

PROGRAMMER TO PROGRAMMER™



Professional .NET for Java Developers  
with C#

# Java 开发人员 .NET 高级编程

——C# 编程篇

Jack Lunn

Dr P G Sarang 等著

张哲峰 庞太刚 等译



清华大学出版社

# Java 开发人员.NET 高级编程

——C#编程篇

Jack Lunn

Dr P G Sarang

等著

张哲峰 庞太刚

等译

清华大学出版社

北 京

北京市版权局著作权合同登记号：01-2002-5387

### 内 容 简 介

本书是专门为 Java 开发人员编写的一本 .NET 高级教程，全面介绍了 Java 开发人员开发 .NET 应用程序时需要了解和掌握的概念、方法和技巧。具体内容涉及到 .NET 平台体系结构、开发工具和语言语法、数据访问、动态 Web 应用程序、ASP.NET 服务、Windows Forms 和智能客户、企业组件、.NET Remoting，以及代码互操作等，并用大量实例和代码进行示范。

本书适合于那些希望了解 C# 和 .NET Framework 更多特性并具有一定的 Java 编程经验的开发人员阅读。

EISBN: 1-86100-791-4

Jack Lunn, Dr P G Sarang et al

Professional .NET for Java Developers with C#

Copyright© 2002 by Wrox Press Ltd.

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

All rights reserved.

本书中文简体字版由英国乐思出版公司授权清华大学出版社出版。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。**

### 图书在版编目(CIP)数据

Java 开发人员 .NET 高级编程——C# 编程篇 / (美) 鲁恩等著；张哲峰等译. —北京：清华大学出版社，2003

书名原文：Professional .NET for Java Developers with C#

ISBN 7-302-06664-7

I. J... II. ①鲁...②张... III. ①JAVA 语言—程序设计②C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 044451 号

**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.com.cn>

**责任编辑：**王晓娜

**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：**24.5 **字数：**627 千字

**版次：**2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷

**书号：**ISBN 7-302-06664-7 TP·4986

**印数：**0001~4000

**定 价：**49.00 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](http://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](http://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

Microsoft 对 .NET 的首创性踌躇满志,深信其他用户将不得不购买其许可证,以此促成 .NET 在业界的霸主地位。毫无疑问, .NET Framework 体现出与传统 Microsoft 模式的较大幅度的分离。 .NET 是完全面向对象的产品,为各种编程问题提供了统一的解决办法。顾名思义, .NET Framework 提供的模式具有很高的网络识别功能,并对一系列技术提供了强有力的支持,如远程处理、Web 服务、XML 以及其他一些技术(如新推出的智能客户范例)。

Microsoft 之所以调整并统一旗下的所有编程工具,其根本原因是它受到了 Java 的威胁。Java 是一门非专用性语言,且受到很好的支持。该语言非常精巧,且完全面向对象。这些特性使 Java 在过去的数年内取得了比 Visual Basic 更大的成功。Bill 及其 Microsoft 帝国不得不被动应战。

众所周知,Microsoft 的竞争对手已经将 Microsoft 诉之以法,控告 Microsoft 非法利用其垄断地位,例如,为了阻止 Java 的发展,Microsoft 控制了 Windows 中所包含的用来支持其他非标准的语言结构的 JVM,而这种行径与 Sun 公司针对 Java 发布的 General Public License(GPL,通用公共许可证)相违背。法庭最终同意 Sun 公司的控诉,责令 Microsoft 更新其当前版本的 JVM(当时的版本是 v1.4.1)。

因为 Microsoft 无法创建特定于 Windows 的 Java 语言,并且其营销策略又主要基于创建并维护对其操作系统的依赖性,所以 Microsoft 创建了它们自己的 C#语言,以享用 Java 语言中众多的特性和优势。尽管 Microsoft 宣称 C#派生于 C++,而不是 Java,但在这种新语言中,有很多内容更类似于 Java,这也令 Java 开发人员处于领先地位,并使 C#成为 .NET 模型的一个逻辑入口。

## 本书读者对象

本书适用于那些希望了解 C#和 .NET Framework 更多特性的 Java 开发人员。C#有很多内容类似于 Java,而本书将利用这一特性来介绍这门新的语言和新的编程模式。本书通过大量的示例突出显示了 Java 和 C#之间的异同。无需安装任何程序包(如 Visual Studio .NET)即可学习本书内容,您可以通过免费下载的命令行编译器和工具使用所展示的全部代码。

