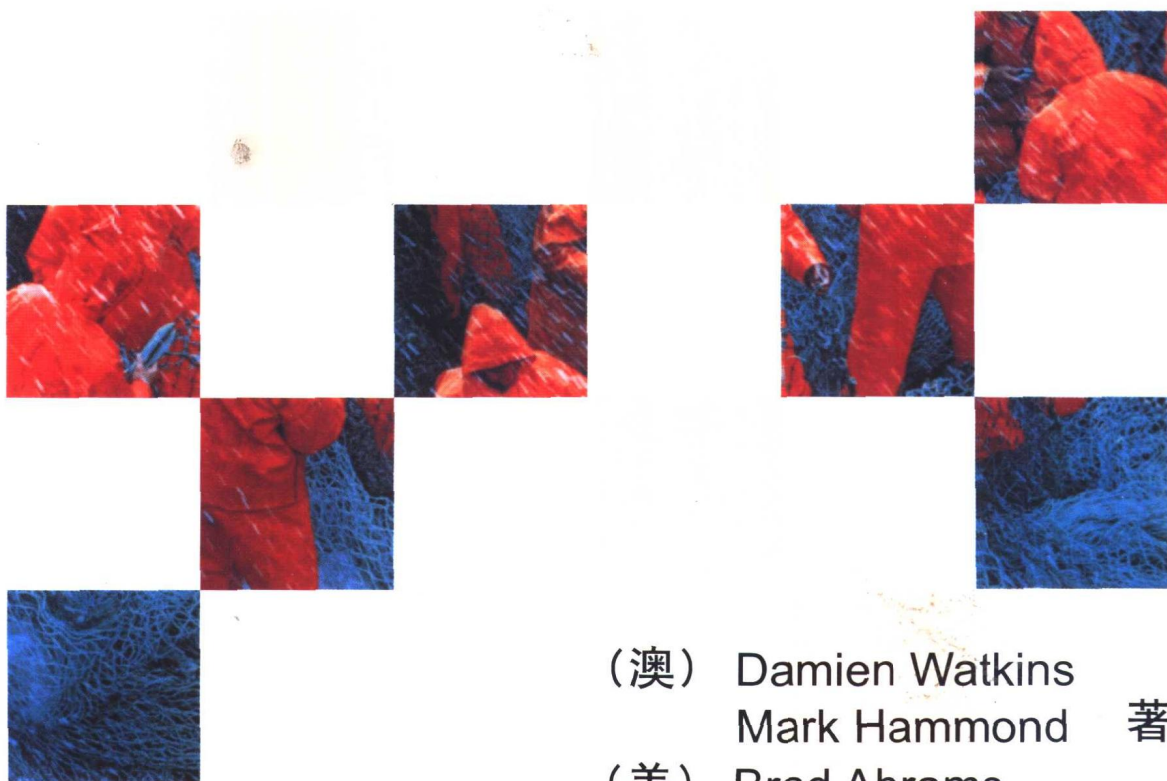


.NET开发丛书



.NET环境编程全景

Programming in the .NET Environment



(澳) Damien Watkins
Mark Hammond 著
(美) Brad Abrams

陈孝强 译



清华大学出版社

.NET 开发丛书

.NET 环境编程全景

(澳) Damien Watkins
Mark Hammond 著
(美) Brad Abrams
陈孝强 译

清华大学出版社
北 京

内 容 简 介

本书介绍了.NET框架的理念、目标和结构。从实践的角度说,这本书将教会读者如何开发框架、类、工具以及如何使用各种不同的编程语言设计.NET组件。

本书特色:内容涵盖面广,包括了CLR和框架类库、CLR的类型、元数据和执行系统、.NET程序集的联编和部署、.NET语言入门知识(包括C#和VB.NET)等。

本书适合初步掌握了C#语言,并且想在.NET程序设计中更上一层楼的读者阅读。同时本书也可供各级开发人员参考,尤其是附录,对于程序员拓展视野、掌握.NET框架最新发展趋势很有帮助。

Simplified Chinese edition copyright © 2003 by **PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.**

Original English language title from Proprietor's edition of the Work.

