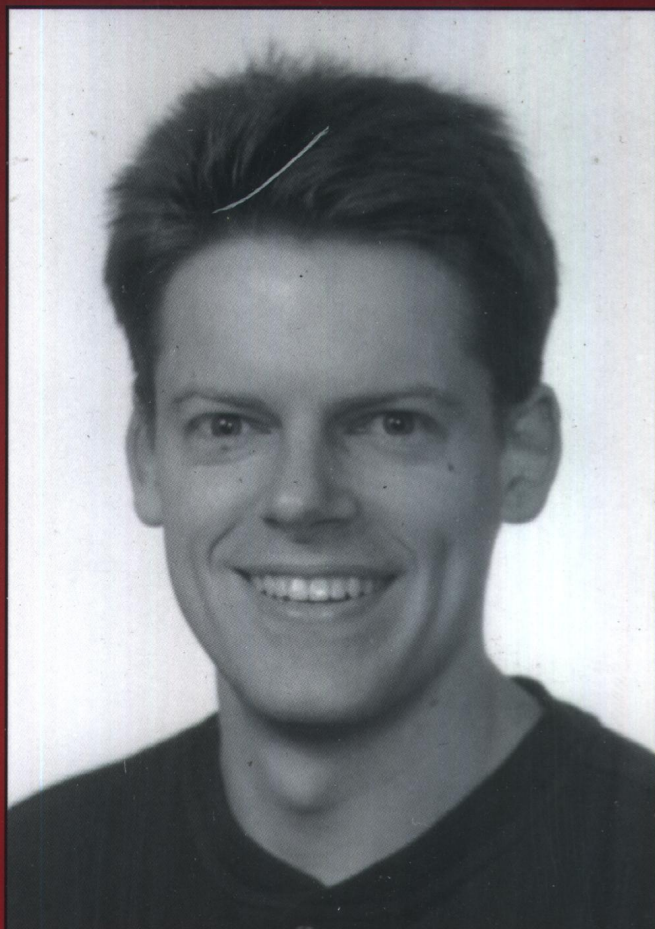
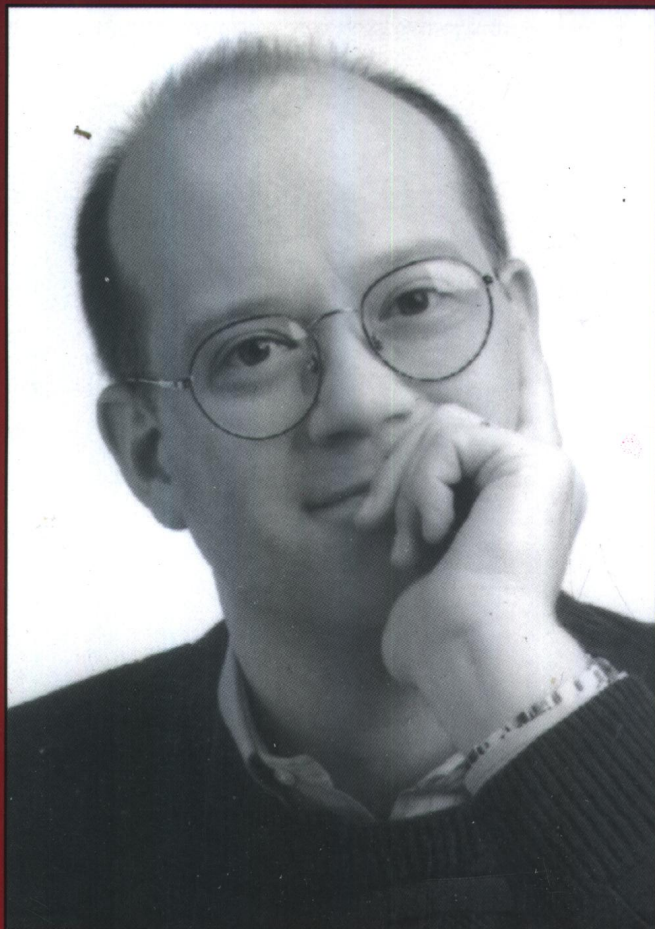


