HZ BOOKS

开发人员专业技术丛书

C# Unleashed

C# 技术内幕

(美) Joseph Mayo 著

王启丁 高锦文 刘明 等译

SAMS

机械工业出版社
China Machine Press

开发人员专业技术丛书

C# 技术内幕

（美） Joseph Mayo 著

王启丁 高锦文 刘明 等译

机械工业出版社
China Machine Press

C#是C语言系列中第一个基于组件的程序设计语言。本书全面介绍了C#语言的语法和编程技术以及C#的类库，使读者迅速掌握这个全新的语言。本书不仅介绍了C#本身，还介绍了如何使用C#将软件作为服务来开发。除了讲述必要的语法之外，本书重点介绍了C#的实际应用。本书循序渐进，每一章都是带你一步一步地从C#的核心进入.NET框架的元素，并介绍了分布式多层Internet应用程序开发中用到的一些高级概念。除此之外，它还展示了如何调试、监视和扩展企业级的应用程序，使你能够应用C#的全部功能。本书注意比较了C#与C++和Java语言之间的异同，以便帮助开发人员顺利地过渡到C#。同时，本书也适合一般用户，只要使用过某种计算机语言开发过程序，就能掌握本书的内容。本书内容全面，讲述透彻，是专业开发人员的优秀参考书。

本书版权登记号：图字：01-2002-0811

图书在版编目（CIP）数据

C#技术内幕/（美）梅欧（Mayo, J.）著；王启丁等译. -北京：机械工业出版社，2003.1
（开发人员专业技术丛书）
书名原文：C# Unleashed
ISBN 7-111-10288-6

Ⅰ. C… Ⅱ. ①梅… ②王… Ⅲ. C语言－程序设计 Ⅳ. TP312

中国版本图书馆CIP数据核字（2002）第032366号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）
责任编辑：朱 劼
北京忠信诚胶印厂印刷·新华书店北京发行所发行
2003年1月第1版第1次印刷
787mm × 1092mm 1/16 · 34.75 印张
印数：0 001 - 5 000 册
定价：59.00 元

前　言

欢迎你阅读本书。这是一本关于C#（英语发音是see sharp）编程语言的指南和参考书。C#是一种崭新的面向对象的编程语言。它强调以组件为基础的软件开发方法。

基于组件的编程方法几年前就已出现，但C#所提供的功能将使我们达到软件开发的新水平。这是向着XML的Web服务转变的新模式——将软件视为一种服务，这种服务是非连接型的、无状态的，并且符合国际的开放式标准。

把软件作为服务是下一代计算机系统的趋势。例如，C#非常适合于建立Web服务，适合于开发在遵循开放式标准的Internet上可重用的组件。软件开发不再限于过去几年中发展起来的单机结构（monolithic architecture）。Web服务使应用程序能够在Web上使用分布式的服务，这就大大简化了软件开发，并提升了软件重用的规模。C#是在Web服务领域的主角，它有力地促进了软件作为服务这一大潮。

本书不仅介绍C#本身，而且展示了如何用C#开发作为服务的软件。纵观软件的发展，我们非常幸运地及时到达了这个转折点。多少年来，程序都是作为目的单一的单机应用程序来编写的。通过研究和经验，我们已经认识到模块化的好处，这就逐渐发展出面向对象方法。它为我们提供了大规模重用的可能性和可维护性。当协作和通信变成了商业的需求时，就自然地发展出了客户端/服务器和连网技术。Internet和Web为我们提供了分布式的、无状态的和安全的软件技术，包括Applet等等。下一步将是C#的用武之地——使Internet自动化。

这本书为什么适合你

如果曾经使用过其他计算机编程语言开发软件，你肯定能够理解本书中的概念。有编程经验的读者肯定懂得怎样编写进行逻辑选择和循环控制的代码，当然也会了解变量和十六进制之类的基本数字系统了。老实说，有雄心的初学者如果努力也能够很好地掌握这本书的内容。

历经了几年的软件开发之后，在编写这本书时我又多次回想起一些往事。我常常自问："如果过去的两年我都用COBOL编程来解决千年虫的问题，看这本书会怎么样呢？"或者"假如我是PROLOG程序员，在进行了多年的科学研究之后又会怎么样呢？""使用X语言的人能了解某些解释或例子吗？"当这些回答是肯定时，我感到充满信心，我已经十分认真地考虑了各种读者的情况。

