PROGRAMMER TO PROGRAMMER™

Professional .NET Framework 2.0

.NET Framework 2.0 高级编程

(美) Joe Duffy 著
王海涛 陈宇寒 译

wrox™
清华大学出版社

.NET Framework 2.0 高级编程

(美) Joe Duffy 著

王海涛 陈宇寒 译

清华大学出版社

北 京

Professional .NET Framework 2.0
Joe Duffy
EISBN：0-7645-7135-4

北京市版权局著作权合同登记号　图字：01-2006-1581

图书在版编目(CIP)数据

.NET Framework 2.0 高级编程/(美)达夫(Duffy,J.)著；王海涛，陈宇寒译. —北京：清华大学出版社，2007.6
书名原文：Professional .NET Framework 2.0
ISBN 978-7-302-15184-5
Ⅰ.N… Ⅱ.①达…②王…③陈… Ⅲ.计算机网络—程序设计 Ⅳ.TP393.09

中国版本图书馆 CIP 数据核字(2007)第 067672 号

责任编辑：王　军　于　平
装帧设计：孔祥丰
责任校对：成凤进
责任印制：王秀菊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　**邮　　编**：100084
c-service@tup.tsinghua.edu.cn
社 总 机：010-62770175　**邮购热线**：010-62786544
投稿咨询：010-62772015　**客户服务**：010-62776969
印 刷 者：北京鑫丰华彩印有限公司
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：35.5　**字　数**：864 千字
版　　次：2007 年 6 月第 1 版　　**印　　次**：2007 年 6 月第1 次印刷
印　　数：1～4000
定　　价：68.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770177 转 3103　产品编号：020273-01

前　言

2002 年 1 月 14 日，我还是一名 Java 开发人员。当时我对 Windows 没有太多好感，主要是因为在那之前的几年我被 COM 和 Win32 惹恼了太多次。那时我不喜欢使用 HANDLES、WinDbg，以及 free 和 delete。20 世纪 90 年代中晚期，我曾花费几年时间使用微软工具和技术进行软件开发，但最终对所开发出来的极其复杂的、堪称 “生态系统” 的产品感到厌烦。虽然那时 Java 也还没有成熟，但它已经提供了某些功能，如沙箱状的执行环境、简单(不使用指针！)的语法和垃圾收集器等。它精心设计的库使喜欢使用纯粹的面向对象程序设计的人们感到非常方便(其实我就是其中的一员)。

但仿佛一夜之间，我再次成为 Windows 开发人员。我再次开始喜爱上这个平台。这段时间除了是我个人的重大转折点外，也反映出整个业界范围的转折点。几年之后再回过头想一下，可以清楚地看出，该转折点将 Windows 平台上的编程模型一下子带回到主流软件开发的最前端。哪些因素促成了这次革命性的方向转移？答案很简单：.NET Framework、C#语言，以及两者的基础——公共语言运行库(Common Language Runtime，CLR)，它们都于 1 月 15 日在 MSDN 上发布并提供下载。

当前，这一平台已经历 3 次大的提升，市场上已经有了 1.0、1.1 版本，现在是 2.0 版本。技术不断地走向成熟，越来越健壮、可靠，并已经跨过了与创新性的(同时也是危险的)新技术竞争的阶段。是的，我必须承认：我青睐 CLR。

本书目标

本书的首要目标是使您喜欢.NET Framework 和 CLR 2.0 技术，并鼓励您在这个平台上编写优秀的代码。由用户编写的卓越的应用程序和库即使不比平台自身更重要，至少也是同等重要。假如因为我在这本书里写的某些内容而激发您在 CLR 上编写下一个 google.com，并随后因此而变得富有，那么我的工作就做到位了。

当然，大多数人需要一本合适的书还是为了实际目的(例如为了更好从事他们的工作)，所以这也是本书的目标之一。这本书应该成为一本出色的基础读物，帮助您迅速了解 2.0 版本提供的功能，在很短的时间内从对该平台一无所知到有基本的了解，并作为需要帮助时的一本参考书。我还相信本书将成为您深入钻研自己感兴趣的特定技术领域的很好的起点。

