

微软.NET程序员系列

Microsoft Press

- ◆ Amazon读者评价 ★★★★★
- ◆ 解说公共语言运行库基本概念
- ◆ 剖析.NET Framework结构与编程原理
- ◆ 大量代码可重用的经典实例

欧美编程类图书  
销量冠军!

Applied  
MICROSOFT

# .NET Framework Programming 程序设计

[美] Jeffrey Richter 著

Wintellect  
Raise Your Windows IQ™

商丽媛 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

# Microsoft .NET Framework 程序设计

[美] Jeffrey Richter 著  
商丽媛 译

清华大学出版社

(京)新登字 158 号

## 内 容 简 介

.NET 框架是一种新的计算平台,它简化了在高度分布式 Internet 环境中的应用程序开发。本书是《微软.NET 程序员系列》丛书之一。全书共 5 大部分,着重讲述 CLR 的工作原理以及它所提供的功能,还将讨论 FCL 的各个组成部分。具体内容包括: .NET 框架的基础知识,如何使用类型和公共语言运行库,如何设计类型和管理类型,并介绍了.NET 中的两类基本类型。本书所涵盖的技术信息,有助于透彻地理解基础的.NET 设计原则。学完本书后,将能够优化并调试.NET 框架应用程序,从而轻松、高效地创建高性能且安全可靠的.NET 应用程序。

本书适合广大编程爱好者阅读,对于使用 VB 或 C++ 的程序开发人员本书更是一本不可不读的好书。

**Applied Microsoft .NET Framework programming**

**Microsoft Press**

**Copyright © 2002 by Microsoft Corporation**

**Original English language edition published by Microsoft Press, a Division of Microsoft Corporation**

**All rights reserved.**

**No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher. For sale in the People's Republic of China only.**

本书中文简体版由 Microsoft Press 授权清华大学出版社出版发行,未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

图书在版编目(CIP)数据

Microsoft .NET Framework 程序设计/(美)理查著;商丽媛译.—北京:清华大学出版社,2002  
(微软.NET 程序员系列)

书名原文: Applied Microsoft .NET Framework Programming  
ISBN 7-302-05834-2

I.M... II.①理...②商... III.计算机网络—程序设计 IV.TP393

中国版本图书馆 CIP 数据核字(2002)第 067458 号

北京市版权局著作权合同登记号:图字 01-2001-2105 号

**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。**

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:许勇光

印 刷 者:北京牛山世兴印刷厂

发 行 者:新华书店总店北京发行所

开 本:787×960 1/16 印张:29.5 彩插页:4 字数:560 千字

版 次:2002 年 9 月第 1 版 2002 年 9 月第 1 次印刷

书 号:ISBN 7-302-05834-2/TP·3453

印 数:0001~5000

定 价:58.00 元

# 《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，也给软件开发商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 40 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，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！

这套丛书中的《C#技术内幕》和《C#编程技术》已于去年与读者见面，并得到读者的广泛好评。

目前，本丛书中已出版和在编的共有 17 本，已从 6 月份起陆续和读者见面。这些书目包括：

- **《Microsoft .NET Framework 程序设计》**  
全面、详细地介绍了 Microsoft .NET Framework，可以帮助开发人员和设计人员轻松、高效地创建高性能且安全可靠的 .NET 应用程序。
- **《应用程序升级——Visual Basic 6.0 到 Visual Basic .NET》**  
升级 Visual Basic 6.0 程序代码的最佳指导。涵盖了 Visual Basic .NET 的全部新功能和各种方案，这些方案能够以最小的中断进行代码的移植并维护混合环境。
- **《ASP.NET Web 应用程序开发新思维》**  
介绍了最新 Web 应用程序构建技术，带给您 Web 应用程序开发的新思想。帮助开发人员充分利用 Microsoft .NET，开发高效、安全的 Web 应用程序。
- **《构建 Web 解决方案——应用 ASP.NET 和 ADO.NET》**  
介绍 ADO.NET 与 ASP.NET 的综合应用，使开发人员可以将 Web 开发技术与数据库开发技术完美地结合起来，构建功能更加强大的 Web 应用程序和服务。
- **《Microsoft Windows 网络编程（第 2 版）》**  
主要讲述 Winsock 网络编程技术。详细介绍了如何编写高性能、可扩展的 Winsock 应用程序，还讲述了如何使用 C# 开发 Winsock 程序。
- **《构建 XML Web 服务——基于 Microsoft .NET 平台》**  
详细深入地讲解了用于创建 XML Web 服务的主要协议和工具，并深入探讨了如何利用它们构建功能强大、高效的 Web 商业解决方案。还介绍了 Microsoft .NET My Services 以及其他基于 XML 的微软新技术。
- **《Microsoft .NET My Services 精解》**  
该书由 My Services 开发组的专家共同编写。详细解释了各项 .NET 服务，并介绍了 .NET My Services 消息界面模型、数据操纵语言、安全授权模型、系统文档结构以及管理模型，助您创建出充分利用 .NET My Services 优点的 Web 服务。
- **《移动设备 .NET 应用程序设计》**  
介绍了如何使用 ASP.NET、Visual Studio .NET 和移动通信 Internet 工具箱创建 Web 应用程序，在各种移动设备上以适当格式动态显示相同页面。有助于学习如何使用微软的信息服务器，以及如何为移动设备提供 E-mail 访问。
- **《SQL Server 2000 与 Visual Basic .NET 编程》**  
介绍如何使用 SQL Server 2000、Visual Basic .NET、ADO.NET、ASP.NET、XML Web 服务和其他数据工具，开发数据库访问应用程序。
- **《Microsoft .NET Server 企业解决方案》**  
介绍如何计划、开发和部署企业电子商务解决方案。将理论与实际紧密结合，详细讲解了如何解决电子商务集成时遇到的问题，并进一步探讨了使用 Microsoft .NET Server 的各种

