

Web与无线

实用技术译丛

使用XML 开发.NET Web服务

全方位指导使用XML开发Web服务

- SOAP、Web服务描述语言及XML模式定义
- 配置与实现、开发与部署以及调试技术
- 大量现成可运行的应用程序

(美) **David Jorgensen** 著

刘颖 刘洋 译
任韦潼 汪佳



科学出版社

www.sciencep.com

Web 与无线实用技术译丛

使用 XML 开发.NET Web 服务

(美) David Jorgensen 著

刘 颖 刘 洋 译
任韦潼 汪 佳

科学出版社

北 京

内 容 简 介

面对当今及未来的网络程序开发所面临的挑战,必须有一种与语言无关、与平台无关的应用程序,它可以让诸多机构继续使用已有的硬件,使用即有的应用程序,并且让现有的开发人员可以在不经培训的前提下继续工作。随着 XML Web Services 的出现,这样的梦想随即成为现实。

本书旨在为广大读者引进国外有关 XML Web Services 的最新知识,让读者通过本书的学习掌握其中的精髓。

本书内容详实丰富,权威性强,并且包含大量的实例。

Developing .NET Web Services With XML

Original English language edition published by Syngress Publishing, Inc.

Copyright © 2002 by Syngress Publishing, Inc.

All rights reserved.

本书中文版由美国 Syngress Publishing, Inc. 授权科学出版社出版,未经出版者书面允许不得以任何方式复制或抄袭本书内容。

版权所有,翻印必究。

图书在版编目(CIP)数据

使用 XML 开发 .NET Web 服务 / (美) 乔格吉 (Jorgensen, D.) 等著; 刘颖, 刘洋等译. —北京: 科学出版社, 2003

(Web 与无线实用技术译丛)

ISBN 7-03-011632-1

I. 使… II. ①乔… ②刘… ③刘… III. 电子计算机—无线网站
IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 046096 号

责任编辑: 朱凤成 / 责任校对: 都 岚

责任印制: 吕春珉 / 封面制作: 一克米工作室

科学出版社 出版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2003 年 6 月第 一 版

开本: 787×1092 1/16

2003 年 6 月第一次印刷

印张: 19 1/4

印数: 1—4 000

字数: 460 000

定价: 36.00 元

(如有印装质量问题, 我社负责调换〈路通〉)

丛书序

十几年前, Internet 对平常人来说还是一个新生事物。如今, 计算机网络技术飞速发展, 其应用已经渗透到各行各业。网络通过信息交换来实现资源共享, 不但能快速传输、集中和综合处理数据信息, 而且也为我们提供了相当灵活的工作环境。网络技术已经成为经济发展的强大动力! 网络正在改变人们的生活、学习和工作方式, 提高人们的生活质量, 推动社会的进步。要想在网络飞速发展的今天大展身手, 就必须了解网络、掌握网络、应用网络! 这套《Web 与无线实用技术译丛》正是为此而量身定做的!

从技术角度来说, 这套丛书的内容涉及网络的构建、管理、安全与维护以及网络应用程序的开发, 涵盖了计算机网络网络技术的各个方面, 提供了全方位的网络技术教程和解决方案。从网络类型上来讲, 这套丛书不仅以 Cisco 网络为例详细讲解了构建局域网技术的方方面面, 而且对日益火爆的无线网络(无线局域网、无线 Internet 和蓝牙技术)作了深入的剖析。

Windows 2000 无疑是微软最为成功的一款操作系统, 其网络特性也十分突出。而在网络的硬件方面, Cisco 也是一支独秀, 其完整的产品结构及其产品本身的强大功能, 使其成为网络设计中路由器和交换机的首选。如何将这两个领域中最优秀的产品结合起来, 让这种组合发挥最大的潜能, 是对网络体系结构设计师的挑战。有了《在 Windows 2000 上构建 Cisco 网络》, 您就可以清楚地了解到何时、何种情况下应该使用何种硬件, 以及该硬件如何与 Windows 2000 操作系统结合。

无线局域网(WLAN)技术的发展使人们摆脱了传统线缆的束缚, 可以更方便、灵活、快捷地访问网络资源。无线局域网具有像以太网和令牌环这样的传统局域网的所有特性和优势, 而且不受电缆连接的限制, 实现了更大的自由和灵活性。由于便携式设备(例如笔记本电脑和 PDA)的普及, 这些设备的使用者要求随处都能够使用网络, 但却不需要“寻找”或“插拔”网线, 这样的需求导致了对无线局域网络需求的不断增加。无线局域网适用于工矿企业、大专院校、科研院所、金融证券、商业网点, 与 Internet 相连可实现数据信息的自动、高速且无地域限制的传输。《构建 Cisco 无线局域网》详细介绍了 Cisco 公司的基于 802.11 的产品, 并且全面讲述了构建 Cisco 无线局域网的技术。虽然无线局域网的应用扩展了网络用户的自由, 然而, 这种同时也带来了新的挑战——网络安全性。本书详细介绍了各种形式的网络攻击以及相应的无线局域网各种级别的安全措施。当然, 读者如果对网络安全尤为感兴趣, 《信息时代的个人安全策略》当是首选。

