

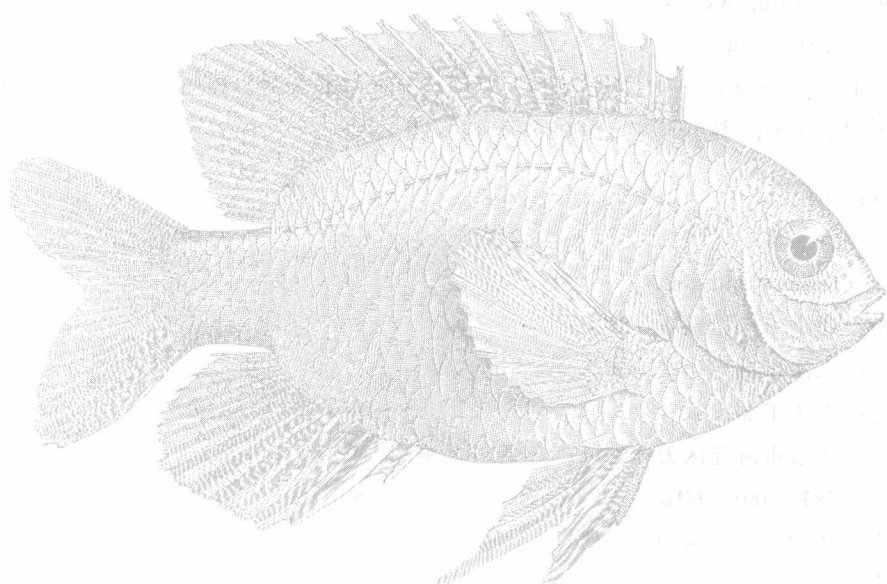
O'REILLY®

学习 WCF：从原理到实践

Learning WCF

Michele Leroux Bustamante 著

庞引明 侯伯薇 译



電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

O'Reilly Media, Inc.介绍

为了满足读者对网络 and 软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权电子工业出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是在线出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为20世纪最重要的50本书之一）到GNN（最早的Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的Web服务器软件），O'Reilly Media, Inc. 一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为 O'Reilly Media, Inc.紧密地与计算机业界联系着，所以O'Reilly Media, Inc. 知道市场上真正需要什么图书。

本书获得的赞誉

“……在你手中（计算机中，或者其他媒介中）的是一本书，作者是一位 MVP 和 RD，并且是一位 WCF 数字精英（我现在敢告诉你了）。该作者实践经验丰富，交流手段精湛，正如你对 WCF 数字精英所期待的那样与产品小组联系颇深。这一切造就了一本众望所归的力作……它是通往 WCF 专家阵营的阳关大道。买一本，带回家，好好从中汲取营养吧。”

——Steve Swartz，微软互联系统部架构师

“就像维吉尔（Virgil）引领但丁（Dante）穿过地狱湖那样，Michele 也将带领你们遍览 WCF，由浅入深，渐入佳境。关注实用和真实的案例，避开神秘莫测的陷阱……当你完成的时候，你将成为 WCF 专业开发者群体中的一员，这些人能够以自己喜欢的方式来使用 WCF。”

——Ted Neward, <http://www.tedneward.com>

“仅仅是契约和安全性章节的价值就已超过你购书的付出了！”

——Sam Gentile，V5 算法公司软件架构师&技术主管

“Michele 做了一件富有成效的工作，以一种平易近人的、实践性很强的方式带领大家走近 WCF。与此同时，本书讲授了技术内部的工作原理，并从一开始，就为更复杂的场合提供实用的解决方案。该书令人震撼！”

——Rick Strahl，西风技术公司

“买这本书吧，我是认真的。我们（微软）对微软通信架构 WCF 的设计目标是，使得分布式系统更加易于构建，并且将你在此过程中需要掌握的技术种类数量从半打减少到唯一。（WCF）作为一种编程方式，我们相信已经取得成功，而使一个飞行器更易于飞行还意味着你必须学会如何来操纵它。分布式系统也一样，我们已经降低了复杂度，不过你还是需要学会如何构建分布式系统，而这正是 Michele 这本书的切入点。我们实现了这个架构，Michele 展示并讲解这一架构——以一种绝妙的方式——如何将所需要的组成部分钩联在一起用于你的解决方案。买这本书吧，我是认真的。”

