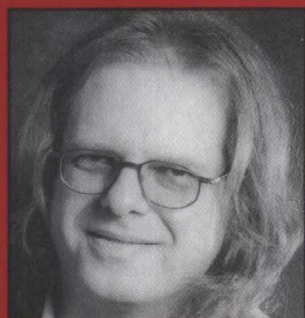
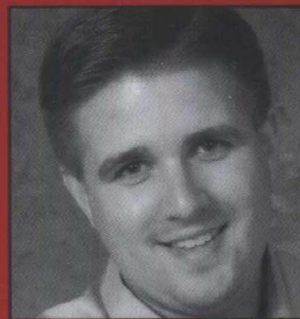
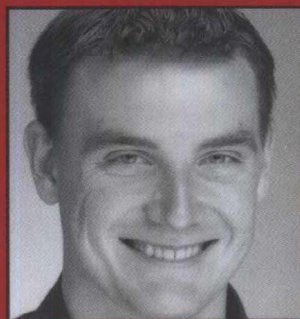
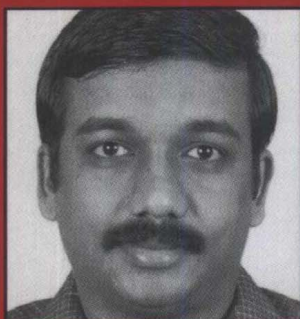
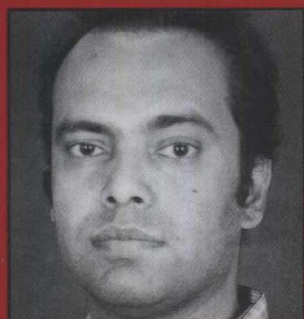


PROGRAMMER TO PROGRAMMER™



C# Web Services

C# Web 服务 高级编程

——使用.NET Remoting 和 ASP.NET
创建 Web 服务

Ashish Banerjee Aravind Corera 等著 康博 译



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



59

TP3/3.09
B21

C# Web 服务高级编程

—— 使用.NET Remoting 和ASP.NET 创建Web 服务

Ashish Banerjee

等著

Aravind Corera

康 博 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: 01-2001-4060

内 容 简 介

Web 服务是 .NET 开发过程中最重要的功能, 它允许我们使用诸如 HTTP 和 SOAP 的标准协议在 Internet 或 intranet 上调用方法, 将 Windows 组件的编程方法提升到一个新的水平。本书详细地介绍了使用 ASP.NET 和 .NET Remoting 创建 Web 服务的技术, 具体内容包括 ASP.NET Web 服务、使用 .NET Remoting 创建 Web 服务、创建 Web 服务客户程序、Web 服务协议(SOAP 和 WSDL)、利用 UDDI 发现 Web 服务和使用 .NET 密码技术保护 Web 服务等, 并展示了大量的 Web 服务应用示例。

本书适合那些有兴趣使用 .NET Framework 所提供的工具创建 Web 服务, 并希望学习 ASP.NET 和 .NET Remoting 这两项技术的读者。本书也是广大 Web 服务开发人员深入掌握 .NET Web 服务开发技能的一本优秀指南。不过, 学习本书读者还必须具备 C# 和 .NET Framework 的基础知识。

Ashish Banerjee, Aravind Corera et al: C# Web Services

EISBN: 1-861004-39-7

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。

图书在版编目(CIP)数据

C# Web 服务高级编程——使用 .NET Remoting 和 ASP.NET 创建 Web 服务/(美)班纳基, (美)卡罗纳等著; 康博译.—北京: 清华大学出版社, 2002

书名原文: C# Web Services——Building Web Services With .NET Remoting and ASP.NET

ISBN 7-302-05685-4

I. C... II. ①班...②卡...③康... 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 050082 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.tsinghua.edu.cn>

责任编辑: 李阳

印 刷 者: 北京大中印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 36.25 **字数:** 927 千字

