

“这世上只有很少几个人能把企业服务解释得足够清楚，而Christian就是其中之一。”
—— 来自Ingo Rammer的前言

Enterprise Services with the .NET Framework
Developing Distributed Business Solutions with .NET Enterprise Services

.NET企业服务框架

——应用.NET企业服务开发分布式业务解决方案

Microsoft®
.net™
开发系列



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

[美] Christian Nagel 著
夏桅 金雪根 译

.NET 企业服务框架

——应用.NET 企业服务开发分布式业务解决方案

Enterprise Services with the .NET Framework

——Developing Distributed Business Solutions with .NET Enterprise Services

[美] Christian Nagel 著

夏 桅 金雪根 译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

随着 Web 技术的发展,大部分桌面软件依然是孤立的,这意味着分布式应用程序具有广阔的发展前景。分布式应用程序主要应用于企业内部和企业与企业之间,当然,它对于普通用户来说也是意义非凡的。企业服务是一种难学易用的框架,因为它面对的是极其复杂的企业环境。

这是一本关于如何利用 .NET 企业服务框架开发分布式业务解决方案的书,本书涵盖了 .NET 企业服务的方方面面,如并发、COM 互操作、事务服务、状态管理等。书中详细讲解了应用 .NET 企业服务框架开发分布式业务解决方案的具体方法和实用技巧,使读者在知道 Know how 的同时还知道 Know why。

本书的另外一个特点是提供了大量的示例代码,以或大或小的实例展示了各种方法的应用场景和技巧。无论你是刚入门的初学者还是经验丰富的高手,本书都是一本权威的指南和参考书。

Authorized translation from the English language edition, entitled *Enterprise Services with the .NET Framework: Developing Distributed Business Solutions with .NET Enterprise Services*, 1st Edition, 032124673x by Nagel, Christian, published by Pearson Education, Inc, publishing as Addison Wesley Professional, Copyright©2005 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 PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright ©2006

本书简体中文版由电子工业出版社和 Pearson Education 培生教育出版亚洲有限公司合作出版。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书简体中文版贴有 Pearson Education 培生教育出版集团激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字: 01-2005-5369

图书在版编目(CIP)数据

.NET 企业服务框架: 应用 .NET 企业服务开发分布式业务解决方案 / (美) 内格尔 (Nagel, C.) 著; 夏槐, 金雪根译.

北京: 电子工业出版社, 2006.11

书名原文: *Enterprise Services with the .NET Framework: Developing Distributed Business Solutions with .NET Enterprise Services*
ISBN 7-121-03071-3

I. N... II. ①内...②夏...③金... III. 计算机网络—程序设计 IV. TP393

中国版本图书馆 CIP 数据核字 (2006) 第 094775 号

责任编辑: 周筠 杨福川

印 刷: 北京市天竺颖华印刷厂

装 订: 三河市金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 36.25 字数: 745 千字

印 次: 2006 年 11 月第 1 次印刷

定 价: 68.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系电话: (010) 68279077; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

专家对本书的赞誉

Praise for Enterprise Services with the .NET Framework

“对于那些沉浸于.NET 社区的人来说，不知道 Christian Nagel 着实是一件很难的事情。Christian 是.NET 领域中绝对重量级的高手，特别是在.NET 企业服务领域。由于.NET 企业服务不仅可以让你避免面对大量的纷繁复杂的细节，而且可以让你在什么也不用做的情况下就轻松增加应用程序的可扩展性。开发者们将会发现，这本书里介绍的技术对于运行着分布式企业应用程序的公司来讲，它是有着显而易见的好处的。”

——William G. Ryan, Microsoft MVP, Senior
Software Developer, TiBA Solutions, LLC

“不管你还是一位资深的软件架构师，还是一个刚入门的开发者，分布式应用程序的开发都绝非易事，毕竟它同时涉及到很多很复杂的技术。直到现在，我们都很难找到这方面的指导——无论是不是完整的。Christian 现在不仅给出了这样的一份参考资料，而且从各种独立的技术开始，逐渐深入讲解如何架构和开发一个高可扩展性的分布式应用程序的整体框架。这绝对是一本货真价实的技术精华集！”