Original English language title: Programming in the .NET Environment, 1st Edition by Damien

Watkins, Mark Hammond, Brad Abrams, Copyright © 2003

EISBN: 0-201-77018-0

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education, Inc.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2003-1767

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

.NET环境编程全景/ (澳)沃特金斯, (澳)哈蒙德, (美)艾布拉姆斯著; 陈孝强译. —北京: 清华大学出版社, 2003

(.NET开发丛书)

书名原文: Programming in the .NET Environment

ISBN 7-302-07279-5

I. N… II. ①沃… ②哈… ③艾… ④陈… III. 计算机网络—程序设计 IV. TP393

中国版本图书馆CIP数据核字(2003)第084070号

出 版 者: 清华大学出版社

地 址: 北京清华大学学研大厦

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

邮 编: 100084

社 总 机: 010-62770175

客户服务: 010-62776969

文稿编辑: 汤涌涛 庞 燕

封面设计: 立日新设计公司

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: 185×230 印张: 24.75 字数: 536千字

版 次: 2003年9月第1版 2003年9月第1次印刷

书 号: ISBN 7-302-07279-5/TP·5284

印 数: 1~4000

定 价: 49.00元

推荐读物

以下列出了作者在书中引用的部分书籍，推荐给读者：

- J.Gough(2002). *Compiling for the .NET Common Language Runtime (CLR)*, Englewood Cliffs: Prentice-Hall.
- M.Henning and, S.Vinoski (1999). *Advanced CORBA Programming with C++*, Boston:Addison-Wesley.
- B.LaMacchia, S.Lange, M.Lyons, R.Martin, and K.Price (2002). *.NET Framework Security*, Boston:Addison-Wesley.
- S.Lidin (2002). *Inside Microsoft .NET IL Assembler*, Redmond: Microsoft Press.
- J.Richter (2002). *Applied Microsoft .NET Framework Programming*, Redmond: Microsoft Press.
- W.R.Stevens (1992). *Advanced Programming in the UNIX Environment*, Reading: Addison-Wesley.
- C.Szyperski (1998). *Component Software—Beyond Object-Oriented Programming*, Boston: Addison-Wesley.

读者对本书的赞誉

本书可以帮助读者深入了解.NET 框架带来的最新变革。尽管该书的示例代码用 C# 编写，但是事实上该书的大部分内容都是和具体语言无关的，适用于所有的.NET 语言。图书的语言精炼，提供了进行.NET 应用程序开发必不可少的信息。书中介绍了一些富于挑战性的概念，而且条理清晰。该书提供了相当丰富的资料，如果你要进行.NET 开发，这是一本很不错的案头参考书。附录介绍了几种.NET 语言，通过这些介绍你可以了解这些语言各有什么优缺点以及在不同的情况下使用哪一种语言。

——James J. Edelen 摘自亚马逊网站

本书的作者是微软.NET 开发团队的成员，通过他们独特的视角，我们可以了解从其他作者那里了解不到的知识。该书介绍了当初设计 C#时所做的权衡比较、设计 Visual Basic .NET 时对于 VB 未来发展的种种设想、垃圾回收机制和确定性终结这两种内存管理机制的利弊以及垃圾回收机制胜出的原因。书中提供了大量的实例，向读者展示.NET 如何使用与 CORBA, COM 以及 Java 不同的方式解决问题。通过阅读，读者会了解，虽然其他技术在某些方面胜过了.NET，但是从综合表现的角度看，.NET 是最终的佼佼者。书中介绍各种概念（比如类型系统、元数据、版本控制、安全性）时都使用了同样的模式：首先介绍核心问题，然后简单地谈一谈其他技术如何解决这个问题，最后通过简洁的实例详细介绍.NET 如何提供了一个最佳的解决方案。本书介绍了一切其他.NET 书籍中常常被忽略的问题。这些问题有的比较艰涩难懂，适合有一定编程经验的读者阅读。比如本书对值类型如何装箱为引用类型进行了精彩的阐述，再比如书中对一个简单程序的 IL 代码进行了逐行解释。

