

PROGRAMMER TO PROGRAMMER™



.NET Enterprise Development with VB.NET:
From Design to Deployment

Written and tested for final release of .NET v1.0

.NET 企业应用 高级编程

—— VB.NET 编程篇

Matthew Reynolds Karli Watson 等著 康博 译



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

.NET 企业应用高级编程

—— VB.NET 编程篇

Matthew Reynolds 等著

Karli Watson

康 博 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: 01-2002-0258

内 容 简 介

Visual Basic.NET 是目前世界上最为流行的编程语言——Visual Basic 的最新版本。全新的 Visual Basic.NET 经历了脱胎换骨的改变,在保留原有易用性的基础上,拥有了众多以前 VB 编程人员所梦想的功能,成为真正的“面向对象”语言,几乎拥有和新型 C#语言相同的技术功能。

本书论述的中心是使用 VB.NET 设计企业应用程序的开发.NET 企业解决方案。首先从程序设计出发,介绍与开发分布式应用程序有关的 N 层模型、Web 开发和 Web 服务等背景知识, Wrox 企业对象,以及如何设计企业应用程序。同时,对随着本书的介绍而不断创建和完善的范例应用程序进行了概述。然后进入到本书的核心部分——开发部分,依次介绍了应用程序浏览器的创建、桌面应用程序的创建、自动化处理过程、使用 Web 服务、使用 ASP.NET、使用移动客户机、服务、安全性和自动化部署、应用程序的管理和性能监控等与企业开发密切相关的内容。

书中附带了大量的示例代码,一方面便于读者更好地理解书中介绍的企业应用程序的设计思想和开发方法,另一方面为读者编写自己的企业应用程序提供了大量翔实的源代码参考资源。无论是初学者还是资深的开发人员,凡是希望能够深入了解企业应用程序内部运行情况的读者,都能够从本书获益。

Matthew Reynolds, Karli Watson et al: .NET Enterprise Development With VB.NET:From Design to Deployment

ISBN: 1-861006-17-9

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任何部分。

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: .NET 企业应用高级编程——VB.NET 编程篇

作 者: Matthew Reynolds Karli Watson 等著 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: 于平

印 刷 者: 北京密云胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 24.75 字数: 633 千字

版 次: 2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

书 号: ISBN 7-302-05682-X/TP·3344

印 数: 0001~5000

定 价: 45.00 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

.NET Framework 是 Microsoft 提供的一个巨大的全新编程世界。.NET Framework 的设计目标是，通过改变服务、设备和用户交互的方式使 Internet 在商务中的应用得到革新。使用 .NET Framework 构建的下一代应用程序将允许用户和计算机之间以及计算机和计算机之间进行更加丰富、更加广泛和更加全面的交互。

.NET 除了提供与设备交互的新方法之外，还为我们提供了创建它们的快捷而有效的方法。通过使用像 XML 这样的业界标准，开发人员能够毫不费力地构建出能够跨越设备和领域界限的应用程序。当然了，除了能够保证商业信息的安全以外，.NET Framework 所提供的这些技术还不是非常完善。Microsoft 在开发 .NET 技术的过程中一直注意到这一点，还提供了在 Internet 上保存数据的安全方法，并允许应用程序为用户提供定制的功能。

本书论述的重点是企业解决方案。读者通过阅读书中一个不断扩充的案例分析，可以对在设计和开发一个完整的企业应用程序的过程中所涉及到的所有问题有一个清晰的理解。并对如何创建 .NET 企业应用程序，以及如何通过各种设备、使用不同的方法并以一种安全的方式与企业应用程序连接等问题产生一个广泛且基本的了解。因此，无论您需要创建什么样规模的应用程序，本书都将帮助您了解如何着手去做，以及如何更清楚地定义企业应用程序的各个部分的区域和范围。

本书的主要内容

第 1 章介绍了本书所涉及的主要内容，并对贯穿于本书后面部分的应用程序进行了概述。尤其是为了给后面的开发范例提供铺垫，对分布式编程和 N 层应用程序进行了总体上的介绍。

