

WCF 核心技术

Essential Windows
Communication
Foundation

Steve Resnick
[美] Richard Crane 著
Chris Bowen
鲁成东 戚文敏 译

Microsoft®
.net™
Development
Series

TP368.5
L084

WCF核心技术

Steve Resnick
[美] Richard Crane 著
Chris Bowen
鲁成东 戚文敏 译

人民邮电出版社

北京

图书在版编目 (C I P) 数据

WCF核心技术 / (美)雷斯尼克 (Resnick, S.), (美) 克兰 (Crane, R.), (美) 鲍恩 (Bowen, C.) 著; 鲁成东, 戚文敏译. —北京: 人民邮电出版社, 2009.10
ISBN 978-7-115-20548-3

I. W… II. ①雷…②克…③鲍…④鲁…⑤戚… III. 网络服务器—程序设计 IV. TP368.5

中国版本图书馆CIP数据核字 (2009) 第143991号

版 权 声 明

Authorized translation from the English language edition, entitled Essential Windows Communication Foundation, 9780321440068 by Steve Resnick, Richard Crane, Chris Bowen, published by Pearson Education, Inc, publishing as Addison-Wesley, Copyright © 2008 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and POSTS & TELECOMMUNICATIONS PRESS Copyright © 2009.

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签。无标签者不得销售。

WCF 核心技术

- ◆ 著 [美]Steve Resnick Richard Crane Chris Bowen
- 译 鲁成东 戚文敏
- 责任编辑 刘映欣
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
- 邮编 100061 电子函件 315@ptpress.com.cn
- 网址 <http://www.ptpress.com.cn>
- 三河市海波印务有限公司印刷
- ◆ 开本: 800×1000 1/16
- 印张: 28.75
- 字数: 591千字 2009年10月第1版
- 印数: 1—2500册 2009年10月河北第1次印刷

著作权合同登记号 图字: 01-2008-3843号

ISBN 978-7-115-20548-3

定价: 65.00元

读者服务热线: (010)67132705 印装质量热线: (010)67129223

反盗版热线: (010)67171154



内容提要

本书全面系统地介绍了如何使用 WCF 进行 Web Service 编程。对于与 Web Service 相关的领域，包括最简单的 SOAP-HTTP 绑定、各种高级 WS-* 扩展协议，以及各种数据编码、消息交换模式，书中都给出了用 WCF 实现的方法和例子。本书还介绍了通过 WCF 用各种方式来扩展原类的 Web Service 服务，使用各种不同的绑定方式在不同的网络环境甚至本机上部署服务等内容。另外，专门讲述了问题诊断与异常处理，还涉及了与微软工作流（WF）和热门技术 AJAX 的集成。

本书示例丰富，讲解透彻清晰，主要面向微软平台上的开发者。对于已有 C# 基础的 WCF 初学者或具有一定经验的 WCF 开发者，本书也大有裨益。另外，对于正在使用其他技术（如 Java）而希望进入 WCF 领域，或者想对其有所了解的 Web Service 技术人员，本书也是适用的参考资料。



对本书的赞誉

“Resnick、Crane 和 Bowen 对微软的 Web Services 平台非常精通。不论你对 WCF 是首度使用还是已驾轻就熟，本书都会对你有所裨益。”

——Nicholas Allen，微软公司 Web Services 项目经理

“我们开发者常被要求在许多领域成为‘即时专家’。如果有一天你要在微软公司 .Net 3.x 的新天地中开始做分布式系统开发和通信，就将面对这个名为 WCF 的庞然巨兽。那时，你若渴望有秘籍在手，那么非本书莫属。”

——Ron Landers，IT Professionals 公司高级技术顾问

“设计和编写分布式应用是 .Net 开发者和架构师面临的最繁复艰深的挑战之一。采用何种技术？选择何其多，而编码时间又何其少也。WCF 解决了这个问题，那就是用统一的平台为 .Net 构建分布式应用。与其他分布式系统类似，WCF 兼具多种选择和可能性。本书用一种浅显易懂的方式以真实的示例解释了选择的范围。基于 WCF 基本知识，书中对当前该如何应用 WCF 给出了解答。无论开发者和架构师构建何种类型的分布式应用，本书都是必读之物。”

——Thom Robbins，微软公司 .NET 平台产品管理部门主管

“本书内容非常翔实，对技术的讲解清晰易懂，面面俱到。无论你是 WCF 的旧雨还是新知，此书都可称为无价之宝。”