PROGRAMMER TO PROGRAMMER™



PROFESSIONAL BizTalk

BizTalk 高级编程

Stephen Mohr

Scott Woodgate 著 康博译



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



164

(-713.36
1786

BizTalk 高级编程

Stephen Mohr

Scott Woodgate

康 博

著

译

清华大学出版社

(京) 新登字 158 号

北京市版权局著作权合同登记号：01-2001-4314

内 容 简 介

现在致力于企业应用集成(EAI)和电子商务的企业都面临的问题就是,如何将使用不同通信和数据格式的应用程序集成在一起并和业务伙伴实现无缝连接。Microsoft BizTalk Server 2000 使用一台集成服务器和一套应用集成工具使企业不需对代码做大的修改就能实现跨平台的应用集成。本书对 BizTalk Server 做了全面细致的介绍,重点讲述了如何使用 BizTalk Server 处理企业应用集成中的各种问题,尤其是如何快速开发电子商务系统。

读者将从书中学习到如何通过配置服务器而不是编写大量的集成代码来完成大部分的集成工作。全书共分 12 章,分别介绍了如何使用 BizTalk Server 提供的各种工具来完成关键任务,如计划应用程序间的数据流、定义数据结构、转换不同数据格式和追踪系统中的数据流等。

本书适合于进行各种 EAI 工程和电子商务系统开发,并具有在 Windows 平台上开发 Web 应用程序经验的程序员阅读。

Stephen Mohr Scott Woodgate: Professional BizTalk

EISBN: 1-861003-29-3

Copyright© 2001 by Wrox Press Ltd.

Authorized translation from the English language edition published by Wrox Press Ltd.

All rights reserved. For sale in the People's Republic of China only.

Chinese simplified language edition published by Tsinghua University Press.

本书中文简体字版由清华大学出版社和英国乐思出版公司合作出版。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,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: BizTalk 高级编程

作 者: Stephen Mohr Scott Woodgate 著 康博 译

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 侯彧

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 38 **字数:** 972 千字

版 次: 2002 年 5 月第 1 版 2002 年 5 月第 1 次印刷

书 号: ISBN 7-302-05439-8/TP·3206

印 数: 0001~3000

定 价: 68.00 元

出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft, IBM, Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前言

日益复杂的 Web 应用程序使程序员面临着更新的挑战。当更多的应用程序在 Web 上运行时，我们不得不利用那些已有的继承下来的应用程序。例如，对于一个电子商务站点来说，支持它的大部分数据来自继承的内部系统。有效的库存检验和订单执行要求与公司供应链软件进行自动链接。在一个组织内部，人们越来越倾向于以前那些独立运行的应用程序进行链接。业务进程中的手动链接正在被自动链接所取代。这些都是企业应用集成(EAI)的例子。EAI 是指将互相分离的应用程序直接链接起来，使之能够作为一个更大系统的一部分协同工作。应用集成要求进行结构化数据的交换，这样数据就能够以一种有意义的方式在组成一个较大的系统的不同的应用程序之间进行传递。

企业对企业(B2B)集成的发展趋势为 EAI 指明了新的方向。组织希望供应商和客户建立紧密的链接，这样货物就能够迅速有效地运转，减少了保持库存和等待送货的需求。这些业务伙伴希望实现电子交易，将一个业务伙伴的企业资源计划(ERP)捆绑到另一个业务伙伴的 ERP 系统上。这时的 EAI 具有很大的规模，已经超出组织界限和计算平台的范围。虽然网络互用性、安全性和数据交换在 B2B 中是很庞大的工作，但是考虑到成功的 B2B 集成所带来的巨大回报，还是值得去努力的。

程序员的任务就是找到高效、可靠的解决方案来实现信息的传送。尤其在处理继承系统或是关键任务系统的时候更具有挑战性。一个公司的库存管理系统或财务系统可能建立得很好且得到了证实，并且对公司的内部运作绝对重要。因此与部门实用软件和应用程序不同，程序员很少能够改动这些系统，更不能将其卸载。在传统的集成工程中，将应用程序捆绑到一起的集成层是嵌入这些应用程序中的。这意味着若要改变大系统就要对组成它的应用程序的源代码进行修改。由于集成代码分布在许多应用程序中，所以这种集成需要花费很大的精力去维护。