——摘自亚马逊网站

你想知道.NET 到底是如何工作的吗？如果你感到不了解 v-table 如何工作就没办法用好 C++，那么这本书就是为你准备的。选这些人来写这本书是恰到好处的。在本书中，他们不仅阐明了.NET 是如何工作的，而且对于.NET 为什么要这样做作了绝妙的阐述。有人说只有编写编译器的人才有必要读这本书，对此我不敢苟同。从那块小小的硅片上面开始，我们就不得不用越来越大的系统开销来制造更多的抽象概念，实现更多的功能。如果不了解功能和资源开销之间的对比关系，你就很难编写出真正出色的代码，所以你应该读一读这本书。

——James Plamondon, 摘自亚马逊网站

序

没有什么可以代替经验。尤其是把传授经验作为工作目的时，这句话显得尤为正确。正如您在阅读本书时将看到的那样，作者 Damien Watkins, Mark Hammond 和 Brad Abrams 都熟悉微软.NET 开发，并且有能力阐述清楚他们所掌握的知识。

Damien Watkins 是参加 Project 7 项目的首批外聘开发人员之一，Project 7 是微软和几所大学的合作研究项目，旨在开发下一代的 COM。

Mark Hammond 因其对 Python 语言的 Win32 扩展所做的大量工作而闻名。他几乎和 Damien 在同一时间被 Project 7 聘用，因为 Project 7 项目组认为 Python 应该是首批移植到 .NET 平台上的语言中的一种。

Brad Abrams 是微软.NET 框架项目组的一个主要项目经理，同时也是公共语言规范 (CLS) 项目组的负责人。从一开始他就参加了 .NET 框架的设计和实现。

虽然 .NET 框架建立在微软设计 Win32 API 和基于组件的程序开发的经验之上，但它还是和它的前辈有许多不同，以至于开发者要花一些时间才能够熟悉这个环境。要想高效运用这个框架，您的思维要进行一些根本的改变。

举一个简单的例子，例如内存管理。在 Win32 编程中，内存和资源的泄漏是一个主要的问题。在 .NET 编程中，垃圾回收托管堆自动解决了大多数这类问题。另一方面，有了垃圾回收并不意味着程序员不再需要进行精心的程序设计。另外，了解 .NET 内存管理的工作原理将会对您进行明智的结构设计提供帮助。

CLR 安全系统提供了两种不同的安全模型，基于角色的安全模型和基于证据的安全模型。通过使用 Windows NT 提供的角色、用户和组，您可能比较熟悉基于角色的安全模型。而基于例证的安全模型对您来说可能是一个完全新鲜的事物。Watkins 和他的合著者们在本书中详细介绍了这种新的安全模型，使读者们能够轻松地充分利用这个新的强大的平台功能。

框架类库 (FCL) 是一个大话题，需要数千页才能够详细介绍所有的内容。本书对框架类库进行了简短的介绍，提供了一个入门的好途径。另外，作者除了告诉您如何使用类，还介绍了进行库设计所应遵循的规则。

这本书将成为您的得力助手。

——Jeffrey Richter

前言

开发大型的分布式软件系统是复杂和有趣的挑战。一些构架可以简化开发工作，减轻了开发者的负担，这些负担和设计过程中遇到的许多互操作性问题有关。本书就将介绍这样一种构架——微软的.NET 框架。