第 2 章讨论应用程序的设计。我们将对 Wrox 企业对象(Wrox Enterprise Objects, WEO)进行全面概括的介绍，它是一个通过 Builder 工具(在附录 A 中进行介绍)创建的丰富的对象层。接下来，还将介绍如何使用工具，以及如何创建和使用存储过程。

在了解了如何创建和运行基本的应用程序之后，第 3 章将带领读者学习创建一个应用程序浏览器。我们首先简要地介绍使浏览器可用于一般应用的各种方法，然后展示如何创建浏览器，这里主要涉及到了诸如身份验证、添加菜单选项以及运行子应用程序等内容。

第 4 章将介绍如何创建一个桌面应用程序，以及如何将桌面应用程序与业务对象挂钩。本章通过一个相关的简单例子演示了桌面应用程序的原理。在这个例子中，我们为您展示了如何编辑客户信息，以及如何使用我们的应用程序在数据库中搜索客户。

第 5 章讨论了自动处理。主要介绍了捕获、传送、处理以及报告订单，并展示如何创建一个订单处理程序。在讨论该主题的时候，我们还会介绍有关体系结构的问题，以及轻松开发这些功能的方法。

第 6 章对 Web 服务进行了讨论。该章内容为我们展示了如何使用 Web 服务发布业务对象的功能。首先，我们介绍了利用桌面应用程序来使用 Web 服务，在第 7 章中，我们介绍了如何利

用 ASP.NET 来使用 Web 服务。我们还将通过例子来介绍一种安全的方式，即使用 SSL 来为 IIS 中的身份认证服务传送用户信息。

由于目前可以查看来自不同客户机的数据，包括移动客户机，因此第 8 章将介绍如何使用移动客户机与我们的应用程序连接。我们将特别介绍 Microsoft 的 Mobile Internet Toolkit，然后演示一个使用了企业业务对象的小型应用程序。

第 9 章讨论了有关服务的内容。首先介绍用于验证远程对象的用户身份的基于标记的身份验证系统。然后介绍创建一个服务并与该服务连接，最后介绍怎样安装和运行这个服务。

第 10 章讨论了安全性和自动化部署。首先，我们深入讨论了代码访问安全方面的问题，其中包括获得证据及评估权限等内容。然后讨论应用程序安全和保障客户编辑的安全。

第 11 章深入介绍了管理的有关内容。在这一章，我们构建了一个管理工具，并且展示了一些同样适用于其他地方的基本要素，例如停止和开始服务。最后，简要介绍了一下 Microsoft 的管理控制台(Microsoft Management Console)。

第 12 章是本书的最后一章，介绍了监控企业应用程序方面的内容，主要讨论的内容包括性能计数器、异常报告、调试和负载平衡。为了使监控方面的内容更加完整，我们还简要介绍一下 Microsoft Application Center 2000。

附录 A 介绍了书中一直使用的 Builder 工具的内部运转方式，以帮助读者对后台的运行情况有一个更好的认识。

本书读者对象

对本书感兴趣的将是那些开发和出售通用软件产品的人，这些软件产品可以用作“业务流程”应用程序(或某种程度上用作“生产力”应用程序)。为自己公司或者是为其他公司定制通用产品，并将其作为“业务流程”应用程序来部署的人们也可以从本书中获取大量非常宝贵的信息。

当然了，所有希望深入了解企业应用程序，以便更好地探究其内部运行情况的人们也会从本书中受益。尤其令人高兴的是，作者为本书开发了一个 Object Builder 工具，该工具允许我们创建用于访问数据表的对象。

本书是一本中中级水平的程序开发类图书，该书也能够帮助初学者和高级开发人员理解创建企业应用程序的活动范围和必需的内在的有关知识。

使用本书的准备条件

为了能够运行本书中的范例，您必须具备以下软件条件：

- Windows 2000 或 Windows XP
- Visual Studio .NET

客户支持

我们一贯重视听取读者的意见，特别希望听到您对本书的反馈意见，例如，您喜欢什么，不喜欢什么，您认为我们下次如何才能做得更好。您可以填写本书后面的反馈卡，将您的意见寄给我们，或者发电子邮件到 feedback@wrox.com。请您在反馈表中务必写清楚本书的书名。

