精通.NET核心技术
——原理与构架

刘晓华　　编著
飞思科技产品研发中心　　监制

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书是.NET 核心技术的基础入门篇，全书共分 19 章，随书所附光盘包含书中的全部实例源代码。

本书首先扼要介绍了整体的.NET 框架，然后介绍了.NET 的基本类库、基本类型的操作，接下来介绍了.NET 程序设计的核心要素也是开发人员最大的学习难点：程序集。围绕程序集讲解了其创建、反射机制、引用、版本控制、资源、特性编程等问题。最后围绕.NET 中的伪进程应用域和线程这两个基本概念，详细介绍了.NET 中进程通信、多线程、异步调用等关键技术。此外本书还有若干章节介绍了.NET 框架提供的服务，包括异常处理、特性编程、垃圾回收、多语言编程、安全性等。

全书内容深刻，实例丰富，是广大程序设计人员学习和深入掌握.NET 技术的必备参考读物。

图书在版编目（CIP）数据

精通.NET 核心技术——原理与构架/刘晓华编著.－北京：电子工业出版社，2002.8
（精通系列）
ISBN 7-5053-7768-X

Ⅰ.精... Ⅱ.刘... 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 045569 号

责任编辑：杨章玉　李志强
印　　刷：北京市增富印刷有限责任公司
出版发行：电子工业出版社　http://www.phei.com.cn
　　北京海淀区万寿路 173 信箱　邮编：100036
经　　销：各地新华书店
开　　本：787×1092　1/16　印张：53.75　字数：1376 千字　附光盘 1 张
版　　次：2002 年 8 月第 1 版　2002 年 8 月第 1 次印刷
印　　数：5000 册　　定价：79.00 元（含光盘）

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系调换。联系电话：（010）68279077

出版说明

“精通”系列是电子工业出版社经典的技术精品丛书，一直受到广大读者，特别是计算机专业技术人员的关注。在这些专业人士的支持和鼓励下，“精通”系列已经成为一个标杆，反映出目前国内外最新计算机技术的动态和发展方向。

“精通”系列中的每部著作完全是作者的呕心之作，代表了作者在该领域的最高成就，集成了作者多年的理论和实践经验，凸显了作者为计算机技术的发展做总结和展望的写作初衷。丛书的作者都是由著名高校的学科带头人、国际上知名的教授学者、权威的计算机专业人士和业界的集大成者组成。他们的知识结构、理论和实践体系有着突出的卓越之处：

- 站在技术的最前沿
- 有最深刻的理论基础
- 实践环境具有广泛的代表性和权威性
- 结论的指导价值

因此，这样雄厚的写作班子保证了本系列丛书的高层次、高质量和高品质，也足以满足国内读者的高品位、高需求和高要求。除了作者之外，审校者同样实力超群，他们从理论的角度、读者需求的角度、技术成熟度的角度等不同的侧面，为作者提出了大量的参考意见和修改建议，使得每部著作的结构更坚实、体系更完整、理论更完善、针对性更强。

电子工业出版社计算机研发部本着服务于读者、服务于科技的精神，在选题上精益求精，综合考虑和平衡了目前技术上的热点、未来发展的重点以及潜在读者需求的卖点等多方面因素，精心推出“精通”系列，并将不断进行补充。

当然，我们的努力与读者的关怀是分不开的，衷心地欢迎读者提出意见和建议，促使我们推出更多、更好的精品书，以飨读者。我们的联系方式：

电　话：　（010）68134545　68134811

E-mail：　support@fecit.com.cn

网　址：　http://www.fecit.com.cn　http://www.fecit.net

通用网址：　计算机图书、FECIT、飞思教育、飞思科技、飞思

电子工业出版社计算机研发部

关 于 飞 思

新世纪之初的北京，一群满怀共同理想的年轻人聚集在飞思教育产品研发中心的旗帜下，他们将新的希望和活力注入了中国 IT 教育产品开发领域。飞思人在为把自己打造成为中国 IT 教育产品研发的精英团队而更加不懈努力。