当上网成为一种生活习惯, 电子化成为一种生活方式时, 我们的目光已从身边的真实世界转移到了一个虚拟的时空。从此我们的思维有了更广阔的飞翔天地, 我们的交流再没有了障碍……但与此同时, 以身份盗用为代表的信息化犯罪也与日俱增, 对人们的信息安全构成了威胁。《信息时代的个人安全策略》对这个问题进行了深入的探讨, 并从硬件、软件 and 用户自身等多个角度讨论了若干解决方案。

伴随无线技术的迅猛发展, 客户端的移动设备也在快速地更新换代。作为 Web 管

理员,应该如何从容面对复杂形势愈演愈烈的场面?如何使自己的工作能真正为用户带来体贴的服务?如何使自己能够在发展空间巨大但同时竞争激烈的市场中站稳脚跟?

《无线网站管理实战》从最基本的无线 Internet 构成,到移动设备的发展,为 Web 管理员提供了无线 Internet 方面的工具和技术,例如向现有的 Web 站点上添加无线 Internet 功能、构建新的无线应用,以及帮助管理员了解无线 Internet 解决方案的部署,并用实用而具体的示例做出详细解释及说明。

蓝牙是一种近程无线互联技术,它使笔记本电脑、移动电话和其他便携式设备及家用设备可以相互交换信息。蓝牙技术被美国《网络计算》杂志评为“十年来十大热门新技术”之一。业界人士预计,继 Internet、3G 移动通信之后,蓝牙技术热将席卷全球。这种新技术的“能量”何在?蓝牙技术能让各种电器之间密密麻麻的连线在您面前消失,蓝牙设备就像一款万能遥控器,将传统电子设备的一对一的连接变为一点对多点的连接。而且,这种连接无需复杂的软件支持。另外,蓝牙设备使用全球通行的、无需申请许可的 2.4GHz 频段,可进行实时数据和语音传输,而且具有较高的传输质量(传输速率可达到 10Mbps,在支持 3 个语音频道的同时还支持高达 723.2Kbps 的数据传输速率)。2000 年,爱立信公司推出不必手拨的手机——R520 手机,这是第一次使用了蓝牙技术的手机和头戴式耳机,只要与该手机的距离保持在 10 米以内,您就能用耳机来接听和拨打电话。现在,已经有越来越多的蓝牙技术产品投放市场。《蓝牙应用开发指南》详细介绍了蓝牙技术以及在各种常用的操作系统上开发蓝牙应用产品的全过程,是蓝牙技术开发人员的必读书。

面对当今网络程序开发的挑战,最好有一种与语言和平台无关的开发方法,能让众多的单位和个人在使用已有硬件和应用程序的前提下享受到更多的网络服务,而且开发人员不必经过培训就可以进行更新。新出现的 XML Web 服务就可以将这样的设想变成现实,这是一种极具有潜力和应用前景的技术。《使用 XML 开发 .Net Web 服务》意在介绍 XML Web 服务的最新知识,让读者顺利掌握开发 Web 服务的利器。

电子商务时代,企业经营的制胜关键在于强调速度和品质,如何善用信息系统提供即时管理信息以协助决策,将从订单到出货的业务流程全部自动化,是电子商务要解决的重要课题。企业的业务经营和内部管理需要各式各样不同的信息系统,通过网络提供应用软件租赁服务的应用服务提供商 ASP (Application Service Provider) 便应运而生。

《应用服务提供商配置宝典》全方位介绍了 ISP 如何成功转型为 ASP,从而使得 ISP 通过集成现有的基础结构实现向未来的技术无缝移植。

总之,技术领先、实用,结构完整、严谨,是我们引进这套书的重要依据;向读者朋友提供创作水平高、翻译质量好的热门图书,是我们的初衷;以书会友,书友互动,是我们的理念。我们热切希望广泛结识技术界、图书界以及社会各界的读者、作者和支持者,同时鞭策自己不断创新,以飨厚爱。

在丛书的出版过程中,得到了许多老师和朋友的热情帮助,不一一赘述,一并致谢!

丛书编辑组

前 言

自从 1998 年 2 月以来,通过万维网联盟(World Wide Web Consortium, W3C)不懈的努力,XML 一直保持着向前发展的势头。最初,许多开发人员对 XML 嗤之以鼻,认为它只不过是书写脚本的另一种方式而已。然而,这些惯于从事数据库管理和开发的开发人员不久便意识到,XML 是能在双方之间提供数据而无需依赖于专门方案的一种方式。

最初,开发人员将 XML 的代码片断写进他们的桌面应用程序中,以便存储配置数据或作为一个输出文件。随着时间的推移,开发人员逐渐将 XML 应用到互联网中。数据库开始借助 XML 实现相互间的通信,公司发现由于使用了 XML,在处理外部数据库数据时轻松了不少。