版 次: 2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

书 号: ISBN 7-302-05685-4/TP·3347

印 数: 0001~4000

定 价: 68.00 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 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 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

什么是 Web 服务

Web 服务是一项最新的应用程序开发技术，它已经引起了在各个平台上工作的开发人员的很大兴趣。Web 服务的基本概念很简单，即允许针对一个对象在 Internet 和本地网络上使用 Remote Procedure Call(远程过程调用，RPC)。Web 服务并不是第一个这样做的技术，但是它不同于早期的技术，它能够通过 HTTP 和 XML 等平台无关标准完全对客户程序隐藏实现细节。客户程序需要知道 Web 服务的 URL 和方法调用使用的数据类型，但是不需要知道 Web 服务是用 Java 创建的并运行在 Linux 上，还是一个运行在 Windows 平台上的 ASP.NET Web 服务。

在 .NET Framework 平台中，Microsoft 为 Web 服务提供了强大的支持。尤其是，它提供了许多能够轻松创建和访问 Web 服务的工具。本书将介绍使用 C# 和 .NET Framework 平台创建 Web 服务的不同方法，并介绍利用这些重要的开发技术所需要使用的一些工具、技术和协议。

本书主要内容

.NET Framework 平台提供了创建 Web 服务的两种途径：ASP.NET 和 .NET Remoting。本书将详细介绍这两种方法，并提供大量 Web 服务示例，以便说明它们是如何综合应用到实际的应用程序中的。另外，还提供一些关于这个重要新技术的潜在用途的看法。

在第 1 章中首先介绍什么是 Web 服务、为什么要使用 Web 服务，以及如何开发 Web 服务，同时介绍了使用 Web 服务所得到的好处，并高度综述了 Web 服务的体系结构。

第 2 章介绍了 Web 服务使用的标准和协议：SOAP 和 WSDL。其中，这些标准允许 Web 服务隐藏其实现细节，并提供跨平台的互操作性，这是 Web 服务最重要的优点之一。

在第 3 章中，首先探讨 Microsoft 实现 Web 服务的细节。Microsoft 环境提供了 4 种创建 Web 服务的方法(除了手动创建 SOAP 和 WSDL 文件)：ASP.NET、.NET Remoting(在本书中将深入探讨)、VB SOAP Toolkit(能够将 COM 组件作为 Web 服务提供)和 ATL Server(提供一种用非托管的 C++ 创建包括 Web 服务的 ISAPI 过滤器的方法)。

下面开始学习本书的核心内容：这部分详细介绍如何使用 .NET Framework 平台所提供的工具创建 Web 服务和 Web 服务客户程序。第 4 章介绍构建一个 ASP.NET Web 服务的过程，通过一个信用卡验证服务示例来展示如何创建、测试和部署 ASP.NET Web 服务。

在创建一个 Web 服务后，接着就可以访问它。为此，第 5 章讨论如何为新的 Web 服务创建一个客户程序。有两种方法可以自动完成这个任务，一种是使用命令行工具 wsdl.exe，一种是使用 Visual Studio .NET。本章就介绍生成 ASP.NET Web 服务客户程序的两种方法。

接下来的 4 章内容主要探讨 .NET Remoting Web 服务。其中第 6 章首先概述 Remoting 和 .NET Remoting 体系结构，同时还介绍了 Remoting 基础结构使用的概念和类。

接着第 7 章介绍使用 .NET Remoting 和 Web Services Anywhere(无处不在的 Web 服务，它



是 Microsoft 使用 Remoting 的 Web 服务的名称)创建 Web 服务。.NET Remoting 允许在任何应用程序中使用我们的 Web 服务(不仅在 IIS 中),而且可以使用任何通道(所提供的是 TCP 和 HTTP 通道)和格式(所提供的是 SOAP 和二进制格式化程序)。

在介绍 Remoting 和 Web Services Anywhere 的概念之后,在第 8 章创建了一个远程 Web 服务示例,它是一个比萨(pizza)投递公司的示例,并且使用了 HTTP 和 TCP 通道。