## 本书主要内容

本书首先将讲述一些重要的基础体系结构和语法问题,所有 .NET 开发人员在学习更高级的技术、成为专业开发人员之前,均需了解这些技术。我们将展示如何实现 Windows 桌面应用程序、Web 应用程序、数据访问、远程处理、智能客户以及互操作性。



本书主要包含以下内容:

**第 1 章——平台体系结构:** 首先简要介绍 .NET Framework。在本章我们将查看构建 Framework 的各个组件,即公共语言运行库、类库和 ASP.NET。此外,将探讨 .NET 的优势,并将其与 J2EE 相对比。

**第 2 章——开发工具:** 虽然可以从 Microsoft 免费下载一些用于编译和运行所有 .NET 应用程序的工具,但是 Microsoft 还提供了一些功能强大的开发工具,它们可以大幅度提高工作效率。我们将在本章介绍这些工具的重要性。

**第 3 章——语言语法:** 本章将讲述 C# 语言的核心内容。我们将对照 Java 语法做一些有用的对比,并创建一些简单的示例,以介绍 C# 的一些基础内容,如运算符、循环、方法和异常。最后将介绍 C# 的一些特有属性,如索引器、属性和委托。

**第 4 章——数据访问:** 几乎每个应用程序都会用到保存在某种类型的数据存储器中的数据,这些数据存储器可能是固有的数据库,也可能是已格式化的文本文件,如 XML。本章将介绍 .NET 提供的访问应用程序数据的一些技术,展示如何显示、编辑和删除数据。

**第 5 章——动态 Web 应用程序:** 本章将介绍在 .NET 环境中创建 Web 应用程序的基础知识,讲述在创建 Web UI 时可供使用的、作为 ASP.NET 应用程序驱动的事件以及控件。我们将创建一些示例,以展示各种控件,包括常用的 HTML 控件以及更为复杂的 .NET 服务器控件。

**第 6 章——高级动态 Web 应用程序:** 本章是对前一章内容的扩展,讲述了更高级的技术。将介绍如何在 Web 应用程序中实现绑定,从而可以快速地在 Web 页面中显示数据,并与之交互。本章还将介绍应用程序状态和会话状态,以便使应用程序可以在用户浏览不同网页期间保留重要的信息。此外,还将查看可以用来扩展标准 Web 控件、创建自定义控件的用户控件,完善本章的内容。

**第 7 章——ASP.NET Web 服务:** 与 Java 环境不同的是, .NET 平台更多地面向 XML Web 服务。本章是介绍 Web 相关内容的最后一章,在本章中,我们将查看 .NET 如何利用 ASP.NET 的内置支持,简化 Web 服务的开发工作。我们将创建并部署一个简单的示例,实现与后台数据库的交互,并展示如何利用命令行工具来生成与 Web 服务进行互操作的代码。

**第 8 章——Windows Forms 和智能客户程序:** .NET Framework 的一个优势是,开发人员可以使用完全相同的语言和非常类似的技术进行 Web 程序开发和桌面程序开发。本章将通过 Notepad 创建一个功能完整的 Windows 桌面应用程序,并在命令行中编译该程序。随后,我们修改这个独立的桌面应用程序,将其转换为智能客户程序(Microsoft 为内置有部署和更新功能的桌面应用程序指定的名称),并将 Web 应用程序的中央控制功能引入到该桌面应用程序中。

**第 9 章——企业组件:** J2EE 环境有很多方面是针对大型分布式企业应用程序的,而 .NET 至少提供了与此相同的功能。本章将帮助您熟悉与企业应用程序开发人员相关的一些 .NET 特性和类。

**第 10 章——.NET Remoting:** Java 机制中提供了一项用于远程运行代码的 RMI(Remote Method Invocation, 远程方法调用)技术, .NET 中也采用了相应的技术。这些技术允许我们无缝访问远程计算机中的代码,而这种功能构成了企业分布式解决方案的重要组成部分。开发人员要实现远程处理解决方案,需要了解众多的技术细节,本章将介绍所有相关内容。

**第 11 章——与现有代码的互操作:** 在采用新的技术(如 .NET)时,大多数企业都拥有一定数

量的旧组件库。要顺利地实现技术转移,关键在于能够利用现有的代码。本章将介绍如何在.NET应用程序中访问 Java 和 COM 组件。

## 学习本书的条件

本书针对那些希望了解 C#的 Java 开发人员。因此,我们尽量减少运行示例所需使用的工具。大部分代码都可以通过.NET SDK(可免费下载)提供的命令行编译程序和工具运行。您可以在 MSDN 下载区左侧的导航栏中单击 Software Development Kits 选项下的 Microsoft .NET Framework SDK,获取该 SDK。该下载区的网址如下:

