

微软.NET程序员系列

- ◆ 欧美读者评价 ★★★★★
- ◆ 微软分布式系统专家精心编写
- ◆ 利用.NET技术迅速进入分布式开发领域
- ◆ 介绍如何集成各种.NET技术
- ◆ 使开发人员提高到一个新水平

Microsoft®

.NET分布式应用程序:

集成XML Web服务与.NET远程处理

MICROSOFT .NET Distributed Applications:
Integrating XML Web Services And .NET Remoting

(美) *Matthew MacDonald* 著

戢中东 周长青 张 晔 常小红 译
Visual Studio .NET产品组 审校



Microsoft
.net

清华大学出版社

微软.NET 程序员系列

.NET 分布式应用程序： 集成 XML Web 服务与.NET 远程处理

(美) Matthew MacDonald 著

戢中东 周长青 张 晔 常小红 译

Visual Studio .NET 产品组 审校

清华大学出版社

北 京

内 容 简 介

本书是《微软.NET 程序员系列》丛书之一, 主要介绍了如何使用.NET 进行分布式编程。书中详细讲解了需要掌握的关键.NET 技术并介绍了将.NET 应用于分布式应用程序体系结构的最佳实践经验。本书的特色在于向您展示了如何将独立的.NET 技术完全整合在一起。本书讲述使分布式应用程序具有高性能、安全性和可缩放性的体系结构和优秀实践经验, 也十分关注应用程序体系结构方面的内容。

本书读者包括已经掌握微软.NET 基础知识并想涉足分布式编程领域的开发人员, 还包括基本掌握各种.NET 技术但是想在实际应用程序中合理利用它们的程序员。

.NET 分布式应用程序: 集成 XML Web 服务与.NET 远程处理

Microsoft .NET Distributed Applications: Integrating XML Web Services and .NET Remoting

Matthew MacDonald

Copyright © 2003 by Microsoft Corporation.

Original English Language Edition Copyright © 2003 by Microsoft Corporation.

Published by arrangement with the original publisher, Microsoft Press,
a division of Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文版由 Microsoft Press 授权清华大学出版社出版。

北京市版权局著作权合同登记号 图字 01-2003-3099 号

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

.NET 分布式应用程序: 集成 XML Web 服务与.NET 远程处理/(美)麦克多纳德(MacDonald, M.)著; 戢中东, 周长青, 张晔, 常小红译. —北京: 清华大学出版社, 2005.3

(微软.NET 程序员系列)

书名原文: Microsoft .NET Distributed Applications: Integrating XML Web Services and .NET Remoting
ISBN 7-302-10291-0

I. N… II. ①麦…②戢…③周…④张…⑤常… III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字(2005)第 000677 号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

文稿编辑: 赵飞虎 张 莉

封面设计: 陈刘源

印 刷 者: 北京牛山世兴印刷厂

装 订 者: 北京市密云县京文制本装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×230 印张: 37.75 插页: 5 字数: 900 千字

版 次: 2005 年 3 月第 1 版 2005 年 3 月第 1 次印刷

书 号: ISBN 7-302-10291-0/TP·7010

印 数: 1~3500

定 价: 66.00 元

Microsoft[®]
.net[™]

通往.NET的成功之路由此开始……

——Visual Studio .NET为未来的十年作好了准备。
——您呢？

从下面一种或几种环境开始您的体验

使用您最擅长的语言学习Visual Studio .NET和.NET框架

快速应用程序开发
Microsoft Visual Basic®
Microsoft Office,
Borland PowerBuilder
和Borland Delphi

Web应用程序开发
ASP, ECMAScript,
Java™, Server Pages, Perl
Server Pages, IBM
Web sphere®, Macromedia
ColdFusion®和Microsoft
Visual InterDev

企业应用程序开发
Microsoft Foundation
Classes, Enterprise Java
Beans™, UNIX C/C++平台,
分布式数据库, 客户
思想和控制策略