再次获得一个 Web 服务后,需要创建一个访问它的客户应用程序。在第 9 章中完成了这项工作,使用 soapsuds.exe 实用程序为 Web 服务创建一个代理,并为 Web 服务创建 Web 和 Windows 客户程序。

如果在访问 Web 服务之前需要知道它们的 URL,则 Web 服务的可用性明显地受到了限制。第 10 章将介绍 Universal Description, Discovery, and Integration (通用描述、发现和集成, UDDI),它提供了一种发布 Web 服务的 URL 和它们的功能描述的方法,能够使一些潜在的用户在预先不知道 Web 服务的 URL 的情况下找到它们。

第 11 章介绍一个重要但又不成文的主题:使用 .NET 密码系统类保护 Web 服务。为此,我们使用一个 Remoting Web 服务,并创建一个定制通道接收器,其中使用了加密和解密方法。这个概念将在案例分析 2 中进行进一步的探讨。

第 12 章说明了一个有趣的可能性:使用 Web 服务传送一个 Windows Forms 程序集,而且它能够在客户程序上被实例化。我们使用一个时间表应用程序的示例,它可以从我们的 Web 服务中下载并保存在本地,或者直接运行。这为桌面应用程序的发布和版本化提供了一个有趣的解决方案。可以允许客户程序保存应用程序的数据,即使在不存在有用的网络连接的情况下,也可以始终运行应用程序。

最后为了介绍许多 Web 服务的可能性,我们介绍两个案例分析。第 1 个案例分析是一个航班预定系统,该系统由一个 ASP.NET Web 服务来创建,并展示了一个 Web 服务被整合到一个大的 Web 应用程序中的过程。第 2 个案例分析使用 .NET Remoting,并介绍如何创建一个自定义通道接收器,利用该接收器实现一个能够插入到任何远程 Web 服务中的密码系统层。

本书读者对象

本书适用于有经验的开发人员,他们或者要具有一些开发经验,或者要有在 .NET Framework 平台上使用 C# 的编程经历,并熟悉 ASP.NET 和 Web 编程。

学习本书所需要的准备工作

为了能够运行本书中的示例,您必须具备下面的工具:

- 带有 IIS 5 或更高版本的 Windows 2000 或者 Windows XP 操作系统。
- .NET Framework SDK Beta 2 或更高版本。本书中的代码不能运行在 .NET Beta 1 上,但它们都被新发布的版本测试过。

本书中的所有源代码都可以从以下站点下载: <http://www.wrox.com/Books/BookDetails.asp?isbn=1861004397>。

用户支持

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能够完善的地方。您可以发送电子邮件(地址为 feedback@wrox.com)向我们反馈意见。请确保反馈信息中一定要提到本书的书名。

如何下载本书的示例代码

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，可以方便地定位您所需要的书目。然后，单击 Code 列中的 Download 超链接，或者单击本书具体页面中的 Download Code 超链接，就可以下载相应的示例代码。

从我们的站点中下载的文件都是使用 WinZip 压缩过的文档。将附件保存到本地磁盘上的文件夹中后，需要使用解压缩程序(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(WinZip、PKUnzip 等)解压缩时被设置为使用原有文件夹名。

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，但是错误肯定还是在所难免。如果您发现本书存在错误，例如拼写错误或不正确的代码片段，请给我们反馈信息，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。我们将检查您的反馈信息，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 Advanced Search 或者书名列表轻松定位到本书页面。然后，单击 Book Errata 超连接即可。

E-mail 支持

如果您希望直接向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，要求在邮件的主题中带上本书的书名和 ISBN(国际标准书号)的后 4 位数字。一个典型的电子邮件应包括下面的内容：