——Clemens Vasters，微软互联系统部高级程序经理

For my dad

从 1996 年 Gartner Group 提出 SOA 概念到现在，已经有十多年了。SOA 已经从最初少数预言家在纸上画出一块理想的饼，渐行渐近，变成了现实的系统。

SOA 不是一种产品，也不是一种技术，而是一种理念，一种愿景；是软件工程和系统架构的一种理想情形，即将业务变更与系统实现分开，在表征业务逻辑的服务之间实现松耦合。可以说，SOA 成了构建综合业务系统的一种诸多理想状态的集大成者。

按照 SOA 的典型层次，一个综合的应用系统，总是从业务建模开始，再映射到服务的描述构建，之后会将服务进行层层粒度细化，直到最后的系统实现，与数据库系统建立存取关系。应该说，SOA 与具体的实现技术是无关的。不过具体到要实现 SOA 理念的厂家来说，就一定是与技术有关的了。没有厂家会抛弃其现有技术框架，去实现 SOA 的经典 5 层结构。因此，几乎所有的厂家都将 SOA 实现诠释为一种类似中间件的技术。于是，现有的 SOA 架构就演化成两个核心：服务的描述和解析实现，以及服务间的通信技术。XML 被作为描述服务的工具和消息编码的格式，成为了不二的选择。

本书就是在这样的背景下，描述在微软的 .NET 架构上实现 SOA 框架的技术基础：WCF。在微软内部，WCF 被称为 Indigo。事实上 WCF 的全称是 Windows Communication Foundation，它其实是 .NET 中支持 SOA 的一个通信架构。在 WCF 的 3 个大写字母中，核心是“C”，也就是通信。前面已经讲过，在当前的 SOA 架构中，就做了两件事情，一件是服务的描述和处理，另一件就是通信机制。

具体到 WCF 中，这个通信机制就是，通过契约来选定通信过程中所使用的协议，通过建立信道和提供消息流机制来实现客户和服务之间的请求/响应过程。其他相应内容都是围绕着这个中心展开的。有了服务就会谈到寄存，契约约定了协议就需要绑定，为了提高吞吐量就需要并发机制，为了保障通信就必然需要安全性、可靠性和故障处理等。

本书在写作上的最大特点就是，十分注重原理的实践性。通过贯穿全书的 lab（操作实例），读者可以一边理解 WCF 原理一边动手实践。读完本书，基本上就可以动手编程了。

需要说明的是，本书讲述的是在 .NET 架构下 SOA 的核心支撑技术 WCF 的机制。对于准备在 .NET 上构建 SOA 应用系统的开发者来说，本书必不可少。不过对于一般 SOA 工作者来说，更应该在阅读本书的基础上，进一步广泛地了解其他厂家的相关产品和实现机制，以求更加全面深入地理解 SOA 架构的精髓。无论如何，在众多技术中，首先阅读理解 WCF 还是很有益的。

翻译本书是一件艰苦卓绝的工作，尤其是要跨越一个春节和近年来少见的严冬。正是对 SOA 的热爱和推崇才使得我们在工作之余坚持完成了对本书的翻译，然而时间仓促，理解有限，在此斗胆献上我们的翻译作品，一是为了使阅读原版有困难的读者也有机会了解该书内容，二是为了与业内同仁共同学习和探讨相关技术。

感谢出版社的编辑们，徐定翔、何艳、晓菲和刘美惠都为本书的翻译出版倾注了大量心血，他们时刻鼓励着我们以更大的责任心投入到本书的翻译中。

我们的家人也在此过程中给予了我们更多的亲情、理解和宽容，使得我们能够在温暖的严冬里完成这一翻译作品。

我们作为本书的翻译者，对于本书翻译中的错误负全部责任。而且，我们也认为，能够有这样的机会与业界同仁共同探讨，是极为难得的机会。欢迎读者朋友们给出建议、批评和指正。