<http://msdn.microsoft.com/downloads/default.asp>

要运行该 SDK,您必须在计算机中安装有 Windows NT、2000 或 XP。

对于 Web 应用程序(包括 Web 服务)而言,在安装.NET Framework 之前,需要在 Windows 2000 或 XP 中安装 IIS。IIS 是 Windows 2000 和 XP Professional 中的一个可选组件,可在安装完操作系统之后独立安装。如果需要,可以通过 Windows 控制面板的添加/删除程序对话框安装 IIS。

## 用户支持

我们一贯重视读者的意见,并想知道每位读者对本书的看法,包括读者喜欢和不喜欢的内容,以及读者希望我们下一次完善的地方。您可以通过发送电子邮件(地址为 [feedback@wrox.com](mailto:feedback@wrox.com))向我们反馈意见。请确保反馈信息提到本书的书名。

### 1. 如何下载本书的示例代码

当您访问 Wrox 公司站点(地址为 <http://www.wrox.com/>)时,通过 Find a Book 工具即可方便地定位需要的书目。然后,单击本书详细信息页面中的 Download Code 超链接,或者 Wrox 主页顶端的 DOWNLOAD CODE 超链接,并在文本框中选中本书,就可以下载相应的示例代码。

在下载本书代码之前,您可以提供您的用户名和当前的电子邮件地址,以注册该书。该步骤完全是可选操作,以便我们在您下载代码出现问题或以后更新代码下载数据包时,方便与您联系。Wrox 公司确保在注册期间,您的详细信息不会泄漏给任何第三方。关于隐私策略的详细信息,请单击下载页面上的相关链接查看。

从我们的站点上下载的文件都是使用 WinZip 压缩过的文档。保存文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 或其他)被设置为使用原有文件夹名。

### 2. 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,请反馈信息给我们,我们将不胜感激。



勘误表的发布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。您的反馈信息将被检查，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 Find a Book 搜索框定位本书页面。然后，单击 View Errata 超链接即可看见本书最新的勘误列表。该链接位于本书的详细页面中的封面图解下面。

### 3. E-mail 支持

如果您希望直接向详细了解本书的专家咨询本书中的问题，可以发送电子邮件到 [support@wrox.com](mailto: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。一封典型的电子邮件应包括下面的内容：

- 在主题栏中必须有本书的书名、ISBN 的后 4 位数字和问题所在的页码。
- 邮件正文应包括读者的名字、联系信息和问题。

我们不会返回给您无用邮件，因为我们仅仅需要有用的详细资料，以便可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支持：

- 用户支持：首先，您的信息将被递送到我们的用户支持人员手中，并由他们阅读。他们手中有最被频繁提到的问题的文档资料，能够立即回答有关本书或者 Web 站点的任何常见问题。
- 编辑支持：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。
- 作者支持：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 公司的支持过程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，对于超出常规书目支持的问题，您可以从 <http://p2p.wrox.com/>论坛的公共列表中获得支持信息。

### 4. p2p.wrox.com 站点

为了便于和作者及同行讨论，请加入到 P2P 站点的邮件列表，而且我们独特的系统除了提供了一对一的邮件支持系统之外，还通过邮件列表，论坛，新闻组等联系方式进一步体现了 Programmer to Programmer™(程序员为程序员而著)的理念。如果您向 P2P 发送一个问题，应该相信一定会得到登录邮件列表的 Wrox 公司作者和其他相关专家的帮助。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](http://p2p.wrox.com) 站点中找到许多对自己有所帮助的邮件列表。

按照下面的步骤可以访问邮件列表：

- (1) 登录 <http://p2p.wrox.com/> 站点。
- (2) 从左边的主菜单栏选择适当的类别。
- (3) 单击希望加入的邮件列表。



- (4) 按照订阅说明填写自己的邮件地址和密码。
- (5) 回复您收到的确认邮件。
- (6) 使用订阅管理程序加入更多的邮件列表并设置自己的邮件首选项。

### 本系统提供最佳支持的原因

您可以加入整个邮件列表，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](mailto:listsupport@p2p.wrox.com)。