工具和技术解决具体的问题。

为了使读者的学习目标更为明确，在本丛书中，又划分出了两个子系列，现有 7 本书：

- **Core Reference 系列**

目前共有 4 本：《Visual Basic .NET 核心编程》、《Visual C#核心编程》、《ADO.NET 核心编程》和《Microsoft .NET 核心编程》。已经列入我们的出版计划的还有另外 3 本：《Visual C++ .NET 核心编程》、《Visual J# .NET 核心编程》和《Microsoft .NET Framework 核心编程》。这些书分别介绍相应编程语言和 .NET 框架类库的指令、示例代码、最佳编程惯例和基于案例的解决方案，内容详尽，讲解深入，集指导性与实用性于一体，最适合相关的开发人员用作专业读本。

- **Language Reference 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》和《Visual C++ .NET 语言参考手册》。以简洁、易于浏览和使用的形式，从 A 到 Z 列出了这 3 种编程语言的技术参考。

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，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。

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

2002 年 6 月

# 前 言

现在，每个人都注意到了 Internet 的价值，我们的生活方式也越来越依赖于基于 Web 的服务。我个人就非常喜欢通过 Internet 购物、获悉交通状况、比较产品、买票、阅读产品评价等。

然而，我发现仍然有很多事情是目前的 Internet 无法做到的。例如，我想在我居住的地区找到一家能提供某种佳肴的饭店。而且，我还希望能够询问饭店晚上 7 点是否有座位。或者，如果我有自己的公司，我可能希望知道哪一家供应商的某种产品有现货。如果有多家供应商都可以提供这种产品，我想找到报价最低的那家，或者是交货最快的那家。

未能提供这些服务的原因主要有两点：首先，没有一个适当的标准来集成这些信息。毕竟，现在的供应商都以自己的方式来描述他们的产品。描述各类信息的现行标准就是可扩展标记语言(XML)。其次，这些服务之所以还不能实现是因为开发集成这些服务所需的代码太复杂。

Microsoft 认为销售服务是未来的发展方向——也就是说，公司将提供各种服务，对此感兴趣的用户可以享受这些服务。很多服务将是免费的，当然也有一些需要通过订购计划才能得到，还有一些则需要按次付费。可以将这些服务视为某种商业逻辑的运行。以下是一些服务的示例：

- 验证信用卡交易
- 获取从 A 地到 B 地的指导
- 浏览一家饭店的菜单
- 预订航班、客房或租一辆轿车
- 更新在线相册中的相片
- 将您的日程结合孩子的日程，制定出一个全家度假计划
- 通过一个活期存款账户付款
- 跟踪发送给您的包裹

还有很多类似的服务，任何一家公司都可以提供这种服务。毫无疑问，Microsoft 将开发其中一些服务，并会很快提供给大家。其他公司也会提供一些服务，与 Microsoft 在自由市场上分庭抗争。

那么，我们如何利用这些将要出现的服务呢？如何生成基于 HTML 或其他服务的应用程序并将更为丰富的功能提供给用户呢？例如，如果饭店提供了能获取菜单的服务，那么应用程序应编写为可以查询每个饭店的菜单的工具，搜索特定的菜肴，然后只呈现那些用户住所周边的饭店。



## 注意

要创建此类功能丰富的应用程序，公司必须提供它们的商务逻辑服务的编程接口。这个编程接口必须可以通过网络(如 Internet)远程调用。这就是 Microsoft .NET 计划的全部含义。简言之，.NET 的目的就是将信息、人和设备互联。