庞引明

侯伯薇

序

Forewords

嗨，我是 Steve Swartz，Michele 请我为她的新书《学习 WCF》写前言，因为我是 WCF V1 的两位广度架构师之一。这显示了一个职位要给人留下深刻的印象是多么容易。我仅仅是软件开发工作中的普通一员。我们部门的技术伙伴——架构师 Brad Lovering，构想出了 WCF，并且指导我们小组的技术工作。我们部门的杰出工程师——架构师 Eric Christensen，设计和实现了 WCF 的核心部分。像 Steve Millet 和我这样的广度架构师负责将 WCF 的各个部分整合成一个统一的整体。WCF 的每一部分，都在一个深度架构师指导下，由一个包括开发人员、产品经理、测试人员和文档工程师的小组负责开发出来。值得称道的是，这 200 多个人中的任何一员，都不会炫耀自己对 WCF 做出的技术贡献。其实，大部分的微软产品组成员都不爱张扬，我们是一个害羞的群体。那些你们最终认识的人就是那些足够外向而出头露面的人。

微软产品小组为潜在的用户群体考虑颇多。我没有做过专门统计，但是，从个人角度来看，我敢确定，当我们开发 WCF 时，WCF 就已经有了数百万活跃的用户。包括：使用 COM、DCOM、COM+ 和 MSMQ 的人们；使用 ASMX、.NET Remoting、Enterprise Services 和 System.Messaging 的人们；使用 WSE 的人们；还有使用 WCF 早期版本的人们。即使尽我们所能，也不可能与所有这些潜在用户们协同交互以完成我们的日常工作。我们的替代策略是，花一些时间和潜在客户群中有影响力的核心成员保持协作，期待他们能够把知识传递给更多的人。我们花时间和微软的“地区总监”（RD）们密切合作。这些总监们遍布世界各地，和微软员工及其他开发者团体有着良好的关系。另外，我们也与 WCF 的数字精英相与甚欢，他们是一支 10 到 20 人的小分队，是这个星球上最聪明最活跃的 WCF 用户。

既然大家出于厚爱，称我为 WCF 产品小组中 WCF 数字精英的第一人，我就有义务告诉大家更多有关这一神秘团队的事情。WCF 数字精英团队的确切成员数目尚属未知数，即使对于 WCF 小组的管理层也是如此。那些高调宣称属于 WCF 数字精英团队的人通常不是，而那些平时对此三缄其口的人倒反而有可能正是团队成员。正式的名单，即使存在，也是误导大家。这千真万确，你可以通过一些线索分辨出一个 WCF 数字精英，通过他们是 WCF 的小组成员，通过他们知道 WCF 开发的林林总总的细节，通过他们在身体的某个部位的面向服务的宗教纹身等来确认。但是这些线索，即使对于他们的朋友和家人来说都或许是隐秘符号。对于我们，就只有通过一些细微线索来分辨了：与业界的联系，在一些团体中的活跃程度，在会议上的演讲，以及到雷蒙德的旅行和公开的出版物等。

言归正传，在你手中（计算机中，或者其他媒介中）的是一本书，作者是一位 MVP 和 RD，并且是一位 WCF 数字精英（我现在敢告诉你了）。该作者实践经验丰富，交流手段高明，正如你对 WCF 数字精英所期待的那样与产品小组联系颇深。这一切造就了一本众望所归的力作。本书唯一值得改进的地方是，应该包含一张 Michele 的生活近照。缺了玉照的本书，只能被视为通向 WCF 技术的一道门而已。买一本，带回家，好好从中汲取营养吧。如果你曾经在身体的某个部位留下过面向服务的宗教纹身，你可以追忆一下那一痛苦时刻，并且认识到阅读本书将是一个起点，把你送上出发之路。你将心怀感激。

