

Windows Communication Foundation 3.5 Unleashed

WCF 3.5

揭秘

Craig McMurtry

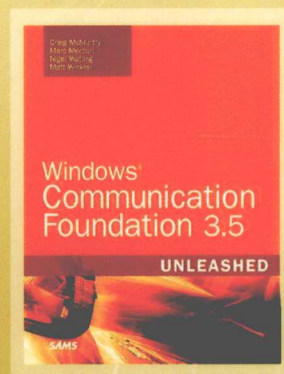
[美] Marc Mercuri 著

Nigel Watling

Matt Winkler

赵科平 龚岑 等译

- 来自微软WCF开发团队的权威之作
- 从这里，开始.NET企业级开发之旅
- 实战性极强的80多个经典示例



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

微软技术系列

Windows Communication Foundation 3.5 Unleashed

WCF 3.5

揭秘

人民邮电出版社
北京

图书在版编目(CIP)数据

WCF3.5 揭秘 / (美) 麦克默特里 (McMurtry, C.) 等著;
赵科平, 龚岑等译. —北京: 人民邮电出版社, 2009.9
(图灵程序设计丛书)

书名原文: Windows Communication Foundation 3.5
Unleashed
ISBN 978-7-115-21180-4

I. W… II. ①麦…②赵…③龚… 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133406号

内 容 提 要

本书是 WCF 领域内少有的经典著作, 以 WCF 的预备知识开篇, 以 REST 和 Plain XML 服务的体案例分析结束全书, 全面而深入地讲解了 WCF 技术的各个知识点。具体包括 WF、使用 WCF 和 WF、WCS、信息卡、标识元系统、与 COM+ 集成、定制行为、定制信道、定制传输、发布 / 订阅系统以及对等通信等。本书不仅系统地阐述了 WCF 的重要理论知识, 而且还提供了大量的实例和解决常见问题的最佳方案。它是 WCF 开发团队集体智慧的结晶, 是进行 WCF 开发必备的参考书。

本书适合于各层次的 .NET 开发者。

图灵程序设计丛书

WCF 3.5 揭秘

◆ 著 [美] Craig McMurtry Marc Mercuri
Nigel Watling Matt Winkler

译 赵科平 龚 岑 等

责任编辑 傅志红

执行编辑 印星星

◆ 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>

北京鑫正大印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 34.5

字数: 858千字

印数: 1—3 000册

2009年9月第1版

2009年9月北京第1次印刷

著作权合同登记号 图字: 01-2009-4224号

ISBN 978-7-115-21180-4/TP

定价: 79.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Authorized translation from the English language edition, entitled *Windows Communication Foundation 3.5 Unleashed* by Craig McMurtry, Marc Mercuri, Nigel Watling, and Matt Winkler, published by Pearson Education, Inc., publishing as Sams. Copyright © 2009 by Sams Publishing.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording or by an information storage retrieval system, without permission from Pearson Education, Inc.

Simplified Chinese-language edition copyright © 2009 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由美国Pearson Education Inc.授权人民邮电出版社独家出版。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，侵权必究。

上一版序

我从2001年开始从事WCF（那时称为“Indigo”）的研发工作，当时我们还是一个团队，我应该是第20位加入该团队的成员。在该团队任职期间，我是存储、可管理性、可靠消息和队列方面的项目经理主管。我们的团队有一个宏伟的愿景：通过为Web服务创建一个基础，使之可以实际应用到广泛的分布式计算问题上，促进下一代Web服务的发展。我们希望确保为各种业务而实现的Web服务可以提供安全的通信——机密性、签名、联合，于是分布式计算客户就可以为现实世界中的通信使用Web服务。我们希望确保Web服务可以融入到ACID模型的事务中，确保其能与数据驱动的系统或那些事务性计算任务进行有效的交互。我们希望确保Web服务可以某种方式编写，从而使广域的松散性不会再约束有意义的分布式应用程序的开发。在这些应用程序中，消息能以发送的次序达到你想发送的地址。这些目标如此涉及底层，现在看起来甚至有点奇怪，但是要知道，在2001年我们都接受这样的事实：当创建分布式系统时，其中的大部分工作都需要自力更生。