- 在主题中必须有本书的书名、ISBN 的后 4 位数字和存在问题的页码。
- 信息的主题应包括读者的名字、联系信息和问题。

我们将不返回您的无用邮件，因为我们仅仅需要有用的详细资料，以便可以节约您和我们的时间。当您发送一个电子邮件信息时，它将得到下面这些支持：

- 用户支持：首先，您的信息将被送到我们的用户支持人员手中，由他们先进行阅读。归档一些被频繁提到的问题，并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。

- 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)，他们将请求本书的作者。我们将尽量保护作者免受干扰，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，从而使所有的读者受益。

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，对于超出常用书目支持的问题，您可以从 <http://p2p.wrox.com/forum> 中的通用列表中获得支持信息。

p2p.wrox.com 站点

为了便于作者和其他人讨论，特将讨论内容加入到 P2P 站点的邮件列表中，而且我们惟一的系统将在邮件列表、论坛、新闻组以及所有其他服务(一对一的邮件支持系统除外)上提供 programmer to programmer™(由程序员为程序员而作)联系。如果您向 P2P 发送一个问题，它一定会被登录邮件列表的 Wrox 公司作者和其他相关专家检查到。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定一个邮件列表：

- (1) 登录 <http://p2p.wrox.com>/站点；
- (2) 从左边的主菜单栏中选择一个适当的类别；
- (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) 使用预定管理程序加入更多的邮件列表并设置自己的邮件首选项。

本系统提供最好支持的原因

您可以选择连接到整个邮件列表，也可以只接收每周的邮件摘要。如果您没有时间和工具来接收邮件列表，可以直接查找我们的在线文档。独特的 Lyris 系统可以将一些无用的垃圾邮件删除，并保护您的电子邮件地址不被侵扰。当存在加入和离开列表、以及任何有关列表的其他常见问题时，请发送邮件到 listsupport@p2p.wrox.com。

目 录

第 1 章 Web 服务概述	1
1.1 分布式计算的历史	1
1.1.1 客户机/服务器模型	2
1.1.2 组件技术	3
1.1.3 分析现状	4
1.2 Web 服务	4
1.2.1 使用 Web 服务的好处	5
1.2.2 Web 服务的标准	8
1.3 Web 服务的体系结构	9
1.3.1 提供 Web 服务	9
1.3.2 使用 Web 服务	10
1.4 Web 服务的前景	11
1.4.1 可编程的 Web	11
1.4.2 .NET My Services	12
1.5 小结	14
第 2 章 Web 服务协议	15
2.1 SOAP 协议	15
2.1.1 格式	16
2.1.2 数据类型和 SOAP 串行化	20
2.1.3 HTTP 协议之上的 SOAP	23
2.2 WSDL 协议	25
2.2.1 格式	26
2.2.2 客户机代理	33
2.3 DISCO 协议	34
2.4 UDDI 协议	35
2.5 小结	35
第 3 章 Web 服务和 .NET Framework	36
3.1 SOAP Toolkit	36
3.1.1 SOAP Toolkit 的概述	37
3.1.2 提供 Web 服务	37
3.1.3 使用 Web 服务	38



3.1.4	使用 SOAP Toolkit 发布 COM 对象	39
3.1.5	何时使用 SOAP Toolkit	45
3.1.6	SOAP Toolkit 概要	45
3.2	ASP.NET	46
3.2.1	提供 Web 服务	46
3.2.2	命名空间	47
3.2.3	使用 Web 服务	47
3.2.4	.NET Remoting	48
3.2.5	何时使用 .NET Remoting	48
3.3	ATL 服务器	48
3.4	小结	49
第 4 章	构建 ASP.NET Web 服务	51
4.1	ASP.NET Web 服务概述	51
4.2	Web 服务协议	52
4.2.1	SOAP 协议	52
4.2.2	HTTP 协议	56
4.2.3	System.Web.Services.Protocols 命名空间	56
4.3	Web 服务的体系结构	58
4.4	发现文件	60
4.5	服务描述	60
4.5.1	查看 WSDL 文档	61
4.5.2	wsdl.exe 实用程序	61
4.5.3	.NET 如何使用 WSDL 文档	64
4.6	创建信用卡验证 Web 服务	64
4.7	不使用 Visual Studio .NET IDE 创建 Web 服务	66
4.7.1	创建 IIS 虚拟目录	72
4.7.2	WebService 属性	73
4.7.3	WebMethod 属性	73
4.8	使用 Visual Studio .NET 创建 Web 服务	74
4.9	测试 Web 服务	77
4.9.1	在浏览器中测试 Web 服务	77
4.9.2	使用 IDE 进行调试	78
4.10	检验和修改 SOAP 响应	79
4.10.1	参数编码	79
4.10.2	SOAP 主体格式化	81
4.11	Web 服务中的状态管理	83
4.11.1	访问和使用 Application 对象	83