Steve Swartz

微软互联系统部架构师

在 2004 年圣地亚哥的微软技术教育大会（TechEd）上，我有幸和一帮非微软雇员的家伙们呆在一起。大会向他们介绍了一个新的且十分神秘的技术，那就是开发代号为“Indigo”的技术。这是微软平台上的分布式编程技术的一项完美的发明创造，是一个受控技术套件的一部分，将会在未来的某个时间发布。我们听了一系列讲座（不时被问题和讨论所打断——毕竟，这个群体几乎不能哪怕仅仅是休息一下，并且对不是十分确信的结论不依不饶），做了一些代码实例，吃了一些备好的食品，喝了很多苏打水，然后，我们步履沉重地回到 TechEd 关注的当前问题上来。

时间快速转过大约十二个月，Indigo 作为社区技术预览（Community Tech Preview，CTP）版本发布问世。我和其他一些演讲者在阿姆斯特丹的一个会议上围坐在圆桌前，其中的一

名演讲者（后来加盟微软）俯下身子看着我，强烈抱怨 Indigo——现在有了个绰号 WCF（Windows Communication Foundation）——太过容易，而对于我们这些靠它来写文章、写书和做培训混饭吃的人来说，这可不是件好事。我还清晰地记得那时的想法，“不，我不这样认为。”但是，在当时，我已经无法争辩，因为啤酒喝多了。

时间再快速地转过一点，Vista 宣布发布，随之发布的是后来被称为“NetFX3”的技术，现在被命名为 .NET Framework 3.0。Indigo 及其姊妹技术由 Microsoft 的 Marketing 部门的伙计们来命名：Indigo 被称为 Windows Communication Foundation 或 WCF，Avalon 被称为 Windows Presentation Foundation 或 WPF，而 WinOE 被称为 Windows Workflow Foundation 或 WWF，这使得很多开发者抓着脑门、迷惑不解，“WTF?”

Windows Communication Foundation 对于培训者和推荐者来说，代表着两难的抉择：从概念上讲，WCF 是一项简单的技术，其重点在于端点的“ABC”——Addresses、Bindings 和 Contracts。WCF 真正显得可恶的是其与这些概念如影随形的大量细节。契约和绑定选项有着近乎无穷多种的可能的组合方式可以应用到给定的端点上，而每一种组合又有着隐含的关联——关于端点的用法和可用性。更不要提还有隐含的关联，关于这些选项之间的互相组合。当潜在的 WCF 用户粗略地浏览一下 System.ServiceModel 命名空间，数量巨大的类通常已经足够吓退那些胆小或内心柔弱之人。

时间再快速转过一些，Michele 在大约 6 个月之前，向我提出了一项请求：看一看本书的一些章节，针对其中的一些内容给出我的反馈。她一定是觉得，我，作为从 WCF 诞生起就宣讲和教授它的人，不要仅仅是确信它没有任何技术上的疏漏（确实没有），而且还要能够帮助落实一些措施，以期找到最佳途径将 WCF 的大量内容传授给读者。出于多方面的考虑，包括我想要助朋友一臂之力的想法，我答应了她的请求。

事实证明，我真的对我的选择感到满意，而不仅仅是因为我得以将名字留在本书中。

我希望本书写作精良，她做到了。我希望 Michele 对 WCF 的讲解丝丝入扣，她做到了。我没有想到的是，我也深陷其中不能自拔，我逐渐感到，她设计的一步一个脚印的基于操作实例的方法，毫无疑问是最好的方式来向程序设计人员传授 WCF 之道。这在很大程度上归功于她所采用的叙述方法——简洁、实用、明快。

就像维吉尔（Virgil）引领但丁（Dante）穿过地狱湖那样，Michele 也将带领你们遍览 WCF，由浅入深，渐入佳境。关注实用和真实的案例，避开神秘莫测的陷阱。假如你像但丁那样在漫漫旅途中一直与向导如影随形，那么，当你浮出水面的时候，就会看到，最好的 WCF