如何下载本书的范例代码

当您访问 Wrox 公司站点(地址为 <http://www.wrox.com/>), 通过 Search 工具或书名列表, 可以方便地定位需要的书目。然后, 单击 Code 栏中的 Download 超链接, 或者单击本书的详细页面中的 Download Code 超链接, 就可以下载相应的范例代码。

从我们的站点上下载的文件都是使用 WinZip 压缩过的文档。保存附件到本地磁盘上的文件夹中后, 需要使用一个解压缩程序(例如 WinZip 或 PKUnzip)来解压缩文件。在解压缩文件时, 通常将代码解压缩到每一章所在的文件夹中。在解压缩的过程中, 应确保解压缩程序(WinZip、PKUnzip, 其他)被设置为使用的文件夹名。

勘误表

我们已经尽最大努力确保本书中的文本和代码没有错误, 但是错误仍然在所难免。如果您发现本书存在错误, 例如拼写错误或不正确的代码段, 请给我们发来反馈信息, 我们将不胜感激。勘误表的发送可以节约其他读者学习本书的时间, 而且能够帮助我们提供更高质量的信息。请将您的反馈信息发送 e-mail 到 support@wrox.com。您的反馈信息将被确认, 如果正确, 将被粘贴到本书的勘误页面上, 或者在本书的后续版本中使用。

要在我们的站点上找到勘误表, 请访问 <http://www.wrox.com/>, 并通过 Advanced Search 或者书名列表轻松定位本书页面。然后, 单击 Book Errata 超连接即可, 该链接位于本书的详细页面中的封面图解下面。

E-mail 支持

如果您希望直接向详细了解本书的专家咨询书中的问题, 可以发送电子邮件到 support@wrox.com, 要求在邮件的主题栏中带上本书的书名和 ISBN(国际标准图书编号)的后 4 位数字。一个典型的电子邮件应包括以下内容:

- 在主题部分中必须有本书的书名、ISBN 的后 4 位数字和问题的页数。
- 在邮件的正文部分应包括读者的名字、联系信息和问题。

我们将不返回给您无用邮件, 以节约您和我们的时间。当您发送一个电子邮件信息时, 它将经过下面一系列的处理:

◆ 客户支持: 首先, 您的信息将被递送到我们的客户支持人员手中, 并由他们阅读。他们手中有最为常见的询问问题文档资料, 能够立即回答有关本书或者 Web 站点的任何常见问题。

◆ 编辑处理: 接着, 一些有深度的问题将被送到对本书负责的技术编辑手中, 他们在程序设计语言或者特定的产品上有着丰富的经验, 能够回答相关主题的详细技术问题。

◆ 作者处理: 最后, 如果编辑不能回答您的问题(这种情况很少发生), 他们将请求本书的作者。我们将尽量保护作者免受干扰, 以便不影响其写作。然而, 我们也非常高兴转寄给他们一些特殊的问题。所有 Wrox 公司的作者都为他们的书提供技术支持。作为回应, 他们将发送电子邮件给用户和编辑, 进而使所有的读者受益。

Wrox 公司的支持部门仅仅对那些与我们出版的书目内容直接相关的问题提供支持, 对于超出常用书目支持的问题, 您可以从 <http://p2p.wrox.com/forum> 中的公共列表中获得支持信息。

p2p.wrox.com

为了便于作者和其他人讨论，特将讨论内容加入到 P2P 站点的邮件列表中，而且我们唯一的系统将 programmer to programmer™(由程序员为程序员而作)的编程理念与邮件列表、论坛、新闻组以及所有其他服务内容(一对一的邮件支持系统除外)相联系。如果您向 P2P 发送一个问题，应该相信它一定会被登录邮件列表的 Wrox 公司作者和其他相关专家所检查到。无论您是在阅读本书，还是在开发自己的应用程序，都可以在 p2p.wrox.com 站点中找到许多对自己有所帮助的邮件列表。