这本书中讲述的很多主题与它们的实际内容比起来要浅显得多。这是比较合理的。我已经在书中介绍了很多有关运行时和库的最重要的内容，但肯定的是至少省略了其中一个方面，因为要想完全覆盖所有内容将给这本书增加大约 10000 页的文字。为节省读者的时间，减轻阅读这么多文字的负担，我优先列出了一部分主题，并将集中讨论这些主题，我认为这些主题对以下 3 个目标而言是最重要的：(1)在该平台上迅速提高开发能力；(2)对体

系结构有持久和基本的理解；(3)提供实际可行的建议以避免常见的错误并能在当前应用开发中编写优秀的代码。

写作初衷

当有了写作这本书的机会时，我进行了长时间的、艰难的思考才决定接受这一任务。我试图搞清楚怎样才能将这本书与其他已经出版的、讲述同样主题的书区别开来。然而不久我就意识到：我甚至连一本市场上销售的有关.NET Framework 的书都没有读过，而我还自认为是这方面的专家。

总结之后我认为，我之所以没有阅读任何一本有关.NET Framework 的其他书籍，其主要原因在于我非常不欣赏大部分这类书籍所涉及的内容层次和采用的写作风格。大多数作者选择了一种与产品附带的软件开发工具包(Software Development Kit，SDK)文档非常类似的方式来写作.NET Framework，即采用一种过于初级和入门化的风格。显然，阅读产品文档有助于人们理解一些面上的东西，但我想要了解得更多。毕竟文档是免费的！

如果我要写这本书，那么它必须是我自己也喜欢阅读的一本书。我认为要达到这一目标，必须满足以下几个条件：

- 不仅介绍技术“是什么”，还要说明隐藏在技术后面的“基本原理”和“采用它的原因”。这意味着要深入讨论有助于深刻了解某个主题的内部工作机制，并且甚至可以否定明显不成熟的设计决策。阅读一本仅仅描述一个平台提供什么样功能的书不仅枯燥无味，而且会使人立刻将这本书降格为查考资料类图书。
- 解释重要概念的同时必须说明它们与其他技术的联系和相互间的对应关系。.NET Framework 和 CLR 并非该领域的第一个平台，忽视前人的工作就好像是亵渎读者。我假定本书的读者已经懂得如何编程，因此将某项技术与读者可能熟悉的其他已有平台进行对比说明，对解释该项技术有所帮助。即使读者对相关技术并不熟悉，了解某些(疯狂)想法并非第一次实现往往是一件好事。
- 尽可能覆盖全面，但也不隐瞒不完整的地方。在不得不简略进行介绍的地方，读者可以根据自己的时间利用指向相关资源的链接进行更为深入的学习。显而易见的是，没有一个作者可以在少于 10000 页的篇幅里，以任何适当的细节程度介绍.NET Framework 或是 CLR 的每个组成部分。与其自认为已很好地做到这一点，不如给读者留下自己进行深入研究的线索，使他们能够按照自己的计划或在必要时继续深入研究。

心里有了这些指导方针后，我接受了这个任务，并进行了为期一年的探索。这的确是一段有趣的旅程。现在重新阅读过去一年里写下的内容，我感到在上述所有方面都做得相当好。我希望读者认同我的观点。

使用本书的前提

要开始使用托管代码进行开发，所需要的就是.NET Framework 软件开发工具包(SDK)。

这个开发包可从 MSDN 站点(http://msdn.microsoft.com)上免费获取。下载包是可以再发布的，除了基本工具和编译器外还包括 CLR 及.NET Framework 库。很多开发人员将会选用 Visual Studio 2005 来替代基于 SDK 的简单命令行式开发环境。同样可在 MSDN 站点(http://msdn.microsoft.com/vstudio)上找到有关 Visual Studio 的信息。

本书主要内容

本书共分 4 个部分，下面会进一步介绍这些内容。除了 4 个部分之外，还有一个单独的附录，该附录描述了公共中间语言(Common Intermediate Language，CIL)的完整指令集。

第 I 部分：CLR 基础

本部分的目标是要了解 CLR 在执行托管代码中扮演的角色。从某种角度来看，是从基础开始并逐步加深学习。有些人也许宁愿略过本部分直接跳到第 II 部分，在学习运行库基本概念之前先了解库。本部分将讨论这样一些话题，如通用类型系统(Common Type System，CTS)为程序提供什么样的抽象，CLR 如何在一台物理机器上运行托管代码，以及它运行代码时使用的垃圾收集和实时编译等服务。