——Willy-Peter Schaub，Barone、Budge 及 Dominick 公司技术专家，微软公司 MVP

“很显然，作者于分布式应用开发一道浸淫多年，才能对 WCF 萃精撷华得此一书。书中有大量实用信息，可以指导用户在 WCF 项目中节约时间，仅‘分析诊断’一章，就足以令用户省下大量用于对艰难问题的调试的时间。极度推荐此书。”

——Yasser Shohoud，微软公司技术中心达拉斯技术主管



现在是 2007 年 12 月,我正在写这篇序言。不久前,是 Windows Communication Foundation (WCF) 作为 .NET 框架 3.0 版本的一部分首次发布一周年。不到一个月前,我们刚刚在 .NET 框架 3.5 版本中为这个平台提供了重大的新功能。毫不夸张地说,在这两个版本中有海量的知识等待使用者理解和吸收。

在 Microsoft 平台上可以构建各种类型的分布式应用, WCF 的目的之一,就是将所有与之相关的开发经验加以整合。我们需要一组核心概念,简单实用同时又具备足够的表述能力,来描述所有我们打算替代的技术所蕴含的语意。在此之前出现的各种 Microsoft 的协议栈 (ASMX、Remoting、COM+、MSMQ 及 WSE),可谓各有优劣。我们的使命则是继承发扬那些工作良好的理念,同时从失败中汲取教训。一旦成功,开发者就可以用它编写多种不同类型的分布式应用,而不需要学习多种各不相同(而且常常差异极大)的编程模型。

为了让这种统一开发体验能够梦想成真,我们需要一个高度灵活的运行架构,以匹配丰富多样的编程模型。关键的可变领域必须被辨识出来,抽象成可扩展的机制,以避免对新平台造成无谓的功能限制。当默认行为无法满足某个应用的要求,或者特定场合需要的功能缺失时,运行架构一定要确保外部开发者能够很自然地在架构中插入定制的解决方案。

对我而言, WCF 中最激动人心的部分在于其应用场合之广泛令人惊讶。我们在 .NET 3.5 中提供的功能毋庸置疑地证实了这一点。这一版本中推出了两种平行的方案,可以覆盖极其丰富的分布式应用场景。一种用于将 WCF 与 Windows Workflow Foundation 的强大功能集成,为长期运行的、声明式 (declarative) 的、相互关联的商业流程提供了一个底层平台。另一种则扩展了 WCF 的外延,以应对今天不断发展的互联网的需求。这些应用场合对运行架构和编程模型都有着独特的要求, WCF 能通过扩展来满足这些要求,而不必对已有的实现进行大的修改。这很好地证明了 WCF 架构在长远的将来仍能够满足分布式应用不断发展的需要。

因为我们的首次发布仅过去了一年,所以看到实际中客户在我们平台上投入之大,着实令人兴奋。更令人兴奋的则是了解到通过这些对 WCF 的投入,他们在开发效率、性能以及互通性等方面获得的收益。在我们判断平台成功与否的标准中,用户的成功是第一位的,也是最重要的,根据这个标准, WCF 将会是一个非常成功的平台。

实际上，这只不过是以一种委婉的说法来说明将时间花在学习 WCF 上是很值得的。从这点来看，现在拥有本书是非常幸运的。Rich、Chris 和 Steve 的工作非常出色，他们从繁冗中将 WCF 的精髓提炼出来，使用户能够在此平台上卓有成效地工作。作者兼具技术敏锐度、行业经验以及同（WCF）产品组的紧密关系，这些独一无二的优势无疑将使得本书足以在所有 WCF 开发者书架上占据显著位置。让这些家伙来阐述我们的产品技术，我无比开心。我坚信，读完此书你也将会有同感。

Steve Maine

美国华盛顿州西雅图市

2007 年 12 月



前 言

Windows Communication Foundation (WCF) 是统一的编程模型，用来为微软平台编写分布式应用。它涵盖了之前出现的多种技术，如 ASMX、.NET 远程调用、DCOM 和 MSMQ 等，并提供了一套可扩展的 API，以满足分布式计算那变化多端的需求。在 WCF 出现之前，用户必须对多种技术一一掌握，才能为特定的分布式应用需求选择一种正确的实现方式。WCF 提供的统一实现途径很大程度上简化了这种选择过程。