我们也知道大部分的计算环境都是异构的，许多厂商的系统同时并存，所以我们希望通过伟大的Web服务技术标准确保互操作性。我们决心实现良好的互操作性，并且全力地实现了目标。要在底层实现广泛的互操作性，WS-Security、WSAtomicTransactions、WS-ReliableMessaging、WS-Management、WS-Policy、WS-Transfer、WS-Eventing等协议都是必需的。但是，我们开始这个项目时什么都没有，它们都是由WCF团队的同事们后来实现的。回顾以前，我们可能会说：“我们当然希望通过被广泛接受的、可以相互组合的多个Web服务标准使用其他系统。”然而，这在2001年却是一个高不可攀的目标。

我们希望支持一种单一的编程模型，使开发者从面向消息转向远程过程性模式，或者从TCP转向HTTP或MSMQ等队列协议时，不需要从头学起。面对.NET Remoting、ASMX、Socket、MSMQ等众多的编程模型，用一套统一的API完成上述各模型的任务显然很困难，但我们仍然迎难而上。我们希望支持可扩展性，这样再出现新的消息交换模式、协议或加密机制时，也无需另外一种编程模式了。

作为首席项目经理，我帮助贯彻了可管理性这一理念，也就是任何应该交给IT专家决定的信息（当前的协议、加密机制、服务地址、监视，等等）都尽可能交给他们。这又是一个极高的目标：我们希望用WCF创建的应用程序具有最好的跟踪、监控和控制功能，易于通过优秀的配置和跟踪工具使用，而且能通过WMI与所有的Windows管理工具集成。简单地说，这个目标就是让使用WCF创建的应用程序比基于其他框架创建的应用程序更具可管理性，而且管理成本更低。

我们希望为现实世界创建重要的分布式应用程序能变得简单而又有趣，这可能是我们最富雄

心的目标。我们希望直观地引导开发者创建符合分布式系统最佳实践的应用程序。正如Steve Swartz（“简单而又有趣”最忠实的倡导者）所告诉我的，我们的目标是创建这样的一个框架，“如果你在山顶放一个球并让它滚下来，它就会自然地在一个地方停下来，这个地方有一个构架优良的服务，它帮你避免了分布式系统开发者在过去20年里犯下的所有错误。”

那我们做得怎么样呢？看看在Vista中和网络上作为.NET 3.0一部分发布的最终产品，我认为我们做得相当不错。WCF是一个统一的、可扩展的框架，它确实可以帮助你以一个统一的框架去创建现实世界中的安全的、可信赖的、互操作的、可管理的分布式应用程序，而且这个过程实际上很有趣（至少对于喜欢编程的人来说是这样的）。这花了我们六年时间，但是我们实现了所有的主要目标。实际上，我非常喜欢这个产品，现在我的“新”工作就是为微软创建完全基于WCF所提供的功能的新产品，以推广WCF（我现在很享受这份工作）。这个团队中的每位开发人员和项目经理的书架上都有这本书，几乎我们所有的人都把这本书作为开发或使用Web服务的必备参考书，其中还包括几位实际上开发了WCF的开发人员和项目经理。

最后说说Craig吧。我和Craig认识的时候，他身为WCF技术的传道者。他的精力和对项目的热情很有感染力，他是WCF的最忠实拥护者。要是有人问起：“我们可以支持这样的场景吗？”在90%的情况下Craig大概会这样回答：“哦，可以的，我上个星期就试过了，这是原型。”站在他的角度，他可以看到我们这些专注于具体功能特性的人所看不到的全貌。WCF能有今天，他直率的反馈、技术深度和热情功不可没。我相信他对WCF的热情和广博的知识会在每一个章节里闪光，我保证你会像我们一样，发现这是一本令人愉快的、有启发性的而且很有用的书。

