

内 容 简 介

本书主要讲解了.NET 框架下 Web 服务的开发和使用,其中包括基于 ASP.NET 框架的、基于远程处理框架的和基于 ATL Server 的 XML Web 服务的开发和使用三部分重要内容。且在每一部分内容中,都讲到了 XML Web 服务的开发步骤、支持的数据类型,以及使用 SOAP 标头、安全性等高级特性。随书附赠的光盘包含了书中全部精彩实例源代码。

本书适用于具有相关知识基础并至少了解一门.NET 平台下的编程语言的.NET Web 服务开发的中、高级用户。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

.NET Web 服务开发指南 / 刘晓华编著. —北京: 电子工业出版社, 2002.10

(开发专家之 MS.NET)

ISBN 7-5053-8045-1

I .N... II.刘... III.计算机网络—程序设计 IV.TP393.09

中国版本图书馆 CIP 数据核字(2002)第 076218 号

责任编辑:王树伟 王 蒙

印 刷:北京兴华印刷厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京海淀区万寿路 173 信箱 邮编:100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:25.375 字数:649.6 千字

版 次:2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

印 数:4000 册 定价:39.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。若书店售缺,请与本社发行部联系调换。电话 68279077

出版说明

“开发专家”是电子工业出版社计算机研发部长期以来精心培育的计算机科学技术类本版品牌。这个品牌是由多个专题系列组成的横向大系列，涵盖了计算机技术的各个方面，特别是一直受到极大关注的程序开发类系列，例如《开发专家之数据库》、《开发专家之网络编程》、《开发专家之 Delphi》、《开发专家之 Sun ONE》，以及《开发专家之 MS.NET》等。这些专题系列基于各自的角度，从纵向上包含了该专题的所有内容。因此，整个“开发专家”的品牌架构纵横交错，囊括了所有的计算机技术和所有的技术层面，海纳百川而又极具可扩展性。

“开发专家”的作者队伍主要依托于“飞思科技产品研发中心”。“飞思科技产品研发中心”是由专业的策划人员、权威的技术专家和精深的作者队伍共同构成。在图书的出版上，形成了以研发为基础、以出版为中心、以服务为支持的专业化出版框架和流程。通过深入的市场调查和技术跟踪，在综合了技术需求和读者焦点等因素的基础上，形成各系列丛书的写作重点和大纲，然后聘请业界的最前沿学者进行写作。同时，策划工作全程介入写作进程，严格控制写作质量，用最专业的技术背景、最深刻的理论基础、最具代表性的案例、最能为专业读者接受的形式，为读者提供品质最佳的图书产品。这体现了出版者和著作者的完美结合。

多年来，计算机研发部始终把创造社会效益摆在首位，秉承一切为国内计算机技术专业读者服务的精神，为推动国内 IT 技术发展、为体现国内技术的原创水平，穷尽所有的创意与努力，将出版者的命运与读者的支持紧紧地连在了一起。

在此，我们临出版之残酷竞争而不惧，旌旗猎猎而异军突起，这与广大读者的支持是分不开的。为了使我们的脚步更坚实、使我们的队伍永葆活力和创造力，我们期待着您能为我们的前进贡献出您的意见和建议。同时，我们也在等待着您的加入。

我们的联系方式：

电 话：(010) 68134545 68131648

电子邮件：support@fecit.com.cn

飞思在线：<http://www.fecit.com.cn> <http://www.fecit.net>

勘误网址：<http://www.fecit.com.cn/question.htm>

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

电子工业出版社计算机研发部

关于本丛书

对开发人员而言，.NET 是一个完美的开发平台。它提供了一套公共的运行库，并制定了一套公共语言规范，所有符合该规范的语言都可以无缝使用这套运行库。在.NET 平台下，除了语法上的区别，各种编程语言没有本质的不同。它们共享公共类库，具有类似的编程模型和相差无几的功能。开发人员可以自由选择自己喜爱的语言开发程序。.NET 平台提供大量的服务，包括垃圾自动收集、面向对象的多线程、基于程序集的部署、异常处理、特性编程、远程处理、ASP.NET 网页框架、互操作、安全性等，使开发人员可以快速构架任何应用，从包括传统的桌面应用到面向 Web 的大型分布式应用。.NET 将彻底改变软件的开发方式、使用方式和发行方式，.NET 将是一场软件革命。

Microsoft .NET 是一个用于构建、运行和体验下一代分布式应用程序的平台。它是跨客户端、跨服务器的开发人员工具，它由以下部分组成：

- .NET 框架编程模型，它使开发人员能够构建 Web 应用程序、智能客户端应用程序和可扩展标记语言 (XML) Web 服务应用程序，它们利用诸如 SOAP、可扩展标记语言 (XML) 和 HTTP 等标准协议以编程方式通过网络开放其功能。
- 开发人员工具，比如 Visual Studio.NET，该工具为用 .NET 框架进行编程提供了一个迅速集成开发应用程序的环境。
- 一组服务器，包括 Windows 2000、SQL Server 和 BizTalkServer，它可集成、运行、操作和管理 XML Web 服务和应用程序。
- 客户端软件，比如 Windows XP、Windows CE 和 Microsoft Office XP，它有助于开发人员跨多种设备和现有产品，为用户提供深切、引人的使用感受。