基于 XML 的 Web 服务是当今应用最普遍的分布式计算技术，这些服务被用于在私网或公网上发布技术性和商业性的功能接口。服务或者使用 SOAP 规范，或者使用其他协议，它们用来传送信息的典型格式是内含尖括号的文本文档，但有例外。这些服务通常使用 HTTP 作为传输协议，但仍有例外。作为一个用于这种基于 XML 的 Web 服务的框架，WCF 与大多数技术协议栈兼容。

Rich、Chris 和我都是从 .NET 出现伊始（1999 年左右）就开始为其进行编码的。我们为微软公司工作，帮助客户使用 WCF 解决实际问题。我们的客户来源广泛，从跨国公司到独立服务提供商（ISV），乃至网络创业公司，他们各有各的问题、需要和优先权，这些我们都要一一应对。我们向他们展示可行之道，推荐运转良好的方案，避开失败。在构建分布式应用方面，我们经验丰富，并据此向他人传授 WCF 的知识。

我们写这本书的目的，是希望以一种可以即学即用的方式向软件开发者展示 WCF。我们提供的材料细节翔实，足以令你在使用各种不同的功能时知其然更知其所以然。很多情况下，我们甚至进一步描述了框架中的某些细微之处，不过也没有琐碎到介绍 API 文档的地步。

Blog 圈中有大量的 WCF 细节资料，其中许多来自 .NET 产品组，还有许多来自其他的一直学习这项技术的开发者。我们大量使用了 Blog 作为资料来源。本书对这一知识库所做的组织与整理，令你可以在书桌前、沙发上或任何习惯阅读的地方轻松地汲取这些知识。

目标读者

本书的目标读者是那些希望在.NET 平台上构建分布式应用的软件开发者。作为资深开发者,我们深知对于学习使用一项新技术来说,可靠的建议和清楚的范例有多重要。我们遍寻 Blog 圈,穷搜微软内部邮件,写下大量代码,就是为了提供最好的例子,助读者为其所欲为。

那些需要了解 WCF 的架构师也可以从本书受益。书中各章节涉及 WCF 的多个方面,涵盖了基本概念、绑定、信道、行为、托管、工作流及安全,描述了使用 WCF 设计与实现服务的各个重要方面。阅读这些章节开头短短数页的介绍文字,是概览这门技术的最好方法。

写作本书的目的之一是期望缩短你的 WCF 学习过程。我们描述并演示了如何完成常见任务,介绍了基本概念,也涉及了高深话题。在本书中,主题是以一系列有待解决的问题呈现的。本书介绍了如何使用 WCF 完成目标,而不是罗列 API。

阅读本书需要的前提适中。如果你对 WCF 感兴趣,大概在.NET 上已颇有根基;你很有可能通晓 C#或 Visual Basic,要不然至少也曾经一度如此;而且,理所当然地,你大概也对 Visual Studio 驾轻就熟。故而我们假定你能够编写像样的.NET 代码,并且乐于花费时间来精通 WCF。

安装要求

WCF 是微软.NET 框架 3.x 版本的关键组件。WCF 首次发布是随.NET 3.0 一起,而后在.NET 3.5 中发布了增强版本。第二个发行版本中增加的内容还算适度:对非 SOAP 的 Web 服务的增强,对 WCF 和 WF 的集成,以及一个健壮的补丁包(service pack)。本书针对.NET 3.5,除非有特殊原因非要使用更老的版本,否则明确推荐这个版本。

.NET 以两种形式打包:可以二次发布的运行库和软件开发包(SDK)。运行库预计安装在目标机器上,这些机器不用于开发,其中包括测试、打包和生产环境等。而 SDK 则应安装在用户的开发机上。SDK 中有代码范例、文档及开发辅助工具。无论是运行库还是 SDK 版本的 .NET 安装包,都可以从微软公司的 MSDN 网站上下载,网址为 <http://msdn2.microsoft.com/en-us/netframework/default.aspx>。Visual Studio 2008 中也包含.NET 3.5 的 SDK。微软.NET 框架 3.5 版本可以安装的操作系统包括 Windows XP SP2、Windows Vista、Windows Server 2003 及 Windows Server 2008。

内容结构

我们并不指望你能从头至尾读完本书。如果对 WCF 所知甚少,那么可能需要首先阅读第 1 章,并尝试其中的例子。之后的每一章依次涵盖了 WCF 的一个主要功能集。每章中都