系统开发
Microsoft Windows DNA,
服务和网络程序, 应用程序/
嵌入式, UNIX, 后台服务
和实时系统等

其他语言和脚本开发
APL, Lisp, JScript®,
COBOL, Perl, Scheme/Tcl,
Erlang及其他第三语言

了解.NET

Microsoft Visual Basic .NET
流行的Visual Basic编程语言的更高级阶段, 它保留了Visual Basic面向对象的特点和运行方式, 并且能够使用.NET平台的全部功能

Microsoft Visual C# .NET
一种结合了Visual Basic易用性和C++的强类型与灵活的编程语言, 其拥有许多功能使开发更快捷、更简单, 尤其在应用COM+和Web服务的情况下

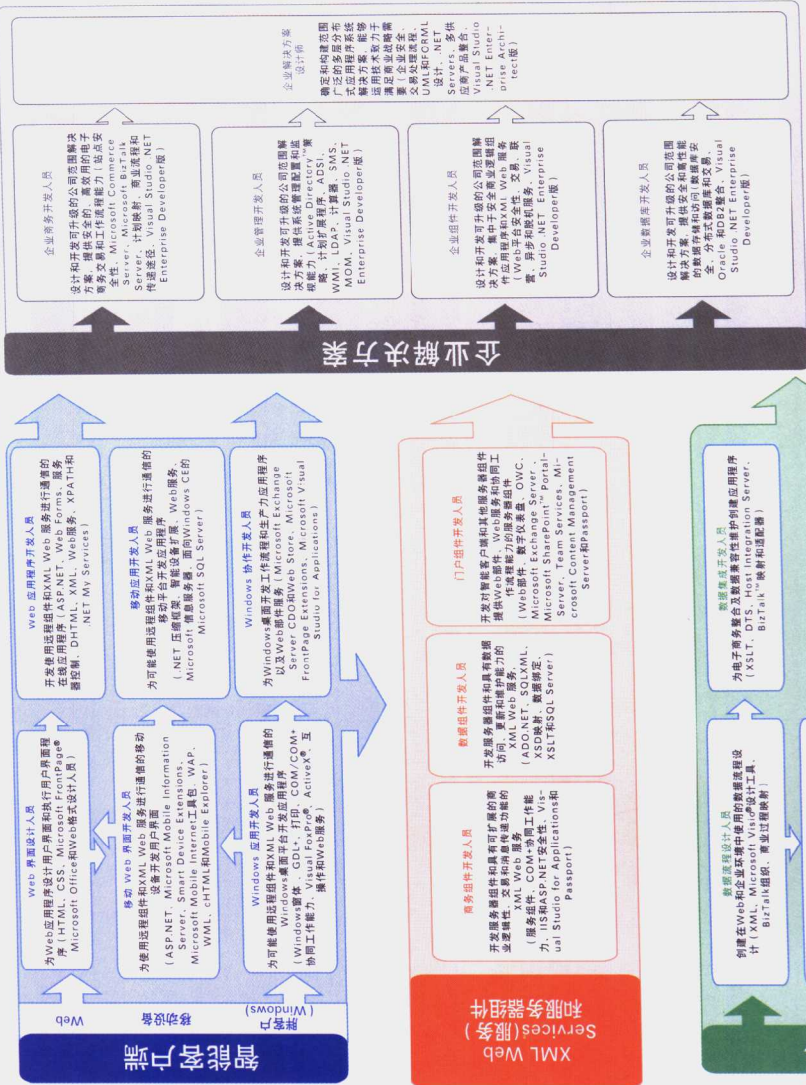
Microsoft Visual C++ .NET
促进Windows开发, 并能开发人感到重新类型的、强有力的.NET应用程序或通过C++语言, 利用.NET现有或应用程序

其他
支持.NET
框架的语言
如COBOL和Perl

技术类:

MSF: Microsoft Solutions Framework
XSLT: eXtensible Style Sheet Language Transformations
DTS: Data Transformation Services
OLAP: Online Analytical Processing
MDX: Multidimensional Expressions
WAP: Wireless Application Protocol
WML: Wireless Markup Language
OWC: Microsoft Office Web Components