《开发专家之 MS.NET》系列丛书针对 Microsoft .NET 整个开发体系和特色组织作者进行了编写，它涵盖了 MS.NET 中的方方面面，为那些希望学习 MS.NET 技术的开发人员展示了非常完整的知识体系。在利用 MS.NET 进行项目开发的过程中，本丛书也可以为各种不同需要提供完美的解答。

关于本书

IT 技术发展一日千里，现在 Web 服务或称为 XML Web 服务作为下一代智能网络的关键技术日益引起人们的关注。它提供在可缩放的、松耦合的和非特定平台的环境下交换信息的能力，信息交换使用诸如 HTTP、XML、XSD、SOAP 和 WSDL 之类的标准协议。使用文本格式（如 XML）的灵活性使信息交换能够以松耦合的方式随着时间而发展。由

于 Web 服务基于标准协议且不特定于平台，因此可以与多种多样的实现、平台和设备进行通信。

Web 服务将改变未来软件的开发和使用模式。通过集成来自不同供应商的 Web 服务，程序设计师可以快速构建新一代应用。软件供应商也乐意以 Web 服务的形式直接在 Web 上发布它们的软件，因为这样可能意味着无限的潜在用户，并且减少软件营销成本。

本书读者对象

本书面向 .NET Web 服务开发的中、高级用户，要求读者有相关基础知识，并且至少了解一门 .NET 平台下的编程语言，如 C# 或 Visual Basic。

本书内容

本书主要讲解了 .NET 框架下 Web 服务的开发和使用，其中包括如下 3 部分重要内容：

- 基于 ASP.NET 框架的 XML Web 服务开发和使用。
- 基于远程处理框架的 XML Web 服务的开发和使用。
- 基于 ATL Server 的 XML Web 服务的开发和使用。

且在每一部分内容中，都讲到了 XML Web 服务的开发步骤、支持的数据类型，以及使用 SOAP 标头、安全性等高级特性。

本书特点

跟同类书相比，本书有如下特色：

（1）内容全面

本书囊括了在 .NET 平台下开发 Web 服务的所有技术：包括基于 ASP.NET 框架的 Web 服务开发、基于远程处理框架的 Web 服务开发和基于 ATL Server 的 Web 服务开发。

（2）实践性强

本书是笔者本人研究 XML Web 服务的经验总结。一路走来，深知学习一门尖端技术的艰辛。为了使读者能用最少的代价掌握这门最新最有潜力的技术，笔者毫无保留的奉献了学习过程中编写的全部实例，并且详细描述了每个实例的实现步骤，并对其中的难点进行了深度剖析。只要动手复现一遍全书的例子，你将跻身 XML Web 服务开发高手行列。

（3）重点突出

实施安全性是 Web 服务开发过程的一个难点，也是影响 Web 服务能否实用的一个关键因素。对 ASP.NET Web 服务安全性的彻底分析是本书的一个特色。

学习建议

在学习本书之前，读者最好对 .NET 应用开发语言，C#、Visual Basic、C++ 中的一种有所了解。因为篇幅所限，本书不可能对编程语言本身作过多的阐述。有关编程语言的书，可以参考我社出版的《C#编程指南》、《Visual Basic.NET 编程指南》或《Visual C++.NET 编程指南》。

对于 XML Web 服务的初学者,请一定确保完全掌握了本书前 6 章的内容。只要跟随书中的讲解,动手实现几个例子,掌握这部分应该没有问题。

第 7 章~第 13 章介绍了 ASP.NET Web 服务开发中的比较高级一些的话题。每章的内容都是比较独立的,读者可以根据自己的需要选读其中的一部分。不过如果要理解基于 SOAP 的自定义安全策略,则要先掌握第 8 章中有关 SOAP 扩展的知识。

对于远程处理开发感兴趣的读者,可以在阅读第 1 章后直接跳到第 14 章,碰到疑难时,可以回头翻阅关于 ASP.NET Web 服务的章节,以对比印证。

第 15、16 章是为 C++ 程序员写的。建议 C++ 程序员在粗略阅读前面的章节后再阅读第 15 章,因为稍加修改,就可以把前面所讲的技术用 C++ 托管扩展实现。第 16 章仅向 C++ 开发人员提供了用 ATL Server 开发 Web 服务的最基本的内容。如果希望更多地了解这方面的知识,可以参考本社的《Visual C++.NET 编程指南》一书。

本书由飞思科技产品研发中心策划并组织编写,由刘晓华主笔,同时参与写作的还有杜广达、李欣、吴大雷、黄科、张翼彬、薛德军、罗霄、牛海霞、徐方坡、朱凯、罗强、周卫、熊薇、曹彬、刘翔、李景彬、黎昌杰、陈海涛、张鹏、杨飞等。由于编者水平有限,再加上时间仓促,虽然经过再三勘误,但仍难免有纰漏之处,欢迎广大读者予以批评和指正。

