

<<WCF 4高级编程>>

图书基本信息

书名：<<WCF 4高级编程>>

13位ISBN编号：9787302266990

10位ISBN编号：7302266999

出版时间：2011-10

出版时间：清华大学出版社

作者：[美]Pablo Cibraro,Kurt Claeys,Fabio Cozzolino

页数：423

译者：吴文国

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<WCF 4高级编程>>

内容概要

本书讨论基于 .NET 4.0 架构的 WCF (Windows Communication Foundation) 开发编程技术。

WCF 是 .NET 架构中的技术，用来创建面向服务的应用程序、交换不同通信方案中的消息，以及执行由服务操作生成的工作流。

通过学习本书，读者将掌握面向服务的基本原理，学习通信模式和发现如何显式定义业务流程。

读者还将学习如何使用这些技术的不同部分来实现上述功能，并清晰地理解 WCF

4 的不同组件是如何相互支撑、协作，成为一个综合框架的，进而支持企业级分布式应用程序的各个方面。

除了介绍 WCF 技术外，本书还采取了一种实用的讲法，介绍了 3 个案例(面向服务、通信和业务流程)并逐步实现了它们。

本书将引导读者如何将 WCF 和 Visual Studio 工具用于实际开发的各个方面。

在构建 WCF 4 知识体系的过程中，读者将掌握如何有效地利用 Visual Studio 2010，以构建能够最大化利用 WCF 4 新增功能的解决方案。

读者将学习如何创建应用程序(作为开发人员和架构师)，如何把这些应用程序集成为 WCF 4 中新的编程范式。

读者将学习如何用建立在 WCF 技术和 .NET 服务之上的新架构模式创建一个解决方案，将学习如何解决在实现 WCF

4 项目所需要的新编程范式和新结构风格的过程中遇到的实际问题。

本书中介绍的实例决不是简单的“hello

world”示例，而是引导读者如何建立架构正确的解决方案，并向读者展示最佳的编程风格。

本书 4 位作者都具有丰富的实际开发经验。

在日常的开发工作中他们都曾遇到过这些问题，并用最佳的指导和实践提出了切实可行的解决方法。

他们把各自的实际经验融入到了本书中。

<<WCF 4高级编程>>

作者简介

作者：（美国）西布拉奥（Pablo Cibraro）（美国）克雷斯（Krut Claeys）（美国）科佐利诺（Fabio Cozzolino）（美国）格拉布纳（Johann Grabner）译者：吴文国 西布拉奥（Cibraro P.），是一位企业架构师、Tellago地区的CTO。

克雷斯（Claeys K.），是.NET企业级应用程序架构师和培训师，热衷于用WCF实现集成，拥有MCT、MCSD认证。

科佐利诺（Cozzolino F.），是FimeSan公司的软件架构师、分析师和开发人员。

格拉布纳（Grabner J.）是微软Learning Partners项目的自由培训师。

<<WCF 4高级编程>>

书籍目录

目录

- 第1章 设计原理与设计模式1
 - 1.1 SOA简介1
 - 1.2 SOA架构的4条原则3
 - 1.2.1 边界显式定义3
 - 1.2.2 服务自动化3
 - 1.2.3 服务共享的是模式和契约，而不是类3
 - 1.2.4 基于策略的服务兼容性4
 - 1.3 服务的内部结构4
 - 1.4 组织业务流程中的服务7
 - 1.5 SOA的底层技术7
 - 1.5.1 SOAP8
 - 1.5.2 WS - * Protocols8
 - 1.5.3 WSDL8
 - 1.6 契约优先原则9
 - 1.7 WCF和.NET服务如何实现SOA模式10
 - 1.7.1 模式10
 - 1.7.2 解耦契约：接口与实现10
 - 1.7.3 代理模式11
 - 1.7.4 OperationContext模式11
 - 1.7.5 并发契约11
 - 1.7.6 数据保密性12
 - 1.7.7 Web服务原子事务12
 - 1.7.8 会话外观12
 - 1.7.9 异常保护12
 - 1.8 通信与集成模式13
 - 1.8.1 集成模式14
 - 1.8.2 消息交换模式16
 - 1.8.3 消息模式22
 - 1.9 业务流程模式26
 - 1.9.1 流程管理器26
 - 1.9.2 在工作流声明中的模式28
- 第2章 服务契约与数据契约31
 - 2.1 服务契约32
 - 2.2 数据契约32
 - 2.3 消息契约32
 - 2.4 契约与代码32
 - 2.5 汽车租赁服务——实现示例33
 - 2.5.1 步骤1：定义服务契约33
 - 2.5.2 步骤2：提取服务元数据34
 - 2.5.3 步骤3：服务的实现38
 - 2.5.4 步骤4：生成客户端代码39
 - 2.5.5 [ServiceContract]和