有个问题常被问及：“.NET 框架里面究竟有些什么新东西？”从某种程度上说，答案很简单：“没什么新东西。”但是如果在一定的背景之下，我们也可以说它是最新的软件发展成果。例如，C++是一个明显的进步——但是C++实际上是一个同时具备 Simula 67 面向对象概念和 C 语言的高效性这两个优势的混合体。类似地，Java 也没有包含多少新发明，只是采用了虚拟机和类库这些存在了很多年的概念。那么，这些进步是如何导致知识复合体的出现呢？通常，实现的方法是利用协同作用，也就是说，对现有的技术用一种新的和以往不同的方式进行组合，使开发者能够在一个构架中结合使用两个强大的概念。.NET 框架正是如此。虽然使用这个框架有很多好处，但是读者在学习的过程中也不会感到吃力，因为.NET 环境中包括了许多熟知的概念，当然，这些概念对应的实现可能会有些变化。

例如，在.NET 框架出现之前就已经存在“面向对象”这个概念。近来，这个概念在从图形用户界面（GUI）设计到网络编程的许多领域都得到了认可。.NET 框架支持所有的面向对象概念，包括分类、信息隐藏、继承和多态性。.NET 框架中的新东西是打破了编程语言之间的界限，这些界限过去曾制约了面向对象概念的发展。对于其他概念，.NET 框架也进行了扩展，例如，继承必须遵循一定的安全约束；您可以使用一个类型，并不表示您可以继承这个类型。

读者对象

了解本书的读者对象是很重要的，这样您才能知道本书是否适合您去读。这本书的目标读者是具有以下要求的软件开发者，他们希望：

- 了解.NET 框架的理念和构架。

- 开发出应用于.NET 框架的泛型框架、库、类以及工具。
- 在.NET 环境中使用多种语言开发应用程序。

这本书是为软件开发人员写的，书中不仅介绍了.NET 框架的目标和构架，而且说明了技术如何实现满足这些目标的功能和服务。对于所有的分布式系统开发人员来说，理解.NET 的构架和理念是非常重要的，即便他们并不使用.NET 框架进行开发。为什么呢？因为.NET 框架代表了微软对于应用于 Internet 的分布式应用程序的开发的设想。通过理解.NET 框架的构架，开发人员可以深入了解与分布式系统开发相联系的一些问题以及微软对这些问题的解决方法。

一旦开发人员对于.NET 框架的构架有了一定的了解之后，下一步就是在这个框架之上开发软件。.NET 框架并非一个抽象的编程模型，而是一个功能完备的系统，它允许开发人员将他们的解决方案付诸实践并且在一个健壮安全的环境中提供给他们使用。.NET 框架并不依赖特定的编程语言，开发人员可以选择合适的语言来开发系统的不同部分，然后在运行时进行合并，而不必理会语言之间的不同之处。

那么，这本书不适合什么样的人呢？这不是一本介绍程序设计的书，读者在读这本书之前应当有开发软件的经验。这本书既不是整个.NET 框架的，也不是.NET 框架的某个部分的面面俱到的权威指南。一部全面介绍.NET 框架的所有内容的书将几乎是无法消化理解的。有许多书介绍了.NET 框架的某一方面，比如 ASP.NET。而这本书则提供了一个完整的.NET 构架基本面貌的概述。要想详细了解.NET 框架的各个部分，比如安全系统，读者应当参考其他图书或者说明文档。

组织结构

本书的结构很简单：

- 第1章介绍基本概念并提供一些开发分布式系统涉及到的背景知识。
- 第2章～第4章是“运行库部分”，介绍被认为是“微观编程”的问题——定义类型、存储元数据以及执行程序等。
- 第5章～第7章是“联编与部署部分”，这一部分讨论分布式系统开发中的难点问题，比如通过组件装配和开发软件以及部署中会遇到的挑战。这一部分还会介绍.NET 基本框架，包括建立应用程序所需要的库。
- 附录中包含第1章～第7章没有涉及到的一些重要的补充知识。这些知识包括一些来自于为.NET 框架开发编译器的开发人员的经验。