在前言的图 1 中，一种新的应用程序——一个到内部系统的基于 Web 的网关——被连接到两个已有的系统上，即组织的财务应用程序和库存应用程序。库存应用程序是关键任务系统，不能被更改。这两个系统是分别开发的，采用不同的通信手段和数据格式。财务应用程序使用 COM，以固定的定位文件格式进行数据交换。而库存应用程序利用 MSMQ，以逗号分隔变量(CSV)的格式进行数据传输。网关利用 HTTP 以 XML 文件格式进行数据传输。在中间控制各个应用程序间数据流的是一个集成服务器，它是本书重点讲述的另外一个应用程序。

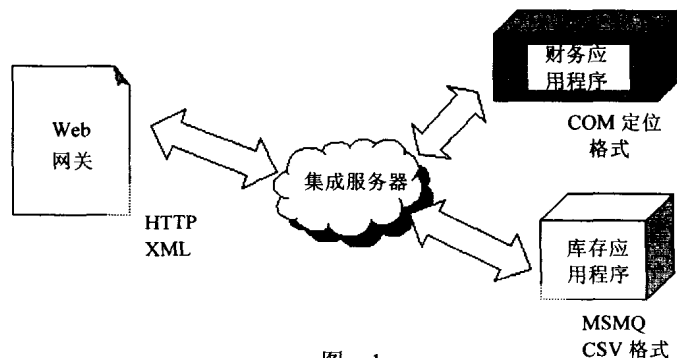


图 1



我们已经提到，遗留的应用程序通常用于关键任务，因此不允许对它的源代码进行改动。这意味着这些应用程序不能用传统的技术进行集成。相反，必须利用这些应用程序在标准操作中使用的编程接口或 I/O 机制对其进行集成。

以这种方式集成也是有优点的。它大大减少了对一个应用程序的代码的集成，从而使该过程变为仅仅是进行配置的过程。在需要编程的时候，它通常是以组件或脚本的形式由集成软件执行。 workflow 集成器对组成系统的应用程序间的数据流进行观察，并且编写一些对应用程序间数据传送进行协调的代码以实现所期望的工作流。当工作流改变的时候，数据和协调模式之一或是两者必须改变。如果这些都在中心集成服务器上，那么改变工作流的任务就会简单得多。在许多情况下，其工作就是对运行在集成服务器上的数据进行重新配置。

在前言的图 2 中，中心流程图表示期望一些分布式系统所做的工作——业务的工作流。流程图中的每一个动作都由一个应用程序来执行。这三个应用程序必须由某个了解它们并且能够协调它们的动作的中心应用程序来协调。理想的协调应用程序是一个由描述所期望的工作流的数据体进行初始化的通用应用程序。另外，协调应用程序必须具有利用这三个应用程序所支持的通信协议和数据格式与这三个应用程序进行通信的机制。

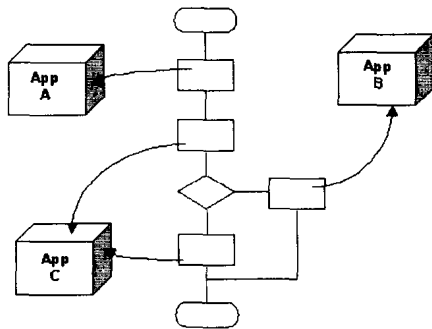


图 2

这就是 Microsoft BizTalk Server 的用武之地。它为我们提供了一个数据驱动的中心集成服务器以及一系列应用集成的工具。虽然它的实现广泛利用了 Windows 2000 的很多专有的特性，但是它能够进行跨平台的应用集成。只利用 Internet 相关的开放标准就可以完全实现集成。程序员们可以做下面这些工作：

- 设计一个系统中的信息流
- 利用 BizTalk 及其接口作为粘接剂实现真正的集成
- 跟踪集成系统中的信息流

BizTalk Server 允许您利用已有的、经过检验的应用程序快速开发 EAI 和 B2B 电子商务业务系统。BizTalk Server 在捆绑内部系统和通过 Internet 链接业务伙伴方面同样具有很大用处。实际上，它可以根据您的意愿随时对重要的应用程序进行连接而不需对代码做大的改动。