21 世纪的今天，飞思人在多元化教育产品的开发和出版等方面已经迈出了坚实的第一步，开拓出属于自己的一片天空，初步赢来了涓涓细流。

如今，本着教育为科技服务的宗旨，飞思教育产品研发中心拓展为飞思科技产品研发中心，并以崭新的面貌等待您的支持与关注。

飞 思 人 理 念

我们经常感谢生活的慷慨，让我们这些原本并不同源的人得以同本，为了同一个梦想走到一起。

因为身处科技教育前沿，我们深感任重道远；因为伴随知识更新节奏，我们一刻不敢停歇。虽然我们年轻，但我们拥有：

“严谨、高效、协作”的团队精神

全方位、立体化的服务意识

实力雄厚的作者群和开发队伍

当然，最重要的是我们拥有：

恒久不变的理想和永不枯竭的激情和灵感

正因如此，我们敢于宣称：

飞思教育＝丰富的内容+完美的形式

这也是你和我共同精心培育的品牌 

的承诺。

“问渠哪得清如许，为有源头活水来”。路再远，终需用脚去量；风景再美，终需自然抚育。

年轻的飞思人愿为清风细雨、阳光晨露，滋润您发芽，成长；更甘当坚实的铺路石，为您铺就成功之路。

前言

关于.NET

.NET 是微软公司下一代的计算计划。其目标是把整个 Internet 整合为一个可计算的统一网络。

对开发人员而言，.NET 是一个完美的开发平台。它提供了一套公共的运行库，并制定了一套公共语言规范。所有符合该规范的语言都可以无缝使用这套运行库。在.NET 平台下，除了语法上的区别以外，各种编程语言没有本质的不同。它们共享公共类库，具有类似的编程模型和相差无几的功能。开发人员可以自由选择自己喜爱的语言开发程序。.NET 平台提供大量的服务，包括垃圾自动收集、面向对象的多线程、基于程序集的部署、异常处理、特性编程、远程处理、ASP.NET 网页框架、互操作、安全性等，使开发人员可以快速构架任何应用，从包括传统的桌面应用到面向 Web 的大型分布式应用。.NET 将彻底改变软件的开发方式、使用方式和发行方式，.NET 将是一场软件革命。

学习建议

.NET 是一个庞大的体系。它提供了多种编程语言，为开发各种.NET 应用提供了全面的、完善的服务。对.NET 了解越深入，就越能体会.NET 的优越性——随心所欲地挑选编程语言，轻松地利用基础类库开发出最前沿的各种应用：控制台应用、基于 Win Form 的窗口应用、基于 Web Form 的 ASP.NET 应用、Web 服务、远程处理等。但.NET 这种强大的功能和无与伦比的优势往往让初学者感到困惑。下面的若干建议，可能对初学者有些帮助：

选择自己喜爱的编程语言。.NET 平台提供了非常多的编程语言供选择。但在.NET 平台下，不同的编程语言除了语法上有一些区别之外，功能上的区别将很小。它们共享.NET 框架的基础类库来实现各种应用。对已经有 Visual Basic 编程经验的人员来讲，可以考虑选择 Visual Basic.NET。因为已经获得经验，可以帮助加快熟悉 Visual Basic.NET 的语法。如果读者已有 Java/C++ 编程经验，可以考虑选择 C#（读做 C Sharp），因为 C#的语法与 Java/C++的语法很相像。本书的绝大部分例子，都提供了 Visual Basic 和 C#的两种实现。读者可以根据选用的编程语言阅读合适的版本。