当然,并不只是开发人员注意到了 XML 的可行性。Microsoft 也同样认识到了 XML 的巨大潜力,并将其作为 .NET Framework 的一块基石。.NET 旨在成为缩小桌面应用程序和在线应用程序之间差异的桥梁,并促进双方对象的通信。与此同时,Web 服务(Web Service)的概念正在发展过程中。

广义地讲,Web 服务就是商业过程在网络上的展现。其内在的含义是基于 XML 的通信流通过 HTTP 协议在一个公共网络(Internet)上传送。当然,Web 服务对一个组织内部同样适用,它可用作密封或暴露传统系统中固有的商业逻辑的一种机制。然后,这些传统系统中,新的应用程序可以利用 Web 服务接口来调节改进过的复杂商业逻辑,有时可以用上数十年的时间。这使得系统在逻辑层得以复用,而不必考虑物理的配置。

Web 服务是全新的 .NET Framework 的基础部分。可以将 Web 服务分为两类:生产者(producer)和消费者(consumer)。生产者 Web 服务用来检索结果集,例如,检索一个电子商务数据库或 Jokes。本书第 8 章将介绍相关技术。消费者 Web 服务用于使用数据结果集,将该集合用作其他用途。本书第 6 章将通过一个购物车的例子对此作详细介绍。本书将详细讨论这两种类型的 Web 服务,并提供例子加以说明。本书探讨的 Web 服务的方面包括:XML、SOAP、WSDL、UDDI,以及 .NET Framework。

当然,Web 服务的应用并不仅限于微软的 .NET framework。IBM、SUN 和 Oracle 等公司也不会袖手旁观。W3C 将继续致力于 SOAP 和 XML 的修订工作,这意味着它们将继续发展壮大。

更进一步,数据库将会把它们对象调整为 Web 服务,这将在第 7 章将有叙述。这是 Web 服务和 XML 不断发展的一个板块。使用 XML 通用的多功能性,一个应用程序能够在多种平台上传送数据并得到相同的结果。将这一逻辑植入 Web 服务中,就增强了应用程序和数据库的功能。

编写本书时假设的前提是读者已具有使用 XML 的经验。例子中的代码是用 C# 和 VB.NET 编写的。同时,我们还将涉及 ADO.NET 和 SQLXML Web 服务。如果读者尽可能全面地研究面向对象的程序设计,掌握其知识,将会从代码实例中获得最大的收益。

创作队伍简介

Mesbah Ahmed (工业管理学硕士及博士) 是美国托莱多大学 (University of Toledo) 信息系统专业的教授。除了教学与研究工作, 他还为俄亥俄州和密歇根州的 IT 及制造行业提供技术咨询及培训。Mesbah 曾为 Ford Motors、Dana Corporation、Riverside Hospital 和 Sears 等公司提供过系统设计和实施项目等咨询服务。目前他提供 Java Server、XML 和 .NET 等技术领域的培训。他还从事数据库系统、制造系统以及分布式环境及 Web 环境下应用程序开发等领域的研究生教学工作。最近, Mesbah 获得了由托莱多大学颁发的杰出教学奖以及由 College of Business Graduate 颁发的教学优秀奖。他目前的研究兴趣在于数据仓库及数据挖掘。Mesbah 已在多家学术刊物上发表过研究论文, 这些刊物包括: 《Decision Sciences》, 《Information & Management》, 《Naval Research Logistic Quarterly》, 《Journal of Operations Management》, 《IIE Transaction》和《International Journal of Production Research》。Mesbah 还在许多国内国际会议中发表过很多的论文并多次举办讲座。

Patrick Coelho (微软认证专家) 是 University of Washington Extension、North Seattle Community College、Puget Sound Center 和 Seattle Vocational Institute 等院所的讲师, 他所教的课程是 Web 开发 (DHTML、ASP、XML、XSLT、C# 和 ASP.NET)。Patrick 是 DotThatCom.com 的创始人之一, 该公司为学生提供咨询、在线程序开发资源以及实习机会。Patrick 目前与本书的合著者 David Jorgensen 和 nLogix 一起, 共同致力于研究 .NET 的解决方案。

Adrian Turttschi (微软认证解决方案开发专家, 微软认证系统工程师) 曾供职于 KPMG International/CERING, 职位是整合体系结构设计者。他主要负责 KPMG 下一代的全球知识管理及协作解决方案 (KnewPro) 的组件、服务和第三方应用程序的整合工作。KnewPro 是用来支持地域上和组织上分散的团队之间协作的应用程序, 它将知识共享与内容管理、团队协作、企业研究、工作流以及遗产数据连接整合到一起。Adrian 还是 KnewPro 体系结构文档的编写人员之一。在加入 KPMG 之前, 他曾在 EBSCO Publishing 做过软件工程师。作为 Early Adopter 计划的一员, Adrian 熟悉 Java、C#、Visual Basic、Pascal 和 .NET Framework 的编程。Adrian 曾做过多次演讲, 并曾在《XML Journal》、《Nature》和《Exchange & Outlook Magazine》发表过许多篇文章。Adrian 获得了瑞士伯尔尼大学 (University of Bern) 科学学院授予的计算机科学及数学硕士学位, 以及位于美国麻省 Brandeis University 艺术与科学研究生院授予的数学文学硕士学位。Adrian 现居住在德国。