要想订阅一个邮件列表，只需按照以下的步骤操作：

- (1) 登录 <http://p2p.wrox.com>/站点。
- (2) 从左边的主菜单栏选择一个适当的种类。
- (3) 单击希望加入的邮件列表。
- (4) 按照说明订阅并填写自己的邮件地址和密码。
- (5) 回复您收到的确认邮件。
- (6) 使用预定管理程序加入更多的邮件列表并设置自己的邮件首选项。

本系统提供最好的支持的原因

您可以连接到整个邮件列表，也可以只接收每周的邮件摘要。如果您没有时间和工具来接收邮件列表，可以直接查找我们的在线文档。独特的 Lyris 系统可以将一些没有用的垃圾邮件删除，并保护您的电子邮件地址不被侵扰。当存在加入和离开列表、以及任何有关列表的其他常见问题时，请发送邮件到 listsupport@p2p.wrox.com。

目 录

第 1 章 绪论	1
1.1 商业环境的改变.....	1
1.2 分布式应用程序.....	2
1.2.1 N 层模型.....	3
1.2.2 Web 开发.....	6
1.2.3 Web 服务.....	6
1.3 .NET 简介.....	7
1.4 范例应用程序.....	7
1.4.1 应用程序浏览器.....	8
1.4.2 Web 服务.....	8
1.4.3 监控和管理.....	9
1.4.4 安全.....	9
1.4.5 移动接口.....	9
1.5 小结.....	10
第 2 章 应用程序设计	11
2.1 Wrox 企业对象.....	11
2.2 使用 WEO Object Builder.....	19
2.3 使用 Object Builder 工具.....	19
2.4 关联实体.....	34
2.5 插入数据.....	37
2.6 用户自己的存储过程.....	38
2.6.1 创建存储过程.....	39
2.6.2 调用存储过程.....	41
2.6.3 为自己的存储过程命名.....	44
2.7 小结.....	44
第 3 章 应用程序浏览器	45
3.1 准备工作.....	45
3.2 浏览器发布.....	46
3.3 开始创建 Web 服务.....	48
3.3.1 构建 Web 服务.....	48



3.3.2	GetFunctionalityCatalog 方法	52
3.4	构建客户程序	53
3.4.1	身份验证	55
3.4.2	增加菜单选项	61
3.4.3	更新标题	63
3.4.4	运行子应用程序	64
3.5	与浏览器进行通信	69
3.6	打开新的浏览器和其他用户界面元素	73
3.7	小结	83
第 4 章	桌面开发	84
4.1	调试	84
4.2	编辑客户	88
4.2.1	Customer 实体	88
4.2.2	基本窗体	88
4.2.3	构建控件库	89
4.3	检索数据	90
4.3.1	搜索客户	91
4.3.2	EntitySetScroller 控件	94
4.3.3	绑定数据	98
4.3.4	移动记录	101
4.4	更改数据	104
4.4.1	标记更改	105
4.4.2	并发处理	108
4.4.3	保存更改	110
4.4.4	有关更新	115
4.5	添加新客户	117
4.6	小结	117
第 5 章	自动化处理过程和事务处理	118
5.1	我们的步骤	118
5.1.1	获取订单	118
5.1.2	订单处理	118
5.1.3	订单送货	119
5.1.4	订单报告	119
5.2	问题的范围	119
5.2.1	获取订单	119
5.2.2	处理订单	120

5.2.3 订单送货	120
5.2.4 订单报告	120
5.3 载入订单	120
5.3.1 载入 XML	122
5.3.2 处理 XML	129
5.4 订单处理和事务处理	132
5.4.1 构建“OrderProcessor”	132
5.4.2 处理订单	133
5.4.3 测试 Processor 项目	145
5.4.4 回滚测试	148
5.5 使用服务	148
5.6 小结	148
第 6 章 Web 服务	149
6.1 为什么要使用 Web 服务	149
6.2 使用 Web 服务访问 BookManager 数据	150
6.3 保障 BookManager 服务的安全	159
6.3.1 身份验证与授权	160
6.3.2 SSL 连接	166
6.3.3 保障“BookManager”服务的安全	167
6.4 小结	181
第 7 章 Internet	182
7.1 ASP.NET 速成	182
7.2 BookManager ASP.NET 应用程序	186
7.2.1 GetAuthorsForBook 存储过程	187
7.2.2 FanMail Web 服务	188
7.2.3 AuthorFanMail Web 应用程序	191
7.2.4 测试应用程序	206
7.3 设计服务器控件	210
7.4 小结	210
第 8 章 移动控件	211
8.1 无线 Internet	211
8.2 移动 Internet 工具箱	213
8.2.1 移动 Web 项目	214
8.2.2 移动控件范例	215
8.3 访问移动的 BookManager	226