我们的联系方式:

电 话:(010) 68134545 68131648

电子邮件: support@fecit.com.cn

飞思在线: <http://www.fecit.com.cn> <http://www.fecit.net>

勘误网址: <http://www.fecit.com.cn/question.htm>

通用网址: 计算机图书、飞思、飞思教育、飞思科技、FECIT

飞思科技产品研发中心

目 录

第 1 章 理解 XML Web 服务.....1	4.2 使用 HTTP-POST 方法 访问.....50
1.1 什么是 XML Web 服务.....1	4.3 创建和使用代理.....54
1.2 XML Web 服务的优势.....1	4.3.1 产生代理类.....55
1.3 XML Web 服务的用途.....3	4.3.2 考察产生的代理类.....56
1.4 XML Web 服务中的 主要协议.....4	4.3.3 使用 HTTP-GET/HTTP- POST 代理访问 Web 服务..62
1.5 XML Web 服务的实现.....6	4.4 小结.....64
1.6 XML Web 服务的使用.....8	第 5 章 用 SOAP 访问 ASP.NET Web 服务.....65
1.7 小结.....8	5.1 关于 SOAP.....65
第 2 章 在集成开发环境下开发 第一个 ASP.NET Web 服务.....9	5.2 产生代理类.....67
2.1 开始之前.....9	5.3 使用代理类.....72
2.2 新建 ASP.NET Web 服务 项目.....10	5.4 实例.....72
2.3 查看 ASP.NET Web 服务 项目的内容.....12	5.5 小结.....74
2.4 实现 ASP.NET Web 服务.....15	第 6 章 ASP.NET Web 服务中的数据 封送.....75
2.5 编译.....23	6.1 基本原理.....75
2.6 测试.....26	6.2 支持的数据类型.....80
2.7 查看服务描述.....28	6.3 定制数据封送.....95
2.8 部署.....29	6.4 小结.....101
2.9 提供发现机制.....32	第 7 章 使用 SOAP 标头.....103
2.10 小结.....33	7.1 在服务器端定义和处置 SOAP 标头.....103
第 3 章 用 .NET 框架 SDK 开发 ASP.NET Web 服务.....35	7.2 生成处理 SOAP 标头的 客户端.....106
3.1 准备工作.....35	7.3 更改 SOAP 标头接收方.....109
3.2 编写 ASP.NET Web 服务.....36	7.4 使用可选的 SOAP 标头.....110
3.3 测试.....37	7.5 处理未知的 SOAP 标头.....112
3.4 动态编译.....39	7.6 处理 Web 客户端要求的 SOAP 标头.....117
3.5 部署.....40	7.7 处理在处理 SOAP 标头时 发生的错误.....119
3.6 支持发现.....40	7.8 小结.....122
3.7 调试.....42	第 8 章 使用 SOAP 扩展.....123
3.8 小结.....45	8.1 理解 SOAP 扩展.....123
第 4 章 用 HTTP-GET 和 HTTP-POST 访问 ASP.NET Web 服务.....47	
4.1 使用 HTTP-GET 访问.....47	

8.1.1	Web 服务的生命周期..	123
8.1.2	SOAP 扩展的工作 时机.....	124
8.1.3	使用 SOAP 扩展的 基本步骤.....	125
8.2	实现 SOAP 扩展.....	125
8.2.1	保存对表示将来 SOAP 消息的 Stream 的引用.....	126
8.2.2	初始化 SOAP 扩展.....	127
8.2.3	处理 SOAP 消息.....	129
8.2.4	调用 SOAP 扩展方法 的顺序.....	131
8.3	配置 SOAP 扩展.....	132
8.3.1	利用配置文件.....	133
8.3.2	使用自定义特性.....	134
8.4	示例.....	136
8.5	小结.....	146
第 9 章	异步调用 Web 服务.....	147
9.1	理解异步调用.....	147
9.1.1	回调技术.....	147
9.1.2	同步.....	149
9.2	Web 服务异步调用接口.....	150
9.2.1	BeginHello 方法.....	151
9.2.2	EndHello 方法.....	152
9.3	Web 服务方法异步调用的 几种方法.....	153
9.3.1	使用回调.....	153
9.3.2	轮询完成.....	155
9.3.3	开始调用, 结束调用 ..	157
9.3.4	开始调用, 等待处理, 结束调用.....	158
9.4	小结.....	160
第 10 章	创建分析 Web 页内容的 Web 服务.....	161
10.1	理解 WSDL	161
10.2	指定输入参数.....	166
10.3	指定从分析的 HTML 页 返回的数据.....	167