实际上，在许多情况下，您不只是在处理自己公司内部的应用程序，可能经常需要与业务伙伴的应用程序进行集成以形成一个真正自动化的系统。比如，拿电子商务的例子来说，您可能需要与供应商的系统进行集成以实现适时订货，这样就能保持较低的库存量。另外您也许需要与订单执行公司进行集成以便将所订购的货物送到客户手中。在所有的这些阶段中，您还会关心运行状态到底如何——也就是要能够对记录进行跟踪。BizTalk 能够为您提供实现这些目的

的所有功能。

本书主要内容

本书介绍了 BizTalk Server 以及如何处理 EAI 中的问题，尤其是电子商务系统(包括 B2B 和 B2C)的快速开发问题。它能够帮助您学会如何利用 BizTalk Server，将不同的应用程序进行链接以形成一个较大的系统。

本书将学习 BizTalk 提供的各种工具以帮助您完成下列任务：

- 为应用程序间的数据流作计划
- 定义将在应用程序间流动的数据的结构
- 对不同的消息格式进行转换，以使数据能够在使用不同数据格式的应用程序间进行传输
- 对系统中的消息进行跟踪

本书还将说明如何利用 BizTalk 建立使用您和业务伙伴的已有的应用程序的集成解决方案。

本书读者对象

本书适合于开发各种 EAI 工程，特别是各种电子商务系统的程序员。它将告诉您如何利用 BizTalk Server 来帮助完成这些任务。如果您具有一些在 Windows 平台上开发 Web 应用程序的经验，将会有助于本书的学习(包括利用 MSMQ 进行消息传递，虽然这一点并不强求)。

如果需要进行应用集成，不管您是想使用 BizTalk Server，还是想与使用 BizTalk Server 的其他人进行集成，本书都将告诉您如何使用这个令人激动的新产品。

本书所使用的技术

在介绍对不同类型的系统进行集成的时候，本书采用了不同的技术。其中包括 XML、运行 IIS 和 ASP 的 Windows 2000、SQL Server、MSMQ 和 Visual Basic。本书还列出了与各种定界和定位的平面文件格式进行集成的例子(您不必具有使用这些格式的经验)。

本书中的许多应用程序都是用 VB 开发的，虽然不要求您必须了解 VB 才能学习本书，但如果想要运行所有的示例，需要利用 VB 6 以上的版本。另外本书还广泛使用了 XML，熟悉它对学习本书有帮助，但这并不是先决条件。

用户支持

我们力图使本书尽可能地精确而且有趣，但最重要的是这本书对您有什么帮助。希望能够让编者了解您的意见，可以使用电子邮件与我们联系，信箱地址是 feedback@wrox.com。

源代码与更新

当您在练习本书示例的时候，可能想要自己亲手键入所有的代码。许多读者都宁可这样做，因为这是熟悉所采用的编码技术的好方法。



不论您是否想要自己键入代码，在我们的站点上都可以得到本书的所有源代码，网址是：
<http://www.wrox.com/>。

对于那些试图自己键入代码的读者来说，可以利用我们的文件来检查应该得到的结果——如果您认为自己可能键入了错误代码，那么您应该首先来浏览这些文件。如果您不想自己键入代码，那么从我们的站点上下载代码则是必做之事。

不管是哪一种方法，都有助于更新和调试。

勘 误 表

我们已经尽了最大努力确保在文本和代码中没有错误。但是，人总是要犯错的，因此我们认识到需要通知您在本书中所出现的错误及校正。我们所有的勘误表都可以在 <http://www.wrox.com> 上找到。如果您发现了没有列出的错误，请通知我们。

我们的 Web 站点是其他信息和支持的中心，包括书中所有的代码、示例章节、出版预告以及相关主题的文章与看法。

目 录

第 1 章	BizTalk 与应用集成	1
1.1	企业应用集成的益处	1
1.2	EAI 的挑战	2
1.2.1	简单的客户机-服务器模型	2
1.2.2	分布式业务系统	3
1.2.3	综述 EAI 所带来的挑战	4
1.3	EAI 和 B2B 的场景和体系结构	5
1.3.1	业务进程集成	5
1.3.2	业务伙伴集成	6
1.3.3	自动化的采购	7
1.3.4	企业对企业(B2B)的集合体	8
1.4	首创 BizTalk Framework	9
1.4.1	BizTalk Framework	10
1.4.2	BizTalk.org	11
1.4.3	BizTalk Server	14
1.5	Microsoft BizTalk Server 2000 介绍	14
1.5.1	BizTalk Server 的特性	14
1.5.2	对 BizTalk Server 工具的总体认识	15
1.6	BizTalk 消息传递服务	15
1.6.1	组织	15
1.6.2	文档定义	16
1.6.3	封包	16
1.6.4	端口和分配表	16
1.6.5	信道	17
1.6.6	事务	17
1.6.7	消息传输	18
1.6.8	消息排队	21
1.7	BizTalk Server 工具介绍	22