8.3.1	BookList 应用程序	227
8.3.2	分页	233
8.4	小结	233
第 9 章	服务	235
9.1	Remoting	235
9.2	基于标记的身份验证	236
9.3	验证用户身份	237
9.3.1	IIS 配置	238
9.3.2	“身份验证”的方法	239
9.3.3	测试身份验证	242
9.3.4	“注销”方法	242
9.3.5	从应用程序浏览器调用“身份验证”	243
9.4	建立服务	247
9.4.1	第 1 步——服务实现库	248
9.4.2	第 2 步——控制台应用程序	251
9.4.3	第 3 步——Windows 服务	252
9.5	与服务连接	253
9.5.1	测试连接	255
9.5.2	工作原理	256
9.5.3	显示连接类型	260
9.6	传递标记	261
9.6.1	使用上下文调用	262
9.6.2	观察服务器端	264
9.7	Windows 服务	266
9.7.1	添加安装程序	267
9.7.2	安装和运行服务	268
9.7.3	继续开发	270
9.8	小结	270
第 10 章	自动部署和代码访问安全性	271
10.1	自动部署	272
10.1.1	设置 IIS	274
10.1.2	全局程序集高速缓存	277
10.2	.NET 的代码访问安全性	278
10.2.1	获取证据	280
10.2.2	安全策略	282
10.2.3	评估权限	287

10.2.4	更多的安全性问题	289
10.2.5	禁用安全性	290
10.3	应用程序安全性	290
10.3.1	用户权限	290
10.3.2	保障客户编辑的安全	291
10.4	断言安全性	298
10.5	小结	300
第 11 章	管理	301
11.1	创建管理工具	301
11.2	创建工具	302
11.2.1	创建项目	303
11.2.2	管理对象	304
11.2.3	开始服务和停止服务	309
11.3	为“服务主机”添加管理对象	317
11.3.1	驻存管理对象	318
11.3.2	调用远程对象	321
11.3.3	默认视图	326
11.4	Microsoft 管理控制台	328
11.5	小结	329
第 12 章	性能监控	330
12.1	性能计数器	330
12.1.1	性能计数器简介	330
12.1.2	.NET 性能计数器	332
12.1.3	实现性能计数器	333
12.1.4	实现使用量计数器	333
12.1.5	可扩展性计数器	342
12.2	报告异常	345
12.2.1	串行化异常	345
12.2.2	“ReportException” Web 服务方法	349
12.2.3	测试异常处理程序	350
12.2.4	报告更多的数据	351
12.3	事件日志	352
12.4	调试和跟踪	354
12.4.1	提供调试信息	355
12.4.2	Debug.WriteLine	356
12.4.3	Trace vs. Debug	356



12.5	负载均衡	357
12.5.1	负载均衡的含义	357
12.5.2	循环法负载均衡	358
12.5.3	单点故障	360
12.5.4	Microsoft Application Center 2000	360
12.6	小结	361
附录 A	WEO 对象构造器	362
A.1	数据库扫描器	362
A.2	代码生成器	371
A.3	对象构造器类的参考	378

第 1 章 绪 论

在本书从头至尾的讲述过程中，将逐步建立起一个能够完全使用的企业应用程序，该应用程序假定是提供给图书零售商使用的。在这个过程中，我们希望演示说明商务应用程序可以使用的领域，如何解决像数据并发操作、版本和安全这样的基本问题。我们还会引入我们已经开发出来的 Object Builder 工具集，它们有助于应用程序软件的开发。这里看到的技术仅是建立企业应用软件方法的一个基本范例。本书可以由读者根据自己的应用程序在全书的章节中选择学习的课程，从而为您设计和构建自己的商业软件提供一些思路。读者不可能只是简单地将书中的代码范例拿出来并直接使用它们。更为妥当的做法是，用这些范例代码作为建立您自己应用程序的基础程序块。