计算机有外围设备与之相连，如鼠标、监视器、键盘、数码相机和扫描仪等。而操作系统，如 Microsoft Windows，提供了一个开发平台，支持应用程序对这些外围设备的访问。在某种程度上，甚至可以将这些外围设备都视为服务。

在这个日新月异的世界里，这些服务(或外围设备)现在都已连接到 Internet。开发人员希望用一种便捷的方式访问这些服务。Microsoft .NET 的部分目的就是提供这种开发平台。图 0.1 说明了这种开发平台。左边的 Windows 表示由硬件外围设备组成的开发平台，右边的 Microsoft .NET 框架表示 XML Web 服务通信的开发平台。

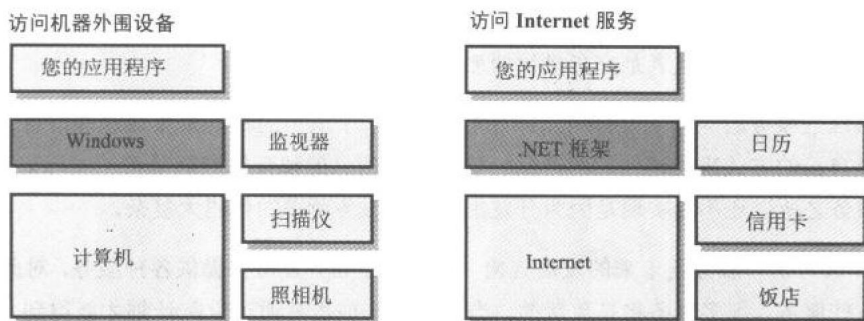


图 0.1 外围设备与 XML Web 服务的开发平台

尽管 Microsoft 在开发和定义使这个新世界成为可能的标准上扮演领导者的角色，但是它尚未拥有任何标准。客户端机器通过创建特殊格式的 XML 并将其通过企业内部网或 Internet 发送出去(主要使用 HTTP)，从而描述服务器请求。服务器知道如何分析此 XML 数据，处理客户端的请求，然后将响应以 XML 的形式返回给客户端。简单对象访问协议(Simple Object Access Protocol, SOAP)就是用来描述通过 HTTP 发送的特定格式的 XML 的术语。

图 0.2 显示了大量 XML Web 服务，它们都通过具有各自的 XML 有效负载的 SOAP 进行相互通信。该图还显示了运行应用程序的客户端，该客户端可以通过 SOAP(XML)与 Web 服务甚至其他客户端进行通信。另外，图中还显示了通过 HTML 从 Web 服务器获取的结果。用户可能填写一份 Web 表单，表单被发回到 Web 服务器。Web 服务器处理用户的请求(这涉及到与某些 Web 服务的通信)，结果最终以标准 HTML 页面的形式发回给用户。

另外，提供服务的计算机所运行的操作系统必须侦听这些 SOAP 请求。Microsoft 希望此操作系统是 Windows，但是并非一定要 Windows，任何可以侦听 TCP/IP 套接字端口并能从端口读/写字节的操作系统都能胜任。在不久的将来，移动电话、呼机、汽车、微波炉、冰箱、电视机、立体音响设备、游戏控制台，以及其他各种设备都可以参与到这个新世界中来。

在客户端或应用程序端，操作系统必须能够支持读写套接字端口的功能以发布服务请求。客户端计算机必须支持用户的应用程序所期望的任何特性。如果用户的应用程序想要创建一个窗口或菜单，



那么操作系统必须提供此功能，否则应用程序的开发人员必须手动实现。当然，Microsoft 希望人们可以编写出可充分利用 Windows 所提供的丰富功能的应用程序。不过，需要重申的是：推荐使用 Windows，但不是必须使用它。