然后, 根据您的职业意向和技能发展需要, 确定工作角色和相关的.NET技术



Web, Windows, and Mobile

Business Development

Enterprise Development

System Development

将技术应用于工作的必备工具

清华大学出版社

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

《微软.NET 程序员系列》序

自 2000 年 6 月微软宣布自己的 .NET 战略以来,在不到两年的时间里,.NET 已经从战略变成现实。.NET 带来了全新的、快速而敏捷的企业计算能力,也给软件开发者和软件开发人员提供了支持未来计算的高效 Web 服务开发工具。作为微软.NET 战略的重要组成部分——Visual Studio .NET (中文版)已经于 2002 年 3 月 22 日正式在中国推出。

Visual Studio .NET 是一个功能强大、高效并且可扩展的编程环境。它充分展现了应用程序开发的潜能,并提供了生成应用程序所需的工具和技术。这些应用程序将给当今的企业、机构提供强大的支持,并推动下一代基于 XML Web 服务软件的发展。

有了 Visual Studio .NET,那些对全世界数百万的专业和业余程序员来说曾一度极端复杂、费时费力,甚至让人望而生畏的编程任务现在已不再神秘。更重要的是,Visual Studio .NET 使开发人员能运用既有的技能和知识来迎接新的编程挑战。

在 10 年前,Visual Basic 1.0 成为数以百万计的开发人员的革命性的应用程序开发语言。现在,Visual Studio .NET 为未来的 10 年做好了准备。

微软出版社为了配合 Visual Studio .NET 的推广以及 .NET 技术的普及,邀请 Visual Studio .NET 项目开发组的核心开发人员和计算机图书专业作家精心编写了英文版《微软.NET 程序员系列》丛书;该丛书自面市以来,在美国图书销量排行榜上一直高居前列,颇受好评,成为程序开发人员和网络开发人员了解 .NET 技术的权威工具书。尤其是《Microsoft .NET Framework 程序设计》一书,长期占据美国及欧洲此类书籍的排行榜冠军位置,程序开发人员不可不读此书。

清华大学出版社为了满足中国广大程序开发人员、网络开发人员学习最新技术的渴望,在微软出版社的配合下,从《微软.NET 程序员系列》这套丛书中精选了 50 余本翻译成中文,以满足国内广大读者的需要。这套丛书阵容庞大(且在不断扩充之中),几乎涵盖了 .NET 技术及其应用的各个方面;也正因为如此,翻译和编辑加工的工作量也大得惊人。但为了保持国外优秀技术图书的魅力,同时使读者领会新技术的真谛,本丛书的翻译和编辑都是经过严格筛选的、具有很高的翻译水平或丰富编辑经验的技术人员;另外,我们还聘请微软公司 Visual Studio .NET 产品组的技术专家审读每一本书,确保在技术上准确无误。

相信这套丛书定会帮助程序开发人员、网络开发人员以及那些具有一定编程基础的中、高级读者,快速、全面地掌握 .NET 技术,协助他们为技术生涯的下一个 10 年做好准备,为培养新一代软件人才,并推动中国软件产业的快速发展起到积极的作用!

这套丛书分为 3 个子系列:技术内幕系列、语言参考系列和程序员系列。目前,已出版和在编

的共有 36 本，已从 2002 年 6 月份起陆续和读者见面。

- **技术内幕系列**

目前共有 7 本：

- ◆ 《Visual C++ .NET 技术内幕(第 6 版)》

本书是 Visual C++ 和 MFC 开发的经典著作。它秉承了第 4 版和第 5 版的风格，已根据该编程语言的最新版本 Visual C++ .NET 进行了全面更新和补充，是 .NET 时代的 C++ 程序员必读的教材。此外，本版仍由第 4 版译者潘爱民先生翻译。