——Clayton Burt, Managing Partner, Onzo, LLC

“从传统方式转换到分布式应用程序的开发，会带来大量的诸如安全、部署这样的新问题，同时它还需要你对事件、事务、消息等概念有新的理解。这本书向开发者和架构师们展示了如何使用.NET 企业服务来创建稳固、安全和可维护的分布式应用程序，它对.NET 企业服务应用程序的开发是极佳的指导。”

——Brian Davis, Director of Software Development,
InfoPro Group, Inc., Co-Creator, KnowDotNet.com

.NET 企业服务框架——应用.NET 企业服务开发分布式业务解决方案

Microsoft .NET Development Series

John Montgomery, Series Advisor

Don Box, Series Advisor

Martin Heller, Series Editor

微软.NET开发系列图书(The Microsoft .NET Development Series)得到了微软技术开发部门的领导和专家的大力支持,这些专家包括微软的架构师以及DevelopMentor公司的资深教授和讲师。这一系列的图书为读者提供了核心的信息资源,充分地满足了开发者在编写高效应用程序和有效管理程序代码方面的需求。这些资深的专家将引导你如何最有效地去使用.NET企业服务框架以及与之相关的程序设计语言。

微软开发系列图书书目

Brad Abrams, .NET Framework Standard
Library Annotated Reference Volume,
01-321-15489-4

Keith Ballinger, .NET Web Services:
Architecture and Implementation,
0-321-11359-4

Bob Beauchemin, Niels Berglund, Dan
Sullivan, A First Look at SQL Server 2005
for Developers, 0-321-18059-3

Don Box with Chris Sells, Essential.NET,
Volume 1: The Common Language Runtime,
0-201-73411-7

Keith Brown, The .NET Developer's Guide
to Windows Security, 0-321-22835-9

Mahesh Chand, Graphics Programming
with GDI+, 0-321-16077-0

Anders Hejlsberg, Scott Wiltamuth,
Peter Golde, The C# Programming
Language, 0-321-15491-6

Alex Homer, Dave Sussman, Rob
Howard, ASP.NET v. 2.0—The Beta
Version, 0-321-25727-8

James S. Miller and Susann Ragsdale,
The Common Language Infrastructure
Annotated Standard, 0-321-15493-2

Fritz Onion, Essential ASP.NET with
Examples in C#, 0-201-76040-1

Fritz Onion, Essential ASP.NET with

Examples in Visual Basic .NET,
0-201-76039-8

Ted Pattison and Dr. Joe Hummel,
Building Applications and Components
with Visual Basic .NET, 0-201-73495-8

Dr. Neil Roodyn, eXtreme .NET:
Introducing eXtreme Programming
Techniques to .NET Developers,
0-321-30363-6

Chris Sells, Windows Forms Programming
in C#, 0-321-11620-8

Chris Sells and Justin Gehtland,
Windows Forms Programming in Visual
Basic .NET, 0-321-12519-3

Paul Vick, The Visual Basic .NET
Programming Language, 0-321-16951-4

Damien Watkins, Mark Hammond,
Brad Abrams, Programming in the .NET
Environment, 0-201-77018-0

Shawn Wildermuth, Pragmatic
ADO.NET: Data Access for the
Internet World, 0-201-74568-2

★Paul Yao and David Durant, .NET
Compact Framework Programming with C#,
0-321-17403-8

★Paul Yao and David Durant, .NET
Compact Framework Programming with
Visual Basic .NET, 0-321-17404-6

(注:带星号★图书的简体中文版已经由电子工业出版社出版)

更多信息请点击: www.awprofessional.com/msdotnetseries/

译序

从个人 PC 问世到现在的二十余年时间里，桌面软件已经发展到了一个相当成熟的阶段。事实上，几乎所有能想到的，并且能用机器自动完成的任务，大部分都已经有了相应的软件对其进行实现。在 GUI 之后，要对桌面软件再进行一次大规模的革命性创新，是相当难的事情，这有赖于人工智能、语音识别等基础性技术的突破。至于另外一个热门——Web 应用程序的开发，只是另一种表现层而已。从软件的本质考虑，Web 应用程序实际上是以浏览器为客户端的另一种桌面程序。