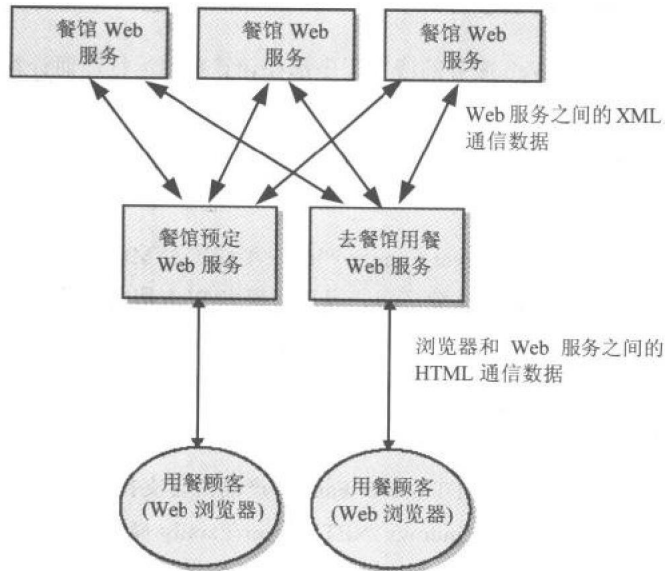


图 0.2 XML Web 服务示例

我想要说的是：无论 Microsoft 是否是参与其中，这个新世界都将实现。Microsoft 的 .NET 计划的目的就是让开发人员方便地创建和访问这些服务。

现在，我们可以编写出自己的操作系统，创建自定义 Web 服务器，去侦听和手动处理 SOAP 请求。但是这太费时间、耗精力了。Microsoft 替我们完成了这些艰难的工作，我们只需借助 Microsoft 的努力，来简化我们的开发任务。现在，作为应用程序开发人员，我们可以将精力集中在商务逻辑和服务上，把所有的通信协议和钻研工作都留给 Microsoft (它有大量的开发人员，非常乐意做这些深奥琐碎的工作)。

## Microsoft .NET 计划

许多年来，我都和 Microsoft 及其技术打交道。我目睹了 Microsoft 引入各种各样的新技术和创意：MS-DOS、Windows、Windows CE、OLE、COM、ActiveX、COM+、Windows DNA 等。当我第一次听到 Microsoft 的 .NET 计划时，我惊讶于 Microsoft 的神话依然坚不可摧。我认为，Microsoft 是有远见且有计划的组织，全体员工众志成城，共同实现这一宏伟计划。

将 Microsoft 的 .NET 平台与 ActiveX 进行对照，ActiveX 实际上只是为陈旧的 COM 起的一个新名字。再将 Microsoft 的 .NET 计划和 Windows DNA (分布式 Internet 结构) 对照，后者是另外一种市场营

销标签，被 Microsoft 贴在现有的很多技术上。但是，我更相信 Microsoft .NET，为了证实这一点，我写了这本书。下面简单介绍一下 Microsoft .NET 计划的组成部分。

## 基础操作系统：Windows

因为这些 Web 服务和使用 Web 服务的应用程序都在计算机上运行，而计算机又都有外围设备，所以，我们仍然离不开操作系统。Microsoft 推荐使用 Windows。而且，Microsoft 还将 XML Web 服务的特定功能加入到 Windows 系列操作系统，Windows XP 和 Windows .NET Server Family 中的服务器最适合这种崭新的服务。

Windows XP 和 Windows .NET Server Family 产品中集成了对 Microsoft .NET Passport XML Web 服务的支持。Passport 是一种服务，用来验证用户身份。很多 Web 服务都要对客户进行身份验证以确保信息访问的安全。当用户登录到一台运行 Windows XP 或 Windows .NET Server Family 的服务器时，他们就已登录到用 Passport 进行身份验证的 Web 站点和 Web 服务。这意味着，用户无需在访问不同的站点时，再次输入用户名和密码。

另外，Windows XP 和 Windows .NET Server Family 产品内置支持加载和执行那些实现 .NET 框架的应用程序。最后，Windows XP 和 Windows .NET Server Family 操作系统还有一种全新的可扩展的即时消息通知应用程序。此程序允许第三方供应商(如 Expedia 和 United States Postal Service 等)与用户无缝通信。例如，用户可以在他们的航班晚点以及有包裹时接到通知，前者来自 Expedia，后者来自 U.S. Postal Service。

## 有益的产品：.NET 企业级服务器

作为 .NET 计划的一部分，Microsoft 提供了几种产品，公司可以选择适合自己的商务逻辑(服务)的产品。这里列举了部分 Microsoft 的企业级服务器产品：

- Microsoft Application Center 2000
- Microsoft BizTalk Server 2000
- Microsoft Commerce Server 2000
- Microsoft Exchange 2000
- Microsoft Host Integration Server 2000
- Microsoft Internet Security and Acceleration (ISA) Server 2000
- Microsoft Mobile Information Server 2002
- Microsoft SQL Server 2000

很可能这些产品的名称最终都会加上“.NET”，用于市场推广。但是，我相信，随着时间的推移，会有更多的 .NET 特性集成进来。

## Microsoft XML Web 服务：.NET My Services

当然，Microsoft 并不希望只局限于提供一些基础技术，Microsoft 希望自己也能参与到这个新世界中。所以，Microsoft 正在构建自己的 XML Web 服务：一些免费，另外一些则需要支付一定的使用费。Microsoft 最初计划提供以下 .NET My Services：

- .NET Alerts
- .NET ApplicationSettings
- .NET Calendar
- .NET Categories
- .NET Contacts
- .NET Devices
- .NET Documents
- .NET FavoriteWebSites
- .NET Inbox
- .NET Lists
- .NET Locations
- .NET Presence
- .NET Profile
- .NET Services
- .NET Wallet