本书目的

“我们为什么要写这本书？”我们不止一次问自己这个问题。

在 1998 年晚些时候，微软询问莫纳什大学是否愿意参加“下一代 COM”（也就是后来为人们所熟知的 COM 对象运行库 [COM Object Runtime, COR]）的开发。Project 7 是这个多国合作项目的名称，微软研究院的 James Plamondon 向莫纳什大学发出参加该项目的邀请。为什么选择莫纳什大学呢？主要是因为这个项目和 Bertrand Meyer 以及他的面向对象程序设计语言 Eiffel 有关。在开发的早期，微软就坚定地将注意力集中到设计一个支持尽可能多的编程语言的运行时系统上。

莫纳什大学接受了邀请，并且在 1999 年初，Damien Watkins 在亚特兰大参加了一个概要会议，正是在那时提出了写这本书的设想。Damien 一看到后来被称为 Lightning 的 COR 的预演就问 Jame Plamondon，是否有人正在写一本关于 Lightning 的书。很显然，需要有一些书来介绍 Lightning，Damien 希望 Project 7 成员们细微而卓有成效的工作都被记录下来。在 James 的鼓励下，经过几年的修改，这本书问世了。这本书的附录中对微软以外的为 Project 7 做出贡献的人们表示了诚挚的感谢。

Mark Hammond 从 1991 年开始参加 Python 的开发，设计并维护 Python 的 Win32 扩展，包括 PythonWin IDE 和 COM 的支持库。Mark 从 20 世纪 90 年代中期开始为微软的 Python 相关项目工作，显著的成果包括 Python 语言的 ActiveScripting 和 ActiveDebugging 扩展。在 2000 年，Mark 和 Andy Robinson 合作出版了他的第一本书 *Python Programming on Win32*。

1998 年，在看到 Python 日益普及后，Project 7 项目组做出决定，Python 应当成为移植到这个平台上的最初几种编程语言之一。Mark 和微软的合作历史使其成为负责这项工作的当然人选。他住在澳大利亚的墨尔本，靠近墨尔本大学和莫纳什大学。这样，Project 7 的核心开发团队在西雅图以及地球对面的墨尔本同时形成了。因此，Damien 遇到了 Mark。

Brad Abrams 非常幸运地参加了公共语言运行库（Common Language Runtime, CLR）的设计，在设计应用程序编程接口（Application Programming Interface, API）和一些基本类型比如 System.Object 和 System.String 的过程中得到了锻炼。他参加了最早的设计决策，这些决策对 .NET 框架产生了广泛的影响，甚至包括所有的源代码。在 Brad 领导公共语言规范（CLS）的开发小组时，他热衷于在执行系统中内建的跨语言支持。

在 1998 年早些时候，来自另外一个开发组的 Adam Smith，问了一个有趣的问题：

如果从他的库中暴露一个属性，Visual Basic（或者其他语言）可以使用这个 API 吗？正如其他任何一个受人尊敬的微软项目经理会做的那样，Brad 召开了 1 小时的会议讨论 CLR 应该提供哪些可以被所有编程语言使用的特性，但这次会议并没有解决问题。事实上，为回答这个问题，微软的主要设计师花了数千小时，前后历时 3 年。这些设计师包括 Anders Hejlsberg, Peter Kukul, Paul Vick, Alan Carter, Scott Wiltamuth, George Bosworth, Lauren Feaux, Ian Ellison-Taylor, Herman Venter, Jonathan Caves, Francis Hogle, Mark Hall, Daryl Olander, Craig Symonds 和 Brain Harry。后来，Brad 和微软之外的一群主要的程序设计语言的改革者，也就是 Project 7 的成员一起重新审查和修改 CLS。通过这次合作，Brad 遇到了 Damien 和 Mark。