10.3.1	预备知识: 正则 表达式.....	167
10.3.2	步骤.....	168
10.4	实例.....	168
10.5	小结.....	173
第 11 章	ASP.NET Web 服务的 安全性.....	175
11.1	基本原理.....	175
11.2	基于 Windows 的身份 验证.....	180
11.2.1	准备	180
11.2.2	集成 Windows 验证 ...	183
11.2.3	基本验证方式	184
11.2.4	模拟	186
11.2.5	授权	194
11.2.6	考虑安全性后的 客户端.....	195
11.3	基于 SOAP 标头的自定义 解决方案.....	197
11.3.1	HTTP 模块和 ASP.NET 基础结构.....	298
11.3.2	方案的实施	199
11.4	自定义基于 Windows 的 身份验证.....	214
11.5	代码级别的访问安全	219
11.6	小结	222
第 12 章	使用 ASP.NET 内部对象	225
12.1	概述.....	225
12.2	使用 Application	226
12.3	使用会话	231
12.4	使用 Server 对象	240
12.5	使用缓存对象	242
12.6	小结.....	254
第 13 章	使用事务	255
13.1	事务的基本概念	255
13.2	ASP.NET Web 服务中的 事务.....	256
13.3	实例.....	257

13.4 小结	265
第 14 章 开发基于远程处理框架的 Web 服务	267
14.1 理解远程处理	267
14.1.1 什么是远程处理	267
14.1.2 什么是远程处理 框架	268
14.1.3 远程处理的过程	268
14.1.4 远程对象	269
14.1.5 远程处理中的数据 封送	271
14.2 基于远程处理的 Web 服务 的开发步骤	272
14.2.1 实现若干个提供 Web 服务的远程处理类	272
14.2.2 把远程对象发布为 Web 服务	275
14.2.3 支持发现	277
14.2.4 客户端通过代理访问基 于远程处理的 Web 服务	277
14.2.5 实例	279
14.3 控制生存期	285
14.3.1 基于租约的生存期 管理	285
14.3.2 实例	286
14.4 数据封送	297
14.4.1 概述	297
14.4.2 返回值	298
14.4.3 传入参数	310
14.5 处理错误	318
14.5.1 基本概念	318
14.5.2 实例	319
14.6 安全性	324
14.7 小结	332

第 15 章 用 C++托管扩展开发 Web 服务	333
15.1 C++托管扩展简介	333
15.1.1 添加对程序集的 引用	333
15.1.2 引入和声明命名 空间	334
15.1.3 使用托管类型	334
15.1.4 声明托管类型	336
15.1.5 应用特性	340
15.1.6 编译和运行	340
15.1.7 实例	341
15.2 开发 ASP.NET Web 服务	343
15.2.1 服务器端	343
15.2.2 测试	348
15.2.3 客户端	350
15.3 开发基于远程处理的 Web 服务	354
15.3.1 实现基于远程处理的 Web 服务	354
15.3.2 实现用户	356
15.4 小结	358
第 16 章 用 ATL Server 开发 Web 服务	359
16.1 概述	359
16.1.1 基本结构	359
16.1.2 开发步骤	360
16.1.3 客户端	360
16.1.4 实例	361
16.2 数据封送	370
16.2.1 概述	370
16.2.2 实例	372
16.3 使用 SOAP 标头	387
16.3.1 基本步骤	387
16.3.2 实例	387
16.4 小结	392

第 1 章 理解 XML Web 服务

下一代网络将演变为可计算的智能网络，而 XML Web 服务将在其中扮演关键的角色。依赖于标准协议，过去组件实现的功能将直接以 XML Web 服务的形式在 Web 上发布，而这种组件可以被无数的用户所用。开发人员也将受益于 XML Web 服务，同样基于标准协议，他们搜索特定的 XML Web 服务，并在软件中直接调用 Web 上的 XML Web 服务。

本章主要内容如下：

- XML Web 服务的定义
- XML Web 服务的优势
- XML Web 服务的用途
- XML Web 服务中的协议
- XML Web 服务的实现
- XML Web 服务的使用

1.1 什么是 XML Web 服务

XML Web 服务的正式定义是：XML Web 服务是一种编程模型，它提供在可缩放的、松耦合的和非特定平台的环境下交换信息的能力，信息交换使用诸如 HTTP、XML、XSD、SOAP 和 WSDL 之类的标准协议。在 XML Web 服务及其客户端之间交换的基于 SOAP 的 XML 信息，可以是结构化、类型化或松散定义的信息。使用文本格式（如 XML）的灵活性使信息交换能够以松耦合的方式随着时间而发展。由于 Web 服务基于标准协议且不特定于平台，因此可以与多种多样的实现、平台和设备进行通信。

从上述定义可以看出 XML Web 服务一个特点是基于开放的标准协议，最大的优点就是实现异构平台间的互通。

下面的介绍将有助于读者进一步理解 XML Web 服务。

1.2 XML Web 服务的优势

在对 XML Web 服务不太熟悉的时候，经常要问的一个问题是，为什么还要 XML Web 服务，难道现有技术还不够吗？回答这一问题的最好方式就是列举 XML Web 服务的优势，并指出现有技术在这方面的表现。

1. 异构平台互通

XML Web 服务的最大优势是提供了异构平台无缝衔接的技术手段。长久以来，人们一直在摸索异构平台互通的解决方案。

当然，如果所有的用户都使用同样的硬件平台、同样的操作系统、同样的软件，采用相同的协议通信，这样或许就根本不会产生互通的问题。但显然，这样的假设在现实中并不成立，互通的需求总是存在。