这些面向个人的 XML Web 服务称之为 Microsoft 的 .NET My Services。可以在 <http://www.Microsoft.com/MyServices/> 上找到与此相关的更多信息。Microsoft 将推出更多的个人服务，而且正在创建面向公司的 XML Web 服务。

除这些公共的 Web 服务之外，Microsoft 还将创建一些内部服务，用于销售数据和记账。只有 Microsoft 的雇员才能访问这些内部服务。我希望各家公司都能迅速采纳这种观点，在企业内部网上使用 Web 服务，使全体雇员可以使用公司内部信息。公开的 Internet Web 服务和使用它们的应用程序的实现将是一个漫长的过程。

## 开发平台：.NET 框架

某些 Microsoft .NET My Services(如 Passport)目前已经在使用中。这些服务在 Windows 上运行，是使用如 C/C++、ATL、Win32、COM 之类的技术创建的。随着时间的推移，这些服务和新服务将最终用更新的技术来实现，如 C#(发音为“C sharp”)和 .NET 框架。



### 要点

尽管通篇介绍的都是创建 Internet 应用程序和 Web 服务，但 .NET 框架的功能远不止

这些。总的来说，.NET 框架开发平台允许开发人员构建以下种类的应用程序：XML Web 服务、Web 窗体、Win32 GUI 应用程序、Win32 CUI(控制台 UI)应用程序、服务(由服务控制管理器控制)、实用工具以及独立组件。本书提供的材料适用于上述所有应用程序类型。

.NET 框架由两部分组成：公共语言运行库(CLR)和框架类库(FCL)。NET 框架有助于简化服务和应用程序的开发。而且，这也是本书的重点：如何开发.NET 框架的应用程序和 XML Web 服务。

最初，Microsoft 计划使 CLR 和 FCL 可用于各种 Windows 版本，包括 Windows 98、Windows 98 第 2 版、Windows Me、Windows NT 4、Windows 2000、32 位和 64 位的 Windows XP，以及 Windows .NET Server Family。NET 框架的“简装”版，称为.NET 压缩框架，可用于 PDA(比如 Windows CE 和 Palm)以及家电(小型设备)。在 2001 年 12 月 13 日，欧洲计算机制造商协会(European Computer Manufacturers Association, ECMA)将 C#编程语言，部分 CLR 以及部分 FCL 作为标准。不久，上述技术的 ECMA 兼容版本将出现在广大操作系统和 CPU 上。



#### 注意

Windows XP(包括家庭版和专业版)未附带.NET 框架。然而，Windows .NET Server Family(Windows .NET Web Server、Windows .NET Standard Server、Windows .NET Enterprise Server 和 Windows .NET Datacenter Server)将包含.NET 框架。实际上，这也是 Windows .NET Server Family 名称的由来。Windows 的下一个版本(代号为 Longhorn)的所有版本都将包含.NET 框架。目前，只能在您的应用程序中重新发行.NET 框架，而且安装程序必须安装.NET 框架。Microsoft 其实已经开发了一套.NET 框架的重新发行文件，允许您在应用程序中随意发行：<http://go.microsoft.com/fwlink/?LinkId=5584>。

几乎所有的程序员都熟悉运行库和类库。至少很多人都使用过 C 运行库、标准模板库(ATL)、Microsoft 基础类库(MFC)、活动模板库(ATL)、Visual Basic 运行库或 Java 虚拟机。实际上，Windows 操作系统本身就可以当作一个运行引擎和库。运行引擎和库向应用程序提供服务，这为程序员节省了重复开发同一算法的时间。

Microsoft .NET 框架使开发人员更喜欢使用现在的(而不是早期的)开发平台技术。尤其是，.NET 框架确实实现了代码重用、代码规范化、资源管理、多语言开发、安全、部署和管理。在设计此新平台时，Microsoft 认为有必要改善目前的 Windows 平台的某些缺陷。以下是 CLR 和 FCL 提供的部分功能。

- **一致的编程模型** 现在的操作系统的某些功能是通过动态链接库(DLL)访问的，另一些功能则通过 COM 对象访问，而 CLR 和 FCL 使所有的应用程序服务都是通过一个公用的面向对象的编程模型访问的。
- **简化编程模型** CLR 致力于简化 Win32 和 COM 所需的基础结构。尤其是，CLR 使开发人员无需理解以下概念：注册表、全局唯一标识符(GUID)、IUnknown、AddRef、Release、

HRESULT 等。CLR 使开发人员无须了解这些概念，这些概念并不以 CLR 的任何形式存在。当然，如果想编写一个与现有的非.NET 框架代码交互的.NET 框架应用程序，则必须熟悉这些概念。