已经呈现在你的面前——以及你的客户和顾客的面前——用于构建分布式的、面向对象的通信层。

请尽情享受它吧。当你完成的时候，你将成为 WCF 专业开发者群体中的一员，这些人能够以自己喜欢的方式来使用 WCF。

Ted Neward

<http://www.tedneward.com>

前言

Preface

我的整个职业生涯就是在学习和应用新技术，也在不断接受与技术和业务相关的挑战。我以学习新事物为乐事，并且喜欢通过文字和公开演讲与他人分享我的学习心得。我乐于帮助别人学习富于挑战性的新课题，尤其是与构建分布式企业系统有关的课题，这是我喜爱的领域。在这几年时间里，我一直在认真考虑写另一本书，直到开始本书的写作。同时，我一直在等待那些能让我真正充满激情的技术出现。这种技术能解决架构师和开发者们在构建企业系统时所面临的问题，它值得我为它放弃数月的灿烂阳光——WCF（Windows Communication Foundation）就是这样的技术。

WCF 之所以令我如此兴奋以至于要它为它写一本书的原因很多。毫无疑问，我多年专注于 Web 服务和可互操作性（interoperability）是动因之一。WCF 深度支持新兴的 WS*（web service standards），并且依靠自身的可延展性随着这些标准的发展而发展。WCF 的另一个令我印象深刻的特性在于，它是一个完美的 SOA 平台，将“服务”放在首位，并且把服务的开发从低端的业务组件中分离出来。令 WCF 魅力十足的最有说服力的原因也许是，当你创建一个服务时，它可以用来在内部网上实现跨进程、跨机器边界的客户—服务器调用。通过这个调用，可以公开队列调用和传送服务保证，并且使用完整的 WS*协议栈来公开可互操作的 Web 服务。简而言之，WCF 统一了早期的多种技术，即 .NET Remoting、Enterprise Services、ASMX（ASP.NET web services）和 WSE（Web Services Enhancements）。

在 WCF 之前，开发人员被迫将分布式功能同调用这些分布式功能的技术耦合起来。这就意味着我们需要提前知道我们使用的到底是远程对象、服务组件，还是 Web 服务。因为它们有着不同的表述模式。我不知道下面的哪一种情况会更糟糕，一种是为了满足不同的通信需求而学习这 3 种不同技术，一种是使用 3 种中的任何一种技术来满足所有不同的通

信需求。有了 WCF，这些问题都迎刃而解了。开发者们可以学习一种编程模式，基于契约来设计服务，在服务部署时，基于协议和策略来做出决定。而且，开发者们不必了解底层的基础管线和通信协议。

现在的企业系统的设计和开发者都应该使用 WCF，这一点无可争议。现在的市场上没有任何一种其他平台能够像 WCF 这样，囊括了 SOA 所有的价值主张，封装了一长串的 WS* 协议，提供了多种功能来支持不同类型的分布式通信，同时支持安全、事务、传输保证和其他系统服务。WCF 是一个 enabler (译注 1)，它是一个完美的系统设计和安全性、可靠性、可扩展部署的集大成者。

我写这本书，是为了告诉系统的设计者和开发者们 WCF 的卓越和高效，是为了给特殊的应用情形提供推荐方案，是为了表达 WS* 协议支持的重要性和相关性，是为了帮助读者们在这个平台上迅速形成生产力。更进一步讲，写作本书也是为了我自己。因为我为 WCF 的问世而如此兴奋，以至于我想徜徉于这个平台的每一个角落，包括它的基础架构和它丰富的功能集。

在这本书里，我将和读者分享自己深入钻研 WCF 平台的收获，比如，我将与大家分享曾经构建大规模、分布式、面向服务系统的经验和知识应用到 WCF 平台的体会，还会摘取一些我和微软产品组的同事们的讨论细节。在 .NET 3.0 的 Beta 1 版本的研发期间，我曾经和 CardSpace 产品组密切合作过。我的工作和基础代码库直接挂钩，重点在联邦安全性功能上，是为了集成 CardSpace 和 PSTS (Portable Security Token Service) 装置。我还在常规的 SDRs (Software Design Reviews) 期间，或者通过其他交流方式与 WCF 的项目组成员有过直接接触。同时，我也将自己在解答问题环节中的心得记录在本书中。