于是第一代网络诞生了，整个网络连接了各种各样的计算机，上面运行着各种系统软件，但它们都采用相同的通信协议 TCP/IP 相互对话，FTP、MAIL 等服务随之诞生。但这种互通的层次是传输层实现的，应用软件之间的直接对话还不可能。

随着 HTTP 协议和 HTML 协议的广泛使用，Web 诞生了。互联网络迎来了它的第二个发展高峰。Web 提供了丰富的信息展示能力，通过将应用程序转变为 Web 应用，只要有浏览器，客户端就可以通过 HTTP 协议在网络中访问应用程序。

但应用程序的用户并不总是满足仅通过浏览器访问。有时候，他们希望有一种更直接的方式把应用嵌入到自己的应用。例如，用户可能不希望每次都访问公司 A 的网站以查看即时股票信息，相反，他们希望在自己的应用程序中集成该功能，这时 HTML 就无能为力了。

现在 XML Web 服务正好满足了这种需求。公司 A 把即时股票信息查询转变为 XML Web 服务 S，然后将该服务信息公布在某个公共的站点 B，用户 C 通过访问 B 了解到服务 S 的信息，其中包括 S 的服务描述。然后 C 将根据 S 的服务描述生成本地代理，借助于该代理用户获取 S 提供的即时股票信息。

有了 XML Web 服务，整个 Web 服务就不止是提供信息，它还将具有处理信息的无限潜力。任何两个应用程序，只要能读写 XML，那么就能互相通信。

2. 更广泛的软件复用

通过组合已有模块搭建应用程序，就是软件设计中的复用技术。通过使用成熟稳健的模块，能大幅度提高软件的生产效率和质量。

在软件开发史中，最先出现的就是源代码级的复用：把经常使用的代码包装为过程而被其他代码调用。这样的复用仅限于同一种开发语言，例如，用 PASCAL 编写的排序函数不能在 C 中直接使用。

为了克服这个问题，人们想到把某些公共功能编写为二进制的动态链接库 DLL。这样的 DLL 可由一种语言编写而可能由另一种编程语言调用。但不同语言编写的 DLL 仍可能存在差异，例如参数传递顺序、数据类型的内存布局等。另外，这种复用仅限于特定的平台范围内，因为 DLL 是二进制的机器代码。此外，为了使用 DLL 中的函数，用户必须了解很多细节信息，如方法名及其参数、参数传递规则等。

COM 随之而来。COM 是基于二进制的对象复用标准，它将接口和接口实现分离开来。借助于系统提供的服务，用户可以透明地引用 COM 对象，而不用实现细节以及确定 DLL 所在的位置。事实上接口的实现可以采取任何编程语言，但由于 COM 是二进制标准，所以仍依赖于特定的平台。也就是说，Windows 系统下的 COM 组件，在 Linux 或 UNIX 下将不可用。

下一个登场的自然就是 XML Web 服务了。可以采用任何语言在任何平台下把一组功能包装成 XML Web 服务，然后将其注册到 UDDI 中心（一个特殊的用以索引 Web 服务的站点）。用户只要获得了描述 XML Web 服务的 WSDL 文件，就可以用代理工具自动产生

合适的客户端代理，然后就可以通过代理访问 XML Web 服务提供的功能。这里用户不必了解 XML Web 服务如何实现、实现的平台等。用户只需确认根据 WSDL 描述的 XML Web 服务能否满足需要，因此 XML Web 服务将软件复用技术推向了一个高峰。

3. 普遍的通信能力

XML Web 服务可用基于 XML 的 SOAP 来表示数据和调用请求(这是被称为 XML Web 服务的原因之一)，并且通过 HTTP 协议传输这些 XML 格式数据。这一点非常重要，因为它意味着 XML Web 服务的调用请求和回应消息可以透过防火墙。出于安全性的考虑，一般系统都关闭了除 HTTP 协议标准端口 80 以外的其他端口。

而在 XML Web 服务出现以前的各种分布式组件技术，都是基于二进制标准的，并且通过特殊的端口进行通信，因此当调用被防火墙隔离时，就可能导致失败。

4. 更迅捷的软件发行方式

在 XML Web 服务出现以前，软件主要通过机器预装、光盘等物理媒介发行。这种发行方式的缺点是：成本高，需要大笔的营销开支，从而也增加了用户的使用费用；软件更新周期长；不便于定制，一般只能针对某些需求制作有限个版本，软件购买者要为他永远不会使用的功能模块支付费用，这显然是不公平的。

XML Web 服务将彻底改变软件的发行方式。软件供应商将把软件分解成若干 Web 服务模块构成的系统，直接在 Web 上发布软件。通过支付少量的费用，用户在网上下载安装基本的客户端模块，这些模块利用远程的 XML Web 服务满足用户的特殊要求。随着软件开发商不断更新 XML Web 服务版本，用户也能及时享受新版本的服务。

5. 方便商务到商务的集成 (B2BI)