4.11.2 访问和使用 Session 对象	84
4.12 部署与配置 Web 服务	84
4.13 小结	87
第 5 章 使用 ASP.NET Web 服务	88
5.1 客户程序如何与 Web 服务通信	88
5.1.1 使用 HTTP 协议调用 Web 服务	88
5.1.2 使用 SOAP 协议调用 Web 服务	89
5.2 发现 ASP.NET Web 服务	90
5.3 查看和使用 WSDL 文档	93
5.4 创建代理	94
5.4.1 使用 wsdl.exe 实用程序	94
5.4.2 使用 Visual Studio .NET 自动生成代理类	97
5.4.3 同步调用和异步调用	98
5.5 创建 Web 服务客户程序	101
5.5.1 用 C# 构建客户程序	102
5.5.2 在代码中使用 Web 服务	103
5.5.3 调试客户应用程序	104
5.5.4 注意事项	108
5.6 小结	109
第 6 章 .NET Remoting 体系结构	110
6.1 什么是 .NET Remoting	110
6.1.1 无处不在的 Web 服务	110
6.1.2 CLR 对象 Remoting	111
6.2 .NET Remoting 的基本原理	111
6.2.1 远程对象	112
6.2.2 服务器	113
6.2.3 客户机	117
6.2.4 运行服务程序和客户程序	118
6.3 进一步了解 .NET Remoting	119
6.4 System.Runtime.Remoting 命名空间	120
6.5 远程对象	121
6.5.1 知名的对象	122
6.5.2 客户激活的对象	126
6.5.3 基于租约的生存期	129
6.5.4 远程对象类型概要	137
6.6 激活	137
6.6.1 RemotingServices.Connect 方法	137



6.6.2	错误消息	137
6.6.3	代理	138
6.6.4	消息	139
6.7	编组	139
6.7.1	ObjRef	140
6.7.2	传递对象	140
6.7.3	跟踪服务	146
6.8	异步远程	148
6.8.1	异步调用本地方法	148
6.8.2	异步调用远程方法	150
6.8.3	OneWay 方法	152
6.9	调用上下文	153
6.10	小结	155
第 7 章	无处不在的 Web 服务	156
7.1	应用程序方案	156
7.1.1	XML Web 服务	156
7.1.2	带有二进制格式化程序的 Web 服务	157
7.1.3	对等式的 Web 服务	157
7.2	驻留服务器	158
7.2.1	在 ASP.NET 中驻留 Web 服务	159
7.2.2	通道	167
7.2.3	Channel 接口	168
7.2.4	ChannelReceiver 接口	170
7.2.5	ChannelSender 接口	170
7.2.6	TCP 通道	171
7.2.7	HTTP 通道	172
7.3	格式化程序	172
7.3.1	串行化	173
7.3.2	二进制格式化程序	173
7.3.3	SOAP 格式化程序	173
7.3.4	比较格式化程序的结果	174
7.4	通道和格式化程序	178
7.5	小结	181
第 8 章	使用 .NET Remoting 建立 Web 服务	182
8.1	简述比萨投递示例	182
8.1.1	用例图	182
8.1.2	部署	183