- ◆ 《Microsoft .NET Compact Framework 技术内幕》

- ◆ 《Visual Basic .NET 技术内幕》

- ◆ 《Visual C# .NET 技术内幕》

- ◆ 《ADO.NET 技术内幕》

- ◆ 《Microsoft .NET 程序设计技术内幕》

- ◆ 《Visual J# .NET 技术内幕》

- **语言参考系列**

目前共有 3 本：

- ◆ 《Visual Basic .NET 语言参考手册》

- ◆ 《Visual C# .NET 语言参考手册》

- ◆ 《Visual C++ .NET 语言参考手册》

- **程序员系列**

目前共有 27 本。详细书目，请参见本书彩插页。

其中，Microsoft .NET Framework 1.1 类库(1-4 卷)是 .NET System 命名空间大全，直接源自 Microsoft .NET Framework。这 4 卷书包含了 .NET Framework Class Library System 命名空间的全部详细信息。帮助开发人员充分利用 .NET Framework 来开发 Web 应用程序、客户端应用程序和 XML Web 服务。

随着技术的发展，我们将根据读者的需要，不断增加新的书目。

丛书版式特色

本丛书在风格上力求文字精炼。并采用小 5 号字编排，内容紧凑，版面清晰美观，易于阅读。此外，书中还安排了一些特色段落，提供正文之外的一些细节知识：



注意：提醒阅读和操作过程中应注意的事项，避免出现错误或问题。



提示：指点一些操作捷径或实用技巧，使您少走弯路，阅读和操作更为高效。



要点：总结关键知识点或操作细节，帮助您适时掌握要领。



注：提示首次出现的编程元素，以及书中涉及到该元素的其他位置以供参考。



警告：阅读和操作过程中，应特别注意的事项，如果使用不当，可能会导致严重后果。

尽管我们倾心相注，精心而为，总有疏忽纰漏，恳请广大读者不吝赐教与指正，我们定会全力改进，以期在后续工作中得以完善。

本丛书在创作过程中得到了微软(中国)公司的大力支持。本丛书能够顺利出版，更是倾注了无数幕后人员的汗水和心力。在此，对他们的辛勤劳动一并表示衷心感谢！

编者

2003 年 3 月

前 言

从业界第一次认识分布式应用程序体系结构到现在已经 7 年了。从那以后,令人兴奋的新技术,例如 COM/DCOM 与 CORBA/IIOP 宣称要革新生成大规模资源密集型应用程序的方法。分布式体系结构将软件模块化成一组跨越不同机器进行通信的对象,而不是尽量在一台计算机中装载一个应用程序。在理想情况下,这些机器不再需要私人拥有大型机——取而代之的是开发人员能够使用廉价的运行微软 Windows 操作系统的服务器。要增加系统总吞吐率通常只要将别的计算机简单地加入到这个系统中就行了。

所有这些使分布式编程成为软件编程中最令人兴奋和热切关注的领域,但是它也制造了一些非常不利的障碍。原因很简单,分布式应用程序比较复杂。在 Windows 平台上编写分布式应用程序需要牢牢掌握微软的 COM 标准及其企业级软件和组件服务(例如 SQL Server 和 COM+),还需要一些通过努力才能获得的经验。不论程序员的技术多么高超,分布式编程项目可能很快会变成由可怕的版本化、让人头痛的互操作和意外性能瓶颈组成的大杂烩。

这些问题就是创建微软.NET 平台的关键因素。微软.NET 为创建组件、计算机间通信和访问数据提供了全新的模式——这种模式在各个层面上对分布式应用程序来说都是比较理想的。编程新手仍需要投入足够的开发时间并经历曲折艰辛的学习才能掌握这个框架。然而掌握基本知识之后,.NET 会明显使创建真正的可缩放软件系统变得更容易。

本书使用.NET 探讨分布式编程。它详细讲解了您需要掌握的关键.NET 技术并介绍了将.NET 应用于分布式应用程序体系结构的最佳实践经验。本书的最优秀之处在于向您展示了如何将各项独立的.NET 技术完全整合在一起。

读 者 对 象

本书面向已经掌握微软.NET 基础知识并想涉足分布式编程领域的开发人员。这些读者包括已经听说过无连接数据、基于 XML 的标准和 XML Web 服务但是没有亲身尝试过的开发人员。这些读者还包括基本掌握各种.NET 技术但是想在实际应用程序中合理利用它们的程序员。