1. 第 1 章：引言

第 1 章介绍了.NET Framework 技术，并阐述了 2.0 版本的关键改进之处。

2. 第 2 章：通用类型系统

在第 2 章中将逐一介绍通用类型系统(CTS)提供的功能。具体而言，将说明类型和它们的组件是如何组织的，值和引用类型之间的区别，以及类型系统的一些交叉特性，例如泛型和验证。读者将会了解 CLR 类型系统提供了哪些特性，以及像 C#和 VB 这样的语言如何利用上述的特性。

3. 第 3 章：CLR 技术内幕

在本章中将花费较大的篇幅介绍 CLR 如何完成其任务的内部细节。在概念层次上，将会使您了解托管代码能够按照正确的方式执行的原因。还将介绍 C#、Visual Basic (VB)和其他托管语言编译成的中间语言(Intermediate Language，IL)、异常子系统以及运行库管理内存的方式。本章的最后讲解 CLR 的 JIT 编译器。

4. 第 4 章：程序集、加载和部署

本章将介绍 CLR 的部署单元和程序集，它们包含的内容，以及编译器是如何生成它们，运行库是如何加载它们的。还将说明可用于部署的一些可选项，例如共享库、私有库和 ClickOnce。

第 II 部分：基础架构库

在第 I 部分介绍了运行库自身的功能后，本书的下一部分将讨论基类库(Base Class Libraries，BCL)的特定部分。记住，这些是编写托管代码时将用到的 Windows API。将仅讨论托管程序中最常见和最重要的一些库，而在本书后面的章节中学习其他一些更高级的库。

1. 第 5 章：基本类型

本章将介绍 Framework 必须提供的最底层的基本类型。除了在几乎所有的程序中都将使用的一些类似的通用类型外，这些类型包括构建到语言和运行库自身的基本类型。具体而言，包括标量、字符串、日期和时间、数学、常用的实用程序和常见的异常类型。.

2. 第 6 章：数组和集合

几乎所有的程序要用到数据集合。System.Collections.Generic API 提供了一种完成此任务的完善的方法，充分利用了泛型的功能。本章将说明除了一些非常基本的集合外这些 API 将提供的集合类型，包括普通的 System.Collections 类型和数组。

3. 第 7 章：I/O、文件和联网技术

到这个时候，应该非常习惯于创建和使用本地 CLR 数据。但是只对基本类型，如字符串、日期等，进行运算的程序非常少。本章将说明如何通过使用 I/O 与外部世界进行交互，包括使用文件系统和通过网络类库(Network Class Libraries，NCL)进行通信。

4. 第 8 章：国际化

在当今全球化世界中一个日益重要的主题是国际化(i18n)，也就是使应用程序适合地区文化和语言的过程。.NET Framework 上 i18n 的中坚是文化和资源，这正是本章的基本主题。将讨论应用程序国际化需要面对的一些技术性和非技术性挑战。

第III部分：CLR 高级服务

本书的第III部分将介绍 CLR 必须提供的一些高级服务，这些服务包括安全的编程模型、隔离和并发的形式以及 CLR 必须提供的各种互操作性特性。尽管这里讨论的许多主题是 CLR 标记的特性，利用库的程序员将会用到几乎所有这些特性。

1. 第 9 章：安全性

CLR 提供了一种安全基础设施，可以根据用户和代码标识来批准特权的操作。代码访问安全性(CAS)允许根据代码的来源限制程序的操作权限，例如代码是源自 Internet、内联网还是本地机器，此外还可使用其他有意义的标准来确定安全权限。

2. 第 10 章：线程、AppDomain(应用域)和进程

本章将会介绍 CLR 必须提供的各种隔离和执行粒度。还会说明 Framework 中的并发

编程模型，例如如何创建、同步和控制并行操作。另外，还会介绍可以用来控制 AppDomain 和进程的各种技术。

3. 第 11 章：非托管互操作性

并非所有的代码都是托管代码。实际上，大量的 Windows 代码都是用 C、C++和 COM 编写的，并且很可能在今后的一段时间仍是如此。CLR 提供了将类型系统、托管代码的二进制格式和这些技术联系在一起的方法。此外，当和非托管代码互操作使用时，它要求超越简单的内存管理的界限。因此，需要额外的技术来确保以可靠的方式释放资源。