在研究 CLS 应该包含哪些功能的过程中出现了一些副产品，这些副产品就是许多“最佳方法”。Brad 将这些最佳方法记录下来，这些记录后来汇集成为文档 *.NET Framework Design Guidelines*（《.NET 框架设计指南》）。这些文档在利用 .NET 框架提供的 API 实现连续性和可用性的努力中起到了指导作用。为 CLS 所做的工作以及指南的编写使 Brad 在这个微软抽调了各方面完全不同的人员组建的 .NET 设计项目组中成为一个统筹全局的人物。通过这些努力，Brad 因为 .NET 框架的各个组成部分体现的价值以及在这些组成部分中使用连贯概念的需要而得到了赞赏。

除了他的日常工作，Brad 还先后通过欧洲计算机制造商协会（European Computer Manufacturers Association, ECMA）和国际标准化组织（International Standards Organization, ISO）参加了一个负责设计 CLI 和 C#语言标准的小组。这个小组再次对 CLS 和指南进行了仔细的复审和调整，并在 Jim Miller 的帮助下将其作为国际 CLI 标准的一部分发布。Brad 喜欢谈论 .NET 框架以及这个框架为开发工作带来的便利，所以他毫不犹豫地答应写这本书。

对我们来说，为 Project 7 工作所带来的收获与其说来自开发了一种新的技术，不如说来自和这样一大群来自世界各地的各具特色的、能干的开发者一起共事。

本书采用这一书名是为了向 W.Richard Stevens 的经典图书 *Advanced Programming in the Unix Environment*（Addison-Wesley, 1992）表示敬意。Rich 的书是一个奇迹，我们怀念他。

.NET 框架现状

.NET 框架在发展过程中经历过多次改名。最初叫 Project 42，后来被改为 COR，再后来叫 Lightning, COM+2.0 和 NGWS（Next Generation Web Services，下一代 Web 服

务)。直到 2000 年 7 月在奥兰多召开的专业开发者大会 (Professional Developers Conference, PDC) 上正式发布前几个星期才被最终定名为 .NET 框架。^①

.NET 框架的核心部分已经由 ECMA 制定为标准。对 .NET 框架进行标准化的目的是允许该框架的其他实现的建立。除了商业版的 Windows 实现, 微软还设计了 Windows 和 BSD Unix 上使用的共享源代码的实现; 相信来自其他团队的实现也会跟着出现。想了解更多了解标准化工作可以访问网站:

<http://www.ecma.ch>

想详细了解 .NET 框架的标准化工作, 可参阅网址:

<http://www.ecma.ch/ecma1/STAND/ecma-255.htm>

想了解 C# 的标准, 可参阅网址:

<http://www.ecma.ch/ecma1/STAND/ecma-334.htm>

想找到更多共享源代码, 可参阅网址:

<http://msdn.microsoft.com/net/sscli>

致 谢

我们想感谢的人真是太多了。首先, 我们要感谢微软的工作人员, 他们允许我们测试 .NET 框架, 向我们解释他们为什么要设计这样一个框架, 倾听我们的反馈意见, 甚至结合我们的意见修改系统。虽然我们可能会遗漏掉一些人——对此我们感到非常抱歉, 但我们还是要列出他们的名字: Nick Abbott, Ihab Abdelhalim, Chris Anderson, Eric Andrae, George Bosworth, Bryan Barnett, Jonathan Caves, Shuk Chan, Steve Christianson, Dennis Crain, Krzysztof Cwalina, Dan Fay, Kit George, Tony Goodhew, David Glance, Carl Grumbeck, Brian Grunkemeyer, Jonathan Hawkins, Mark Hall, Brain Harry, Anders Hejlsberg, Nick Hodapp, Jim Hogg, Rob Howard, Tom Kaiser, Loran Kohnfelder, Pranish Kumar, Ronald Laeremans, Sebastian Lange, Jason McConnell, Erik Meijer, Todd Merrell, Brad Merrill, Jim Miller, Oshoma Momoh, John Montgomery, Anthony Moore, Jon Nicponski, James Plamondon, Peter Plamondon, Susan Radke-Sproull, Jayanth Rajan, Craig Schertz, Don Syme, Clemens Szyperski, Peter Torr, Paul Vick, Scott Wiltamuth 和 Eric Zinda。