有几页介绍性文字，描述引入所介绍功能的动机与某些设计目标，然后我们会在该章的后续阐述细分的主题。

第1章“基础”阐述了构建和调用 WCF 服务相关的基础知识。在这一章中，讨论并展示了如何实现不同类型的接口，以及如何在各种接口中进行选择。读完本章后，你将有能力使用 WCF 来产生和调用服务。

第2章“契约”介绍了 WCF 中契约的3种主要类型：服务契约、数据契约和消息契约。每种契约都可以用来在代码中定义复杂的数据结构和接口。数据契约将.NET 数据类型映射到 XML，服务契约通过 WSDL 以可跨平台调用的方式发布服务接口端点，而消息契约则让开发者可以直接操作消息中的 XML 而不是操作.NET 数据类型。对每种契约，WCF 的工具集都会生成基于标准的 WSDL 并发布到外界。

第3章“信道”讲述的内容包括信道和信道协议栈。信道模型架构是 WCF 通信框架得以构建的基础。信道架构使得在客户端与服务间发送和接收消息成为可能。信道协议栈可以严格地按需构建。

第4章“绑定”描述如何配置通信协议栈以便严格按需使用协议。例如，如果通信在企业内部进行并且不会跨越防火墙，同时需要最快的性能，使用名为 `netTcpBinding` 的绑定就能达到目的。而如果期望能与流行的 Web 客户端通信，则 HTTP 与文本编码的 XML 就是必需的，故此 `basicHttpBinding` 将是不二之选。“绑定”其实就是预先配置好的信道协议栈。

第5章“行为”讲解服务的“行为”。在 WCF 中，“行为”是指在实际的消息处理过程之外影响服务操作的一种机制。从一条消息被接受开始，直到它被发送给服务操作代码之前的一切动作都属于“行为”的范围。在 WCF 中，并发、实例管理以及事务支持都是在此时被处理的。这一章还展示了如何通过构建定制的“行为”为服务增加额外的控制流程。

第6章“序列化与编码”描述了数据被从.NET 类型（类）序列化成一组 XML 信息集（Infoset）的过程，以及如何在通信中表示 XML 信息集。通常认为 XML 是文本文档，内有尖括号包围的字段名称和取值，而 XML 信息集则是一种更基本的数据结构。这一章讨论了各种方法，可以将这种结构转化为一种能够通过网络交换的格式。

第7章“托管”讲述了托管 WCF 服务的多种多样的选择。其中描述了最常见的托管环境 IIS，但这并不是唯一选择。WCF 服务的托管环境可以是受控.NET 应用程序，可以是 Windows 进程激活服务，也可以是任何.NET 程序。这一章讨论了托管相关的可选方案和技术。

第8章“安全”篇幅很长，涉及了多方面的安全选项。其中讨论并展示了不同的认证模式，对传输级和消息级安全做了比较并各举多例，还描述了内联网和因特网的运行场合。

第9章“诊断”描述如何使用.NET 内建的追踪工具捕获 WCF 事件，举例讲解了如何设置追踪监听器以监听各种不同的事件。追踪查看器是一个强大的工具，它是随 WCF 一起提供的，这一章讲解了使用追踪查看器追踪跨服务调用边界的获得。

第 10 章“异常处理”在 WCF 内部如何处理异常方面给予了读者务实的指导。SOAP 错误可以通过错误契约来描述,而示例则演示了如何抛出和捕获这些错误来尽可能地减少失误。

第 11 章“工作流服务”。在 Visual Studio 2008 和 .NET 3.5 中引入了 Windows 工作流基础库 (WF),这一章讲解了集成 WCF 与 WF 的关键点,还描述了如何从 WF 调用 WCF 服务,以及如何在 WCF 中对外发布 WF 工作流。

第 12 章“对等网络”。网络啮合 (mesh) 技术令客户端之间可以彼此搜寻,这一章展示了如何利用网络啮合构建客户端到端应用,还阐述了啮合寻址 (mesh addressing) 以及客户端地址被解析后如何建立点到点的连接。

第 13 章“可编程 Web”讲述如何使用 WCF 实现非 SOAP Web 服务,所举示例展示了如何使用 AJAX,以及 JSON 这种更简单且 JavaScript 友好的数据格式。此外,这一章还描述了专门用于非 SOAP 托管类,像 WCF-WF 集成一样,这也是 .NET 3.5 新引入的。