<<WCF 4高级编程>>

- [OperationContract]特性40
- 2.6 数据契约42
 - 2.6.1 数据契约详解47
 - 2.6.2 KnownTypes特性49
- 2.7 服务契约与数据契约的版本控制52
 - 2.7.1 数据契约的版本控制52
 - 2.7.2 双向版本控制54
 - 2.7.3 服务契约版本控制的最佳实践56
 - 2.7.4 数据契约版本控制的最佳实践56
- 2.8 消息契约57
- 第3章 绑定63
 - 3.1 绑定的工作原理64
 - 3.2 地址66
 - 3.3 行为67
 - 3.3.1 服务行为67
 - 3.3.2 操作行为70
 - 3.3.3 终结点行为71
 - 3.3.4 契约行为74
 - 3.4 绑定75
 - 3.4.1 basicHttpBinding和wsHttpBinding76
 - 3.4.2 netTcpBinding77
 - 3.4.3 netMsmqBinding77
 - 3.4.4 基于上下文的绑定77
 - 3.4.5 如何选择要使用的绑定78
 - 3.5 配置绑定79
 - 3.5.1 基址80
 - 3.5.2 默认配置82
 - 3.5.3 设置多绑定85
 - 3.6 修改绑定86
 - 3.6.1 绑定的属性86
 - 3.6.2 创建自定义绑定88
 - 3.6.3 重用自定义绑定90
 - 3.7 持久双工服务93
 - 3.8 PollingDuplexHttpBinding绑定：HTTP轮询95
- 第4章 客户端97
 - 4.1 Basic Profile 1.1标准98
 - 4.2 .NET客户端98
 - 4.2.1 共享WSDL-契约98
 - 4.2.2 共享WSDL契约和数据契约-DLL101
 - 4.2.3 共享接口和数据契约-DLL102

<<WCF 4高级编程>>

- 4.3 REST104
 - 4.3.1 REST与WCF106
 - 4.3.2 使用REST客户端112
 - 4.3.3 使用REST Starter Kit工具113
- 4.4 AJAX与WCF的关系114
- 4.5 WCF 4与Silverlight118
- 第5章 实例化121
 - 5.1 实例上下文模式122
 - 5.1.1 PerCall模式122
 - 5.1.2 Single模式125
 - 5.1.3 PerSession模式127
 - 5.2 服务的生命周期131
 - 5.3 性能137
 - 5.3.1 限流137
 - 5.3.2 最佳做法144
 - 5.3.3 负载均衡144
- 第6章 工作流服务147
 - 6.1 剖析工作流服务148
 - 6.2 声明式服务149
 - 6.3 接收与发送活动152
 - 6.3.1 接收活动152
 - 6.3.2 发送活动155
 - 6.3.3 SendAndReceiveReply和ReceiveAndSendReply活动157
 - 6.4 实现工作流服务的第一个示例157
 - 6.5 配置工作流服务161
 - 6.6 实现消息的关联164
 - 6.7 托管工作流服务174
- 第7章 理解WCF安全177
 - 7.1 Web服务安全的历史演变177
 - 7.2 Web服务安全的基本原则178
 - 7.2.1 验证178
 - 7.2.2 授权179
 - 7.2.3 消息的完整性179
 - 7.2.4 消息的机密性179
 - 7.3 传输安全与消息安全180
 - 7.3.1 传输安全180
 - 7.3.2 消息安全180
 - 7.4 WCF安全概述182
- 第8章 WCF安全实战195
 - 8.1 验证的起步195
 - 8.2 基于声明的身份验证模型196
 - 8.3 验证实战199
 - 8.3.1 建立在消息安全之上的