.NET 基础,例如事件、元数据和公共语言运行库(CLR)不在本书讨论范围之内(对大多数专业开发人员来说这些内容是让人非常厌烦的重复内容)。如果您想设计分布式.NET 应用程序但却对.NET 感觉陌生,您最好先从了解 C#或 Visual Basic .NET 开始。已经出版的许多好

书中包含这些内容(其中包括我编著的, *The Book of VB .NET*, No Starch 出版社)。

主要内容

本书审视了使分布式应用程序具有高性能、安全性和可缩放性的体系结构和优秀实践经验。出于这种想法, 不仅如何使用 .NET 技术的讲解变得很关键, 而且何时使用某种技术更为恰当以及如何将这些技术在现实世界的解决方案中进行整合的讲解也很关键。

您可能已经见过关于 .NET 功能的市场宣传, 例如无连接数据和 XML Web 服务。尽管这些技术令人瞩目, 但它们有各自的优点和缺点。要想明智地使用它们就得了解何时使用它们最合适——还得了解它们何时只会带来令人头痛的麻烦。

本书也十分关注应用程序体系结构方面的内容。它会探讨从两层体系结构向 n 层体系结构转移时获得的实际利益或损失, 并介绍如何使您的设计适应手头的问题或项目。分布式应用程序设计有时需要进行权衡折衷, 例如平衡可缩放性和复杂性以及平衡性能和标准。本书包含了确定最佳比例的方法策略。

本书结构

微软 .NET 是一个内容丰富的框架, 其中包含数以百计的令人感兴趣的鲜为人知的内容。对计算机图书作者来说这具有独特的危险: 讲述这些鲜为人知的功能太容易分散注意力, 并且容易使整本书不能投入到真正的核心内容中。本书为了避免这种危险而设计成具有逻辑关系的下列 3 个部分。

第 I 部分：关键技术

本书的第一部分从分布式编程的角度介绍各个 .NET 技术。为了使本书始终以主题为中心, 每章的开头小节都讲解了此项功能在分布式系统中扮演的角色。这样做的副作用是 .NET 中很多令人感兴趣的内容不会出现在本书中——其中包括从自定义控件到文件输入/输出的所有内容。然而这样做使本书为更深入地介绍企业级应用程序主题留下了空间。下面是一些要介绍的主题内容:

- 理解组件和组件宿主
- 使用 XML 和 XML Web 服务桥接与其他平台之间的缝隙
- 使用 .NET Remoting 在分布式组件之间通信

- 使用无连接数据开发数据传送计划并处理并发问题
- 提高对多线程客户端的响应并处理竞争和同步问题
- 创建线程池和 Windows 服务
- 将 Message Queuing 用于异步的解耦通信
- 将 COM+ 组件服务中的事务、实时激活、对象池等功能用于自己的组件

本书第一部分中的章节中代码示例很少，将重点放在了概念上而不是放在像玩具一样简单的应用程序上，这些应用程序只会淡化概念。然而，在合适的地方，即在本书的实例学习部分(第三部分)您会完全投入到示例中。

第II部分：体系结构开发

本书的第二部分介绍应用程序体系结构：如何建立分布式应用程序的各个部分，以便它们能够可靠地运行并且能改变规模来支持数量巨大的客户端。这里包括几个主题：

- 面向对象设计的性能限制
- 无状态编程的基础
- 传送数据、报告错误和监视应用程序性能的最好方式
- 如何按理想的性能设计数据库——以及如何为它编码而不破坏封装
- 从传输层到应用程序代码如何进行加密并使代码安全
- 如何通过使用自动更新应用程序和 XML Web 服务管理分布式应用程序部署

第III部分：案例研究

本书的最后部分介绍了几个完整的学习实例，这些实例说明了如何在分布式应用程序中综合运用本书前两部分介绍的.NET 技术。这 3 个实例是：

- **Invoicer .NET** Invoicer .NET 协调了领域中的雇员销售。其中令人感兴趣的部分是这个实例显示了从“传统”方法——使用 Excel 和 Access——到.NET 解决方案的分阶段移值途径，而不需要完全重写。这个学习实例也考虑到了处理“无连接”系统的问题。
- **Transact .NET** Transact .NET 是订单履行系统，该系统显示了为提供实时通知而集成 XML Web 服务与消息队列的关键技术。XML Web 服务提供了客户端订单接口，而消息队列是传递通知消息的中枢。
- **SuperCompute .NET** SuperCompute .NET 是工作需求应用程序，它异步执行所需求的任务并且随后让客户端采用。它演示了分布式应用程序体系结构是如何让客户端利用服务器的 CPU 能力的。