这本书为谁而写

Who This Book Is For

本书的主要目标读者是中高级的开发者。他们期望学习关于 WCF 的全面知识以构建 SOA 系统，这意味着开发分布式企业系统需要一种可行的技术。本书从 SOA 演变的背景介绍开始，讨论了 SOA 架构的 4 个原则和 WCF 如何满足这些原则。之后，我将自顶向下介绍 WCF 的概念。读者应该事先知道这些概念，以进一步理解与契约设计、寄存、绑定、实例与并发、分流、可靠性、安全性和错误处理等相关的特定功能。

我的目标是使读者深刻理解每一个功能和它的作用，提供给读者自己动手做的机会以巩固

译注 1：开放移动联盟 OMA 定义的一种技术，用于开发、部署、运行服务。

对概念的理解，讨论功能的实际用处和操作规程。我将回答与每一个功能相关的、基于常见的不同设计场景的问题。所以，书中会有少许的架构和设计讨论，以及一些针对每一种场景设计目的的解释。

WCF 可以应用于很多实际问题领域，包括分布式通信、消息传递和 Web 服务等都会在本书的不同场景中被讨论。我会尝试捕捉你将 WCF 编程模型应用到这些领域中的美妙瞬间。本书并不会涵盖分布式通信的方方面面，然而，在讨论 WCF 在因特网上的应用时我一定会给出很多背景材料。本书不打算全面讨论消息系统，不过，我将向你展示如何将 WCF 和 MSMQ 应用到消息环境中。本书也不打算讲授 Web 服务的点点滴滴，但是会深入探讨用到的标准及其工作方式，如何控制消息传输，以及如何在多个层次上激活可操作性。

如果你是一个初学者，在基于分布式通信或 Web 服务之上构建系统方面经验较少，那么，你应该有计划地按顺序从头至尾阅读本书。通过按出现顺序学习一些动手就能做的操作实例，你将会被一步一步带入解决一些相对困难的应用情况，会帮助你不仅仅了解 WCF 是如何工作的，还会让你认识到它解决了什么问题。

尽管本书较为全面，但为了能够界定一个合理的范围，本书没有涉及 WCF 下面的一些细节：

- 贯穿本书，我提到了 WCF 的很多可扩展性的知识点，然而，我并没有单独列出一章来讲解可扩展性或定制信道。
- 没有涉及端到端 (peer-to-peer) 通信，因为这是需要若干章节才能透彻地讲清楚的概念。在你使用端信道实现方案之前，需要先了解本书中讨论的基础知识。
- Visual Studio 下一个版本的开发代码为 “Orcas”，将包含很多用于构建 WCF 应用的生产力工具 (productivity tools)。本书仅略略谈到这些工具，由于它们还没有发布，所以，我所使用的临时版本在最终版本中可能会改变。

本书的结构

How This Book Is Organized

本书包括详细讲解、代码范例和动手就能做的操作实例。同时，第 1 章给出了对重要基础概念的综述，后续章节分别讨论了相关功能的细节。我推荐你首先通读第 1 章，然后就可以随兴趣在各章之间跳来跳去了。这就是说，我对章节的安排使得每一章并不涉及另外一层的复杂内容。不过，按序阅读还是最合适的。

章节概览

Chapters Overview

这里会概述你从每一章中能学到些什么。

第 1 章：Hello Indigo

因为很早就开始使用这一平台了，我变得十分依恋 WCF 的开发代码：Indigo。当然，我们最终都不得不放弃开发代码，转而使用产品代码——不过，这还是不能阻止我在第 1 章中向其表示敬意。在这一章中，我将把 WCF 介绍给你，同时讲述它如何解决分布式企业系统中的问题，并讲解它是如何支持 SOA（Service Oriented Architecture）的。同时，你将学习很多基础概念，使你马上可以上手使用 WCF，包括创建服务契约和服务、寄存服务，以及生成代理调用服务。