随着电子商务的发展，各个商务公司需要更有效的方式来推动它们的商务活动，这就产生了商务到商务集成的趋势。公司需要与它的合作伙伴建立更为紧密的联系，以促进向现有用户销售彼此的产品。但这个过程受到如下因素的干扰：各个企业有自己的一套专用的数据格式和自己的业务软件，紧密集成这些软件需要大量的中间转换。

但有了 XML Web 服务，这种情况将得到改观。各个企业把业务软件的核心模块以 XML Web 的形式向其合作伙伴发布，这样以较低的代价实现了商务到商务的集成。

1.3 XML Web 服务的用途

为了更好地了解 XML Web 服务，首先来了解一下 XML Web 服务将要扮演的角色，下面是 XML Web 服务可能的用武之地。

1. 商务组件

XML Web 服务的最基本用途是包装商务逻辑，以保证能为更多的电子商务活动所用。例如，保证所实现的最基本的方案向它的客户端提供某个基本功能。托运商可以把计算各种送货选项的费用包装为 XML Web 服务，并向其用户公布。用户的应用程序把包装的重

量与尺寸、装货地点与运送目的地，以及其他参数（如服务的种类）保证为基于 XML 的简单消息并使用标准传输协议（如 HTTP）通过 Internet 将它传递给托运商的 XML Web 服务。然后，托运商的 Web 服务将使用最新成本表计算送货费用，并用一个基于 XML 的简单响应消息将费用数额返回给调用的应用程序。

2. 应用集成

由于大多数公司几乎在每个部门都广泛采用了自定义软件，因而导致产生了大量实用但孤立的数据和业务逻辑块。由于开发每个应用程序所处的环境是多种多样的，而技术在永不停歇地发展，因此，利用这些应用程序来创建一个可以工作的集合曾是个令人生畏的任务。

使用 XML Web 服务，可以将每个现有应用程序的功能和数据以 XML Web 服务的形式公开。然后，便可以创建使用此 XML Web 服务集合的复合应用程序以实现各个构成应用程序之间的互操作性。

3. 软件开发

未来软件将是若干 XML Web 服务有机组合而成的复合应用。未来会诞生专门提供 XML Web 服务的软件商，它们直接在 Web 上出售具有专业水准的 XML Web 服务，例如翻译、实时天气预报、搜索引擎等服务。

1.4 XML Web 服务中的主要协议

XML Web 服务基于一系列开发的和标准的协议。理解这些协议以及这些协议如何配合工作，可能是理解 XML Web 服务的关键。

1. XML

如果读者对 HTML 有所了解，就应该清楚，HTML 适合展示数据，指定数据的表现形式、布局等，但从 HTML 中提取数据相当麻烦，因为数据跟格式标志混在一起，而且即使从 HTML 提取了数据，仍缺少一套解释数据的规则。

为了克服 HTML 的上述不足，XML 诞生了。跟 HTML 相同的是，XML 也是由若干标志构成的。但 XML 本身专注于描述数据本身而不是其外在的表示形式，例如，跟用高级语言定义新的类型然后声明该类型的变量类似，XML 用一些元标志定义了数据类型，然后可以用自定义的标志声明数据类型的实例。这样 XML 中的每一项数据都有明确的规定，其含义本身可以用 XML 来描述。

在 XML Web 服务中，XML 有如下作用。

（1）简单对象访问协议 SOAP 用 XML 描述每个 SOAP 消息

也就是说，用户和 XML Web 服务之间交互都采用 XML 这种开发的文本语言。具体而言，客户端的代理将把用户的调用请求按照 SOAP 包装成 XML 格式的消息，然后通过某种协议例如 HTTP 通过 Internet 传到 XML Web 服务所在的服务器。服务器解读 XML 形式的请求，并把它翻译为 XML Web 服务所理解的调用形式。接下来，XML Web 服务的

调用结果被翻译为 XML 消息并透过网络传到客户端，客户端再把响应的 XML 回应翻译为用户应用所能理解的形式。

(2) WSDL 用 XML 描述服务

WSDL 是 XML Web 服务的描述，相当于 COM 对象的接口定义。WSDL 用 XML 描述 XML Web 服务提供的服务端口（相当于一个 Web 方法）、每个端口接收的消息和返回的消息等。

在 XML Web 服务中，XML 扮演了幕后英雄的角色。XML Web 服务的开发者和使用者一般都不用跟 XML 直接打交道。但是，请牢记，XML Web 服务中，所有的数据都表示为 XML！

2. 简单对象访问协议 SOAP

SOAP 是一个基于 XML 的简单协议，用于在 Web 上交换结构化的类型信息。该协议不包含应用程序或传输语义，这使它高度模块化并具有强可扩展性。

注意

SOAP 本身只是一个协议，它定义了如何格式化请求和响应，从而使两个支持 SOAP 的应用程序互通。

之所以被称为“简单”协议，是因为它和用以此目的的其他协议如 DCOM、IIOP、ORB、CORBA 等相比，是简单的。

SOAP 提供了调用 XML Web 服务的一种方法。为此，所有的 XML Web 调用请求和回应都被格式化为 SOAP 消息。

注意

这些 SOAP 消息不是别的，只不过是一个普通的 XML 文本。

虽然，通常采用 HTTP 协议传输 SOAP 消息，但这不是惟一的选择。事实上，SOAP 消息只不过是格式化的 XML 文本，可以由任何传输协议承载，例如 FTP、SMTP 等。