8.1.3	程序集	184
8.1.4	类	184
8.1.5	订购披萨	185
8.2	数据服务	186
8.2.1	数据库图	186
8.2.2	存储过程	187
8.2.3	数据访问组件	189
8.3	PizzaOrder 程序集	192
8.3.1	PizzaOrderDetail 类	192
8.3.2	PizzaOrder 类	194
8.4	比萨订购 Web 服务	195
8.4.1	MarshalByRefObject 类	195
8.4.2	错误处理	195
8.4.3	远程方法	196
8.4.4	测试服务程序	198
8.5	PizzaDeliveryService Web 服务	199
8.5.1	投递比萨——次序	200
8.5.2	PizzaDeliveryService——远程对象类	200
8.5.3	用于两个远程对象的两个通道	201
8.5.4	支持事件	203
8.6	Windows 服务	209
8.7	小结	210
第 9 章	构建一个 .NET Remoting 客户程序	211
9.1	Web 服务	211
9.2	ASP.NET 客户程序	212
9.2.1	Web.Config 配置文件	213
9.2.2	远程服务和 WSDL	213
9.2.3	Soapsuds 实用程序	214
9.2.4	使用 soapsuds 实用程序创建代理	216
9.2.5	填充 DataGrid 对象	218
9.3	Windows Forms 客户程序	222
9.3.1	用户界面	222
9.3.2	客户程序通道配置	223
9.3.3	实现 Windows 事件处理程序	224
9.3.4	运行客户应用程序	227
9.3.5	添加事件处理	228
9.4	小结	229



第 10 章 通用描述、发现和集成(UDDI)	231
10.1 什么是 UDDI	231
10.1.1 Web 服务的发现机制	231
10.1.2 什么是 DISCO 文件	232
10.1.3 UDDI 构件(Building Blocks)	232
10.2 在 Web 上访问 UDDI	233
10.3 以编程的方式访问 UDDI	233
10.3.1 UDDI 数据结构	233
10.3.2 Microsoft UDDI SDK	238
10.3.3 构建一个查询	239
10.3.4 发布到 UDDI	242
10.4 小结	249
第 11 章 .NET 安全与密码技术	251
11.1 Web 服务安全	251
11.1.1 .NET 安全概况	252
11.1.2 .NET 安全框架 API	252
11.1.3 安全工具	255
11.2 Web 服务安全层	257
11.2.1 IP 安全	257
11.2.2 TCP 层安全	257
11.2.3 应用层安全	258
11.3 密码技术基础	258
11.3.1 Hash 算法	259
11.3.2 信用卡验证示例	282
11.3.3 增强信用卡验证示例的功能	300
11.4 小结	309
第 12 章 作为应用程序插件的 Web 服务	310
12.1 设计 GUI	311
12.2 编写代码	313
12.2.1 开端	314
12.2.2 向数据库添加数据	322
12.2.3 数据访问方法	328
12.3 版本化与发布的关系	331
12.3.1 在 WinForm 代码中创建类库	332
12.3.2 创建 Web 服务	332
12.3.3 创建 Windows 客户程序	335
12.4 小结	342

第 13 章 案例分析 1: ASP.NET	343
13.1 应用程序概述	343
13.1.1 业务过程	343
13.1.2 局限性	344
13.1.3 先决条件	344
13.1.4 系统的体系结构	344
13.2 实现	345
13.2.1 数据库设计	345
13.2.2 实现 Web 服务	351
13.2.3 创建代理类库	360
13.2.4 .NET 组件的实现	362
13.2.5 Windows 服务应用程序	369
13.2.6 实现库存查询服务	369
13.2.7 实现 WroxOnline Web 应用程序	374
13.3 整合所有内容	394
13.4 小结	394
第 14 章 案例分析 2: P2P .NET Remoting	395
14.1 任务和设计目标	397
14.2 进行开发的规划	397
14.3 标识组件	398
14.3.1 抽象化设计	398
14.3.2 消息设计	399
14.4 第 1 次迭代——样板通道接收器	399
14.4.1 ISpeak 接口	400
14.4.2 MyProxy.cs 文件	401
14.4.3 配置文件	403
14.4.4 TraceSink.cs 文件	405
14.4.5 运行示例	421
14.5 第 2 次迭代——串行化和 SOAP 封箱	422
14.6 第 3 次迭代——加密模型化	424
14.6.1 取消认证授权	424
14.6.2 签名和加密数据	424
14.7 第 4 次迭代——在通道接收器中应用密码技术	442
14.8 合并密码技术类	442
14.8.1 将公钥映射到 Web 服务	443
14.8.2 手动管理密钥存储	445
14.9 将内容合并——安全通道接收器	450