- **一次运行，总能运行** 所有 Windows 开发人员都知道“DLL hell”版本控制问题。当新安装的应用程序的组件覆盖旧有的应用程序的组件时，导致旧应用程序无法正常运行或停止运行。现在的.NET 框架的结构已与应用程序组件隔离，这样，应用程序就总要加载它在生成和测试时所用的组件。如果应用程序安装后可以运行，那么，该应用程序应该是可以运行。此举将“DLL hell”拒之门外。
- **简化部署** 现在的 Windows 应用程序很难安装和部署。需要创建若干文件、注册表设置和快捷方式。而且，彻底卸载一个应用程序几乎是不可能的。在 Windows 2000 中，Microsoft 引入了一种全新的安装引擎，有助于解决这些问题，但是，授权的 Microsoft 安装程序包仍然可能无法正确处理每件事情。.NET 框架力求消灭这些问题。.NET 框架组件(简称为类型)未被注册表引用。实际上，安装大多数.NET 框架应用程序只是将文件复制到某个目录，将快捷方式添加到【开始】菜单、桌面或快速启动栏中。卸载应用程序像删除文件一样简单。
- **广泛的平台支持** 在编译.NET 框架的源代码时，编译器生成的是通用中间语言(CIL)，而不是传统的 CPU 指令。运行时，CLR 将 CIL 翻译成本机 CPU 指令。因为这种转换是在运行时完成的，因此用于主机 CPU。这意味着可以在任何一台运行与 ECMA 兼容的 CLR 和 FCL 版本的机器上部署.NET 框架应用程序。这些机器可以是 x86、IA64、Alpha、PowerPC 等。如果用户更换计算硬件或操作系统，就可以立即感受到这种广泛支持的好处。
- **编程语言集成** COM 允许各种编程语言互操作。.NET 框架允许这些语言相互集成，这样，您可以把其他语言的类型当作您所用语言的类型来使用。例如，CLR 可以在 C++中创建一个从 Visual Basic 实现的类中派生的类。CLR 之所以能做到这一点，是因为它定义并提供了一个公共类型系统(CTS)，所有针对 CLR 的编程语言都必须使用这个系统。公共语言规范(CLS)描述了编译器实现者要将自己的语言很好地与其他语言集成所必须遵守的规则。Microsoft 提供了几种编译器，它们生成的代码都是针对运行库的，这些编译器是：C++托管扩展、C#、Visual Basic .NET(现在又细分为 Visual Basic Scripting Editio，简称 VBScript，和 Visual Basic for Applications，简称 VBA)和 JScript。另外，除 Microsoft 之外的一些公司和学术机构都开发了针对 CLR 的其他语言的编译器。
- **简化代码重用** 可以使用前面提到的机制创建自己的类，为第三方应用程序提供服务。这使得代码重用极为简单，从而也给组件(类型)供应商提供了巨大的市场。
- **自动内存和管理(垃圾回收)** 编程需要大量技巧和规则，尤其是在管理诸如文件、内存、屏幕空间、网络连接、数据库资源之类的资源时尤为明显。很常见的一个错误就是忘记释放这些资源，最终导致应用程序在某个未知时刻不正常执行。CLR 可自动跟踪资源使用，确保应用程序永远不会泄漏资源。实际上，没有办法可以显式“释放”内存。第 19 章将详细解释垃圾回收的工作原理。

- **类型安全验证** CLR 可以验证所有代码是否类型安全。类型安全能确保总是以兼容的方式访问被分配对象。所以，如果方法输入参数被声明为接受 4 字节的值，那么 CLR 将会检测并防止此参数被当作 8 字节值访问。同样，如果一个对象占有 10 字节的内存空间，应用程序就不可能使该对象出现允许读出 10 个以上字节的形式。类型安全还意味着执行流只会传输到众所周知的位置(即，方法入口点)。没有任何方法可以做到构造某个对内存位置的任意引用并使该位置的代码开始执行。总而言之，这些措施确保类型安全可以消除许多常见的编程错误和常规的系统攻击(例如，利用缓存超限运行)。
- **丰富的调试功能** 因为 CLR 适用于很多编程语言，所以，可以方便地采用某种最适合特定任务的语言来实现应用程序的某一部分。CLR 完全支持跨语言调试应用程序。
- **一致的方法故障范例** Windows 编程中最让人头痛的一个方面就是报告故障的函数的风格不一致。一些函数返回 Win32 状态码，一些返回 HRESULT，还有一些则引发异常。在 CLR 中，所有故障都是通过异常——period——报告的。异常允许开发人员将故障修复代码和完成此项工作所需的代码隔离开。这么做大大简化了代码的读写与维护。另外，异常是跨模块和编程语言工作的。而且，与状态码和 HRESULT 不同的是，异常不能被忽略。CLR 还提供了内置的穿越堆栈的功能，使得查找错误和故障变得更加简单。
- **安全性** 传统的操作系统安全提供了基于用户账户的隔离和访问控制。该模型很有用，但是它的核心是假定所有代码都具有相同的可信任性。这一假设只有当所有代码都是从物理介质(如 CD-ROM)或可信的公司服务器安装时才有意义。但是，随着对移动代码(如 Web 脚本)、Internet 上下载的应用程序、电子邮件附件等的依赖，我们需要以代码更加集中的方式来控制应用程序的行为。代码访问安全提供了这方面的解决途径。
- **互操作性** Microsoft 意识到开发人员手头已经拥有大量的代码和组件。为了充分利用 .NET 框架平台的技术优势而重新改写这些代码将影响该平台的普及速度。所以，.NET 框架完全支持开发人员访问现有的 COM 组件，以及调用现有 DLL 中的 Win32 函数。