第Ⅳ部分：高级 Framework 库

在本书的第Ⅳ部分中，介绍一些更高级的 Framework API，它们尽管没有第 II 部分中介绍的那些 API 常用，但是在托管代码中也会常常用到。

1. 第 12 章：跟踪和诊断

CLR 和相关的工具，如 Visual Studio 集成开发环境(IDE)，提供了强大的调试功能。但是除此之外，使程序和库配备跟踪代码有助于测试和故障分析。另外，跟踪还允许诊断代码中非常细微的问题，如因果关系、性能和可靠性问题。本章将较为全面地说明 Framework 中的跟踪基础设施。.

2. 第 13 章：正则表达式

本章将对正则表达式进行概括性的介绍——除了 System.Text.RegularExpressions 命名空间中的.NET Framework API 外，还会介绍正则表达式的特性、语法和功能。通过本章的讲解，用户应该能够将正则表达式合理地集成到应用程序中。

3. 第 14 章：动态编程

在第 II 部分中说明了元数据如何增强 CLR 和.NET Framework 的功能。第 14 章将介绍如何在动态编程场合中利用这些元数据。这意味着基于元数据驱动的功能呈现在结合了运行时信息的程序中，而不仅仅是编译时已知的信息。这将涉及到使用 Reflection(反射)子系统。 除此之外，还将说明如何使用 System.Reflection.Emit 命名空间来生成元数据。

4. 第 15 章：事务

自 Framework 的 2.0 版本起，添加了一个新的统一事务的 API。这将 ADO.NET、消息传递和 Enterprise Services (COM+)事务集成在一个单独的紧密联系的保护系统中。System.Transactions 提供了一组非常简单的类型并支持本地和分布式事务。

5. 附录

附录中列出了 CIL 和 MSIL 指令集中全部的 IL 指令。

源代码

在您登录到 Wrox 站点 http://www.wrox.com/时，只需使用 Search 工具或使用书名列表就可以找到本书。接着在 Code 栏中单击 Download 链接，或单击本书信息页面上的 Download Code 链接，就可以获得本书所有的源代码。另外，您也可以从本书的合作站点 www.tupwk.com.cn/downpage 上下载本书的所有源代码。

从以上站点上下载的文件使用 WinZip 进行了压缩。在把文件保存到硬盘的一个文件夹中时，需要使用解压缩软件(如 WinZip 或 PKUnzip)对该文件解压缩。在解压缩时，代码常常放在各自的章节文件夹中。在开始解压缩过程时，一定要将解压缩软件 WinZip 或 PKUnzip 设置为使用文件夹名。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。请给 wkservice@tup.tsinghua.edu.cn 发电子邮件，我们就会检查您的信息，如果是正确的，就把它发送到该书的勘误表页面上，或在本书的后续版本中采用。

要在网站上找到勘误表，可以登录 http://www.wrox.com，通过 Advanced Search 工具或书名列表查找本书，然后在本书的信息页面上，单击 Book Errata 链接。

E-Mail 支持

如果您希望直接就本书的问题向对本书知之甚多的专家咨询，那么，就向 support@wrox. com 发电子邮件，在电子邮件的“主题”(Subject)栏中，加上本书的名称和 ISBN 的最后 4 位号码。典型的电子邮件应该包括下列内容：

- 在“主题”栏加上英文书的书名、ISBN 的最后 4 位数字(1354)和问题所在的页码。
- 在邮件的正文中加上您的姓名、联系信息和问题。

我们不会发给您垃圾邮件。我们只需要详细的情况以节省您的宝贵时间和我们的时间。当您发送电子邮件时，它会直接链接到以下支持链：

- 客户支持——您的消息会传送到我们的客户支持人员，他们是阅读信息的第一人。他们有常见问题的文件，会迅速回答一般性的问题。他们回答关于本书和网站的一般性问题。
- 编辑支持——更深的问题会转发到负责本书的技术编辑处。他(或)她具有编程或特殊产品的经验，能够回答某个主题的详细技术问题。
- 作者支持——最后，在编辑都不能回答问题的情况下(这种情况很少出现)，这些问题将转发到作者。我们试图保护作者不要从写作中分心，但是，我们也很愿意将