网络对于普通用户的意义是不言而喻的，它使得用户之间（准确地说是机器与机器之间）进行交互成为可能。对于两个应用程序来说，不管它们是否运行于同一台计算机，这种意义同样非同小可。这实际上是另外一个层次的互操作——从机器级别到应用程序级别。这种互操作不仅可以为软件增加一些新的功能，更重要的是软件架构模型的变化可以让软件做到许多以往做不到的事情，这也是分布式应用程序如今越来越热门的原因之一。与此同时，现实情况是，大部分桌面软件依然是孤立的，这意味着分布式应用程序具有广阔的发展空间。

不过，千万不要一下子就走到另外一个极端——网络就是计算机。也许当网络传输速度、响应时间、可靠性能达到计算机总线性能的十分之一时，这个口号才能变成现实。

我坚持认为分布式应用程序对普通用户（比如说家庭用户）来说也是意义非凡的，不过到目前为止，它最大的市场还在于企业内部和企业之间。这大概也是谈论企业级应用程

序时，往往不得不谈分布式计算的原因之一。企业级应用程序面对着众多挑战，在银行、电信、机场这样的实际环境中，软件起着至关重要的作用。这样的软件开发不但难度大、周期长、成本高，而且项目成功率也较低。大型公司的规模决定了其 IT 部门和设施也会有巨大的规模，分布式应用程序是必然的选择。而分布式应用程序需要更精良的软件设计和编码，因为你必须面对各种各样的问题，包括安全性、可靠网络传输、远程事务、并发操作、性能、可扩展性、可伸缩性（Scalability）等等。单独解决这些问题已非易事，它们合并在一起之后更是成了巨大的难题。

另外一个方面，所有大规模的企业级应用程序都面临着这些共同的问题。这样，最好的选择就是使用一个成熟的框架来应付这些基础问题，让开发者把精力集中于上层代码的设计和实现。这对缩短项目开发周期、成本，提高项目成功率都有着极大的好处。微软早已想到了这一点，COM+技术正是为此而生。

到了.NET 时代，传统的 COM 模型已经略显过时，但微软成功让 COM+和.NET 融为一体。现在我们可以用类似.NET 的方式和习惯操作 COM+服务，这就是这本书的主题——企业服务（Enterprise Services）。

企业服务是一种难学易用的框架，因为它面对的就是极其复杂的实际企业环境，其背后的复杂性是难以估计的。不过好在 Christian 为我们准备了这样一本书，它既可以作为逐步学习企业服务的教程，也可以作为实际开发过程中的技术参考书。来自 Thinktecture 的 Ingo Rammer 给出了这样的评价：

在这个世界上，只有很少几个人能把企业服务解释得足够清楚，Christian 就是其中之一。

Christian Nagel 是一位软件架构师，同时也是培训师和咨询顾问。作为 Thinktecture 公司的成员，他还提供微软.NET 技术相关的培训。由于在开发者社区中的卓越表现，他获得了 Microsoft Regional Director 和 MVP（Visual C#组）的称号。他著有多本.NET 书籍，包括 *Professional C#* (Wrox, 2004)、*Pro .NET Network Programming* (Apress, 2004)和 *C# Web Services* (Wrox, 2001)，并且因此而深受好评。他还有着超过 15 年的软件开发设计经验，对各种编程语言和平台均有所涉猎。

于此同时，.NET 企业服务非常易于使用，它把.NET Attribute 的特性发挥得淋漓尽致，代码不但高效而且简洁，你也许只有学过.NET 企业服务之后才能知道“原来.NET 程序还可以这么写”！

微软从未停止过在企业服务方面的工作，在 WinFX 框架（现在已更名为 .NET 3.0）中，企业服务仍然受到第一位的支持（Windows Communication Foundation/WCF 也完全兼容现在的 .NET 企业服务）。此外，理解 .NET 企业服务（COM+）之后，理解 WCF 会变得易如反掌，因为 WCF 中很多概念和企业服务都有对应关系。此外，微软的 Longhorn Server 预计要到明年下半年才能推出。根据以往的经验，新的操作系统要被企业客户接受，最起码需要一两年的时间。它带的 WCF 要形成产业标准，也不是一件容易的事情，这个过程可能更加缓慢。在最近几年内，对于企业级应用程序的开发，.NET 企业服务将一直扮演至关重要的角色。