^① 关于 .NET 框架历史的一个小插曲: 每一个 .NET 框架可执行文件都含有 “BSJB”。这个字符串指的是 .NET 框架的最初几位开发者——Brian Harry, Susan Radke-sproull, Jason Zander 和 Bill Evans。

我们的 Project 7 的同伴们积极参与了 Redmond 开发实验室的工作, 谢谢他们富有价值的见解。再次列出他们的名单, 尽管会有遗漏: Gerry Butler, Chee Yeen Chan, Simon Cuce, John Daintree, Tyson Dowd, Jan Dubois, John Gough, Jürg Gutknecht, Fergus Henderson, Basim Kadhim, Keif Kornstaedt, Bertrand Meyer, Christine Mingins, Nick Nicoloudis, Nigel Perry, Paul Roe, David Simmons, Raphael Simon, Emmanuel Stapf, Pinku Surana 和 Dean Thompson。

本书的评论者提供了无价的反馈, 并且指出了书中的许多错误, 非常感谢。我们要特别感谢 Martin Heller, Michi Henning, Jim Hogg, Laurie McGuire, Erik Meijer, Scott Meyers, Christophe Nasarre, Jeffrey Richter 和 Purush Rudrakshala。书中剩下的错误责任都在作者。

我们要向 Addison-Wesley 出版社的工作人员表示特别的谢意, 是他们将我们的文字变成了图书出版: Kristin Erickson, Jill Hobbs, Curt Johnson, Chris Kief, Debbie Lafferty, Chanda Leary-Coutu, Joan Murray, Robin O'Brien, Simone Payment, Stephane Thomas 和 Katherine Tristaino。

还要特别提到的是 Thai Kitchen 和 Kim Chi 这两家餐馆, 许多复杂的问题, 都是我们在这两家餐馆享受美味午餐时解决的。

我们还要谢谢我们的家庭, 在过去 4 年中, 我们整天忙碌, 他们要忍受我们不在时的寂寞。

致我的妻子: 你的爱和支持给予我力量, 你的校对工作减少了我出丑的机会!

Brad

致我的女儿 Sacha, 她不得不容忍他的父亲长时间占用她的电脑。致 Dave's Jungle 酒吧的每一个人, 他们让我的生活不至于只剩下电脑。致我的父母 Lee 和 John 及我的兄弟 Paul, Scott 和 Jason, 他们让我在一个自由的环境里成长, 成为一个我愿意成为的人。谢谢你们, 我爱你们。

Mark

致我的妻子 Mäire 和我们的 5 个孩子 Joseph, Jane, John, Leah 和 Katherine, 他们要忍受我长时间不在他们身边。现在, 这本书和论文写完了, 我们又可以在一起了。我爱你们。

Damien

目 录

第 1 章 介绍.NET 框架	1
1.1 编程问题	2
1.2 解决方法	4
1.3 对比.NET 框架和基于 IDL 的系统	5
1.4 .NET 框架的组成	6
1.5 揭示.NET 框架	8
1.6 应用程序和.NET 框架	9
1.7 术语	10
1.8 实例: Hello World	11
1.9 小结	14
第 2 章 类型系统	15
2.1 编程语言和类型系统之间的关系	15
2.2 CLR 类型系统的组成部分	18
2.3 值类型	20
2.4 引用类型	30
2.5 实例: 用户定义对象类型	38
2.6 实例: 值类型上接口的使用	40
2.7 赋值兼容性	42
2.8 嵌套类型	43
2.9 可见性	44
2.10 可访问性	44
2.11 小结	45
第 3 章 元数据系统	47
3.1 元数据问题	48
3.2 保存类型的元数据: IDL 文件	48
3.3 反射: 类型元数据的检查	49