Alex Weinert
微软公司团队项目经理

前言

WCF (Windows Communication Foundation, 开发代号Indigo) 是一种实现软件中的不同部分相互通信的技术。当前有许多其他类似的技术, 包括COM (Component Object Model, 组件对象模型) 和DCOM (分布式组件对象模型)、RMI (Remote Method Invocation, 远程方法调用)、MSMQ (Microsoft Message Queuing, 微软消息队列) 和WebSphere MQ。这些技术在某些应用场景中是有很有效的, 但是在其他场景中并不见得有效, 甚至有的时候根本无法使用。WCF意在成为一种可以在微软.NET程序集 (assembly) 需要和其他软件实体交换数据的任何场景中都适用的技术。实际上, WCF要成为最好的解决方案, 其性能与其他技术相比至少相当, 通常还要更好, 它提供的功能特性一点也不少, 只会更多, 而且它的确是最简单的解决方案。

具体地说, WCF由数个.NET库组成, 这些库中包含了几组添加到微软.NET Framework类库中的新类, 并且它们运行在.NET CLR (Common Language Runtime, 公共语言运行库) 2.0上。它添加了一些工具, 以使内建在Windows操作系统中的Web服务器——IIS 5.1及其以上版本能运行WCF解决方案。

WCF是和WPF (开发代号Avalon)、WCS (Windows CardSpace, 开发代号InfoCard) 以及WF一起免费发布的。在正式发布前, 这一组技术称为WinFX, 后来在2006年6月改名为.NET Framework 3.0。尽管名字这么叫, .NET Framework 3.0和3.5其实只是一组添加到.NET Framework 2.0中的仍然运行在2.0版.NET CLR上的类, 它还对Windows操作系统做了一些相应的改进, 如图0-1所示。^①

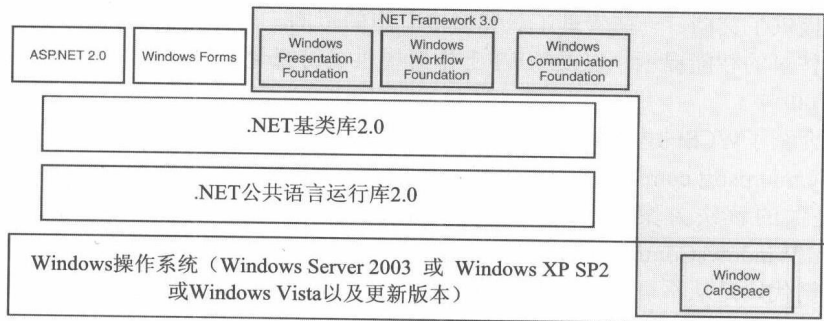


图0-1 .NET Framework 3.0

^① .NET Framework 3.5只是在.NET Framework 3.0基础上进行了扩展, 主要是引入了LINQ和ASP.NET AJAX, 核心部分只进行了微调。——编者注

可以在Windows XP SP2、Windows Server 2003和Windows Server 2003 R2上安装.NET Framework 3.0和3.5, Windows Vista上已经预装了运行库组件。在Windows Server 2008中,你可以通过应用服务器基础角色服务来添加.NET Framework 3.0。.NET Framework 3.0中只有一小部分功能特性是Windows Vista和更新版本操作系统所特有的。

.NET Framework 3.5是在.NET Framework 3.0之上扩充的。本书提及的特性包括构建WCF服务需要的Web协议支持,包括AJAX、JSON、REST、POX、RSS和ATOM,以及启用工作流(workflow-enabled)服务和Visual Studio 2008的完全工具支持。在开发期间,.NET Framework 3.5被分为“红”位和“绿”位^①。“红”位指.NET Framework 3.0的一些特性,目的是提供服务包级别的兼容性。所有可以运行在3.0中的代码均可运行在3.5中。“绿”位则提供新增特性。此外,程序集包含新功能的新增部分将不会影响现有代码。也就是说本书的所有代码都将工作在.NET Framework 3.5中,并且本书中的所有代码应该可以工作在.NET Framework 3.0中(除了那些介绍.NET Framework 3.5新增代码的部分)。