夏桅（Microsoft MVP C#, 2005）

2006 年 7 月

关于作者

About the Author

Christian Nagel 是一位软件架构师，同时也是培训师和咨询顾问。作为 Thinktecture 公司的成员，他提供微软.NET 技术相关的培训。由于在开发者社区中的卓越表现，他荣获了 Microsoft Regional Director 和 MVP（Visual C#组）的称号。他著有多本.NET 书籍，包括 *Professional C#* (Wrox, 2004)、*Pro .NET Network Programming* (Apress, 2004) 和 *C# Web Services* (Wrox, 2001)，并且因此而深受好评。

Christian 有超过 15 年的软件开发和软件架构经验。他的编程生涯从一份关于 PDP 11 和 VAX/VMS 平台的工作开始，到现在，他对各种编程语言和平台均有所涉猎。从 2000 年开始，他就已经开始使用 .NET 和 C# 开发和设计分布式业务解决方案了。

<http://www.christiannagel.com>

<http://www.thinktecture.com>

前言

Foreword

编写软件基础结构的代码，是最有价值、最有意思、最具挑战性的工作之一。软件基础结构往往需要包括复杂的事务系统（包括分布式事务，甚至要包括事务补偿处理而不仅仅是简单的回滚操作）、足够的权限点粒度、基于角色的安全验证框架、透明地支持基于异步队列的互操作过程、自适应的线程模型、完整的过程和控制模型以及一个可以支持不同级别的间接调用（来自进程内、本地和远程的调用）的宿主环境等等。与此同时，我们的客户往往不希望看到我们把时间花费在这样的工作上，毕竟客户的业务并不直接体现在软件的基础结构中。甚至有人可能认为手工实现这样的框架结构意味着项目的失败，因为这样的基础结构理应有一种惯用的实现，而不是每次都由您自己手工编写。

这就是企业服务（Enterprise Services）的由来。它是微软的应用程序服务器提供的一种功能（主要是指 COM+ 框架）。是的，微软并没有过多地谈论它们的操作系统与应用程序服务器之间的关系，但事实上，服务器版的 Windows 系统本质上就是应用程序服务器。注意，我们不是在讨论文件服务器或者是打印服务器。如今，Windows 系统（甚至包括 Windows CE）都已经内置了一些特别的部件，这些部件在别的平台上会被称为应用程序服务器，并且它们通常会被单独出售。

在 Windows 平台上, 应用程序服务器部件并不是操作系统的主要组成部分。由于它们本质上就是 Windows 系统的一些免费的部件, 所以你根本没有必要为它们的价格而担忧。它们都是现成的一些 Windows 常用部件而已。

不幸的是, 这些年来我注意到只有少数开发者使用过这些标准化服务, 并没有把它们当作应用程序架构的基础设施。大部分软件工程师仍然在用他们自己的代码来处理这些基础问题。很难说这样做就是不对的, 毕竟这并不是什么难事。对于一个事务管理系统来说, 有 80% 的部分是较容易实现的, 但以我的经验看来, 剩下的 20% 则非常难以实现, 甚至近乎不可能做到。另外一方面, 对于事务管理系统这样的基础设施, 只有 100% 实现了所有特性后才能充分发挥它的价值。(否则, 难道你愿意去客户那里解释, 在某些特定的情形下, 你建立的记账程序在没有报告错误的情况下也有丢失数据的可能? 我想这绝对是个糟糕的体验。) 同样, 在安全性、可扩展性、队列组件等方面也一样, 它们作为基础服务设施, 必须 100% 完备才行。

也许你和我一样, 大概也想通过简单而自然的方式¹ 学习新技术: 到互联网上找一些示例, 运行之, 改进之, 研究系统究竟是如何运作的, 从而得到更深刻的理解, 然后再继续研究, 最终完全理解整个系统。不过, 对于企业服务来说, 你大概不能按照这样的路线来学习和开发。你可以创建你的第一个组件, 然后标记为 [assembly: Activation (ActivationMode.Server)], 你会注意到, 与原先想象的不同, 系统在后台做了很多的事情。