1.7.1	定义业务工作流: Orchestration Designer	22
1.7.2	定义消息规范: BizTalk Editor	24
1.7.3	消息规范间的转换: BizTalk Mapper	27
1.7.4	消息传递服务的配置: BizTalk Messaging Manager	28
1.7.5	查看运行时消息: 文档跟踪	29
1.7.6	BizTalk Server 的管理: Server Administration	32
1.8	小结	33
第 2 章	应用程序和调度	35
2.1	编排的必要性	36
2.2	编排的实现	37
2.2.1	业务进程图	38
2.2.2	XLANG 调度	39
2.3	BizTalk Orchestration Designer	39
2.3.1	Orchestration Designer 用户界面	40
2.3.2	调度的设计	43
2.3.3	消息实现的绑定	55
2.3.4	动作与 BizTalk 或 MSMQ 实现的连接	65
2.3.5	动作与 COM 和脚本实现的连接	68
2.4	Wrox Site Manager 应用程序	69
2.5	小结	79
第 3 章	规范	81
3.1	什么是规范	82
3.2	消息格式	83
3.2.1	XML 文件	83
3.2.2	定界平面文件	85
3.2.3	定位平面文件	86
3.3	将模式扩展为规范	87
3.3.1	基于 XML 的消息的规范	87
3.3.2	用于平面文件格式的消息规范	88
3.4	设计规范	90
3.4.1	改编遗留格式	90
3.4.2	构建分层形式	90
3.4.3	BizTalk Framework 的规范形式	93
3.5	BizTalk Editor	97

3.5.1	用户界面	97
3.5.2	要点概括	97
3.5.3	使用规范	98
3.5.4	用于 Wrox Site Managers 范例的规范	108
3.5.5	Site Service Description: 一个规范示例	108
3.5.6	定位和定界规范	112
3.5.7	高级特性	116
3.5.8	CableInstallation: 定界文件规范	118
3.5.9	Yard Work: 定位文件规范	119
3.5.10	EDI 规范	122
3.6	小结	123
第 4 章	映射 BizTalk 消息	124
4.1	BizTalk 需要映射的原因	124
4.2	运行时消息的转换	126
4.2.1	映射过程	126
4.2.2	XSLT	128
4.2.3	分析器和串行化器	133
4.3	BizTalk Mapper	135
4.3.1	用户界面	135
4.3.2	构造映射	137
4.3.3	functoid	139
4.3.4	编译和测试映射	148
4.4	对 Wrox Site Managers 进行映射	150
4.4.1	Site Service Description 向 Cable Installation 的映射	150
4.4.2	Site Description 向 Yard Care 的映射	153
4.5	开发自己的 functoid	154
4.5.1	functoid 的工作原理	154
4.5.2	IFunctoid 接口	156
4.5.3	将有线业务规则实现为 functoid	160
4.6	小结	167
第 5 章	BizTalk 消息传递的管理	168
5.1	配置而不是编程	168
5.2	BizTalk 消息传递和传输协议	169
5.2.1	与 BizTalk 直接通信	169



5.2.2	一般协议通信	170
5.3	BizTalk 的管理概念	171
5.3.1	组织	172
5.3.2	文档定义	173
5.3.3	封包	173
5.3.4	消息传递端口	174
5.3.5	分配表	174
5.3.6	信道	175
5.4	BizTalk Messaging Manager	176
5.4.1	用户界面	176
5.4.2	配置服务器选项	176
5.4.3	进行搜索	177
5.4.4	配置组织	178
5.4.5	配置文档定义	181
5.4.6	封包配置	184
5.4.7	消息传递端口向导	185
5.4.8	分配表	190
5.4.9	信道向导	191
5.5	简单的 BizTalk 编程	197
5.5.1	IIInterchange 接口	198
5.5.2	信道测试器实用程序	201
5.6	配置 Wrox Site Management Application	208
5.6.1	组织	208
5.6.2	文档和封包	209
5.6.3	消息传递端口	209
5.6.4	信道	211
5.6.5	测试信道	212
5.7	Wrox Site Management Client	213
5.7.1	用户界面	213
5.7.2	生成 XML	214
5.7.3	工程相关性	214
5.7.4	窗体处理	215
5.7.5	提交消息	216
5.7.6	监视调度执行	218
5.8	小结	219