<<WCF 4高级编程>>

- 用户验证199
- 8.3.2 建立在传输安全之上的用户名验证209
- 8.3.3 利用消息安全实现X509证书的相互验证213
- 8.3.4 建立在消息安全之上的Kerberos验证221
- 8.4 声明转换与安全上下文的初始化226
- 8.5 服务授权228
 - 8.5.1 基于角色的授权228
 - 8.5.2 基于声明的验证和验证上下文232
 - 8.5.3 授权管理器233
- 第9章 WCF联合验证237
 - 9.1 联合验证237
 - 9.1.1 STS服务简介238
 - 9.1.2 多域之间的联合验证238
 - 9.1.3 SAML语言239
 - 9.2 WIF架构241
- 第10章 Windows Azure Platform AppFabric255
 - 10.1 服务总线 and 访问控制简介256
 - 10.2 使用服务总线259
 - 10.3 中继服务264
 - 10.4 WCF中继绑定265
 - 10.4.1 netOneWayRelayBinding 绑定266
 - 10.4.2 netEventRelayBinding 绑定268
 - 10.4.3 netTcpRealyBinding 绑定270
 - 10.4.4 HTTP中继绑定272
 - 10.5 使用访问控制服务(ACS)274
 - 10.5.1 服务名称空间275
 - 10.5.2 作用域276
 - 10.5.3 发送者276
 - 10.5.4 规则276
 - 10.5.5 把第一个服务集成到访问控制中277
- 第11章 创建一个SOA案例285
 - 11.1 需求分析285
 - 11.2 建立解决方案286
 - 11.3 创建接口288
 - 11.3.1 创建CarManagement 接口291
 - 11.3.2 创建Customer接口292

<<WCF 4高级编程>>

- 11.3.3 创建Rental接口293
- 11.3.4 创建External接口294
- 11.4 创建服务296
- 11.5 创建宿主程序298
- 11.6 创建数据库306
- 11.7 实现服务306
 - 11.7.1 为CustomerService和RentalService服务创建数据库访问307
 - 11.7.2 创建CarManagement服务308
- 11.8 公开元数据310
- 11.9 创建CarManagement客户端313
- 11.10 创建RentalApplication应用程序320
- 11.11 添加错误处理功能325
- 11.12 模拟客户端328
- 11.13 扩展CarManagement接口以接受汽车子类328
- 11.14 实现ExternalInterface-Facade330
 - 11.14.1 调用ExternalInterface-Facade331
 - 11.14.2 给参与事务的方法设置事务支持333
 - 11.14.3 为servicehost配置额外的终结点333
- 第12章 创建通信和集成案例335
 - 12.1 需求分析335
 - 12.2 建立解决方案337
 - 12.3 创建HQOrderEntryService-Interface接口项目338
 - 12.4 创建HelperLib类库340
 - 12.5 创建HQOrderEntry-Implementation项目341
 - 12.6 创建HQOrderEntryServiceHost项目342
 - 12.7 创建OrderEntryApplication项目343
 - 12.8 创建LocalOrderEntryInterface接口项目345
 - 12.9 继续HQOrderEntry-Implementation项目346
 - 12.10 创建HQProductServiceASMX项目348
 - 12.10.1 创建Web服务348

<<WCF 4高级编程>>

- 12.10.2 把HQProductService-ASMX作为服务引用添加到OrderEntryService-Implementation项目中349
- 12.10.3 为CheckIfOrderIsValid方法编写代码350
- 12.10.4 为TranslateProduct-Description方法编写代码351
- 12.10.5 为ConvertOrderEntry-Schema方法编写代码351
- 12.11 创建HQLocalizationService服务352
- 12.12 为RouteOrderEntry方法编写代码354
- 12.13 创建RealTimeOrderTracking-Application应用程序355
 - 12.13.1 为RealTimeOrder-TrackingApplication方法编写代码355
 - 12.13.2 添加IsubscribeToOrder-TrackingInfo 接口356
 - 12.13.3 实现SubscribeService方法356
 - 12.13.4 在订单处理时调用订阅服务357
 - 12.13.5 打开SubscribeService服务358
 - 12.13.6 订阅来自RealTimeOrder-TrackingApplication的事件359
 - 12.13.7 配置HQOrderEntry-ServiceHost宿主359
- 12.14 创建路由360
- 12.15 配置HQOrderEntry-ServiceHost宿主362
- 第13章 创建业务流程365
 - 13.1 需求分析365
 - 13.2 建立解决方案366
 - 13.3 创建数据契约367
 - 13.4 创建CalculateReferenceID-Service服务369
 - 13.5 创建ReceiveApprovedHoliday-RequestsService项目370
 - 13.6 给HolidayRequestActivity-Library项目添加服务引用373