当然，学习实例不是完整实用的专业应用程序。然而，就普通问题来说，每个学习实例的代码已经足够用来说明什么是优秀的解决方案框架设计了。通过添加另外的安全性、性能优化或附加功能(特别是在客户端应用程序中，这里缺乏多数用户期望的如同 Windows GUI 一样的易用性)来增强这些框架。实际上，实例学习部分会说明如何选择并集成不同的 .NET 技术并且使您能够以最佳的 .NET 解决方案体系结构实践作为起点。

最后一章(第 19 章)提供了几个有价值的 URL，您可以从这些 URL 中下载别的分布式应用程序学习实例。这些学习实例由不同的微软小组设计，其中包含了有关 ASP.NET、ADO.NET 和 XML Web 服务的最佳实践。其中某些实例在 MSDN 上甚至有说明其设计过程的白皮书。

系 统 需 求

本书的示例需要使用下列软件：

- Visual Studio .NET 或 Visual Studio .NET 2003
- Windows 2000、Windows XP 或 Windows Server 2003

开发所需的最低硬件配置是 450MHz Pentium II 系列处理器，如果运行 Windows 2000 则最少需要 128MB 的内存；如果运行 Windows XP、Windows 2000 Server 或 Windows Server 2003 则需要 256MB 的内存。还需要大约 5GB 的剩余硬盘空间来安装 Visual Studio .NET 2003。这些数值是刚好能满足需要的最小值，如果系统有更多的内存和剩余磁盘空间，开发工作会更容易。

代 码 示 例

本书中出现的多数代码以单独片断的形式被介绍和分析。为了使内容简短(让您不必费力去阅读那些基本的会反复出现的代码)，一些更常见的细节被省略掉了，例如构成控件时需要创建的 Windows 设计器代码。为了下载最新格式的完整学习实例，请浏览与本书相关的站点：<http://microsoft.com/mspress/books/6723.asp>。单击页面右边 More Information 框中的 Companion Content 链接可以进入含有相关内容的 Web 页面。您能够在作者的 Web 站点 <http://www.prosetech.com> 中找到其他内容。



注意

本书中的代码已经用 1.0 和 1.1 版本的 .NET 框架进行过测试。在大多数情况下，这两个平台几乎完全相同地进行工作，尽管 .NET 1.1(它是 Visual Studio .NET

2003 的一部分)作了一些小 bug 修复和性能提升。Visual Studio .NET 2003 和最初的 Visual Studio .NET 的执行出现不同的情况非常少, 文中特别提到了这些不同情况。这里不会使用在 .NET 1.0 中不可用的代码快捷方式。

变 量 命 名

过去, 微软命名变量时喜欢使用各种各样的匈牙利命名法。那意味着, 如果是 Visual Basic 整型变量就会以 int 前缀开始(例如 intCount)。但是在 .NET 环境中自动进行内存管理, 改变数据类型不会导致任何严重后果, 并且大多数变量被保存成完整对象的引用。匈牙利命名法开始显得有些陈旧了。

新的微软标准将避免变量前缀, 特别是对其他类可见的属性与方法。这种形式与 COM 组件和控件使用的标准相似, 它对代码透明化具有很大意义。数据类型不再像过去那样引起问题, 因为 Visual Studio .NET 将发现非法转换并拒绝编译。

本书中的变量不使用数据类型前缀。但控件变量则是个仅有的有意义的例外, 前缀对它来说仍旧是在不同类型控件(例如 txtUserName 和 lstUserCountry)和某些数据对象之间进行区分的有用方法。当然, 在创建自己的程序时, 可以自由选择喜欢的变量命名规则, 使您能够在所有的项目中(实际上是您公司的所有项目)尽量保持一致。