3. XML Web 服务描述语言 WSDL

WSDL 是基于 XML 的协定语言，用于描述服务器提供的网络服务。尽管有了 SOAP，你有能力发出各种 SOAP 消息来请求 XML Web 服务，但你怎么知道 SOAP 消息具有什么内容，才能使 XML Web 服务正确地执行从而得到需要的结果呢？答案就在于 WSDL。它规定了用户调用 XML Web 服务所应了解的一切，包括位置、参数信息和支持的协议。客户端的 XML Web 服务代理能依据 WSDL 准确地产生格式适当的消息，并能无误地理解响应。读者可以把 WSDL 简单地理解为 XML Web 服务的接口定义，它是用户调用 XML Web 服务所需了解的全部。

一般不用专门撰写 WSDL 文件。相反，开发工具提供了依据 XML Web 服务自动产生对应的 WSDL 的方法。

4. 通用说明、发现和集成 (UDDI)

UDDI 是提供程序公布 Web 服务的存在以使 Web 服务使用者找到感兴趣的 Web 服务的一种机制。

在开发出了 XML Web 服务后,自然希望更多的用户能使用它。用户使用你的 XML Web 服务的方法是获得该服务的 WSDL。你可以用 E-mail 把 WSDL 发送给你的用户,或者通知他们在哪个地方可以获得 WSDL。除了 E-mail 这种方式,你甚至可以用电话。方式并不重要,重要的是让用户获得 WSDL (或执行什么命令可以自动产生你的 XML Web 服务的 WSDL)。接下来,用户就依据获得的 WSDL 产生合适的代理访问你的 Web 服务。

但上面的方式可能会遗漏大量潜在的用户。一种更好的方式是吧 XML Web 服务发布到一个公共的大家都知道的站点,用户能够通过查询该站点发现合适的 XML Web 服务。这个过程就是 UDDI。

总之,对 XML Web 服务开发商而言,UDDI 提供了一种发行 XML Web 服务的可选 (一般情况下,是最好的) 方式,这种方式有可能吸引无限的潜在用户。对用户而言,UDDI 提供了 XML Web 服务的索引服务,利用它,可以找到感兴趣的 XML Web 服务。

5. DISCO

DISCO 是 Discovery of Web Services 的缩写。跟 UDDI 规范类似,它也是为帮助用户发现 XML Web 服务而提出的协议。它用 XML 格式定义了对多个 XML Web 服务的 WSDL 的引用。

1.5 XML Web 服务的实现

协议本身并没有对 XML Web 服务的实现作任何规定,这就意味着可以采用多种方式来实现 XML Web 服务——它们可以是一个 CGI 应用、一个 ISAPI 扩展 DLL 或最新的 ASP.NET Web 服务。

读者也面临着选用何种语言实现 XML Web 服务的问题。这可能首先决定于现有的 XML Web 服务开发工具对编程语言的支持。目前对 XML Web 服务开发支持最好的集成开发环境还不多见,最令人满意的当推 Visual Studio.NET,它支持下列编程语言来开发 XML Web 服务: C#、Visual Basic、C++、托管的 C++、J# (微软最近推出的 .NET 平台下的 JAVA)、JScript...随着 .NET 平台的发展,上面的清单还会加长,谁能保证某天微软不会推出 .NET 平台下的 PASCAL 语言呢?读者可以在这些语言中根据自己的喜好选择一种语言开发 XML Web 服务。为了满足不同读者的需要,对于本书中的大部分示例,都提供了两种语言的实现。

除了 Visual Studio.NET 外,Delphi 6 也是一个不错的 XML Web 服务开发平台。Borland 特别提供了 7 个直接的 Web Service 组件、三个 Web Service Wizard,以及其他数个相关的 VCL 组件来帮助开发人员快速开发 Web Service,再结合 Delphi 原有的 MTS/COM+/CORBA/EJB 组件模型开发更具延展性的 Web Service。

但本书不是一本介绍用 Delphi 开发 XML Web 服务的书，提及 Delphi 6 只想说明，在实现 XML Web 服务时，开发人员实际上面临着多种选择的机会。

下面我们将讨论 .NET 平台为 XML Web 服务的实现提供了哪些支持。这一话题的讨论将贯穿全书，这里只是提供一个摘要性质的概览。

.NET 平台提供了两种实现 XML Web 服务的方式：ASP.NET Web 服务和 ATL Server Web 服务。ASP.NET Web 服务通过后缀为 asmx 的文件实现 Web 服务，而后者则通过传统的 DLL 形式提供。

asmx 文件可以用任何托管的编程语言，如 Visual Basic、C#、托管 C++ 等编写，而开发 ATL Server Web 服务则是 C++ 程序员的专利。因此笔者推测，ASP.NET Web 服务将成为未来 .NET 平台下开发的主流 XML Web 服务。