David Jorgensen (微软认证专家) David 在位于华盛顿州首府奥林匹亚的 Alliance Enterprises 公司供职, 该公司专门为诸如州职业康复组织之类的社会服务机构开发基于 Web 的个案管理软件。他的近期项目是将一个州立组织数据进行转换, 包括复杂 SQL Server 数据转换包。David 拥有 St. Martin's College 的计算机科学学士学位, 现居住在华盛顿州的普牙卢普。

目 录

第 1 章 什么是 Web 服务	1
1.1 概述	1
1.2 Web 服务	2
1.3 在 Web 服务中使用 XML	10
1.4 System.Web.Services 名称空间概述	11
1.4.1 System.Web.Services.Description 名称空间	11
1.4.2 System.Web.Services.Discovery 名称空间	11
1.4.3 System.Web.Services.Protocols 名称空间	12
1.5 类型编组 (Type Marshalling)	13
1.6 使用 DataSets	14
1.7 小结	16
1.8 内容速查	16
1.9 常见问题解答	17
第 2 章 Microsoft .NET Framework 简介	19
2.1 概述	19
2.2 获取 .NET Framework SDK	20
2.2.1 系统要求	20
2.2.2 下载网址	21
2.3 安装 .NET Framework	22
2.4 通用语言运行库	23
2.5 用 .NET Framework 开发应用程序	27
2.6 .NET Framework 中的组件	35
2.6.1 ASP.NET	36
2.6.2 ADO.NET	36
2.6.3 VB.NET	37
2.6.4 C#	38
2.6.5 Windows Forms	38
2.6.6 Web 服务	39
2.7 小结	40
2.8 内容速查	41
2.9 常见问题解答	42
第 3 章 XML 基础	43
3.1 概述	43
3.2 XML 综述	43
3.2.1 XML 文档的外观	44

3.2.2 创建 XML 文档.....	45
3.2.3 XML 文档的组成.....	45
3.2.4 格式正确的 XML 文档.....	48
3.2.5 架构 (Schema) 和有效的 XML 文档.....	48
3.2.6 XML 文档的结构.....	51
3.3 用 .NET 处理 XML 文档.....	52
3.3.1 读写 XML 文档.....	52
3.3.2 存储和处理 XML 文档.....	53
3.4 用 XmlTextReader 类读取和解析 XML 文档.....	54
3.4.1 解析 XML 文档.....	54
3.4.2 检索 XML 文档的数据.....	56
3.5 用 XmlTextWriter 类书写 XML 文档.....	58
3.6 探索 XML 文档对象模型.....	60
3.6.1 通过 XmlDocument 对象导航定位.....	61
3.6.2 用 XmlDocument 对象解析 XML 文档.....	62
3.6.3 使用 XmlDataDocument 类.....	63
3.7 用 XPathDocument 和 XPathNavigator 查询 XML 数据.....	70
3.7.1 使用 XPathDocument 和 XPathNavigator 对象.....	71
3.7.2 使用 XPathDocument 对象和 XPathNavigator 对象.....	73
3.8 用 XSLT 转换 XML 文档.....	75
3.8.1 把 XML 文档转换为 HTML 文档.....	76
3.8.2 不同格式的 XML 文档之间的转换.....	78
3.9 XML 怎样与数据库协调工作.....	82
3.9.1 用数据库查询创建 XML 文档.....	82
3.9.2 读取 XML 文档到 DataSet 中.....	84
3.10 小结.....	85
3.11 内容速查.....	85
3.12 常见问题解答.....	87
第 4 章 利用简单对象访问协议 (SOAP) 进行信息交换.....	89
4.1 概述.....	89
4.2 Web 服务范例.....	89
4.2.1 SOAP 的作用.....	89
4.2.2 为什么用 SOAP.....	91
4.2.3 为什么用 Web 服务.....	91
4.2.4 互连分布式对象——SOAP 协议.....	91
4.3 用 Web 服务工作.....	105
4.3.1 传送复杂数据类型.....	105
4.3.2 错误处理.....	108
4.3.3 编写 SOAP 客户端应用程序.....	111
4.3.4 传送对象.....	115
4.3.5 传送关系数据 (DataSets).....	119