请允许我为它简单说几句: 企业服务框架做得已经很好了, 它为复杂的现实提供了非

¹ 这可能是 Tim Taylor 的开发方式: Real men don't need instructions

常简单的接口。一直以来，你经常需要处理分布式应用程序的复杂性。这不是企业服务框架的问题，事实上，这种复杂性来自于高可扩展性分布式应用程序本身——这又是另外一个复杂的话题。而高度抽象的企业服务框架同时也具备良好的易用性，这使你必须知道比表面更深的知识。这就是为什么这样的书籍如此重要的原因。

在.NET 技术早期，我就已经认识 Christian 了。事实上，那时候.NET 的名称还只是 NGWS (Next Generation of Windows Services)，而 C#则还只是代号 *Cool*。如今，Christian 已经成了著名的作家、卓越的培训师和具备丰富实际经验的高级咨询顾问。他也是我很好的朋友，值得信赖的同事。在这个世界上，只有很少几个人能把企业服务解释得足够清楚，Christian 就是其中之一。他能抽出时间来编写这本书，实乃广大开发者的福音。

Ingo Rammer

Vienna, Austria

<http://www.thinktecture.com>

序

Preface

这本书是我多年心血的结晶。当微软于 1996 年发布 Microsoft Transaction Server 时，我就开始在微软的平台上开发分布式业务解决方案了。从那以后我参与了许多项目，帮助客户设计和实现他们的业务解决方案。其间，我做过各种各样的事情——比如，集成旧有系统，创建一个自定义的框架，可以以一种很灵活的方式为某些应用程序进行扩展等等。

微软于 2000 年的 PDC 会议上首次展示 .NET 的预览版时，我就被这项新技术深深地吸引了。从那时起我参与了多本 .NET 书籍的编写，包括 *Professional C#*（Wrox, 2004）、*Beginning Visual C#*（Wrox, 2002）、*C# Web Services*（Wrox, 2001）和 *Prof .NET Network Programming*（Apress, 2004）。

这些年来，我学到的很多东西也都已总结在这本书中了。

为什么必须使用 .NET 企业服务？.NET 企业服务是 .NET 程序使用 COM+ 服务技术的途径，而 COM+ 作为操作系统的一部分，它正在被世界范围内的大量公司所使用。.NET 企业服务实际上扩展了 COM+ 服务，使之具有了 .NET 的功能。不管你是使用 Windows Forms 还是 ASP.NET 来创建业务解决方案，只要是一种前端加后端的应用程序模型，那么在服务端（后端）总有一些公共的特性是需要实现的，并且可能还须要保证可扩展性和互操作性等等。

在 .NET 企业服务中，这些特性都是现成可用的，包括对象池、线程池、自动的分布式事务，还有服务端和客户端之间的异步连接、回调机制、队列组件（可以作为消息队列的一个抽象层）、基于角色的安全等其他更多的特性。

Who Should Read This Book 谁应该读这本书

这本书并不是.NET 程序设计的入门书籍，现在有很多其他的书已经扮演了那个角色。在写这本书时我假设你已经有了.NET 开发的一些经验。

尽管 COM+是.NET 企业服务的基础，但是 COM 技术对于理解这本书或者创建.NET 企业服务解决方案并不是必须的。另一方面，如果你已经有了 COM 的经验，并且希望把已有的 COM 组件和.NET 应用程序进行集成，那么这本书也会给你绝对有用的信息。

Organization 内容编排

这本书一共有 15 章，包括了.NET 企业服务所提供的各种服务。这本书开头先介绍了所有服务的概览和一些相关的核心技术，它们都是企业服务的基础，然后在各个章节逐一描述所有的服务。每一章的开头都有服务的架构性描述，这项服务存在的原因，典型的使用场景等等。在架构描述之后你将看到很多代码示例，这样你不仅能知道这些特性，还能知道如何运用它们。这本书中贯穿了大量的技巧和提示，向你展示了这项技术最好的那些方面。最后一章提供了一个案例，它同时使用了多种服务，这样你可以看到它们是如何在一个真实的环境中协同工作的。