用户不会马上接受 CLR 及其功能，但是他们肯定会发现使用 CLR 的应用程序的品质和特性。至少，用户和公司会欣赏 CLR 更加快捷方便地开发和部署应用程序的能力，这是以前的 Windows 所不可企及的。

## 开发环境：Visual Studio .NET

最后要讲述的 .NET 的组成部分就是 Visual Studio .NET。Visual Studio .NET 是 Microsoft 的开发环境。Microsoft 多年来一直致力于此，并且在其中结合了大量的 .NET 框架特性。Visual Studio .NET 可以运行在 Windows NT 4、Windows 2000、Windows XP 和 Windows .NET Server Family 服务器上，并将运行在以后的 Windows 版本上。当然，Visual Studio .NET 生成的代码可以运行在上述所有 Windows 平台上，另外还有 Windows 98、Windows 98 第 2 版和 Windows Me。

和任何一种深受好评的开发环境一样，Visual Studio .NET 包括项目管理器，源代码编辑器，UI

设计器，大量向导、编译器、链接器、工具和实用程序、文档，以及调试器。它支持在 32 位和 64 位的 Windows 平台，以及 .NET 框架平台上生成应用程序。另外很重要的一点改进是：所有编程语言仅使用一个集成开发环境。

Microsoft 还提供了 .NET 框架 SDK。这个免费的 SDK 包括所有的语言编译器、大量工具和文档。使用 SDK，可以在不使用 Visual Studio .NET 的情况下开发出 .NET 框架的应用程序。只需使用自己的编辑器和项目管理系统，也不必生成拖放的 Web 窗体和 Windows 窗体。我经常使用 Visual Studio .NET，并且整本书都将提到它。然而，本书主要是大致讲述编程，所以在学习、使用、理解每章的概念时，无需学习 Visual Studio .NET。

## 本书目标

本书的目的是解释如何开发 .NET 框架应用程序，着重讲述 CLR 的工作原理以及它所提供的功能，还将讨论 FCL 的各个组成部分。没有一本书可以全面解释 FCL——它几乎包含成千上万的字节，而且该数字正以惊人的速度增长。本书将集中讲解每个开发人员都需要知道的核心类型。尽管本书并非专门针对 Windows 窗体、XML Web 服务、Web 窗体等，但是本书提出的各项技术都适合于上述所有的应用程序类型。

本书并不是讲述特定编程语言知识的教材。读者应熟悉诸如 C++、C#、Visual Basic 或 Java 之类的编程语言，还应了解面向对象编程的概念，如数据抽象、继承和多态(polymorphism)。理解这些概念是至关重要的，因为所有 .NET 框架特性都是通过面向对象的范例提供的。如果您不熟悉这些概念，最好先找一本讲解这些概念的书看看。

尽管本书的宗旨不是讲解基本的编程知识，但仍然涉及各种 .NET 框架编程话题。所有 .NET 框架开发人员都必须意识到这些话题，它们贯穿于全书之中。

本书是一本有关 .NET 框架的公共语言运行库的书，而不是针对某种特定编程语言的编程技术。然而，本书将提供大量代码示例说明工作原理。如果对编程语言不甚了解，这些例子中我认为最适合我的语言是 IL(intermediate language, 中间语言)汇编语言。IL 是 CLR 理解的惟一编程语言。所有的语言编译器都将源代码编译成 IL，IL 再被 CLR 处理。通过 IL，可以访问 CLR 提供的各种特性。

然而，使用 IL 汇编语言是一种相当低级的编程方式，不是说明编程概念的理想途径。所以，我决定在整本书都选用 C#编程语言。因为它是 Microsoft 专门设计用来开发 .NET 框架代码的语言。如果您不想使用 C#编写项目，也没什么问题，您只要能读懂 C#就可以了。