第 2 章：契约

这一章重点讲述契约和串行化。它讲述如何设计服务契约，如何为复杂类型串行化创建数据契约，以及如何使用其他可串行化类型来解决特殊挑战，包括契约优先设计方法。你将学习到服务如何将原数据公开给客户用以生成代理，以及原数据交换是如何支持这一点的。同时，这一章还为你提供了用于服务契约和数据契约版本控制的指导方针。

第 3 章：绑定

绑定是 WCF 的核心，运用绑定，你可以配置服务所支持的通信协议，包括那些与可互操作的通信相关的协议。这一章讲解了每一种核心绑定的实际用法：如何为内部网和因特网应用、单向通信和回叫，以及为处理大型消息载荷而定制这些绑定设置。你还将学习绑定如何为客户和服务配置通信信道，并且学习如何及何时使用定制绑定来处理特殊问题。

第 4 章：寄存

这一章探讨了 WCF 服务的不同寄存选择，包括 Windows 应用、Windows 服务、IIS 和 Windows Activation Service。你将学习每一种环境所支持的相关寄存功能和协议，学习它们所共享的基础寄存架构，以及选择每一种寄存环境的现实理由。

第 5 章：实例化与并发

实例和并发模式管理服务的分配以支持请求，并对应用扩展性施加现实影响。这一章讲解了如何配置服务使其作为单实例（singleton）运行，如何为应用会话提供支持并关注其输入，或者作为可扩展的无会话服务而运行。还讨论了影响并发请求数量和其他服务分流行为的并发模式。你还将学习每一种选项的可扩展性含义，并学习关于负载平衡服务的建议。

第 6 章：可靠性

WCF 的功能（比如可靠会话、分布式事务和队列通信）从整体上提高了企业系统的可靠性。基于可互操作协议的可靠会话使我们能够克服暂时的网络中断，基于 TCP 和 HTTP 的事务支持提升了系统的一致性，而队列通信则提供了持久和实在的可靠性。这一章探讨如何使用 WCF 的这些功能，以及何时使用功能的指导原则。

第 7 章：安全

这一章涵盖了较大的范围，因为安全性功能在 WCF 中介入较深。本章首先描述 WCF 所实现的基础的安全性概念，包括身份 (identities)、双向验证 (mutual authentication) 和消息保护，因为它们都为核心 WCF 绑定而实现。然后，为了尝试简化这一令人生畏的另类问题，我给出了不同的方案，针对内部网、因特网、搭档或附带证书的机器认证，以及联邦安全性。针对如何配置这些方案，同时又减少功能中无用的冗余机能，我给出了自己的处理原则。你需要从本章中汲取的就是对于带有范例的几个重要安全方案的理解，这些范例可以作为你的应用的开始模板。

第 8 章：异常和故障

尽管我在本书的前面谈到过异常和故障，然而这一章才专门将重点放在错误处理概念上。包括调试技巧、故障契约设计和错误处理组件。我将本章置于最后是因为，从前面几章与可能出错的功能类型相关的内容中，你可以知道更多的上下文背景知识，同时因为本章涉及了一些 WCF 的复杂的扩展性功能。

附录 A：设置指南

这一附录为数据库的配置和建立、ASP.NET 成员关系提供了模型安装，以及如何使用授权书的方法，都给出了详细的安装指南。在建立操作实例和代码范例的过程中，你将会不定期地被引导到这里来寻求帮助。

附录 B：当 ASP.NET 遇到 CardSpace

这一附录包含的文章是用于 Windows CardSpace 上的 *asp.netPRO*，也为第 7 章中所讨论的技术提供了额外的背景知识。

操作实例和代码范例

Labs and Code Samples

每一章中都包含了一些动手就能做的操作实例以帮助你熟悉方案环境，而相关的代码范例则展示了更细粒度的知识点及复杂场合。操作实例和代码范例都是用 C#写的，但是，本书中所涉及的概念对于 VB 开发环境也一样。对于操作实例，任何安装指南都是内联的，

而对于代码范例，你可以找到“Read Me”文件，显示了各自需要的安装方法（如果合适的话）。两者的详细安装指南都呈现在附录 A 中。

实例和代码示例文档可以从如下地址下载：www.oreilly.com/catalog/9780596101626。需要补充材料，可以访问我的相关博客：<http://www.thatindigogirl.com>。