# 目 录

<b>第 1 章</b>	<b>平台体系结构</b> .....	<b>1</b>
1.1	Microsoft .NET 的含义 .....	1
1.1.1	.NET Enterprise Server .....	2
1.1.2	Microsoft .NET 智能客户 .....	3
1.1.3	Microsoft .NET XML Web 服务 .....	3
1.1.4	Microsoft .NET 开发人员工具 .....	4
1.2	XML Web 服务 .....	4
1.2.1	XML Web 服务和 SOAP .....	4
1.2.2	WSDL、DISCO 和 UDDI .....	5
1.3	.NET Framework .....	6
1.4	ASP.NET .....	9
1.5	Windows Forms .....	10
1.6	公共语言运行库 .....	11
1.6.1	类型安全 .....	12
1.6.2	公共语言规范(CLS) .....	13
1.7	基类库 .....	14
1.8	J2EE 和 .NET 的对比 .....	17
1.8.1	J2EE 对 .NET 的反击 .....	18
1.8.2	Microsoft 的反辩 .....	19
1.9	小结 .....	19
<b>第 2 章</b>	<b>开发工具</b> .....	<b>20</b>
2.1	Notepad 和 csc.exe .....	20
2.1.1	控制台应用程序 .....	20
2.1.2	运行应用程序 .....	21
2.2	Visual Studio .NET .....	21
2.2.1	Web 窗体 .....	22
2.2.2	Windows Forms .....	22
2.2.3	XML 支持 .....	22
2.2.4	Web 服务 .....	22
2.2.5	创建 Windows 应用程序 .....	22
2.2.6	使用 VS.NET 创建 Web 应用程序 .....	27



2.2.7	部署并运行应用程序	30
2.2.8	创建 ASP.NET Web 服务	30
2.2.9	运行 Web 服务	32
2.2.10	在 Visual Studio .NET 中进行调试	35
2.3	WebMatrix	36
2.3.1	界面	36
2.3.2	Web Matrix 特性和语言支持	37
2.3.3	设计	37
2.3.4	Toolbox	39
2.3.5	Properties 窗口	41
2.3.6	编码和开发	42
2.4	Dreamweaver MX	44
2.4.1	界面	44
2.4.2	文档和模板	45
2.4.3	特性和语言支持	46
2.5	SharpDevelop	46
2.6	小结	47
<b>第 3 章</b>	<b>语言语法</b>	<b>49</b>
3.1	源文件约定	50
3.1.1	文件命名	50
3.1.2	顶级声明	50
3.2	语法	52
3.2.1	基本数据类型	52
3.2.2	转换和强制转换	54
3.2.3	值类型和引用类型	55
3.2.4	运算符	56
3.2.5	流控制	61
3.3	类的基础知识	63
3.3.1	访问修饰符	63
3.3.2	Main() 方法	65
3.3.3	其他方法	66
3.3.4	属性	69
3.4	结构	70
3.5	C#中的数组	71
3.5.1	一维数组	71
3.5.2	多维数组	72
3.5.3	不规则数组	73

3.5.4	System.Array 类	73
3.6	继承类和派生类	74
3.6.1	对基类进行类型转换	76
3.6.2	方法重写	78
3.6.3	抽象类	81
3.6.4	接口	82
3.6.5	运算符重载	84
3.7	异常	86
3.8	高级 C# 技术	87
3.8.1	索引器	87
3.8.2	属性	89
3.8.3	委托	90
3.8.4	无用单元收集	93
3.8.5	安全代码和不安全代码	93
3.9	小结	95
<b>第 4 章</b>	<b>数据访问</b>	<b>96</b>
4.1	ADO.NET 的结构	96
4.1.1	ADO.NET 托管提供者	99
4.1.2	DataAdapter 类	101
4.1.3	DataSet 类	102
4.2	对 DataSet 进行过滤和排序	105
4.3	实际操作	106
4.3.1	取数据	108
4.3.2	处理数据	111
4.3.3	ADO.NET 事件	125
4.4	ADO.NET 异常	127
4.5	输出 DataSet 的内容	130
4.5.1	XML	130
4.5.2	串行化	133
4.6	小结	136
<b>第 5 章</b>	<b>动态 Web 应用程序</b>	<b>137</b>
5.1	页面请求及其工作原理	138
5.2	ASP.NET 页面事件	139
5.3	HTML 服务器控件	141
5.4	Web 服务器控件	152
5.5	验证控件	183
5.6	内联或后台编码	194