本书并不是包罗万象的WCF百科全书,而是提供了在大多数实际应用中使用WCF所需的知识。

本书通过演示如何使用WCF来解释它的各个方面。所以,一般每一章都会给出一个能展示这项技术某方面应用的实例,并且会提供创建该实例每个步骤的详细解释。所有会用C#编程的读者若想边做边学,都可以遵循这些步骤。而那些只是希望阅读本书的读者也可以详细了解到WCF的各个特性以及如何使用它们。

为了运行各章节提供的例子,需要安装带有C#编译器的任何版本的Visual Studio 2005或2008。<http://msdn.microsoft.com/vstudio/express/>上有免费版本供下载。同时还需要安装IIS、ASP.NET和MSMQ。

当然,.NET Framework 3.0或者3.5是必需的,可以从<http://www.microsoft.com/downloads/>下载。各章中的操作指南都假定读者已经安装了.NET Framework 3.0或者3.5的所有运行库和开发组件。运行库组件由Windows Vista预装,或在Windows Server 2008上通过服务器管理器添加。开发组件由一个SDK(Software Development Kit,软件开发工具包)和两个Visual Studio 2005的升级包组成。SDK提供了文档、一些管理工具和很多有用的示例。而升级包对Visual Studio 2005增加了编辑配置文件时的智能提示,以及为WF工作流提供了可视化的设计器。这些特性同样包含在Visual Studio 2008中。

本书同样介绍了WCS的内容。为了能充分利用WCS提供的功能,读者需要安装IE 7。IE 7可以从<http://www.microsoft.com/downloads>下载。

每章所提供的解决方案的源代码可以从出版商的网站上下载^②,也可以从<http://www.cryptmaker.com/WindowsCommunicationFoundationUnleashed>上下载。为了避免Visual Studio弹出对话框说明示例代码是从不可信地址获得的,在解压下载的压缩文件包之前,读者可以右键单击

① 参看Somasegar的一篇博客<http://blogs.msdn.com/somasegar/archive/2006/05/18/601354.aspx>。Somasegar是微软全球副总裁,负责微软公司开发部的工作。——译者注

② 本书所附代码也可以从图灵公司网站(www.turingbook.com)本书配套网页下载。——编者注

压缩包, 在弹出的菜单中选择Properties (属性), 然后再单击Unblock (解除锁定) 按钮 (如图0-2所示)。

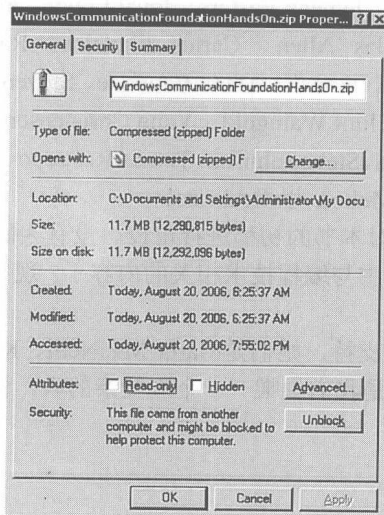


图0-2 解锁下载的代码压缩包

Vista操作系统支持用Visual Studio 2008进行开发, 而当安装SP1中的一个更新后, 它也将支持使用Visual Studio 2005进行开发。在Vista操作系统中使用较早版本的Visual Studio 2005的开发者会碰到一些兼容性问题。在这种情况下, 可以做两件事情尽量减少这些问题。第一, 禁用Vista的用户账户保护 (User Account Protection) 功能; 第二, 在启动Visual Studio 2005时, 总是从可执行文件或快捷方式的右键菜单中选择Run As, 然后从Run As对话框中选择一个管理员账户。

正如.NET Framework 3.5和.NET Framework 3.0的差别一样, 本书与上一版的差别也体现在这些方面。新增内容有Visual Studio 2008支持, 第3章现在包含了持久化服务。WCS章节则更新了操作界面以及一些新特性。第18章新增了新的联合以及JSON API。最重要的改变则是重写了第6章, 使得WCF和WF的整合上有了大量的改进。