有关读者的假定

Assumptions About the Reader

本书假定你已经熟悉 .NET 2.0 和 Visual Studio 2005。本书前面的操作实例指导中，将会在你首次实施一个动作（如添加项目引用）时，提供详细的步骤提示。而在后面的操作实例中，关于步骤的描述将会比较笼统，就是假定你已经逐渐熟悉了有关过程。因此，如果你是这些技术的初学者，建议你从前往后阅读本书。

有一些操作实例和代码范例还将使用 Windows Forms、ASP.NET 和 Windows Service 应用。你不需要知道这些技术的细节，因为操作实例提供了指导，但如果了解细节确实是有益的。

系统要求

System Requirements

这一节描述操作系统支持、数据库引擎和为完成操作实例并运行代码样例所需要的软件。你应该按照它们在如下列表中出现的顺序来安装所需软件：

- Windows XP 和 SP2、Windows Server 2003 或 Windows Vista 附带 IIS 和激活的 MSMQ。
- Microsoft SQL Server 2000、Microsoft SQL Server 2005 或 Microsoft SQL Server 2005 Express Edition（SQL Express）。
- Microsoft Visual Studio 2005。
- Microsoft .NET 3.0 Framework Runtime。系统已经预装了 Vista。
- 针对 Windows Vista 的 Microsoft Windows Software Development Kit 和 .NET 3.0 Runtime Components（针对 .NET 3.0 的 Windows SDK）。
- .NET Framework 3.0（WCF 和 WPF）的 Visual Studio 2005 “Orcas” 扩展。或者安装最新版的 Visual Studio “Orcas”，在本书写作时，它还仅仅是技术预览版。

你的工作目录

Your Working Directory

随本书提供的 ZIP 文件包含了操作实例、范例代码和其他支撑文件，比如，数据库脚本、证书和媒体。建议你将这个文件解压到名为 `c:\LearningWCF` 的目录下，或者不同驱动器上的等价位置。这个目录将贯穿本书作为 `<YourLearningWCFPath>`，在引用目录和文件时读者们可随时查阅。

在你解压该文件之后，将会生成下述子目录：

路径	目录
\Labs\Chapter1、\Labs\Chapter2 等	每一章都有一个子目录，在多数情况下，其中提供了该操作实例的初始方案。在每一个操作实例的开头，你都会被告知需要打开哪一个方案作为开始
\Labs\Chapter1\CompletedLabs、 \Labs\Chapter2\CompletedLabs 等	每一章中的每一个操作实例的完整版本都位于 \CompletedLabs 目录下。这些样例直到你完成了相关操作实例指导中的安装步骤之后才能运行
\Samples	这个文件夹包含了本书中用到的所有样例，它们是按照标题而不是按照章节来组织的
\SupportingFiles\Certificates	包含了待安装的证书以支持操作实例和样例。安装证书的指导可以在附录 A 中找到。每一个操作实例和代码样例都声明了其证书需求
\SupportingFiles\DatabaseScripts	包含了数据库脚本以支持操作实例和样例代码。每一个操作实例和代码样例将声明其数据库需求
\SupportingFiles\SampleImages	包含了样例图像以支持操作实例和代码样例
\SupportingFiles\SampleMedia	包含了样例媒体以支持消息流操作实例和代码样例

本书中使用的惯例

Conventions Used in This Book

本书中使用如下的排版惯例：

Italic

用于 URL 文件名、文件扩展名、路径名和目录。

Courier New

用于代码范例、声明、命名空间、配件名称、类型、方法名称、参数、值、XML 标签、HTML 标签，或者命令行的输出。

Courier New

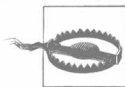
用于在代码范例中表示强调，或者显示应该由用户键入的文本。

Courier New

用于显示应该被由用户提供的值替代的文本。



该图标表示提示、建议或一般注解。



该图标表示警告或注意事项。

写作本书的目的就是帮助你干好工作。通常，你可以在程序和文档中使用本书的代码，不