最后是附录“高级主题”,涉及了一些不适合放入其他章节的高级话题。我们觉得与其让它们埋没于不当之所,不如令其独立门户。

鉴于 WCF 这个主题的广泛性,并非所有话题都以同样的深度展开。本书的目标是帮助开发者使用 WCF 工作时能获得超高的效率,读者将可以通过本书来学习这项技术。本书并不试图成为 WCF 的文档,微软公司那些优秀的技术作者已经在帮助文件和 MSDN 里写过了,但是将那些文档与本书结合使用,将会很好地指引开发者用 WCF 快速有效地构建健壮的应用程序。

注意: 代码延续箭头

当一行代码过长而无法放在一行文字中时,我们将其折行到下一行。这时,会在续行之前加一个代码延续箭头的标志 (➞)。



致 谢

经由许多人的努力，本书方得以面世。两年多以前，Indigo 1 刚刚发布第一个公开测试版本时，我们开始了本书的写作。从那时以来，我们编写了示例代码，每次 WCF 更新版本后，还要对示例进行测试和改进，最后一次使用的是 .NET 3.5 和 Visual Studio 2008。除了忙于编码以外，我们还撰写了本书，不过这也正是使用改变如此之快的技术的乐趣所在。

如果没有来自 WCF 产品组与其他一些微软公司才智之士的大力支持，本书将成为泡影。他们每人都查阅了文字与代码，在我们离题之时予以匡正。我们谨对下列人士致以谢意：Wenlong Dong、Bill Evjen、Steve Maine、Doug Purdy、Ravi Rao、Yasser Shohoud 和 David Stampfli，感谢他们付出的时间、想法和耐心。

我们还要感谢那些技术评论家，正是他们的评论和意见使得本书更臻完善。我们很幸运，得到这些顶级 WCF 专家的帮助。致评论家 Nicholas Allen、Jeff Barnes、Ron Landers、Sowmy Srinivasan、Tom Fuller 及 Willy-Peter Schaub，请接受这份对你们所给予的帮助的谢意。我们要特别对 John Justice 表示感谢，他搜遍产品组帮助我们找到评论者。同样，特别感谢 Thom Robbins 教我们使用通俗易懂的语言写作。

此外，还必须感谢 Liam Spaeth 以及整个微软技术中心团队对我们的支持。感谢世界范围内的全体 MTC 团队和波士顿 MTC 的同事与客户，多谢他们提出的想法。

除了微软公司的人以外，我们还要感谢 PluralSight 公司的 Keith Brown 和 Matt Milner，多谢他们对安全与工作流部分的详细审阅。这两个主题对于我们十分艰深与陌生，因此我们非常需要从他们的专长中得到帮助，结果是获益匪浅。在最后，感谢 Addison-Wesley 出版公司的优秀员工，是他们将所有内容整合在了一起。我们也许了解如何编程和写作，但他们更了解如何制作一本图书。为此，多谢 Joan Murray、Betsy Harris 以及整个团队。



关于作者

Steve Resnick 从 20 世纪 90 年代中期开始就在微软公司工作，担任过从架构师到开发人员以及所在领域中的推广专家等职务。他的专长是因特网技术，精于设计构架高容量高价值的 Web 应用。Steve 是美国微软技术中心的国家级技术总监，他负责制定技术策略与方向，指引其团队满足最棘手的客户要求。他开始使用 .NET 的时间可以上溯至其刚出现时，他是 Web 服务、BizTalk、事务处理以及相关技术方面的专家。他在 Delaware 大学和波士顿大学分别获得了计算机学士与硕士学位。

Rich Crane 是一名技术架构师，在位于马萨诸塞州沃尔瑟姆的微软技术中心工作。Rich 是拥有超过 18 年工作经验的软件架构师和工程师，最近 6 年来一直致力于帮助客户在微软平台上设计和构建解决方案。他曾使用无数微软产品和技术工作过，他是 BizTalk、SQL Server、SharePoint、集群计算服务器以及 Visual Studio 和 .NET 框架方面的专家，曾多次在诸如 TechEd 和 Code Camp 的会议与集会上发表演讲。他以最优等的成绩毕业于 Drexel 大学，获得了电子与计算机工程专业学士学位。