许多人为本书的出版做出了贡献。作者希望感谢以下人士: Joe Long、Eric Zinda、Angela Mills、Omri Gazitt、Steve Swartz、Steve Millett、Mike Vernal、Doug Purdy、Eugene Osvetsky、Daniel Roth、Ford McKinstry、Craig McLuckie、Alex Weinert、Shy Cohen、Yasser Shohoud、Kenny Wolf、Anand Rajagopalan、Jim Johnson、Andy Milligan、Steve Maine、Ram Pamulapati、Ravi Rao、Mark Garbara、Andy Harjanto、T. R. Vishwanath、Doug Walter、Martin Gudgin、Marc Goodner、Giovanni Della-Libera、Kirill Gavrylyuk、Krish Srinivasan、Mark Fussell、Richard Turner、Ami Vora、Ari Bixhorn、Steve Cellini、Neil Hutson、Steve DiMarco、Gianpaolo Carraro、Steve Woodward、James Conard、Nigel Watling、Vittorio Bertocci、Blair Shaw、Jeffrey Schlimmer、Matt Tavis、Mauro Ottoviani、John Frederick、Mark Renfrow、Sean Dixon、Matt Purcell、Cheri Clark、Mauricio Ordenez、Neil Rowe、Donovan Follette、Pat Altimore、Tim Walton、Manu Puri、Ed Pinto、Erik Weiss、Suwat

Chitphakdibodin、Govind Ramanathan、Ralph Squillace、John Steer、Brad Severtson、Gary Devendorf、Kavita Kamani、George Kremenliev、Somy Srinivasan、Natasha Jethanandani、Ramesh Seshadri、Lorenz Prem、Laurence Melloul、Clemens Vasters、Joal Lowy、John Justice、David Aiken、Larry Buerk、Wenlong Dong、Nicholas Allen、Carlos Figueira、Ram Poornalingam、Mohammed Makarechian、David Cliffe、David Okonak、Atanu Banerjee、Steven Metsker、Antonio Cruz、Steven Livingstone、Vadim Meleshuk、Elliot Waingold、Yann Christensen、Scott Mason、Jan Alexander、Johan Lindfors、Hanu Kommalapati、Steve Johnson、Tomas Restrepo、Tomasz Janczuk、Garrett Serack、Jeff Baxter、Arun Nanda、Luke Melton 以及Al Lee。

这里还要感谢John Lambert对本书的初稿进行审校。没有人能比一个专业而又富有经验的软件测试工程师更加适合审阅本书中与编程技术相关的内容了。现在，书中遗留的任何错误都归咎于我们作者自己。

我们还要特别感谢妻子们的支持，她们是Marta MacNeill、Kathryn Mercuri、Sylvie Watling和Libby Winkler。在目前，Matt是我们中唯一一个当父亲的人，他还要感谢他的女儿Grace。

目 录

第一部分 WCF 简介

第 1 章 预备知识	2
1.1 分部类型	2
1.2 泛型	3
1.3 可空值类型	5
1.4 轻量级事务管理器	7
1.5 角色提供程序	8
1.6 小结	10
1.7 参考文献	10
第 2 章 基础知识	11
2.1 背景知识	11
2.2 输入服务 (Enter Services)	13
2.3 WCF	15
2.4 服务模型	16
2.4.1 软件资源	20
2.4.2 创建一个访问资源的服务	22
2.4.3 使用服务	36
2.4.4 在 IIS 中承载服务	46
2.4.5 改变服务通信方式	50
2.5 Visual Studio 2008 工具支持	52
2.6 小结	57
2.7 参考文献	57
第 3 章 数据表示和持久化支持	59
3.1 背景知识	59
3.2 XmlSerializer 和 DataContractSerializer	60
3.3 XML 物神	63
3.3.1 创建服务	65
3.3.2 创建客户端	66
3.3.3 满足查看 XML 的欲望	67
3.3.4 DataContractSerializer 值得吗	67
3.4 使用 DataContractSerializer	67
3.5 异常处理	79
3.6 持久化服务	82
3.6.1 为什么要持久化服务	82
3.6.2 实现持久化服务	83
3.7 小结	88
3.8 参考文献	88
第 4 章 会话、可信赖会话、队列与事务	89
4.1 可信赖会话	89
4.2 会话管理	92
4.3 队列交付	93
4.4 事务	96
4.5 小结	103
第 2 部分 WF 介绍	
第 5 章 WF 简介	106
5.1 WF 是什么	106
5.2 活动	107
5.2.1 开箱即用活动	109
5.2.2 创建定制服务	110
5.2.3 活动通信	115
5.2.4 设计行为	121
5.2.5 事务和补偿	122
5.3 workflow 模型	124
5.3.1 顺序 workflow	124