## 系统要求

.NET 框架可以安装在 Windows 98、Windows 98 第 2 版、Windows Me、Windows NT 4(所有版

本)、Windows 2000(所有版本)、Windows XP(所有版本), 以及 Windows .NET Server Family 服务器。您可以从 <http://go.microsoft.com/fwlink/?LinkId=5584> 下载 .NET 框架。

.NET 框架 SDK 和 Visual Studio .NET 的系统要求是 Windows NT 4(所有版本)、Windows 2000(所有版本)、Windows XP(所有版本), 以及 Windows .NET Server Family 服务器。可以从 <http://go.microsoft.com/fwlink/?LinkId=77> 下载 .NET 框架 SDK。当然, 您必须购买 Visual Studio .NET。

您可以从 <http://www.Wintellect.com> 下载与本书相关的代码。

## 勘误信息

作者和编辑都希望带给您最准确、最新、有深度、通俗易懂、无误的信息。但即使是最神奇的团队, 也有可能出现疏漏。如果您在本书中发现任何错误, 请通过 <http://www.wintellect.com> 网站获得下载资源或更正信息。

## 支持信息

我们尽力确保本书的准确性。Microsoft Press 通过以下地址提供本书的勘误信息：  
<http://www.microsoft.com/mspress/support/>

如果要直接连入 Microsoft Press 的知识库, 输入一个与您的疑问相关的查询, 请访问:

<http://www.microsoft.com/mspress/support/search.asp>

如果您有与本书相关的评论、问题或意见, 请通过以下两种途径发送给 Microsoft Press:

通信地址:

Microsoft Press

Attn: Applied Microsoft .NET Framework Programming Editor

One Microsoft Way

Redmond, WA 98052-6399

电子邮箱:

MSPINPUT@MICROSOFT.COM

请注意, 上述邮件地址不提供产品支持。如果要获取有关 C#、Visual Studio 或 .NET 框架相关的支持信息, 请访问 Microsoft 产品标准支持 Web 站点: <http://support.microsoft.com>。



# 目 录

前言 .....	ix
----------	----

## 第 I 部分

### Microsoft .NET 框架的基本原理

<b>第 1 章 .NET 框架开发平台的体系结构</b> .....	<b>3</b>
1.1 将源代码编译成托管模块 .....	3
1.2 将托管模块组合为程序集 .....	5
1.3 加载公共语言运行库 .....	7
1.4 执行程序集代码 .....	9
1.5 .NET 框架类库 .....	15
1.6 通用类型系统 .....	18
1.7 公共语言规范 .....	19
1.8 非托管代码的互操作性 .....	23
<b>第 2 章 生成、打包、部署和管理应用程序及类型</b> .....	<b>26</b>
2.1 .NET 框架的部署目标 .....	26
2.2 将类型加入到模块中 .....	27
2.3 将模块组合为程序集 .....	34
2.3.1 使用 Visual Studio .NET IDE 将程序集添加到项目中 .....	39
2.3.2 使用程序集链接器 .....	40
2.3.3 在程序集中包括资源文件 .....	42
2.4 程序集版本资源信息 .....	42
2.5 区域性 .....	47
2.6 简单应用程序部署(私有部署 的程序集) .....	48
2.7 简单管理控件(配置) .....	49

<b>第 3 章 共享程序集</b> .....	<b>53</b>
--------------------------	-----------

3.1 两种程序集、两种部署方式 .....	54
3.2 给程序集起一个强名称 .....	54
3.3 全局程序集缓存 .....	59
3.4 生成一个引用强命名程序集 的程序集 .....	65
3.5 强命名程序集可以防篡改 .....	67
3.6 延迟签名 .....	68
3.7 秘密部署强命名程序集 .....	71
3.8 并行执行 .....	72
3.9 运行库如何解析类型引用 .....	73
3.10 高级管理控件(配置) .....	76
3.11 修复存在错误的应用程序 .....	82

## 第 II 部分

### 使用类型和公共语言运行库

<b>第 4 章 类型的基本原理</b> .....	<b>87</b>
----------------------------	-----------

4.1 所有的类型都从 System.Object 派生 .....	87
4.2 类型间强制转换 .....	89
4.3 命名空间和程序集 .....	92

<b>第 5 章 基元、引用和值类型</b> .....	<b>97</b>
------------------------------	-----------

5.1 编程语言的基元类型 .....	97
5.2 引用类型和值类型 .....	102
5.3 装箱和未装箱值类型 .....	107

<b>第 6 章 通用对象运算</b> .....	<b>117</b>
---------------------------	------------

6.1 对象相等和恒等 .....	117
-------------------	-----