熟悉命令行工具。精心设计的 Visual Studio.NET 集成开发环境使用户能够快速构建.NET 应用。但对初学者而言，最重要的可能不是如何高效地开发应用，而是了解集成开发工具做了哪些工作以及是如何做的。这时最好的办法就是用.NET 的命令行工具手工实现相关任务。其实最重要的命令行工具就是编译命令。每种语言都有一个编译命令，例如 Visual Basic.NET 的编译命令是 vbc.exe，C#的编译命令是 csc.exe 等，它负责将特定语言编写的代码编译成程序集。其他比较重要的命令行工具还有 sn.exe（制作强名程序集时要用到）、al.exe（将模块装配成程序集）等。为了增强学习效果，读者可以先用集成开发工具实现，然后用命令行工具再做

一遍。例如，本书所举的例子代码，笔者一般是在集成开发环境下创建并编译通过，然后经过适当修改，用命令行再编译执行的。

重点掌握程序集和应用域。笔者认为，对程序设计师理解.NET 框架而言，程序集和应用域处于核心地位。掌握.NET 的第一个层次是了解.NET 本身，包括其框架构成、公共类型、框架类库等，特别要注意了解那些最常用的类型及其操作，例如字符串、数组、集合等；第二个层次就是理解.NET 中的核心元素，包括程序集、应用域、线程、命名空间等，而程序集和应用域则是这一阶段的重点和难点；第三个层次就是掌握.NET 框架提供的一些核心服务，如跨语言编程、异常处理、安全性等；第四个层次就是熟悉.NET 的高级特性，例如远程处理框架、互操作服务、ADO.NET 数据访问、GDI+绘图等。而第二个层次的学习效果将直接影响到读者对.NET 的理解程度。读者可以判断自己对.NET 的学习达到何种层次，再从本套书中选择合适的内容。

关于本套书

本套书分为《精通.NET 核心技术——原理与构架》与《精通.NET 核心技术——高级特性》两册，涵盖了.NET 核心技术的各个层面，为开发人员提供了完整的知识架构，无论是开发何种.NET 应用的开发人员，都可以在这套书中找到详实的技术参考。一旦通晓本书所介绍的内容，读者自会有 “会当凌绝顶，一览众山小”的感觉。

本套书具有以下特色：

- 迎难而上　本套书披露的内容是.NET 的精华所在，但同时又是一些难啃的硬骨头，掌握起来又有相当难度。单靠自己摸索，开发人员必将付出相当多的宝贵精力。本套书反映了作者本人对.NET 技术的艰难摸索的过程，大多数是对某些问题深度思考后豁然开朗所得到的启示的总结。自从去年 3 月，笔者就开始追踪.NET 技术，在摸索的过程中付出了相当多的热情和精力，逐渐领悟了.NET 的精髓。在艰难的探索过程中，有问题未解时的困惑，更有解开疑团后的云淡风轻。
- 循序渐进　.NET 所涉及的内容非常丰富。面对众多的新概念、新技术，不少读者会有茫然不知从何入手的感觉。本套书精心安排了章节顺序，以尽可能地符合.NET 学习曲线。对于新手，遵循本书安排的阅读顺序将把学习障碍减到最小。
- 手工打造　集成开发环境能简化应用程序的开发，但也会使开发人员不去思考简化背后的事情，从而妨碍对某些关键技术的理解。相反，通过命令行工具进行程序开发，包括编写源代码、编译等，程序员可能要做更多的工作，但更灵活，并就学习一种技术而言，这种方式是可取的。本书穿插讲解了大量的.NET 命令行工具的用法，通过本书的学习，读者可以掌握常用的命令行工具。另外，为了彻底帮助大家理解某些“技术内幕”，本套书还提供了若干技术框架的手工实现。在清晰的论证和详细的实例分析前，读者的一切疑惑将一扫而光。

关于本书

本书是.NET 核心技术的基础入门篇，首先扼要介绍了整体的.NET 框架，然后介绍

了.NET 的基本类库，接下来介绍了.NET 程序设计的核心要素也是开发人员的最大的学习难点：程序集。围绕程序集讲解了其创建、反射机制、引用、版本控制、资源、特性编程等问题。一旦掌握了程序集，读者对.NET 的掌握将上升到一个新的境界。最后带领开发人员熟悉.NET 中的伪进程应用域和线程。围绕这两个基本概念，详细介绍了.NET 中进程通信、多线程、异步调用等关键技术。本书还包括其他一些介绍.NET 框架提供服务的相对独立的章节。例如异常处理、特性编程、垃圾回收、多语言编程、安全性等内容。随书所附光盘包含书中的全部实例源代码。