Chris Bowen 是微软公司美国东北区域的开发推广专家，专长于开发工具、平台以及最佳架构实践。Chris 拥有 15 年的软件架构师和工程师工作经验，他在加入微软公司以前曾在 Monster.com、VistaPrint、Staples 和 IDX Systems 等公司任高级职位，从事网站及电子商务项目的咨询工作。他是 *Professional Visual Studio 2005 Team System* (WROX, 2006) 一书的作者之一。他在 Worcester 工学院获得了管理信息系统专业学士学位以及计算机科学硕士学位。



目 录

第 1 章 基础	1	1.7 小结	26
1.1 为何选择 WCF	1	第 2 章 契约	28
1.2 简介	2	2.1 服务契约	30
1.3 实现 WCF 服务	5	2.1.1 同步请求—响应操作	31
1.3.1 服务的 ABC	5	2.1.2 异步请求—响应操作	34
1.3.2 完全用代码编写 WCF 服务	5	2.1.3 单向操作	36
1.3.3 使用代码和配置文件编写 服务	7	2.1.4 双工操作	37
1.3.4 再谈配置文件	9	2.1.5 单个服务中的多契约与 多端点	44
1.3.5 再谈服务托管	10	2.1.6 在 WSDL 中如何命名操作、 类型、行为及命名空间	47
1.3.6 暴露元数据交换端点	11	2.2 数据契约	49
1.4 实现 WCF 服务的客户端	14	2.2.1 为 .NET 类定义 XML Schema	51
1.4.1 完全用代码编写 WCF 客户端	14	2.2.2 定义类的层次结构	53
1.4.2 使用代码和配置编写 WCF 客户端	15	2.2.3 用 KnownType 属性在 WSDL 中发布其他数据类型	55
1.5 在 IIS 中托管服务	18	2.2.4 数据契约的版本控制	60
1.5.1 讨论	19	2.2.5 数据契约等价性	62
1.5.2 3 步完成 IIS 中的服务托管	19	2.2.6 使用容器	63
1.6 实现 ASMX 服务的 WCF 客户端	22	2.3 消息契约	64
1.6.1 支持工具	22	2.3.1 有类型消息	65
1.6.2 生成客户端代理类和配置 文件	23	2.3.2 无类型消息	68

2.3.3 使用无类型消息的 SOAP	
消息头	70
2.4 小结	72
第3章 信道	74
3.1 信道范型	76
3.1.1 单向通信模式	76
3.1.2 双工通信	77
3.1.3 请求—响应通信	78
3.1.4 范型变换	80
3.2 操作契约与信道范型	80
3.3 信道监听器	81
3.4 信道工厂	83
3.5 ICommunicationObject	86
3.6 小结	89
第4章 绑定	90
4.1 选择合适的绑定	93
4.2 .NET 应用程序间的跨机通信	99
4.3 .NET 应用程序间的本机通信	102
4.4 使用基本的 Web Service 进行通信	104
4.5 使用高级 Web Service 进行通信	107
4.5.1 wsHttpBinding	108
4.5.2 ws2007HttpBinding	110
4.5.3 wsDualHttpBinding	112
4.6 绑定的性能与可伸缩性的比较	119
4.7 使用队列服务进行通信	121
4.7.1 netMsmqBinding	122
4.7.2 msmqIntegrationBinding	130
4.8 创建定制化绑定	132
4.9 绑定元素	135
4.9.1 传输机制	135
4.9.2 编码器	136
4.9.3 安全	137
4.9.4 传输机制的增强/助力	137
4.9.5 范型变换	138
4.9.6 其他协议	138
4.10 通过多种绑定发布服务契约	139
4.11 小结	140
第5章 行为	141
5.1 并发与实例化(服务行为)	143
5.1.1 默认的并发、实例化设置与无会话的绑定	144
5.1.2 单实例使用多线程	146
5.1.3 实现单例	147
5.1.4 会话级实例	149
5.1.5 控制并发实例的数量	151
5.1.6 控制并发调用的数量	155
5.1.7 控制并发会话的数量	157
5.2 导出和发布元数据(服务行为)	159
5.3 实现事务(操作行为)	162
5.3.1 服务中的事务化操作	163
5.3.2 在操作之间的流转事务	168
5.3.3 选择事务协议——OleTx或 WS-AT	173
5.3.4 事务服务行为	174
5.4 实现自定义行为	176
5.4.1 为服务端点行为实现消息检验器	178
5.4.2 作为属性暴露服务操作行为的参数检验器	180