<<WCF 4高级编程>>

- 13.6.1 添加CalculateReferenceID-Service项目373
- 13.6.2 添加对ReceiveApproved-HolidayRequestsService服务的引用374
- 13.6.3 开发HolidayRequestProcess项目375
- 13.6.4 添加工作流375
- 13.6.5 创建变量376
- 13.6.6 配置Receive活动378
- 13.6.7 配置Send活动379
- 13.6.8 配置ApproveRequest操作的ReceiveAndSendReply活动383
- 13.7 开发HolidayRequest-ProcessHost项目388
- 13.8 测试这个服务宿主能否正确公开元数据389
- 13.9 开发ManagersHoliday-RequestApprovalApplication项目391
- 13.10 创建SqlWorkflowInstance-Store项目391
- 第14章 托管服务393
- 14.1 自托管394
- 14.1.1 ServiceHost和服务Host-Base394
- 14.1.2 实现一个自定义的ServiceHost397
- 14.2 IIS托管399
- 14.2.1 ServiceHostFactory与服务HostFactoryBase401
- 14.2.2 使用CustomService-HostFactory类401
- 14.2.3 不通过svc文件承载服务402
- 14.2.4 Windows激活服务403
- 14.3 用Windows AppFabric管理和跟踪终结点406
- 14.3.1 建立Windows Server AppFabric407
- 14.3.2 使用AppFabric监视服务409
- 14.3.3 启动事件查看器412
- 14.4 路由服务413
- 14.4.1 基于内容的路由413
- 14.4.2 协议和安全桥接417

<<WCF 4高级编程>>

14.4.3 错误处理418

14.5 云托管419

14.5.1 在Windows Azure中

托管WCF服务420

14.5.2 Windows Azure Platform

AppFabric服务总线420

14.5.3 通过云中继服务421

章节摘录

版权页：插图：在这个架构中，读者可以看到是由代理把消息插入到通道中。

在通道中，消息在传递给服务之前，要经过其他层，如协议层和编码层等。

在服务端，消息由一个相应的通道栈接收，接收后把消息通过相应的层向上传递，经过编码层、协议层，再由分配器接收，分配器对消息进行分析，然后确定要调用哪个方法。

2.内容富集器 内容富集器是运行在功能层的一个模式。

当应用程序发送数据，而数据并不包含接收者所需要的全部信息时，便需要一个内容富集器。

信息缺失的原因可能是历史因素——可能是在设计的时候还不需要这些信息作为消息的一部分，因为这些信息总是出现在数据库中。

但是在分布式环境中，并非所有服务都可以访问数据库。

在发送者与接收者之间，另一个服务需要读取这个消息，对它进行分析，并添加必要的信息。

对这个额外信息的查询可以通过查询数据库、执行业务规则或直接使用硬编码来实现。

富集数据可以说是一个服务操作的执行过程。

富集内容意味着数据契约的动态变化。

这种富集数据的行为会导致给数据契约增加额外的属性，并赋给这个新增的属性某些值，然后把修改后的数据契约发送给其他服务。

用通用方法执行这个操作，意味着WCF需要采用一种方式，来接收还没有在操作签名中定义数据契约的消息，并用代码未知的数据契约发送此消息。

大多数时候，富集器只是一些通用函数，它们以通用格式（XML）接收数据，改变格式后再把这个数据发送给其他服务。

WCF支持使用消息类接收无类型数据。

使用函数接收消息类的一个参数，并用一个消息响应这个调用。

如果是一个单向无返回值的方法，则不发送任何响应消息。

消息类是一类包罗万象的数据契约。

在一个以消息类为输入参数的方法中，每一个还没有映射到另一个操作的消息都可以接收。

因此建立通用型的内容富集器可以通过创建一个如下这样的服务来实现：此服务中有一个使用消息参数的操作。

当然，这样的代码不能直接访问这个数据契约，因为它并不知道数据契约的细节。

但是这样的代码可以读取XML数据，可以修改数据内容，还可以在需要时添加XML结点。

3.消息路由 在许多情况下，消息的发送者并不知道消息最终要发送到哪里，客户端也不知道消息的实际接收者，因此需要一个路由方法来决定终结点的实际地址。

在路由过程中，可以执行不同类型的处理。

这个决定要受消息内容和一组事务规则的影响，或者实际的终结点本来就只是一个硬编码的字符串。

所有这些情况，基本思想都是用一种灵活的方式解除发送者与接收者之间的耦合关系。

<<WCF 4高级编程>>

编辑推荐

《WCF 4高级编程》共14章节，内容包括设计原理与设计模式、服务契约与数据契约、绑定、客户端等。

《WCF 4高级编程》专门针对中级.NET开发人员和解决方案架构师。

他们对用WCF4创建面向服务的应用程序、实现可靠的通信、宿主业务流程、在云计算中实现安全及可扩展的集成应用比较感兴趣。

<<WCF 4高级编程>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>