本书面向广大程序设计人员，适合于作为学习和深入掌握.NET 技术的参考读物，指导中高级技术人员进行开发工作。

本书由飞思科技产品研发中心策划并组织编写，刘晓华编著，李华、李书德、张宏伟、薛德军、李国志、肖云、卢红娜、林军、万军等人也参加了本书的写作工作。杨艳女士认真阅读了本书的初稿，并提出了部分重要的参考意见。郭晶女士对本书的内容选材提出了很好的建议。范同祥、李志强、杨章玉先生对本书做了大量的文字处理工作，也使本书增色不少，在此表示衷心的感谢。

笔者在写作过程中，参考了微软公司的相关资料，在此特做说明。限于作者水平，加之时间仓促，书中不足之处难免，敬请读者批评指正。

我们的联系方式：

电　　话：　（010）68134545　68134811

E-mail：　support@fecit.com.cn

网　　址：　Http://www.fecit.com.cn　http://www.fecit.net

通用网址：　FECIT、飞思、飞思科技、飞思教育、计算机图书

编　　者

目录

第 1 章　.NET 和.NET 框架概览

在本章里，我们将详细讨论微软的.NET 和.NET 框架的组成、特征和优势。通过本章的学习，读者将获得对.NET 和.NET 框架的整体认识。下面是本章重点讲述的内容：

- .NET 组成及优势
- .NET 框架的构成
- 公共语言运行库的主要功能
- .NET 框架类库的特点和优势
- 一致的编程模型
- .NET 框架支持的应用程序的类型

1.1　什么是.NET

.NET 是为简化在第三代因特网的分布式环境下的应用程序开发，基于开放互联网标准和协议之上，实现异质语言和平台高度交互性而构建的新一代计算和通信平台。也是 Microsoft 以服务的方式递交软件的一种策略。它主要由如下三部分构成：

- .NET 框架
- Web 服务
- .NET 企业服务器

这些部件一起提供了按照用户的需要创建 Web 的方法。图 1-1 展示了.NET 框架的部件构成。

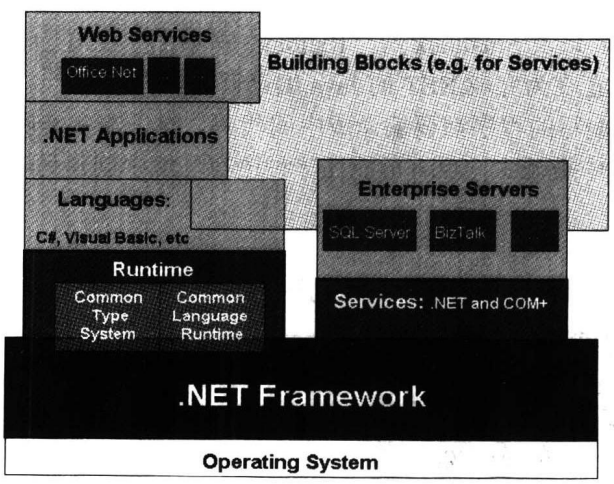

图 1-1　.NET 框架

下面对.NET 的各个部分进行简单说明。

1. 操作系统

微软宣称.NET 平台将独立于操作系统。不仅视窗系列支持.NET 平台，将来 Linux，UNIX 也将支持.NET 平台。为了实现跨平台的战略目标，.NET 所有语言编写的应用不是转化为本地代码，而是编译为微软中间代码，即 Microsoft Intermediate Language，简记为 MSIL。它将立即由 Just in Time（JIT）编译器转换成机器代码。在此过程中，CLR 的角色基本上和 Java 平台中的 JVM 相似。只要实现特定平台的.CLR，就可以在该平台上执行.NET 应用。