5.4.3 通过配置文件暴露服务 行为	183	第 7 章 托管	230
5.5 行为安全	187	7.1 在 Windows Process Activation Services 中托管服务	231
5.6 小结	187	7.2 在 IIS 7 中托管服务	233
第 6 章 序列化与编码	189	7.3 在 IIS 托管的服务中启用 ASMX 特性	235
6.1 序列化 vs 编码	189	7.4 自托管	240
6.2 比较 WCF 中可以选择的序列化 方式	190	7.5 在 Managed Windows 服务中 进行自托管	241
6.2.1 DataContractSerializer	191	7.6 在单进程中托管多个服务	244
6.2.2 NetDataContractSerializer	194	7.7 定义服务和端点定制	247
6.2.3 XmlSerializer	195	7.8 小结	249
6.2.4 DataContractJsonSerializer	198	第 8 章 安全	251
6.2.5 选择序列化器	199	8.1 WCF 安全概念	251
6.3 保留引用与循环引用	200	8.1.1 认证	251
6.4 用 NetDataContractSerializer 共享类型	205	8.1.2 授权	252
6.5 用 IExtensibleDataObject 进行 往返式序列化	209	8.1.3 机密性	252
6.6 用替身对类型进行序列化	214	8.1.4 完整性	252
6.7 为大型数据使用流模式	220	8.1.5 传输和消息安全	252
6.8 用 XmlSerializer 定制序列化	221	8.2 基于证书加密	253
6.8.1 用属性定制 XML 序列化	222	8.2.1 概念	254
6.8.2 用 IXmlSerializable 定制 XML 序列化	223	8.2.2 设置	254
6.9 选择编码	225	8.3 传输级安全	256
6.9.1 文本编码 vs 二进制编码	225	8.3.1 用 SSL 加密	256
6.9.2 通过 MTOM 编码发送 二进制数据	227	8.3.2 客户端认证	259
6.9.3 了解 WebMessageEncoder	228	8.3.3 服务的身份	264
6.10 小结	228	8.4 消息级安全	266
		8.5 用 Windows 内置安全机制加强 服务安全	271
		8.5.1 本节例子简介	271
		8.5.2 用 Windows 证件认证用户	273

8.5.3 使用 Windows 证件为用户授权	276	9.5.4 配置监听器	312
8.5.4 用 AzMan 做授权	278	9.6 服务踪迹查看器	313
8.5.5 扮演用户	282	9.6.1 活动视图	314
8.6 加强互联网服务的安全	286	9.6.2 项目视图	315
8.6.1 ASP.NET 集成	287	9.6.3 消息视图	315
8.6.2 用成员资格供应器进行认证	288	9.6.4 图形视图	316
8.6.3 用角色供应器进行基于角色的授权	291	9.6.5 分析来自多个源的日志	317
8.6.4 使用表单认证	293	9.6.6 过滤结果	319
8.7 日志与审计	297	9.7 小结	320
8.8 小结	299	第 10 章 异常处理	321
第 9 章 诊断	300	10.1 WCF 异常处理简介	321
9.1 WCF 应用程序范例	300	10.1.1 SOAP 通信中的 WCF 异常	322
9.2 追踪	301	10.1.2 未处理异常的例子	322
9.2.1 端对端追踪	301	10.1.3 检测和恢复出错的信道	325
9.2.2 活动和相关性	302	10.2 通知异常的细节	326
9.2.3 启用追踪	303	10.3 通过 FaultException 管理服务异常	327
9.2.4 详细程度方面的建议	304	10.4 简单 FaultException 的限制	330
9.3 消息日志	304	10.5 创建和使用强类型的故障	331
9.4 额外的配置选项	306	10.5.1 在 FaultContract 中声明故障的定义	331
9.4.1 共享监听器	306	10.5.2 定义 FaultContract	332
9.4.2 消息过滤器	307	10.5.3 以定义的 FaultContract 抛出 FaultException	333
9.4.3 踪迹源自动刷新	308	10.5.4 故障契约策略	334
9.4.4 性能计数器	308	10.6 实现客户端故障处理器	334
9.4.5 Windows Management Instrumentation	309	10.7 错误处理应用程序块	336
9.5 使用服务配置编辑器	309	10.8 小结	337
9.5.1 追踪选项	310	第 11 章 工作流服务	338
9.5.2 日志选项	310	11.1 集成点	339
9.5.3 配置源	312		