5.3.2 状态机工作流	131	7.3.4 传输安全示例	186
5.3.3 定制根活动	132	7.4 使用消息安全	191
5.4 工作流承载	132	7.5 身份模拟与授权	196
5.4.1 承载运行时	133	7.6 身份模拟	196
5.4.2 运行时服务	134	7.7 授权	199
5.4.3 定制服务	141	7.8 取消对Windows的更改	205
5.5 规则引擎	143	7.8.1 卸载证书	205
5.5.1 作为条件的规则	144	7.8.2 从IIS中删除SSL配置	206
5.5.2 ConditionedActivityGroup 活动	146	7.8.3 从HTTP.SYS中删除SSL配置	207
5.5.3 作为策略的规则	147	7.8.4 恢复服务器的标识	207
5.6 小结	149	7.9 小结	207
5.7 参考文献	149	7.10 参考文献	207
第6章 使用WCF和WF	150	第8章 WCS、信息卡和标识元系统	209
6.1 消费服务	150	8.1 标识的角色	209
6.1.1 在自定义活动中调用服务	150	8.2 Microsoft Passport和其他标识验证 方法	211
6.1.2 以.NET Framework 3.5的方式 使用Send活动	154	8.3 标识规则	213
6.1.3 扩展Send活动	156	8.4 标识元系统	214
6.2 组合多个服务	158	8.5 信息卡和CardSpace	218
6.3 将工作流公开为服务	158	8.6 管理信息卡	220
6.3.1 在WCF服务中承载 (.NET 3.0)	159	8.7 架构、协议和安全	225
6.3.2 将工作流公开为服务 (.NET 3.5)	163	8.8 CardSpace与企业	235
6.3.3 创建工作流服务	167	8.9 .NET Framework 3.5新特性	238
6.3.4 上下文	168	8.10 .NET Framework 3.5的HTTP支持	239
6.3.5 通信模式	170	8.11 小结	241
6.4 小结	179	8.12 参考文献	241
6.5 参考文献	180	第9章 使用信息卡保护应用程序的 安全	243
第三部分 安 全		9.1 标识元系统开发	243
第7章 安全基础	182	9.2 CardSpace的简单示例	245
7.1 确保通信安全的基本任务	182	9.3 CardSpace示例的准备工作	246
7.2 传输安全和消息安全	183	9.3.1 开启IIS和ASP.NET 2.0	246
7.3 使用传输安全	183	9.3.2 获取X.509证书	246
7.3.1 安装证书	183	9.3.3 将证书导入证书存储区	247
7.3.2 指定服务器即将提供的证书	185	9.3.4 更新Hosts文件使DNS项与证书 匹配	247
7.3.3 配置服务器的身份	186	9.3.5 IIS设置	248
		9.3.6 读取证书私钥	248