2. .NET 框架

这是位于操作系统之上的.NET 最重要的基础构架。它提供了创建、部署和运行.NET 应用的环境。图 1-1 中的 Runtime 和 Services 实质上是.NET 框架的一部分。

3. 编程语言

从本质上而言，.NET 就支持一种语言即 MSIL。因为任何源程序最终都要通过适当的编译器编译成 MSIL 代码。MSIL 是与 CPU 无关的指令集。它包含加载、存储、初始化和调用对象方法的指令。

要让.NET 平台支持一种编程语言，首先该语言必须满足公共语言规范（COMMAN LANGUANGE SPECIFICATION，CLS）。公共语言规范是一组结构和限制，用做库编写者和编译器编写者的指南。它使任何支持 CLS 的语言都可以完全使用库，并且使这些语言可以相互集成。公共语言规范是公共类型系统的子集。对于那些需要编写代码供其他开发人员使用的应用程序开发人员来说，公共语言规范也非常重要。如果开发人员遵循 CLS 规则来设计公共访问的 API，那么就可以在支持公共语言运行时的任何其他编程语言中很容易地使用这些 API。

如果一种语言满足 CLS，那么只要提供该语言的编译器后就可以在.NET 平台下使用该语言了。目前 Microsoft 在.NET 平台上提供了好几种语言以及相应的编译器，例如，C++，Jscript，Visual Basic.NET（也称为 VB 7）以及 C#。C#是随同.NET 出现的一种新语言。Microsoft 合作的第三方供应商正在为大量其他语言开发编译器，其中包括 Cobol，Eiffel，CAML，Lisp，Python 以及 Smalltalk 等。Rational 是著名的 UML 工具 Rose 的开发商，它也加入了这个队伍，正在为完成.NET 的 Java 编译器而努力。

所有.NET 语言编写的程序都要编译成中间代码，这一事实意味着用一种语言写的类或许会在另一种语言中被派生，也有可能在一种语言环境中创建另一种语言编写的类的实例。

4. 企业服务

目前微软.NET 企业服务器主要有如下几种，如图 1-2 所示。

- BizTalk Server 2000
- SQL Server 2000
- Exchange 2000 Server
- Host Integration Server 2000
- Internet Security and Acceleration Server 2000
- Application Center 2000

- Commerce Server 2000
- Mobile Information 2001 Server
- SharePoint Portal Server 2001

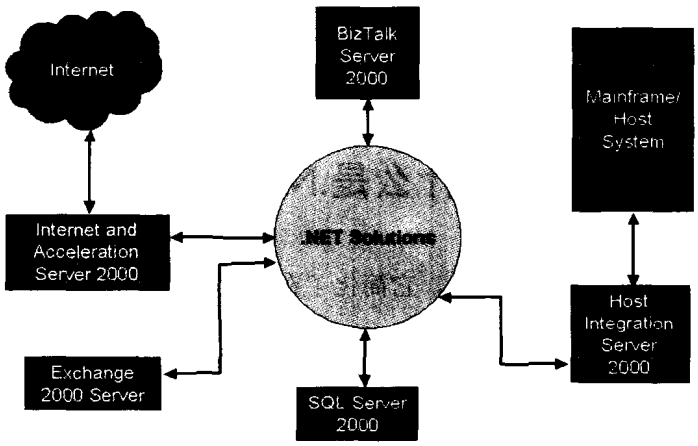

图 1-2　微软.NET 企业服务器

5．**构造块服务**（Building Blocks Service）

构造块服务是一个以用户为中心的 XML Web 服务集，它把对用户数据的控制权从应用程序转移给了用户。以用户同意作为所有事务的基础，实现跨应用程序、服务和设备的个性化的简单性和一致性。构造块服务包括 Passport（用于用户标识）和用于消息传递、文件存储、用户首选项管理、日历管理的服务以及其他功能。Microsoft 将提供对于 .NET 结构非常关键的构造块服务，广大的合作伙伴与开发人员将会大大地扩大构造块服务集合。