5.6.1	后台编码原理.....	195
5.6.2	ASP.NET 和 JSP.....	196
5.7	小结.....	197
<b>第 6 章</b>	<b>高级动态 Web 应用程序.....</b>	<b>199</b>
6.1	运行示例.....	199
6.2	ASP.NET 中的数据绑定.....	199
6.2.1	属性.....	200
6.2.2	集合.....	201
6.2.3	DataTables.....	203
6.3	管理会话和应用程序状态.....	215
6.3.1	会话状态.....	215
6.3.2	购物车页面.....	216
6.3.3	应用程序状态.....	223
6.4	构建自定义控件.....	226
6.4.1	基础自定义控件.....	226
6.4.2	被派生的自定义控件.....	229
6.5	复合自定义控件.....	232
6.6	小结.....	239
<b>第 7 章</b>	<b>ASP.NET Web 服务.....</b>	<b>240</b>
7.1	创建 ASP.NET Web 服务.....	240
7.1.1	调用 Web 服务.....	242
7.1.2	创建 ASP.NET Web 服务.....	243
7.1.3	测试 ASP.NET Web 服务.....	246
7.1.4	使用 Web 服务.....	247
7.1.5	在 .NET 客户程序中使用外部 Web 服务.....	252
7.2	Web 服务链.....	255
7.3	小结.....	256
<b>第 8 章</b>	<b>Windows Forms 与智能客户.....</b>	<b>257</b>
8.1	Windows Forms 应用程序简例.....	257
8.1.1	背后隐藏的秘密.....	258
8.1.2	处理属性.....	260
8.1.3	向窗体添加控件.....	262
8.1.4	控件的布局选项.....	264
8.1.5	事件处理.....	265
8.1.6	完成应用程序.....	266
8.1.7	构建应用程序.....	276

8.2	智能客户	277
8.2.1	创建程序集	278
8.2.2	发布程序集	278
8.2.3	信任程序集	278
8.2.4	创建加载器	278
8.2.5	测试智能客户	280
8.2.6	更新智能客户	281
8.3	WinForms 数据绑定控件	283
8.3.1	与单个元素绑定	283
8.3.2	与多个元素绑定	285
8.4	小结	286
<b>第 9 章</b>	<b>企业组件</b>	<b>288</b>
9.1	J2EE 对企业应用程序的支持	288
9.1.1	J2EE 平台	289
9.1.2	J2EE 技术	292
9.1.3	J2EE 部署	294
9.2	.NET 对企业应用程序的支持	294
9.2.1	System.EnterpriseServices 库	295
9.2.2	组件服务管理器	296
9.3	COM+ 的应用	300
9.3.1	编写 ServicedComponent	300
9.3.2	配置程序集	303
9.3.3	编译应用程序	304
9.3.4	添加对象入池和 JIT 激活支持	304
9.3.5	编写客户应用程序	305
9.4	小结	307
<b>第 10 章</b>	<b>.NET Remoting</b>	<b>308</b>
10.1	远程对象体系结构	308
10.1.1	RMI	308
10.1.2	.NET Remoting	309
10.2	探索 .NET Remoting	310
10.2.1	Remoting 命名空间	310
10.2.2	.NET 中的串行化	311
10.2.3	编组概念	314
10.2.4	信道	316
10.2.5	格式化程序	317
10.2.6	激活	317



---

10.2.7	调用上下文.....	326
10.2.8	租约和发起人.....	327
10.2.9	跟踪和日志记录.....	333
10.3	构建 Remoting 应用程序.....	337
10.3.1	委托和事件的远程处理问题.....	338
10.3.2	构建聊天服务器.....	339
10.4	小结.....	353
<b>第 11 章</b>	<b>与现有代码互操作.....</b>	<b>355</b>
11.1	与 Java 互操作.....	355
11.1.1	Microsoft Biztalk Server.....	355
11.1.2	使用 Web 服务进行集成.....	356
11.1.3	Microsoft Visual J#.....	356
11.2	与 COM 互操作.....	365
11.2.1	通过.NET 访问 COM 组件.....	366
11.2.2	将 ActiveX 控件合并到.NET 应用程序中.....	370
11.3	小结.....	371

# 第1章 平台体系结构

随着 2000 年 Microsoft .NET 平台的发布,全世界迎来了剧烈的信息风暴。但同时这样大规模的信息也使得人们难以判断和理解哪些是重要信息,哪些不是重要信息。当前,大量产品和技术的名称中都添加了.NET,包括:

- .NET Enterprise Server(如 SQL Server 和 Exchange Server)
- Windows .NET Server
- Visual Studio .NET
- .NET Framework