9.3.7 HTTP配置	249	11.2 使用COM+服务模型配置工具	314
9.4 为WCF应用程序添加信息卡	250	11.3 将COM+组件以WCF Web服务提供	315
9.5 添加信息卡	253	11.4 从COM调用WCF服务	320
9.6 使用Federation绑定	258	11.4.1 创建服务	320
9.7 捕获异常	259	11.4.2 创建客户端	322
9.8 处理发布的令牌	260	11.4.3 创建VBScript文件	324
9.9 使用Metadata Resolver	261	11.4.4 测试解决方案	324
9.10 为浏览器应用程序添加信息卡	262	11.5 与MSMQ集成	324
9.11 创建托管卡	271	11.6 创建与MSMQ集成的WCF服务	325
9.12 创建简单的安全令牌服务	273	11.6.1 创建请求	325
9.13 通过HTTP使用CardSpace	275	11.6.2 创建服务	326
9.14 小结	276	11.6.3 创建客户端	328
9.15 参考文献	276	11.6.4 测试	331
第 10 章 高级安全	277	11.7 小结	332
10.1 序幕	277	第 12 章 互操作性	333
10.2 利用声明保证资源安全	278	12.1 小结	335
10.2.1 基于声明授权和基于角色 授权	278	12.2 参考文献	335
10.2.2 基于声明授权和访问控制 列表	279	第五部分 扩展 WCF	
10.3 使用XSI运用基于声明的安全	281	第 13 章 定制行为	338
10.3.1 利用Windows标识授权对局 域网资源的访问	281	13.1 扩展WCF	338
10.3.2 改进初始方案	286	13.2 用定制行为扩展服务模型	339
10.3.3 添加STS作为联合的基础	292	13.2.1 声明你要提供的行为的类型	339
10.3.4 重新配置Resource Access 服务	302	13.2.2 将定制行为附加到操作或终 结点上	342
10.3.5 重新配置客户端	305	13.2.3 将定制行为告知WCF	342
10.3.6 体验带有XSI的、联合的、基 于声明的标识的强大	307	13.3 实现定制行为	342
10.4 基于声明的安全和联合安全	308	13.3.1 声明	343
10.5 小结	309	13.3.2 附加	343
10.6 参考文献	309	13.3.3 告知	344
第四部分 集成与互操作		13.4 实现各种类型的定制行为	350
第 11 章 遗留集成	312	13.4.1 操作选择器	350
11.1 COM+集成	312	13.4.2 参数检查器	351
11.1.1 支持的接口	313	13.4.3 消息格式器	353
11.1.2 选择承载模式	313	13.4.4 消息检查器	355
		13.4.5 实例上下文提供者	357
		13.4.6 实例提供者	358
		13.4.7 操作调用器	359
		13.5 实现WSDL导出扩展	360

13.6 定制行为实战	362	15.7 参考文献	400
13.7 小结	362		
13.8 参考文献	363		
第 14 章 定制信道	364	第六部分 具体案例	400
14.1 绑定元素	364	第 16 章 发布/订阅系统	402
14.1.1 出站通信	365	16.1 使用回调契约的发布/订阅系统	402
14.1.2 进站通信	365	16.2 使用 MSMQ 实际通用组播的发布/ 订阅系统	408
14.2 信道是有形状的	366	16.3 使用流的发布/订阅系统	414
14.3 可能需要信道支持会话	367	16.3.1 流传输模式	414
14.4 根据契约匹配信道	367	16.3.2 使用流传输模式传送定制流	418
14.5 通信状态机	369	16.3.3 使用流传输模式和定制流实 现发布/订阅系统	421
14.6 创建定制绑定元素	369	16.4 小结	425
14.6.1 理解起点	370	16.5 参考文献	425
14.6.2 提供一个支持出站通信的定 制绑定元素	371	第 17 章 对等通信	426
14.6.3 更改自定绑定元素以支持入 站通信	376	17.1 在点对点应用中使用结构化数据	426
14.6.4 通过配置应用定制绑定元素	381	17.2 利用 Windows 对等网络开发平台	427
14.7 小结	384	17.3 理解 Windows 对等网络	427
第 15 章 定制传输	385	17.4 使用对等信道	428
15.1 传输信道	385	17.4.1 终结点	428
15.1.1 进站通信	385	17.4.2 绑定	428
15.1.2 出站通信	386	17.4.3 地址	431
15.2 消息编码器	386	17.4.4 契约	431
15.3 完善栈	386	17.4.5 实现	432
15.4 实现传输绑定元素和编码器绑定 元素	387	17.5 对等信道实战	432
15.4.1 场景	387	17.5.1 设想解决方案	432
15.4.2 要求	388	17.5.2 设计数据结构	435
15.4.3 TcpListener 类和 TcpClient 类	388	17.5.3 定义服务契约	437
15.5 实现支持任意 TCP 的定制绑定 元素	390	17.5.4 实现服务契约	439
15.5.1 配置	390	17.5.5 配置终结点	440
15.5.2 定制传输绑定元素	392	17.5.6 将消息发给特定的节点	441
15.5.3 信道监听器	394	17.5.7 定制对等名称解析	444
15.5.4 传输信道	396	17.5.8 查看对等信道的工作	447
15.5.5 消息编码器	398	17.6 对等信道和 People Near Me	449
15.5.6 使用定制传输绑定元素	399	17.7 小结	449
15.6 小结	400	17.8 参考文献	449
		第 18 章 REST 和 Plain XML 服务	450
		18.1 REST	450