第 1 章，“.NET 企业服务简介”提供了一个和企业服务相关技术的概览。在这里你能了解到你的业务解决方案能用和应该用什么样的技术。而从 MTS 到 COM+和.NET 企业服务的革命性进步的背景信息，会令你更好地理解这些技术。

第 2 章，“对象激活和上下文”包括了 COM+和.NET 企业服务的基础技术。它会告诉你这些服务究竟是如何做到的。你也可以了解到拦截技术的基础知识。第一个服务组件将在这章介绍，你将能看到上下文和对象激活的实际场景。

第3章，“并发”将教你如何并行地运行线程，以及如何避免竞争条件和死锁。在讨论完 COM 单元模型之后，这一章开始阐述 COM+活动的概念。无组件服务——COM+ 1.5 的一个新特性——将在这章扮演重要角色。

第4章，“COM 互操作”是为那些需要在 .NET 应用程序中使用已有 COM 组件的人准备的。在这里你将了解到如何集成 COM 和 .NET 这两种技术。

第5章，“网络访问”提供了一些关于如何使用 DCOM、.NET Remoting 和 Web Services 访问 .NET 企业服务的信息。你还能知道根据应用程序上下文的不同，如何选择最合适的技术。

第6章，“数据访问”描述了如何使用 ADO.NET 访问数据库。不过这章并不只是 ADO.NET 本身的介绍，它还会告诉你一些经验和技巧，也就是在一个分布式环境中使用 ADO.NET 的最佳实践。

第7章，“事务服务”解释了用于描述事务的 ACID 四个属性（Atomicity、Consistency、Isolation 和 Durability），向你展示了如何使用 ADO.NET 的事务，描述了 .NET 企业服务中所有可用的事务选项和设置。你将能学到这几种不同特性的功能，以及从事务性的角度决定你的组件应该如何配置。

第8章，“补偿性资源管理”示范了怎样创建你自己的能参与在 DTC 管理下的事务资源管理器。

第9章，“状态管理”——创建有状态的还是无状态的服务组件？你应该把状态数据放在哪里？——客户端还是服务端？这一章指出了在多种应用程序模型下，状态管理的多种选择。

第10章，“队列组件”在分布式解决方案中往往是一项被忽略的技术。在这一章你将能了解到消息队列的基础功能，以及作为抽象层的队列组件如何利用消息队列实现在断开连接的环境中进行组件上的方法调用。

第 11 章,“松耦合事件”对 COM+和.NET 事件模型进行了比较。LCE(Loosely Coupled Events) 框架使用了一个拦截模型,这样组件不会直接和客户端进行交互。你将在这里读到松耦合事件应该如何应用到实际的场景中,以及如何定义事件过滤器。

第 12 章,“安全”解释了企业服务应用程序中身份认证、权限验证、假冒、机密性等概念。由于很多其他应用程序都要和.NET 企业服务业务解决方案相交互,所以和.NET 企业服务一起,这章还同时讨论了 SQL Server、ASP.NET Web 应用程序、Web Service、.NET Remoting 的安全问题。

第 13 章,“部署和配置”包含了如何安装、配置服务端、客户端(带有客户端代理以访问服务组件)应用程序的内容。

第 14 章,“分布式应用程序的未来”提供了一些未来技术的概述,以及你如何在今天就为它们做好准备的指导。

第 15 章,“案例研究”提供了一个综合性的示例,它利用到了前面所有章节中所涉及的技术,这样你可以看到这些.NET 企业服务特性将如何在一起协同工作。

Sample Code 示例代码

本书的示例代码都是基于 C#的——只不过第 4 章还用用到了一些 C++、Visual Basic、JavaScript 的代码来演示 COM 互操作。你可以从本书的网站上下载这些示例代码:
<http://www.christiannagel.com/EnterpriseServices>。这些代码的 Visual Basic .NET 版本也能从这个网站上下载到。

Contact 联系方式

如果你有关于这本书的反馈意见，或者希望得到关于.NET 企业服务的培训或咨询，
那你可以通过我的网站与我联系：<http://www.christiannagel.com>。

欢迎您的反馈意见！

Christian Nagel

<http://www.christiannagel.com>

<http://www.thinktecture.com>