org 的模式注册库	377
12.5	探讨商业 ASP.NET Web 服务	377
12.6	规划 XMLWeb 服务的未来	378
<b>附录 A</b>	<b>安装样本数据库</b>	<b>380</b>
A.1	下载和安装 OakLeaf Consumer Electronics 数据库	380
A.1.1	添加 SQL Server 登录和数据库 Users	381
A.1.2	验证数据库的正确安装	383
A.2	安装 CFRSQL 数据库	384
A.2.1	下载及连接数据库	385
A.2.2	建立全文本搜索	385
<b>附录 B</b>	<b>用 CFRClient 扩展 CFRSQL 数据库</b>	<b>387</b>
B.1	安装 CFRClient 项目和解决方案	388
B.2	运行 CFRClient 程序	389
B.2.1	把部件记录添加到数据库	390
B.2.2	将区域文本添加到数据库	390
B.2.3	生成内容表	393
B.3	客户仿真程序表单的使用	394
B.3.1	显示区域文档	395
B.3.2	显示内容表	396
B.3.3	执行全文本检索	396

# 第1章 XML Web服务和.NET Framework简介

本章主要内容:

- 认识 Web 服务的实质
- Web 服务体系结构
- 核心 XML Web 服务“标准”
- .NET 中的 XML Web 服务支持

目前没有哪一周不是在准备铺天盖地的新闻稿、联机文章和书面文章,以及“Web 服务革命”的白皮书等工作中度过的。信息技术的权威人士以及计算机专栏作家迫不及待地宣称 Web 服务已成为计算技术的“下一次重大突破”。例如, Gartner Group(一家业界领先的技术分析和咨询公司)认为 Web 服务是“未来 10 年中即将出现的 4 项关键技术”之一。下面的一段内容摘录于 Gartner 2001 年 10 月的出版物。

通过把业务处理完整地包装为软件组件, Web 服务推动目前仍然处于发展阶段的电子商务蓬勃发展。Web 服务极大地便利了软件开发和集成工作,还使企业变得更灵活,并使得企业集中精力关注其核心业务,而无须关心其他事物。Web 服务很可能是 2001 年和 2002 年度最热门的技术潮流,但我们仍然可能没有正确全面地认识到其重要意义。

注意:

登录 <http://www3.gartner.com/>并搜索“Web Services”关键字,即可得到大量 Gartner 公司的 Web 服务的报告和新闻稿。

“把业务处理包装为软件组件”不是新理论。从 1995 年起 Visual Basic 开发人员就开始把业务处理包装为业务逻辑组件,那时 Microsoft 公司在 Visual Basic 4.0 中添加了创建 COM DLL 的功能。Microsoft Transaction Server 可以使多个 COM 组件在分布式数据库事务处理中注册,这就实现了 3 层应用程序体系结构,并且使 Visual Basic 开发人员可以编写企业级应用程序。活动服务器页面(Active Server Pages)和服务器端的 VBScript 或 ECMAScript 把表示层传递给基于浏览器的“瘦”客户机,该客户机与中间层的业务处理组件相连接。基于 Internet 的应用程序只要具有下面情况都可以被认为是一般意义上的 Web 服务:能够利用面向对象的体系结构,并使用 HTTP 或者其他标准 Internet 协议与另外的服务器进行通信,或者可以把表示层传递给客户机。基于 Web 的合作模式和空闲计算机资源的对等共享也可以被认为是基于 Web 的服务。这样看来,普通 Web 服务不能算是一项新技术。

现在,很多介绍 Web 服务的文章提到 Web 服务时总是缺少关键性前缀。作者和分析家们滔滔不绝地议论的技术实际上是 XML Web 服务。工业标准的可扩展标记语言(Extensible Markup Language, XML)文档,简单对象访问协议(Simple Object Access Protocol, SOAP)以及 Web 服务描述语言(Web Services Description Language, WSDL)文件将 XML Web 服务与它们





的前身区别开来——这些前身是特定于操作系统的，受协议限制并依赖于语言。本书主要讲述在 Windows 2000 下运行的 XML Web 服务的知识，适用于 Visual Basic 6.0 和当前正在推出的 Visual Basic .NET 和 ASP.NET 开发人员。在本书的学习过程中所创建并测试的 XML Web 服务对于符合下面条件的客户机或服务器来说都是可访问的：能够通过 HTTP、HTTPS 或者 SMTP 协议通信，并且可以处理 WSDL 文件中的 XML 内容。

## 1.1 认识 Web 服务的实质

在许多出版物和分析性文章中，甚至包括 Microsoft 公司自己关于 .NET 的宣传材料中都把电子商务突出为推动 [XML] Web 服务开发的动力。我们无法精确估算究竟那些已经破产的 “.com” 公司已经将多少亿美元投入到 business-to-consumer(企业对用户, B2C)模式的电子商务无底洞中。到目前为止，business-to-business(企业对企业, B2B)电子商务的参与者们，例如专用于某种商品和某个行业的商务网(以及它们的软件供应商)的经营状况并不好于他们的 B2C 先驱们。今天 IT 管理层采用 XML Web 服务时所遇到的最严重的障碍恰恰就是早期电子商务失败的阴影。

企业(应用程序)集成是 XML Web 服务初始部署的关键。当前，企业集成的应用范围比电子商务应用程序更广泛。企业集成的目标是实现所有类型和规模的分布式计算系统的互操作性，而不仅仅限于 B2C 或者 B2B 项目。分布式系统主要涉及驻留在大型机、UNIX 系统以及 Windows NT/2000 服务器中的数据库之间的交互。XML Web 服务指引了实现数据库交互操作以及软件复用目标的发展方向，并保证软件开发项目的投资回收率(ROI)大幅度增长。

ARC Advisory Services 是一家战略规划和技术评估公司。其在截止至 2006 年的中期预测中，估计在 2000 年企业集成软件和服务销售额将达到 39 亿美元，在 2001 年达到 48 亿美元，到 2006 年将超过 110 亿美元。在 2001 年底至 2002 年初期间，企业集成销售额的迅速增长与全部信息技术开支的减少形成了鲜明的对比。ARC 公司在 2001 年 10 月的出版物中声称：

Web 服务受到了基于 J2EE 的供应商和拥有 .NET 平台的 Microsoft 公司的大力支持。Web 服务将为不同种类的平台提供一个标准的通信接口，非常适合于基于 Internet 的应用程序和企业集成。

在 2002 年 3 月，Gartner Group 预计 Web 服务的全部销售额在 2006 年将达到 280 亿美元。如果 ARC 公司和 Gartner Group 估计结果都是正确的话，那么 B2B 和 B2C Web 服务在 2006 年的市场份额将超过 170 亿美元。众所周知，新技术的预测销售额准确度很低，但是到 2004 年，B2C，乃至 B2B 服务，将可能占据 Web 服务总体 IT 开支中最大的份额。

### 提示：

如果您(或者更重要的是您的老板)对《哈佛商业周刊》(Harvard Business Review)上的商业管理文章的信任程度就像对《新英格兰医药周刊》(New England Journal of Medicine)上的健康保健论文的信任程度相同，您就可以花费 6 美元从 [http://www.hbsp.harvard.edu/hbsp/prod\\_detail.asp?R0109G](http://www.hbsp.harvard.edu/hbsp/prod_detail.asp?R0109G) 下载由 John Hagel III 和 John Seeley Brown 于 2001 年 10 月撰写的《Your Next IT Strategy》一文的 PDF 版本。该文对于公司和 IT 管理都具有指导意义，而不仅