14.9.1	PeerClientSinkProvider 类	451
14.9.2	PeerClientSink 类	454
14.9.3	PeerServerChannelSinkProvider 类	459
14.9.4	PeerServerChannelSink 类	461
14.9.5	加密接收器概览	465
14.10	最后的集成	466
14.11	任务列表	472
14.12	应用程序的应用领域	472
14.12.1	P2P 应用程序的特性	472
14.12.2	P2P 即时消息传递	473
14.12.3	匿名支付系统	473
14.12.4	P2P 发票和订购系统	473
14.12.5	供应链的管理	474
14.13	小结	474
附录 A	.NET Remoting 对象模型	475
A.1	.NET Remoting 构件	480
A.2	远程驻留选项	481
A.3	远程传输通道	481
A.4	远程消息串行化和并行化格式化程序	482
A.5	带有租约远程生存期管理	483
A.6	远程激活策略	485
A.7	.NET Remoting 框架类	486

第1章 Web服务概述

如果您正在阅读本书，那么您一定会对 Web 服务的使用很感兴趣。也许您希望通过为第三方创建外部 Web 服务的方式来向市场提供您的业务解决方案；或许您正在考虑如何通过 Internet 上的现有 Web 服务更快地开发出更高效的软件解决方案；或许您对 Web 服务一无所知，只是想了解一下这项大家都在讨论的新技术。

本章将介绍 Web 服务的概念，以及为什么近来 Web 服务会受到如此青睐？Web 服务是分布式计算的一个重要里程碑，它很可能会彻底地改变当前企业软件解决方案的开发方式。

本章将介绍以下内容：

- 分布式计算的起源，以及它是如何演变为“软件即为服务”这一阶段的
- Web 服务是什么，为什么要使用 Web 服务
- Web 服务模型的优点
- Web 服务运作的标准
- 如何将 Web 服务与目前的企业体系结构进行集成
- Web 服务的前景

在本章的开始，我们先回顾一下分布式计算的起源。

1.1 分布式计算的历史

通过研究分布式计算的发展史，我们可以知道，Web 服务是自然发展的必然结果。随着时间的推移，应用程序越来越变得松散耦合，并被分成多个组件。这使应用程序的不同部分能够分布在多台不同的计算机上。这样，就可以利用多台计算机的资源向一个应用程序提供尽可能多的资源。

最初，分布式计算在商业方面的应用都是在大型计算机上进行的。这些昂贵的计算机可以处理通过哑终端进行登录的多个用户。这些哑终端本身并没有任何处理能力，即它们仅允许用户访问大型计算机上的资源。这种方式的优点是，哑终端可以分散在一栋建筑物中，这就意味着使用大型计算机的每个用户不必都挤在这台计算机前。相对于大型计算机的价格来说，那些哑终端设备真是太便宜了。但是应用程序还是驻留在大型计算机上，这样应用程序本身就要受到大型计算机的运算能力和资源的限制。

随着计算机越来越便宜，以及处理能力的不断提高，为每一个职员都配置一台计算机已成为了可能。这就改变了计算机的使用方式。目前由于人们可以很容易地使用这些功能相对强大的计算机，因此，人们就可以部分依赖这些计算机的处理能力，而无需过多依靠大型机去处理全部的请求。