4.3.6 传送 XML 文件.....	121
4.3.7 SOAPHeader	124
4.4 高级 Web 服务	125
4.4.1 维护状态	125
4.4.2 安全	135
4.5 小结	136
4.6 内容速查	137
4.7 常见问题解答	138
第 5 章 WSDL 和 UDDI	139
5.1 概述	139
5.2 Web 服务标准	140
5.3 描述 Web 服务——WSDL	140
5.4 发现 Web 服务——DISCO	144
5.5 发布 Web 服务——UDDI	145
5.6 小结	151
5.7 内容速查	151
5.8 常见问题解答	152
第 6 章 用 Web 服务打造 ASP.NET/ADO.NET 购物车	153
6.1 概述	153
6.2 建立数据库	154
6.2.1 建立表 Books	155
6.2.2 建立表 Categories	155
6.2.3 建立表 Customer	155
6.2.4 建立表 Orders	156
6.2.5 建立表 BookOrders	156
6.2.6 创建 Access 数据库	156
6.2.7 SQL Server 数据库	158
6.3 创建 Web 服务	164
6.3.1 Book Shop Web 服务概览	164
6.3.2 建立数据连接	166
6.3.3 创建 Web 服务	166
6.3.4 在 ASP.Net 中测试 Web 服务	170
6.4 使用 WSDL Web 引用	173
6.5 建立站点	175
6.6 站点管理	176
6.6.1 创建管理登录 (adminLogin.aspx)	176
6.6.2 创建管理页面 (adminPage.aspx)	177
6.6.3 创建 addBook 页面 (addBook.aspx)	182
6.7 顾客管理	182
6.8 创建 ADOCatalog	184

6.9 建立 XMLCart.....	188
6.10 创建用户接口.....	191
6.10.1 创建 start.aspx 页面.....	191
6.10.2 显示目录.....	193
6.10.3 显示购物车.....	193
6.10.4 编写代码.....	193
6.11 小结.....	194
6.12 内容速查.....	195
6.13 常见问题.....	196
第 7 章 开发 SQLXML Web 服务应用程序.....	198
7.1 概述.....	198
7.2 SQLXML Web 服务.....	199
7.3 开发 TimeTrack 应用程序.....	199
7.3.1 创建数据库.....	199
7.3.2 创建存储过程.....	201
7.4 创建 SQL 服务器虚拟目录.....	202
7.5 在 ASP.NET 中创建客户端应用程序.....	208
7.6 小结.....	221
7.7 内容速查.....	222
7.8 常见问题解答.....	223
第 8 章 创建 Web 服务 Jokes.....	224
8.1 概述.....	224
8.2 创建 Web 服务 Jokes 的动机和要求.....	224
8.3 功能应用程序设计.....	225
8.3.1 定义公用方法.....	225
8.3.2 定义数据库架构.....	226
8.3.3 定义 Web 服务体系结构.....	227
8.4 实现 Jokes 数据资源库.....	229
8.4.1 安装数据库.....	229
8.4.2 创建存储过程.....	231
8.5 实现 Jokes 中间层.....	240
8.5.1 建立 Visual Studio 项目.....	240
8.5.2 开发错误处理程序.....	243
8.5.3 开发数据库访问组件.....	246
8.5.4 开发用户信息管理服务.....	247
8.5.5 开发 Jokes 服务.....	260
8.6 创建客户端应用程序.....	282
8.7 小结.....	293
8.8 内容速查.....	294
8.9 常见问题解答.....	295

第 1 章 什么是 Web 服务

本章内容

- 了解 Web 服务
- 在 Web 服务中使用 XML
- System.Web.Services 名称空间概述
- 类型编组
- 使用 DataSets

1.1 概 述

Web 服务为各种应用程序提供了全新的交互方式。通过访问和使用远程 Web 服务来执行应用程序的功能，可以使程序员在更短的时间内快速传送一个更复杂的应用程序。程序员不再需要创建和维护应用程序的所有功能。通过创建多个 Web 服务来执行多个应用程序中的功能，大大提高了可重用性，从而可腾出时间和资源进行具体项目的其他方面工作。图 1-1 显示了这一过程。