意 见 反 馈

本书试图被设计成为学习 .NET 分布式应用程序技术与概念的最佳向导。为了这个目的, 您的建议和意见非常有用。您可以直接向作者发送意见或褒扬等内容, 地址: distribapps@prosetech.com。作者不能解决您的 .NET 问题或评论您的代码, 但是作者将从有关本书的各种好的或坏的评价(或者有关作者的混淆之处的评价)中获得益处。

技 术 支 持

为了降低书的成本, 减轻读者负担, 对于因内容很少而不值得单独配盘的图书, 我们将其范例代码或练习文件放在我们的网站上, 供读者下载。敬请访问以下网址: <http://www.wenyuan.com.cn>, 查找本书的有关链接。

如果您对本书或配书盘文件有任何建议、意见或想法，请通过以下电子邮件与清华大学出版社计算机第三事业部客户服务部取得联系：

service@wenyuan.com.cn

或致函：

北京 100084-157 信箱

读者服务部

邮编：100084

也可打电话：010-62792098-220

请注意，上述地址并不提供软件产品的支持。

目 录

前言	XIII	2.7.1 共享程序集以及 GAC	34
第 1 部分 关键技术		2.7.2 延迟签名	36
第 1 章 了解分布式体系结构	3	2.8 使用配置文件和共享程序集	37
1.1 什么是分布式应用程序	3	2.8.1 强名称程序集的版本策略	39
1.2 客户端/服务器体系结构	4	2.8.2 共享程序集的代码基础	40
1.2.1 客户端/服务器体系结构存在的问题	6	2.8.3 关于程序集绑定的决定性内容	41
1.3 分布式体系结构	7	2.9 小结	42
1.3.1 分布式体系结构的优点	8	第 3 章 无连接数据: 通用语言	43
1.3.2 DCOM 和分布式应用程序的历史	9	3.1 分布式系统中 ADO.NET 的角色	44
1.3.3 .NET 分布式技术	10	3.2 ADO.NET 对象族	44
1.4 小结	14	3.2.1 数据对象	45
第 2 章 .NET 组件	15	3.2.2 直接数据源交互	46
2.1 什么是组件	15	3.2.3 直接查询和数据读取对象	49
2.2 组件在分布式系统中的角色	16	3.2.4 存储过程	52
2.2.1 服务提供者	16	3.2.5 不知道提供者的 ADO.NET 代码	55
2.2.2 信息包	17	3.2.6 事务	57
2.3 示例组件剖析	17	3.3 DataSet	58
2.4 IComponent 接口	22	3.3.1 创建 DataSet	60
2.4.1 在设计时使用组件	23	3.3.2 DataSet 间接列映射	62
2.4.2 资源与可使用类	24	3.3.3 浏览 DataSet	64
2.5 连接字符串和配置文件	27	3.3.4 关系	65
2.5.1 Visual Studio .NET 中的配置文件	29	3.3.5 从 DataSet 更新	69
2.6 部署程序集	30	3.3.6 更新的问题	70
2.6.1 共享程序集	31	3.3.7 处理错误	72
2.6.2 搜索私有程序集	32	3.4 XML 与平台间的数据交换	73
2.7 程序集的版本	33	3.5 小结	75

第 4 章 .NET 远程处理: 更持久

的 DCOM	76
4.1 分布式系统中 .NET 远程处理的作用	76
4.2 .NET 远程处理基本知识	77
4.2.1 可序列化的类	78
4.2.2 可远程调用的类	80
4.2.3 远程组件宿主	82
4.3 关键的 .NET 远程处理设计决策	83
4.3.1 激活模式	83
4.3.2 对象生存期	84
4.3.3 服务器和客户端激活	84
4.3.4 通信	85
4.4 简单的远程处理服务器	87
4.4.1 远程对象	87
4.4.2 组件宿主程序	87
4.4.3 客户端应用程序	90
4.4.4 具有用户接口的 远程组件	92
4.4.5 使用 Visual Studio .NET 进行测试	93
4.4.6 使用一个不同的 格式化器	94
4.4.7 使用多信道	95
4.5 单独调用对象	96
4.6 单独对象	97
4.7 双向通信	98
4.7.1 使用事件	99
4.7.2 使用委托	103
4.8 配置对象生存期	105
4.8.1 修改生存期	108
4.8.2 使用租约发起者	109
4.9 常见的远程处理异常	111
4.10 小结	112