本章将具体介绍.NET 平台以及构成该平台的技术和产品。随后,我们要讲述.NET 策略中的关键部件——.NET Framework,这也是本书的中心内容。最后,我们将讨论 Java 2 Enterprise Edition(J2EE)和 Microsoft .NET 平台之间的差异。

Microsoft .NET 希望能利用 Java 的功能,改进自己的不足,并同时添加大量优秀的功能。因为很多基础概念是相同的,所以到本章结束时,读者应该可以较透彻地理解 Microsoft .NET 的工作原理。

本章内容包括:

- Microsoft .NET 的根本含义
- .NET Framework
- 对 J2EE 和.NET 进行比较

## 1.1 Microsoft .NET 的含义

Microsoft .NET 战略是基于以下原因而实现的:

- 通信开销的降低
- 客户端处理能力的提升
- 越来越多的用户以 Internet 作为中介来运行软件程序。

Microsoft 希望创建一个平台,可以在运行软件程序时利用客户端强大的处理能力和 Internet 强大的通信能力。使用分布式功能的标准的 n 层设计方法需要很多改进,因为客户端中仍有很多的功能尚待利用。

Microsoft 最终的解决方案是一个松散的、分布式的体系结构,它通过一组核心技术,使用 W3C Standardized XML Web 服务连接 Web 中所有的应用程序,并尽可能地利用客户端的处理功能。通过本章的学习,我们将深入地了解 Microsoft .NET 中每个部件。

Microsoft .NET 包括下列 4 个核心组件:

- .NET Enterprise Server



- .NET 智能客户
- XML Web 服务
- 开发人员工具

所有这些组件都运行在同一组技术环境下——即.NET Framework, 该架构包括 3 个核心组件:

- 公共语言运行库(CLR)
- .NET 类库
- Microsoft ASP.NET

这 3 个组件是实现.NET 战略的基础。我们将在本章后文中介绍每个组件的详细信息。

如图 1-1 所示, XML Web 服务是在 Microsoft .NET 平台中将该体系结构保持为一个整体的核心组件。

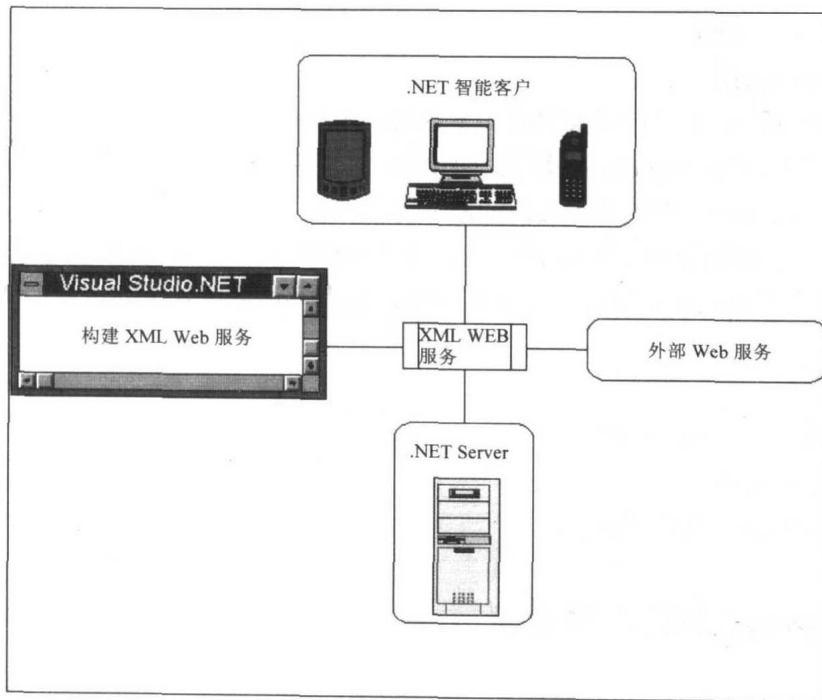


图 1-1

### 1.1.1 .NET Enterprise Server

.NET 当前的服务器基础结构包括 Windows 2000 Server 系列、.NET Enterprise Server 系列和 Windows .NET Server 系列。

.NET Enterprise Server 系列包括:

- Microsoft Application Center 2000
- Microsoft Biztalk Server 2000
- Microsoft Commerce Server 2000
- Microsoft Content Management Server 2001
- Microsoft Exchange Server 2000