特殊的问题转发给他们。所有的 Wrox 作者帮助支持他们的书籍。他们向客户和编辑回复电子邮件，所有的读者都会从中受益。

Wrox 支持过程只能提供直接与已出版的图书相关的问题。对于超出此范围的问题可以通过 http://p2p.wrox.com/论坛的团体列表来提供支持。

p2p.wrox.com

P2P 邮件列表是为作者和同行的讨论而设立的。我们在邮件列表、论坛和新闻组中提供“程序员到程序员的支持”(programmer to programmer support)，还包括一对一的电子邮件支持系统。如果把问题发送给 P2P，就可以相信，您的问题不仅仅是由支持专家解答，而且还要提供给我们邮件列表中的许多 Wrox 作者和其他业界专家。在 p2p.wrox.com 上，可以从许多不同的列表中获得帮助，不仅在阅读本书时获得帮助，还可以在开发应用程序时获得帮助。在网站的.NET 类别中，最适合本书的是 beginning_vb 和 vb_dotnet 列表。

要订阅一个邮件列表，可以遵循下面的步骤：

(1) 进入 http://p2p.wrox.com。

(2) 从左侧的菜单栏中选择合适的列表。

(3) 单击想加入的邮件列表。

(4) 按照指示订阅和填写电子邮件地址和密码。

(5) 回复接收到的确认电子邮件。

(6) 使用订阅管理器加入更多的列表，设置自己的邮件设置。

为什么这个系统提供最好的支持

您可加入该邮件列表中，也可以每周分类接收它们。如果您没有时间或设备接收该邮件列表，可以搜索我们的在线文档。垃圾邮件和广告邮件会被删除，您自己的电子邮件地址会被独特的 Lyris 系统保护起来。任何加入或退出列表的查询，或者与列表相关的一般问题，都应发送到 listsupport@p2p.wrox.com。

目　录

第Ⅰ部分　CLR 基 础

第II部分 基础架构库

第III部分 CLR 高级服务

第 I 部分

CLR 基 础

第 1 章

引　言

我们是从失败中成长起来的，而不是源于成功！

—— 引自 Bram Stoker 的 Dracula

这些年来，Microsoft Windows 平台得到了巨大的发展。尽管在发展过程中也经历过兴衰起伏，但是 Microsoft 的平台总体上在业界保持领导地位。在它的低落期，Microsoft 开发了本书目录表中涵盖的技术。本章简要说明这个发展历程，并概述本书会通篇讨论的技术体系结构。从第 2 章开始介绍通用类型系统(Common Type System)，它是在其上构建平台上所有代码的基础。

1.1　平台的发展历史

在 1985 年，将 Windows 引入到 IBM-PC 平台彻底改变了人们使用计算机的方式。许多人将这种变化归结为 GUI、鼠标指针和时尚的新型应用程序界面等方面，但是我所介绍的变化实际上是指出现了 Windows 应用程序接口(Application Program Interface，API)。16 位的 Windows API 能够充分利用 Windows 平台的潜能来支持完成强大的新任务，并提供了部署构建在动态链接之上的应用程序的新方法。大约 8 年后，也就是在 1993 年，发布了 Windows NT 平台，这是现在称为 Win32 API 的第一个版本。除了支持 32 位和数千个新函数外，Win32 API 几乎和 Windows 1.0 API 相同。

传统上，早期 Windows 平台上的编程是采用 C 语言的系统级编程。但是，在 20 世纪 90 年代后期逐渐分成 3 种不同的类别：系统、应用程序和业务脚本编程。每种编程类别需要使用一组不同的编程语言、工具和方法。这种分化现象随着时间的推移愈演愈烈，导致 Windows 开发人员群体出现分裂并困扰所有使用 Windows 编程的人。

对于系统编程和非常复杂且健壮的应用程序，通常使用 C 或 C++编写代码，直接采用 Win32 编程模型并可能使用类似组件对象模型(Component Object Model，COM)的方法来设计和发布可重用的组件。内存管理是需要解决的问题，必须深入了解 Windows 的工作原理。需要了解的主题包括分割内核空间和用户空间、USER32 和 GDI32 的区别等。必须熟练使