最近，大家会看到一种运用 Internet 技术(尤其是 Web 浏览器和 Web 服务器)，将单独的应用程序迁移到更为分布的模型的趋势。然而很可能再经过两、三年后，会因为我们稍后讨论的几个因素而再一次倾向回到桌面开发。因此在本书的讲述过程中，将桌面应用程序的开发与基于 intranet / extranet 的解决方案放在同等重要的位置。然而，在本书中当然也会介绍非桌面应用程序，尤其是 Web 和 Web 服务应用程序的开发。

我们将要介绍的用于建立一个企业应用程序的一些重要主题包括：

- 带有大量的、可用商业对象的 N 层体系结构
- 桌面(Windows Forms)开发
- Web 开发(intranet / extranet 和 Internet)
- 移动设备的访问
- Web 服务
- 部署
- 集中式监控和管理
- 安全

在开始介绍应用程序的设计之前，首先从总体上概要讲述一下企业应用程序的背景知识。

1.1 商业环境的改变

现代的公司会发现它们的商业前景在最近的 10 年中经历了根本性的改变。随着国内市场和全球市场的开放，商业竞争变得更加激烈。为了生存，公司必须能够按照自己的处理方式充分利用好工具。最近技术上的进步为更高的工作效率提供了保证，并且为客户获得产品或所需的服务提供了更为廉价和易于得到的新方法。

在 IT 企业应用的早期，应用程序只能运行在单独的计算机上，从而极大地限制了它们能够运作的领域。许多早期的桌面解决方案依赖于存储在每台计算机上的公司的数据库的本地副本，



每台计算机都是在相互独立的情况下对数据进行访问和操作。这种过分简化的模式带来了大量的维护问题，比如说在每台计算机上的本地数据库副本要在一些适当的时候进行同步更新，通常是在营业日下班之前，并且更新后的数据库必须再重新分布给每一台工作站。随着技术的进步，应用程序变得逐渐成熟，开始出现了以在线中心数据库为基础，通过网络进行数据访问的解决方案。通常情况下，完成任务所需的工作量可以由用户在桌面终端和服务器之间进行均衡。这种模式就是通常所说的“分布式计算”。

1.2 分布式应用程序

近来的许多企业解决方案都是由几个分布式应用程序组成的，可以根据任务的类型和客户计算机的运算能力在客户机和服务器之间分配任务量。

运行在快速本地网络中的这些应用程序，迫切需要将其扩展为具有通过更为宽广的 Internet 进行访问的能力。我们认为这种 intranet / extranet 应用程序是非常受欢迎的，因为它不存在相关的部署问题。要想使一个基于 intranet 的应用程序扩展到本地局域网之外使用，需要做的全部工作就是使得用户界面适合作为 Web 浏览器使用和提供一种连接到服务器的方法(有一些新的与不同 Web 浏览器的性能相关的巧妙办法，但是在这里我们并不作详细阐述)。由于免除了部署的“痛苦”，使得这种 intranet / extranet 应用程序非常受欢迎，并且说明了我们近来看到的向这种企业应用程序发展的趋势。

这种应用程序的不利方面在于，必须经常要在要适合 Web 浏览器的性能和服务器连接的限制之间找到平衡，这就限制了应用程序的功能和可用性。从另一个角度考虑，这种方法在很大程度上忽略了新式桌面计算机的整体性能。现在桌面计算机的性能是非常强大的，但是想要使用它们的性能非常困难，因为事实上所有的应用程序处理都是由服务器来完成的。

这就导致了向通常所说的“富客户机”的发展。在这种模式中，客户应用程序更为复杂完善，主要依赖客户机自身的性能进行工作，可以独立完成许多任务。

- 瘦客户机——客户机主要用于显示服务器发送的数据，接收用户的输入数据，并将数据传送给服务器。

- 富客户机——客户机除了使用数据以外还有能力做许多事情。例如，服务器可以返回一组原始值，客户机可以将数据转换为一个图表。客户机在将数据发送给服务器之前可以验证用户输入的数据，等等。