18.2	REST服务	451	19.3	小结	497
18.3	REST服务和Plain XML	451	第 20 章	版本处理	498
18.4	REST服务的优点和局限	451	20.1	版本术语	498
18.5	使用WCF创建REST POX服务	452	20.2	版本问题集	499
18.5.1	REST POX服务终结点的地址	452	20.2.1	添加新的操作	499
18.5.2	REST POX服务终结点的绑定	452	20.2.2	更改操作	501
18.5.3	REST POX服务终结点的契约	453	20.2.3	删除一个操作	504
18.5.4	实现	453	20.2.4	更改绑定	504
18.5.5	应用程序范例	454	20.2.5	决定废除一个终结点	504
18.6	.NET Framework 3.5中的RSS和ATOM联合	458	20.2.6	更改服务终结点的地址	504
18.7	JSON	463	20.3	集中的生命周期管理	505
18.8	小结	466	20.4	小结	506
18.9	参考文献	466	20.5	参考文献	507
第七部分 WCF 应用程序的生命周期					
第 19 章	可管理性	470	第八部分 规 范		
19.1	检测功能与工具	470	第 21 章	规范	510
19.1.1	配置系统和配置编辑器	471	21.1	采用WCF	510
19.1.2	Service Configuration Editor	473	21.2	使用WCF地址	512
19.1.3	安全事件的可配置审计	477	21.3	使用WCF绑定	513
19.1.4	消息日志、活动跟踪和Service Trace Viewer	480	21.4	使用WCF契约	515
19.1.5	性能计数器	488	21.4.1	使用结构化契约	517
19.1.6	WMI提供程序	489	21.4.2	使用行为契约	519
19.2	完善管理工具	497	21.5	使用WCF服务	521
			21.6	使用WCF客户端	528
			21.7	使用大量数据	533
			21.8	调试WCF应用程序	534
			21.9	小结	535
			21.10	参考文献	535

Part 1

第一部分

WCF 简介

本 部 分 内 容

- 第1章 预备知识
- 第2章 基础知识
- 第3章 数据表示和持久化支持
- 第4章 会话、可信赖会话、队列与事务

本章内容

- 分部类型
- 泛型
- 可空值类型
- 轻量级事务管理器
- 角色提供程序

为了能正确理解并有效使用WCF，必须熟悉.NET Framework 2.0和.NET CLR的一些功能特性。本章将介绍书中需要用到的分部类型、泛型、可空值类型、轻量级事务管理器和角色提供程序。对这些特性的讲解并不是为了面面俱到，仅仅只是充分阐明它们在后面章节中的用途。

1.1 分部类型

Microsoft Visual C# 2005允许一个类型的定义由多个分部定义组成，这些分部定义可以分布到任意多个属于同一模块的源代码文件中。可以通过修饰符`partial`使用这个特性，即在类、接口或结构的定义前添加该修饰符。下面是一个类的部分定义：

```
public partial MyClass
{
    private string myField = null;

    public string MyProperty
    {
        get
        {
            return this.myField;
        }
    }
}
```

下面是另一部分定义：

```
public partial MyClass
{
    public MyClass()
```