第 5 章 XML Web 服务

(RPC 便捷之路)	114
5.1 XML Web 服务在分布式 系统中的作用	114
5.2 .NET 远程处理与 XML Web 服务	115
5.3 XML Web 服务管道	116
5.3.1 IIS 的作用	118
5.3.2 新建虚拟目录	118
5.3.3 XML Web 服务 文件格式	119
5.4 创建 XML Web 服务	120
5.4.1 数据序列化	125
5.5 测试 XML Web 服务	127
5.5.1 WSDL 文档	129
5.6 使用 XML Web 服务	129
5.6.1 检测代理类	131
5.6.2 使用代理类	133
5.6.3 调试 Web 服务	135
5.7 ASP.NET 平台服务	137
5.7.1 SOAP 头与 SOAP 扩展	138
5.8 发布 XML Web 服务	139
5.9 跨平台 XML Web 服务	140
5.10 小结	141
第 6 章 线程化客户端(响应界面)	143
6.1 什么是多线程?	143
6.2 分布式系统中线程客户端的角色 ..	144
6.3 .NET 线程	146
6.4 异步委托	147
6.4.1 异步远程调用	150
6.4.2 异步 Web 服务调用	150
6.5 等待句柄	152
6.6 回调	153
6.7 异步 I/O	156

6.8 定制线程	157	8.2 发送与忽略通信	214
6.8.1 线程优先级	160	8.3 Message Queuing 介绍	216
6.8.2 线程管理	161	8.3.1 Message Queuing 服务器	217
6.8.3 线程调试	163	8.3.2 从属客户端	217
6.9 加锁	165	8.3.3 独立客户端	217
6.9.1 竞争条件	167	8.4 Message Queuing 基本知识	218
6.9.2 死锁	168	8.4.1 消息队列类型	218
6.9.3 使用 Monitor 类的 高级锁	169	8.4.2 消息队列服务	220
6.10 定制线程对象	172	8.4.3 配置消息队列	220
6.10.1 线程和用户界面	175	8.4.4 消息解析	221
6.10.2 使用回调和加锁	177	8.5 在 .NET 中操作队列	222
6.10.3 向线程发送指令	181	8.5.1 选择队列	223
6.11 小结	183	8.5.2 搜索队列	224
第 7 章 线程池与服务 (可缩放编程)	184	8.5.3 创建队列	225
7.1 分布式系统中线程处理的角色	184	8.6 发送消息	225
7.2 单独的基础	186	8.6.1 发送对象消息	227
7.2.1 跟踪客户端	188	8.6.2 高级消息配置	228
7.2.2 管理任务与清除任务	190	8.6.3 接收消息	230
7.2.3 Ticket 系统	192	8.6.4 浏览消息	232
7.2.4 加锁	193	8.6.5 响应和回执消息	233
7.3 一个单独的 XML Web 服务	196	8.7 分布式应用程序中的消息 队列	235
7.3.1 线程池	197	8.7.1 COM+ 队列组件	235
7.4 Windows 服务	198	8.8 小结	236
7.4.1 创建 Windows 服务	199	第 9 章 COM+ (组件服务)	237
7.4.2 安装 Windows 服务	204	9.1 分布式系统中 COM+ 的角色	237
7.4.3 调试 Windows 服务	207	9.2 COM 和互用	238
7.4.4 控制 Windows 服务	209	9.3 COM+ 注册基础	239
7.4.5 为组件宿主使用 Windows 服务	211	9.3.1 给程序集一个强名称	241
7.5 小结	212	9.3.2 动态注册	241
第 8 章 消息 (轻量通信)	213	9.3.3 手工注册	243
8.1 分布式系统中消息的角色	213	9.3.4 COM+ 与声明模型	244
		9.3.5 COM+ 版本控制	245