第 6 章 高级编排	221
6.1 体系结构：调度与应用程序的比较	221
6.2 载入调度	222
6.2.1 消息抵达时载入调度	222
6.2.2 别名语法	222
6.2.3 以编程方式载入	225
6.3 调度的水合	235
6.3.1 工作原理	236
6.3.2 消息延迟和水合	238
6.3.3 为长事务配置数据库	238
6.4 事务和错误恢复	240
6.4.1 参加 COM+ 事务	240
6.4.2 短期事务	241
6.4.3 长期事务	247
6.4.4 嵌套事务	248
6.4.5 计时事务	252
6.5 并发	252
6.5.1 并发和事务边界	253
6.5.2 伸缩性问题	253
6.6 动态端口绑定	254
6.7 小结	259
第 7 章 消息传递与应用程序的连接	261
7.1 与 BizTalk Server 的应用集成	262
7.1.1 消息到应用的集成	262
7.1.2 业务进程到应用的集成	263
7.1.3 案例研究：Bob's Bolts——将消息直接传递到 Microsoft Access	263
7.2 使用 IBTSAppIntegration AIC	267
7.2.1 创建 IBTSAppIntegration AIC	267
7.2.2 部署 IBTSAppIntegration AIC	274
7.3 使用 IPipelineComponent AIC	280
7.3.1 建立 IPipelineComponent AIC	283
7.3.2 部署 IPipelineComponent AIC	289
7.3.3 配置 IPipelineComponent AIC	290
7.4 使用 BizTalk Scriptor AIC	292



7.5	使用 AIC 的注意事项	297
7.6	小结	297
第 8 章	消息传递与调度的连接	299
8.1	从消息传递中激活一个新的编排	300
8.1.1	为例子创建一个编排调度	301
8.1.2	为例子配置 BizTalk Messaging	304
8.1.3	运行实例	307
8.1.4	工作原理	307
8.2	与现有的编排相关	309
8.3	在两台 BizTalk 服务器之间基于 HTTP 的相关	311
8.3.1	提取内部的 Instruction 消息, 并传送到新的 Instruction 业务进程上	313
8.3.2	在 BizTalk Orchestration 中接收内部消息 Instruction, 发送 Instruction 并接收 Acknowledgement	314
8.3.3	通过 HTTP 使用 BizTalk Messaging 把 Instruction 发送给新的 BizTalk Orchestration	318
8.3.4	在 BizTalk Orchestration 中接收 Instruction, 发送 Acknowledgement	320
8.3.5	把 Acknowledgement 消息返回给 Instruction 调度最初的实例 (Acknowledgement 端)	322
8.3.6	把 Acknowledgement 消息返回给 Instruction 调度最初的实例 (Instruction 端)	323
8.3.7	运行例子	325
8.4	两个 BizTalk Server 之间的基于非 HTTP 的相关	327
8.4.1	使用 BizTalk Messaging 提取内部的 Instruction, 并传输到 Instruction 业务进程	329
8.4.2	把内部的 Instruction 接收到 BizTalk Orchestration 中, 发送内部的 Instruction, 接收 Acknowledgement	330
8.4.3	通过非 HTTP 使用 BizTalk Messaging 把 Instruction 发送给新的 BizTalk Orchestration	332
8.4.4	在 BizTalk Orchestration 中接收 Instruction 消息, 发送 Acknowledgement 消息	335
8.4.5	把 acknowledgement 消息返回给 Instruction 调度的原实例 (acknowledgement 端)	337
8.4.6	把 Acknowledgement 消息返回给 Instruction 调度的原实例 (Instruction 端)	338
8.4.7	运行例子	339
8.5	在 BizTalk Server 和外部进程之间基于 HTTP 的 XLANG 相关	341