在实践中，事情很少能划分得这样清楚，在瘦客户机和富客户机之间还有许多划分。尽管如此，但是基本原理是正确的。对于赞成富客户机的人来说，也许最具有说服力的论据就是客户机能够完成更多的本地处理工作量，减少对服务器的资源消耗，从而允许服务器支持更多的并发用户。

“客户机——服务器”的术语对于建立在现今商业世界上的联网结构来说有一些过分简单化，但是客户机和服务器之间划分的底层概念仍然是有效的。许多应用程序仍然可以被认为是客户机——服务器结构的，尽管它们的底层模型更为完善，像我们下面将要介绍的 N 层结构模型。

1.2.1 N层模型

Microsoft 不是“n层”概念的发明者，但是它长期以来都在 Microsoft 软件开发中占有很大的份额，以至于现在很多人都认为它是由 Microsoft 提出的，同时 Microsoft 也是以这种身份来将它推向市场的。事实上，n层模型仅是一种将应用程序划分为多个部分(“层”或“级”)的方法，划分的每一部分完成一项具体的工作。

n层模型是客户机/服务器技术的扩展。在该模型中，单个服务器为若干客户机提供数据(尽管我说“单个服务器”，实际上可能是多于一个的。在某些情况下，一个应用程序可以使用多个服务器。这里的提法只是帮助读者将客户机/服务器模型考虑为“一个服务器，对应多个客户机”的样子)。

客户机—服务器技术最初的发展，是作为工作在大型机环境中的软件的一种自然扩展方法。在大型机环境中，由某种功能强大的大型中央处理器(大型机本身)来完成应用程序必需的运算处理，由性能差的小型计算机与用户进行交互。随着大型机对客户的吸引力逐渐减少，主要是由于小型计算机增长(更低的价格，更高的性能)所形成的冲击，产生了一种变更趋势。实际上，用户终端(或称为“客户机”)计算机变得性能更为强大，并且价格更为便宜，意味着中心计算机发挥的作用更少了。

随之发生的是客户端代码变“胖”了，从理论上来说，服务器端的代码应该是变得更“瘦”了。然而，随着客户端软件变得更加复杂，客户端软件的部署工作也就变得更加难以解决。为了解决该问题，Internet 技术(尤其是 Web 服务器和 Web 客户机)逐渐变得更为复杂，这意味着我们又转回到了与大型机环境非常相似的一种情形，即功能强大的中心服务器和多台非常低端的客户机(即“哑终端”)。

目前，在使用.NET的情况下，我们可以看到软件开发和运行的模型重新向着“密集型”客户端代码转变，这意味着将有更多的处理过程由客户机来处理。这种转变是极有可能发生的，因为 Microsoft 针对各种技术所做的大量改进工作，都是为了使部署工作能够变得轻而易举。在第 10 章中我们将详细介绍这部分的内容。

今天，绝大多数的分布式应用程序都是建立在 3 层结构的基础上，即数据层(Data)、业务层(Business)和表示层(Presentation)。如果用客户机/服务器的标准来衡量，数据层和业务层可以被认为是服务器端，表示层可以被认为是客户端(尽管在 Web 开发中表示层的任务是由服务器和客户机共同完成的，但是我们仍然坚持上述划分)。

- 数据层——该层负责从数据库中取出数据并将其提供给业务层，反之亦然。
- 业务层——该层负责应用程序的实际处理。例如，它负责确保新的订单正确地存储在数据库中，以便用户能够搜索客户信息等等。
- 表示层——该层负责在业务层和用户之间调度数据。

典型的 3 层设计主要有以下两个优点：它使创建最初的应用程序变得更为容易，还有就是使扩充应用程序也变得很容易。下面我们就来对这 3 层分别作进一步的介绍。

1. 数据层

如果我们依次分开介绍每个层，数据层通常就是指 SQL Server，或是您根据自己的应用程序选择的 DBMS(数据库管理系统)。在企业环境中，可能常常会发现已经为您选择了所要使用