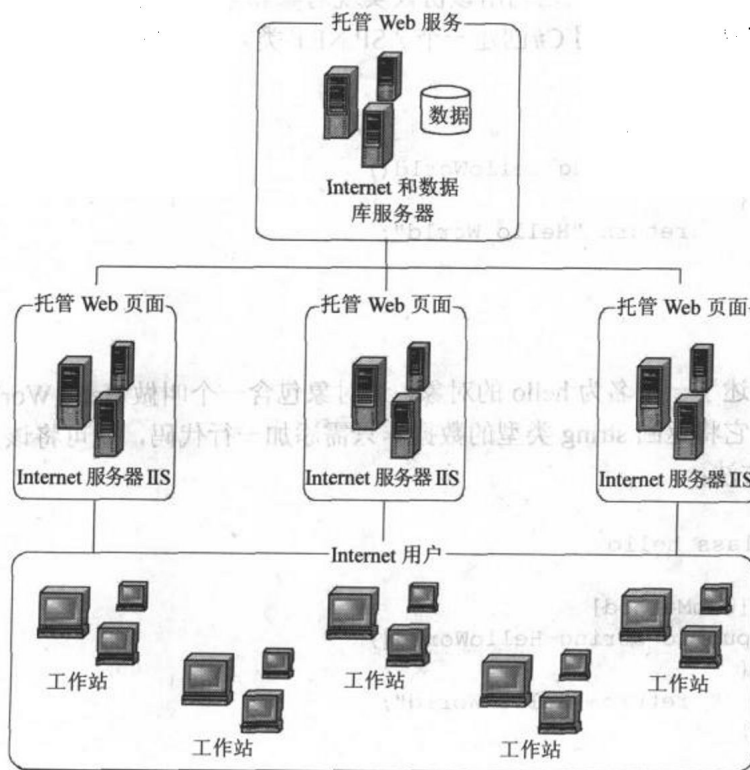


图 1-1 Web 服务适用于何处

本章将分析一个通过 ASP.NET 发送的简单 Web 服务 Hello World。任何能够处理简单对象访问协议（Simple Object Access Protocol, SOAP）的应用程序都能够访问这个 Web 服务的实例。

Web 服务主要通过 XML 运行，以便通过超文本传输协议（Hypertext Transfer Protocol, HTTP）往返传送信息。.NET Framework 给程序员提供的最重要的部分就是 Web 服务。实现基于 XML 的数据传输，就可以通过 Web 服务把基元类型、枚举甚至类传输给执行请求的应用程序。这使应用程序具有全新的可重用性。XML 是建立整个 Framework 的主干。用户接口（User Interface, UI）既可通过执行可扩展样式表转换语言（Extensible Stylesheet Language Transformations, XSLT）创建，也可以通过将数据加载到 DataSets 中和捆绑到 Web 控件（Web Controls）上创建。以 XML 为媒介给客户端设计提供了新的途径。

1.2 Web 服务

Web 服务是任何客户都能通过 HTTP 调用的对象和方法，它基于简单对象访问协议（SOAP）创建。不同于分布式组件对象模型（Distributed Component Object Model, DCOM）和公共对象请求代理程序体系结构（Common Object Request Broker Architecture, CORBA），SOAP 能够通过 HTTP 在端口 80（大多数 Web 服务器使用该端口）传送信息，并且使用标准的数据描述方法。SOAP 使通过 Web 轻松地发送数据和结构成为可能。Web 服务就是利用该协议实现对象和方法的传送。在 VS.NET 中创建 Web 服务很容易。下面是用 C# 创建一个 ASP.NET 类，名称为 Hello World:

```
public class hello
{
    public string HelloWorld()
    {
        return "Hello World";
    }
}
```

这个类描述了一个名为 hello 的对象，该对象包含一个叫做 Hello World 的方法。调用该方法时，它将返回 string 类型的数据。只需添加一行代码，就可将该方法转换为一个 Web 服务方法:

```
public class hello
{
    [WebMethod]
    public string HelloWorld()
    {
        return "Hello World";
    }
}
```

将上述方法转换为 Web 服务方法，需要包括较多的代码。这些代码就是当我们创建新的 .asmx 页和 Hello World 方法时 VS.NET 自动生成的代码：

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Web;
using System.Web.Services;

namespace Hello
{
    public class Hello : System.Web.Services.WebService
    {
        public Hello()
        {
            InitializeComponent();
        }
        private void InitializeComponent()
        {
        }
        protected override void Dispose( bool disposing )
        {
        }
        [WebMethod]
        public string HelloWorld()
        {
            return "Hello World";
        }
    }
}
```

可以在 VS.NET 中快速创建这个类，方法是创建和打开 C# Web 应用项目或者 Web 服务项目，添加新的 Web 服务页。

也可以通过如下代码来创建一个 VB.NET 服务：