8.6 相关的其他方法	345
8.7 小结	345
第 9 章 维护	347
9.1 维护和管理	347
9.2 BizTalk 组、服务器和共享数据	348
9.2.1 BizTalk Server	349
9.2.2 BizTalk Server 组	349
9.2.3 BizTalk 数据库	351
9.2.4 队列	352
9.3 BizTalk Server Administration 用户界面	353
9.3.1 显示共享队列的特性	354
9.3.2 显示接收函数的特性	355
9.3.3 显示服务器的特性	355
9.4 接收函数	356
9.4.1 配置接收函数	357
9.4.2 使用接收函数	361
9.5 一般维护任务	366
9.5.1 权限和远程维护	367
9.5.2 设置企业特性	367
9.5.3 设置组的特性	368
9.5.4 添加一个服务器组	370
9.5.5 为 BizTalk 服务器组添加一个服务器	371
9.5.6 删除和移动服务器	371
9.5.7 启动和停止服务器	371
9.5.8 释放交换消息	372
9.5.9 设置服务器的特性	372
9.5.10 在队列间移动文档	373
9.5.11 导出维护信息	374
9.6 通过维护界面跟踪错误	374
9.6.1 检查组和服务器状态	374
9.6.2 查看 Suspended 文档的错误信息	375
9.6.3 事件日志中可用的信息	376
9.7 BizTalk 的安全性	376
9.7.1 用户角色	377



9.7.2 保护编排	379
9.8 伸缩性的设计	380
9.8.1 部署数据库	380
9.8.2 部署服务和组件	382
9.9 小结	384
第 10 章 路由和确认回执	386
10.1 基本路由要求	387
10.2 基于调用的路由	388
10.3 信道规范	390
10.4 自我路由	390
10.4.1 自我路由的文档规范	391
10.4.2 自我路由的一个例子	392
10.4.3 使用开放的端口	394
10.5 非自我路由的接收函数	395
10.6 路由和 BizTalk Framework	396
10.6.1 BizTalk Framework 2.0	396
10.6.2 Framework 中的路由	401
10.6.3 回执	406
10.7 确认回执	407
10.8 配置 BizTalk Server 处理回执	408
10.8.1 使用可靠的消息传递来处理回执	408
10.8.2 处理跟踪和回执	416
10.9 定制组件	418
10.9.1 建立定制分析器和串行化器的原因	418
10.9.2 定制分析器	418
10.9.3 IBizTalkParserComponent 接口	419
10.9.4 定制串行化器	419
10.9.5 IBizTalkSerializerComponent 接口	420
10.9.6 定制相关器	420
10.9.7 IBizTalkCorrelation 接口	420
10.9.8 IBizTalkAcknowledge 接口	421
10.10 一个 BizTalk Framework 路由和回执示例	424
10.10.1 场景示例	425

10.10.2	客户应用程序的用户界面: BTFCClient.exe	426
10.10.3	窗体加载事件	427
10.10.4	设置端点	428
10.10.5	设置特性	429
10.10.6	置换消息的主体	434
10.10.7	通过 HTTP 传送文档	435
10.10.8	HTTP 接收函数	436
10.10.9	查看回执	438
10.10.10	测试样例和 BizTalk Framework 路由	439
10.11	小结	442
第 11 章	文档跟踪	444
11.1	捕获跟踪信息	444
11.2	显示跟踪信息	455
11.2.1	查看交换的元数据	456
11.2.2	查看文档专用的数据	457
11.2.3	跟踪界面解密	459
11.3	编程访问跟踪信息	460
11.4	在数据库的级别上理解文档跟踪	471
11.5	预备处理器和文档跟踪	475
11.6	管理跟踪数据库	475
11.7	配置 BizTalk Orchestration 跟踪	476
11.8	小结	476
第 12 章	配置编程	477
12.1	配置编程的使用	477
12.1.1	配置导出	478
12.1.2	动态配置	478
12.2	BizTalk 配置对象模型	479
12.2.1	了解配置对象模型	480
12.2.2	处理配置对象时维护引用的完整性	483
12.2.3	IBizTalkBase 接口	484
12.2.4	IBizTalkConfig 接口	485
12.2.5	IBizTalkOrganization 接口	488
12.2.6	IBizTalkDocument 接口	490
12.2.7	IBizTalkPort 接口	491