6．**.NET 应用**

在.NET 平台上，可以开发下面类型的.NET 应用。

- 控制台应用程序
- 脚本和宿主应用程序
- Windows Forms 应用程序（Windows 桌面 GUI 应用程序）
- ASP.NET 应用程序
- Web Services 应用程序
- Windows 服务程序

7．Web Services

构建在.NET 平台上交付最终用户使用的产品如 Office.NET 等。这些产品深刻体现了微软“软件就是服务”的思想。这样的 Web Services 又叫.NET 体验。

.NET 是一个有相当吸引力的战略平台，无论对开发者还是最终用户都是如此，因为.NET 有如下主要优点：

- 跨语言　.NET 支持多种语言的互操作，即在一种语言下开发的组件，可在另一组件下通过面向对象的继承而得以重用。
- 跨平台　.NET 通过将各语言先编译成中间语言（IL），然后再执行时用即时（Just In Time）编译器将之编译成本地平台代码来实现异构平台下对象的互操作，目

前.NET 支持的平台有 Windows，Linux 和 Unix 的支持正在开发中。不仅如此，将来甚至还会出现各种支持.NET 的智能终端。

- 安全 .NET 通过公共语言运行库来实现资源对象、类型的安全。
- 对开放互联网标准和协议的支持 .NET 通过对 HTTP，XML，SOAP，WSDL 等 Internet 标准的强劲支持提供在异构网络环境下获取远程服务，连接远程设备，交互远程应用的编程界面。

1.2 什么是.NET 框架

.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。.NET 框架旨在实现下列目标：

- 提供一个一致的面向对象的编程环境，而无论对象代码是在本地存储和执行，还是在本地执行但在 Internet 上分布，或者是在远程执行的。
- 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- 提供一个保证代码（包括由未知的或不完全受信任的第三方创建的代码）安全执行的代码执行环境。
- 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- 使开发人员的经验在面对类型大不相同的应用程序（如基于 Windows 的应用程序和基于 Web 的应用程序）时保持一致。
- 按照工业标准生成所有通信，以确保基于 .NET 框架的代码可与任何其他代码集成。

.NET 框架具有两个主要组件：公共语言运行库和 .NET 框架类库。公共语言运行库是 .NET 框架的基础。可以将运行库看做一个在执行时管理代码的代理，它提供核心服务（如内存管理、线程管理和远程处理），而且还强制实施严格的类型安全以及可确保安全性和可靠性的其他形式的代码准确性。事实上，代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码，而不以运行库为目标的代码称为非托管代码。.NET 框架的另一个主要组件是类库，它是一个综合性的面向对象的可重用类型集合，利用它可以开发包含从传统的命令行或图形用户界面（GUI）应用程序到基于 ASP.NET 所提供的最新的应用程序（如 Web 窗体和 XML Web Services）在内的应用程序。

.NET 框架可由非托管组件承载，这些组件将公共语言运行库加载到它们的进程中并启动托管代码的执行，从而创建一个可以同时利用托管和非托管功能的软件环境。.NET 框架不但提供若干个运行库宿主，而且还支持第三方运行库宿主的开发。

例如，ASP.NET 承载运行库以为托管代码提供可伸缩的服务器端环境。ASP.NET 直接使用运行库以启用 Web 窗体应用程序和 XML Web Services。

Internet Explorer 是承载运行库（以 MIME 类型扩展的形式）的非托管应用程序的一个示例。使用 Internet Explorer 承载运行库便能够在 HTML 文档中嵌入托管组件或 Windows 窗体控件。以这种方式承载运行库使得托管移动代码（类似于 Microsoft ActiveX 控件）成为可能，但是它具有只有托管代码才能提供的重大改进（如不完全受信任的执行