```
Imports System.Web.Services

Public Class Service1
    Inherits System.Web.Services.WebService
    <WebMethod()> Public Function HelloWorld() As String
        HelloWorld = "Hello World"
    End Function
End Class
```

设置起始页

当在包含其他 .aspx 或者 .asmx 文件的项目中测试 Web 服务时，一定要在测试前把

正在调试或测试的文件设置成起始页。在 Solution Explorer 中右击文件名并且选择 Set as start page 可完成设置。

按 F5 键就能在 VS.NET 中运行这个例子。一般创建和编译过程需要一段时间。在这个过程完成后，就能看到 Hello 服务的屏幕。

屏幕最上面的一行声明支持下面列出的操作。接着就是带项目符号的链接列表，可链接到每个属于 Web 服务的 Web 方法。在当前实例中，只创建了一个名为 Hello World 的 Web 方法。单击链接 HelloWorld，将出现该服务的描述页。

为测试 Web 服务 Hello 中的 Web 方法 HelloWorld，只需单击 Invoke 按钮，调用这个方法。再次调用该方法会返回字符串“hello world”，返回的结果显示在 XML wrapper 中（如图 1-2 所示）。

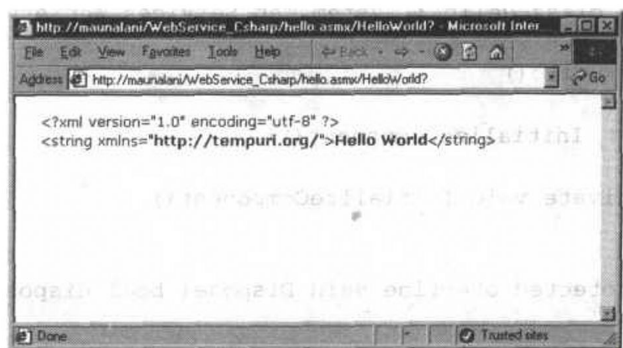


图 1-2 调用 Web 方法 HelloWorld 的结果

注意，XML 节点反映的是方法返回值的数据类型，即 string 型。这条 XML 信息被接收并且转换为字符串“Hello World”。这意味着代码中的任何变量（string 型）均能被指定为 Web 方法的结果。

创建和编译

有 C/C++ 或者 Java 编程经验的读者，对创建和编译的步骤会比较熟悉。以前没有使用过编译语言的 Web 开发人员，可能会感觉这些步骤比较陌生。可以将这个过程想像为：编译器把所有的代码集中起来，并且检查代码中是否存在未定义的变量，变量类型是否匹配，以及是否有其他的句法错误。在该过程中，编译器同时把代码转换为通用语言运行库（Common Language Runtimes, CLR）的中间语言（Intermediate Language, IL），然后再转换成机器语言。这就使得这些代码运行起来比未经编译的脚本更快、更有效。这个阶段完成后，代码将在浏览器中运行。因此，在 .NET 环境中测试 Web 页输出需要花费更长的时间。