开发 ASP.NET Web 服务并不复杂。开发人员不用了解 XML、SOAP、WSDL、UDDI 就可以实现专业水准的 XML Web 服务，因为 ASP.NET 提供了大量的基础服务而掩盖了让人困惑不安的技术细节：

- 自动将用户调用请求分配到 asmx 文件定义的 Web 服务。
- 自动调用请求，翻译为 ASP.NET Web 服务可以理解的格式，例如从中提取出调用参数数据并翻译为 ASP.NET Web 服务接收的参数值。请求格式可以是 XML 格式的 SOAP 消息、HTTP-GET 或 HTTP-POST 等。这个过程称为反序列化。
- 自动把 ASP.NET Web 服务的响应打包为 SOAP 消息或其他消息。这个过程又称为序列化。
- 负责传递序列化后的消息。
- 根据 asmx 文件自动产生 WSDL，开发人员不用手工撰写任何 WSDL，ASP.NET Web 服务的描述文件由 ASP.NET 框架自动根据 asmx 文件产生。
- 基于拷贝的部署。ASP.NET Web 服务可以方便地由开发服务器部署到发行服务器上。因为 ASP.NET Web 服务不用注册任何组件，惟一的要求是发行服务器上必须安装 .NET 框架。
- 支持静态和动态 Web 服务发现。ASP.NET 通过 .disco 文件和 vsdisco 文件分别支持静态和动态的 Web 服务发现。如果用 Visual Studio.NET 开发 ASP.NET Web 服务，将为每一个 ASP.NET Web 项目自动产生一个动态发现文件，并且提供了 .disco 文件模板。

具体到 ASP.NET Web 服务的实现，读者又有两种选择：直接用 .NET 框架 SDK 或用 Visual Studio.NET。我相信 99.9% 的读者会选用后者，因为集成开发环境提供了高亮度的语法显示、智能编码提示、集成调试工具及应用向导等一系列功能。本书中的大部分例子都是在集成开发环境中实现的，但几乎不用做多少工作，就可以把集成开发环境中实现的 ASP.NET Web 服务快速改造成 .NET 框架 SDK 样式的 Web 服务。

对于大批的 Visual C++ 开发人员，他们可能会选择用 ATL 来开发 Web 服务。.NET 对这种方式的支持也很棒，开发人员编写的代码量跟实现同样功能的 ASP.NET Web 服务所用的代码量相差无几。

1.6 XML Web 服务的使用

XML Web 服务的使用几乎不受任何限制,对用户最基本的要求就是能理解 SOAP 消息,而这是项非常容易满足的要求。下面是几类使用 XML Web 服务的用户。

1. 浏览器

大多数计算机上都装有浏览器,因此让 XML Web 服务在浏览器下执行可能是个不错的主意。只要 XML Web 服务支持 HTTP-GET 或 HTTP-POST 的绑定方式,那么就可以通过浏览器执行。ASP.NET Web 服务自动支持这两种方式的绑定,因此可以通过浏览器访问 ASP.NET Web 服务。

通过浏览器使用 XML Web 服务还不用产生任何客户端代理。

2. .NET 应用

这时可以用 .NET 的 WSDL 工具根据 Web 服务的 WSDL 自动产生客户端的代理类,然后通过代理类访问 XML Web 服务。还可以直接用 Visual Studio.NET 提供的添加 Web 引用对话框自动在项目中添加代理类。

3. 非.NET 应用

其他开发工具,例如 Delphi 提供了访问 XML Web 服务的客户端工具。

后两种方法都要求 XML Web 服务支持 SOAP 绑定。客户端的代理类或工具把调用请求序列化为 XML 形式的 SOAP 消息并传送给远方的 XML Web 服务,并把接收到的 SOAP 消息反序列化为应用可以理解的形式。

本书后续章节将主要讨论用户为 .NET 应用时的 XML Web 服务的使用。

1.7 小 结

XML Web 服务是一种编程模型,它提供在可缩放的、松耦合的和非特定平台的环境下交换信息的能力,是下一代智能网络的关键技术。

XML Web 服务的主要优势是:轻松实现异构平台互通、便于实现更广泛的软件复用、具有普遍的通信能力、提供更快捷的软件发行方式和更紧密的商务到商务的集成。

XML Web 服务的主要用途是:保证商务组件、应用集成和开发松耦合的复合软件。

理解 XML Web 服务的关键之处是理解它所依赖的协议。主要有表示消息和数据的 XML 协议、定义消息和响应格式的简单对象访问协议、描述 XML Web 服务器提供的网络服务的 WSDL 协议和两个支持 XML Web 服务发现的 UDDI 和 DISCO 协议。

可以采用任何技术实现 XML Web 服务。.NET 平台下可以编写两种 XML Web 服务:ASP.NET Web 服务和 ATL Server Web 服务。

可以通过浏览器访问支持 HTTP-POST 或 HTTP-GET 绑定的 XML Web 服务。.NET 应用通过 WSDL 或添加 Web 应用的方式产生客户端代理类,并通过代理访问 XML Web 服务。其他非.NET 应用则借助专门的客户端工具访问 XML Web 服务。