3.4	元数据工具和扩展	57
3.5	类型的动态解析	64
3.6	程序集和程序集清单	66
3.7	元编程	70
3.8	元数据文件格式	73
3.9	COM Interop	76
3.10	小结	79
第 4 章	执行系统	81
4.1	执行系统与其他组件模型的比较	81
4.2	中间语言	82
4.3	启动 CLR 程序	89
4.4	COR 调试器	90
4.5	应用程序域	92
4.6	内存管理	94
4.7	安全性	97
4.8	策略管理器	103
4.9	堆栈步	109
4.10	声明式和命令式	112
4.11	小结	113
第 5 章	联编应用程序	115
5.1	解决应用程序相关问题的现有技术	116
5.2	程序集	118
5.3	实例：简单的程序集	119
5.4	嵌入资源和链接资源	122
5.5	程序集链接器	126
5.6	公有程序集和私有程序集	127
5.7	程序集版本控制	134
5.8	国际化和本地化	137
5.9	应用程序域	144
5.10	小结	149
第 6 章	部署应用程序	151
6.1	配置文件	151
6.2	下载 Web 内容	158

6.3 安装应用程序.....	168
6.4 在非 Windows 平台上使用 CLI	170
6.5 小结	175
第 7 章 框架类库.....	177
7.1 历史回顾.....	178
7.2 设计目标.....	180
7.3 设计原则.....	184
7.4 命名空间.....	191
7.5 回顾和前瞻.....	229
7.6 小结	230
附录 A Visual Basic .NET	231
A.1 类型系统的扩展.....	231
A.2 类型系统的修改.....	237
A.3 平台的改变.....	243
A.4 语言的清理.....	251
A.5 新功能.....	252
A.6 未来的方向.....	252
A.7 结论.....	253
附录 B C#	255
B.1 历史和设计目标	255
B.2 C#类型系统	257
B.3 面向组件的开发.....	258
B.4 非安全代码.....	264
B.5 几个小问题.....	265
B.6 堆栈组件实例	268
B.7 未来方向.....	271
B.8 C#和标准化	272
B.9 结论.....	272
附录 C Python for .NET	273
C.1 Python 概览.....	273
C.2 Python for .NET	275
C.3 使用 Python for .NET	278

C.4	Python for .NET 的局限	280
C.5	.NET 和 Python 可能的改进	282
C.6	其他的实现策略	284
C.7	结论	286
附录 D	Perl 和 .NET 运行环境	287
D.1	Perl for .NET 试验编译器	287
D.2	PerlNET 组件联编器	291
D.3	得到支持的 .NET 功能	293
D.4	PerlNET 的状态	294
D.5	实例: Windows 窗体程序	294
D.6	结论	296
附录 E	CLR 中的组件 Pascal	297
E.1	关于组件 Pascal	298
E.2	映射至 CLR	301
E.3	语义的挑战	307
E.4	结论	308
附录 F	Hotdog: 为面向对象虚拟机进行 Scheme 编译	311
F.1	Hotdog Scheme 编译器简介	311
F.2	Scheme 简介	312
F.3	面向对象虚拟机	313
F.4	实现	313
F.5	局限	318
F.6	性能的改进	319
F.7	结论	320
	参考文献	320
附录 G	.NET 框架中的函数型语言	321
G.1	Mondrian 简介	322
G.2	Mondrian 中的类型	323
G.3	Mondrian 中的函数	327
G.4	调用其他 CLR 语言	333
G.5	.NET 的功能: 多语言实例	334
G.6	结论	338

参考文献	338
附录 H Active Oberon for .NET: 语言模型映射的案例	341
H.1 ETH 编程语言的历史	341
H.2 活动对象系统	342
H.3 映射至公共类型系统	354
H.4 小结和结论	363
H.5 致谢	364
参考文献	365
术语表	367
推荐读物	377