服务器之间的通信。在服务器之间发送信息或者使用远程调用功能已经不是新概念了。一些技术，比如 DCOM 和 CORBA，都是众所周知的专用协议，并且已使用多年。现在较新的是通过 HTTP 传输信息的标准协议，这里的协议指的是 SOAP。SOAP 使得用不同的语言开发的、在不同的平台上运行的应用程序，能够有效地进行远程过程调用（Remote Procedure Call, RPC），甚至可以通过防火墙进行调用。DCOM 不用端口 80，

端口 80 是为 HTTP 网络通信保留的；这导致 DCOM 的调用会被防火墙阻塞。SOAP 调用使用端口 80，能够调用位于防火墙后面的过程。图 1-3 对怎样使用 Web 服务进行了高层次的概述，既适用于不同客户类型的公司与客户之间的交互，也适用于在公司的所有服务器（包括传统系统）之间收集和发布公司内部的数据。

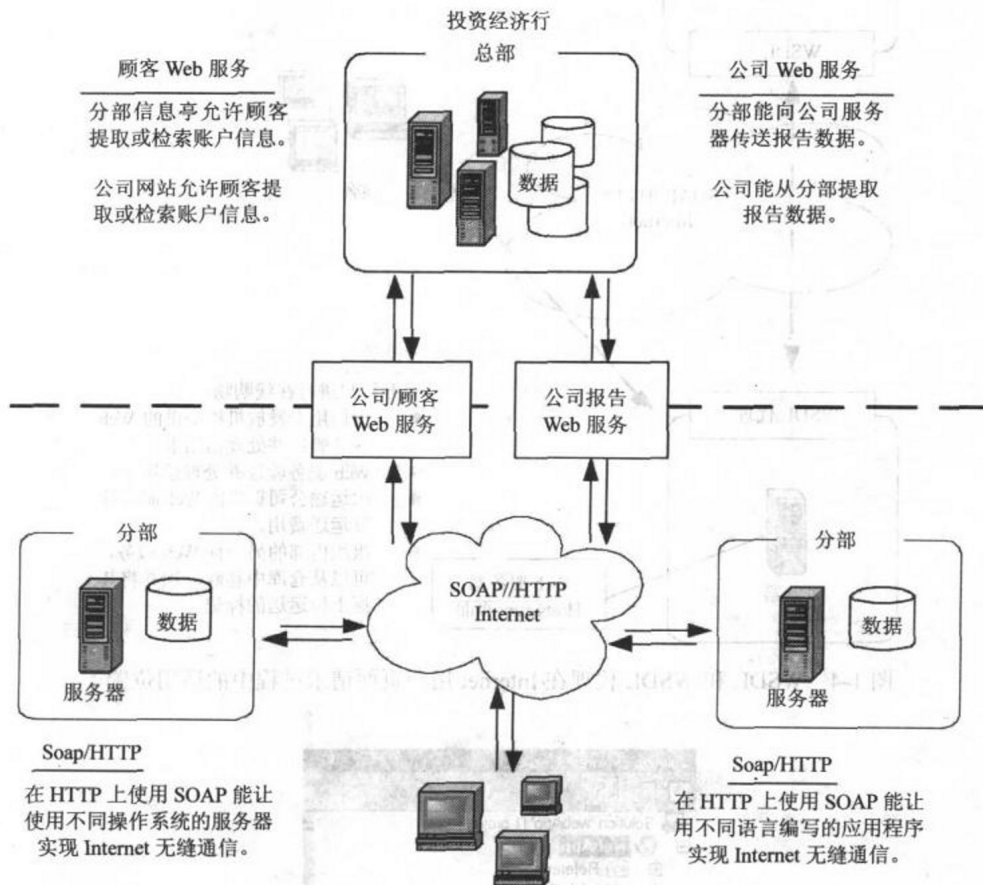


图 1-3 Web 服务的适用环境

在 ASP.NET 中，Web 服务及其方法在扩展名为 .asmx 的页面中定义。创建 Web 服务时，.NET Framework 会在托管该服务的服务器上生成 Web 服务描述语言（Web Services Description Language, WSDL）文件。该 WSDL 文件描述 Web 服务接口。在 Solutions Explorer 中单击 Add Web reference 并选择服务器和服务（如图 1-4 所示）时，VS.NET 将在托管 .aspx 页的 Web 服务器上生成 WSDL 代理。

图 1-5 给出了用于本地主机的 Web 引用和该服务器上的各 Web 服务的 WSDL 代理。

注意

Web 服务器上的单独的应用程序可以访问不同服务器上的多个 Web 服务。同样，多个 Web 服务器也可以访问同一个 Web 服务。

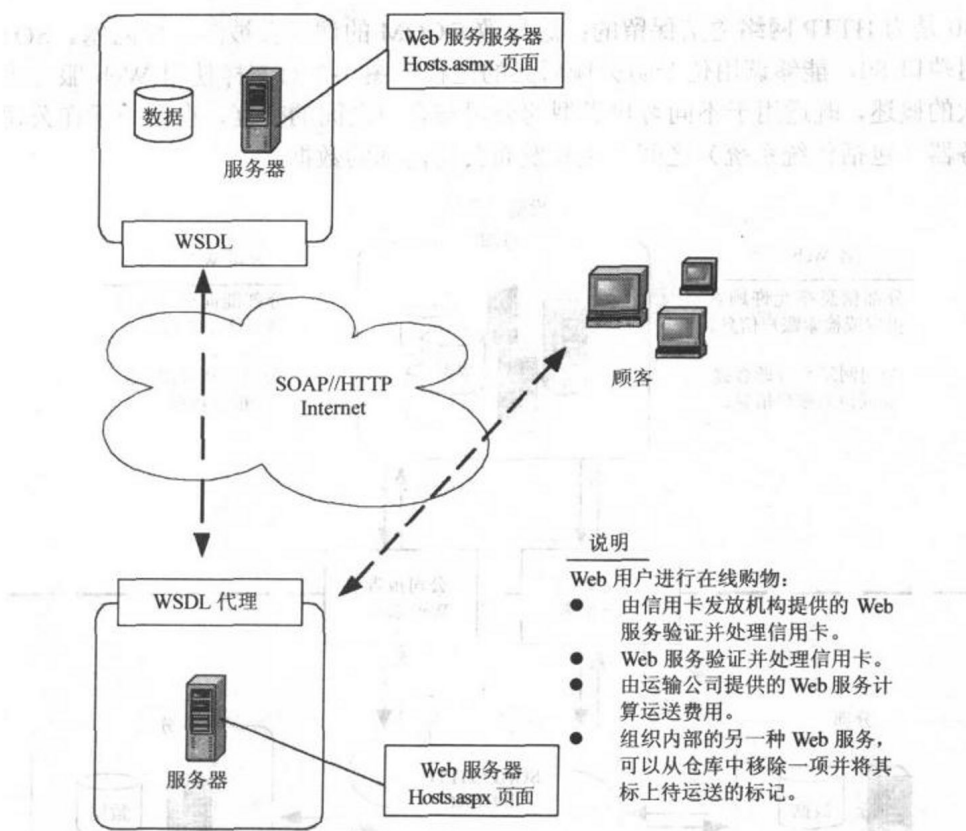


图 1-4 WSDL 和 WSDL 代理在 Internet 用户页面请求过程中的适用位置

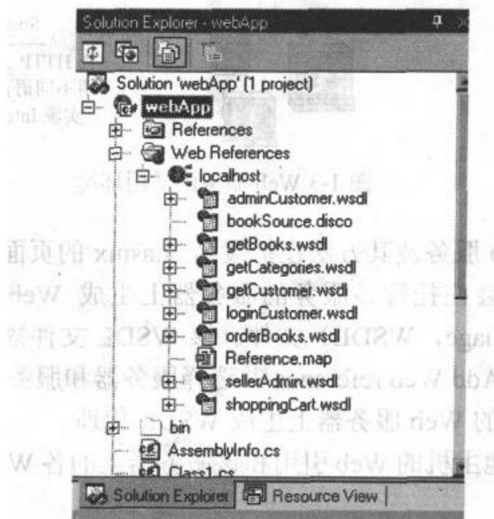


图 1-5 VS.NET 的 Solution Explorer 窗口中的 Web 引用

1. .asmx 文件

ASP.NET 使用 .asmx 文件扩展名定义 ASP.NET Web 服务。在 C# 和 VB.NET 中, 代