本书是为每个程序员编写的。虽然它针对两个较大的潜在的读者群——C++和Java程序员有专门的提示，但是它也考虑了所有程序员的需求。它所讲述的是很基本的东西，使你能够了解C#的每个方面知识。同时，它也充分地研究了你每天需要处理的现代企业级的任务。通过本书的学习，我希望你有一个感觉，那就是现在你的背包里已经有了极好的开发Web服务和分布式解决方案的新工具，而开发Web服务和分布式解决方案将是我们无法回避的主要工作。

组织和目标

本书中的全部内容都是关于用C#编程语言编写代码的问题。许多高级主题内容本身就足够写一本书了。但是，本书的重点并不是讲解这些高级主题的细节，虽然本书对其中的一些领域也有一定深度的介绍。每一章的目的都是展示怎样用C#来完成给定的任务。即使是最理论化的章节，其重点仍然是如何将每个主题应用于使用C#来编写代码。

本书共分为六个部分。从比较简单的材料和与C#语言直接有关的内容开始。随后，介绍与C#相关的领域，并展示如何使用库。之后，才介绍更多高级的主题，并展示怎样使用C#为各种技术开发代码。

第一部分 C#的基础知识

第一部分提供了C#语言语法中最基本的元素。从介绍C#操作的环境开始，然后展示怎样创建两个简单的C#程序。它介绍了各种不同的C#类型以及怎样使用它们来创建表达式和语句。接着讲解用分支和迭代来控制程序流的问题。在介绍了足够的材料使读者能够了解相对复杂的程序之后，再介绍怎样调试C#程序。

- **第1章** C#并不是在程序直接编译成机器码的典型环境中操作。它是在管理代码运行的虚拟执行系统中运行。这个环境之所以能按其运行方式运作，需要用到一些特殊技术。C#已经作为开放标准提交给标准化组织了。该章选材非常精简，这是为了使读者能够迅速地深入到本书的主题——编写C#代码。
- **第2章** 介绍C#的基础，其中包括怎样建立一个程序、基本语法和关于C#类型的信息。C#的语法和类型与它的母语言C和C++的语法和类型很像。
- **第3章** 在任何语言中，计算的中心都是操纵带有表达式的数据。C#包括了一整套一元的、二元的和三元的表达式。重点介绍表达式的生成、运算符优先级和使用枚举及数组类型。
- **第4章** C#语言中用于控制程序流的有几种结构。除了传统的if和select语句、while和for循环语句之外，还介绍了新的foreach循环。我们还会讲到臭名昭著的goto语句。
- **第5章** 高级程序员别不耐烦，调试非常重要，特别是对中级程序员来说尤其如此。因此，我专辟一章用来讲解调试。其中还有一节是介绍预处理指令。

第二部分 用C#编写对象和组件

第二部分介绍如何用C#编写对象和组件。对于一些人来说，这一部分可能是最难学习的内容。因此，我从很基本的知识开始讲，然后逐步深入。事实上，有三章都是专门介绍对象和组件编程的概念，其余各章用来介绍C#对象的其他类型及如何使用C#的面向对象特性。

- **第6章** 应用C#编程的关键部分是了解面向对象的编程（OOP）。如果你纯粹使用OOP方法，那很好。该章是着重从方法、逻辑或过程来讲解程序设计。许多人在入门之前都会对面向对象的编程感到头痛。该章就是为帮助初学者尽快地入门而设计的。
- **第7章** C#中最常用的对象是类。该章就着重讲述创建类及其成员的机制，包括构造函数、析构函数、字段、方法、特性和索引器。其中还有怎样使用XML注释的完整的程序演示。

- **第8章** 类定义了对象。反过来，类也有面向对象的行为。该章的讨论和例子都深入介绍了如何使用类来完成面向对象的编程。
- **第9章** C#中的重载的方法和操作符类似于C和C++中的相应方法和操作符，它们之间只有一些很小的差异。该章中举出了几个例子来说明怎样重载成员和操作符。这些例子也介绍了一些新的约束和安全防护措施。
- **第10章** C#具有广泛的错误处理和异常支持。该章所讨论的有关内容包括用try/catch/finally块进行异常处理，异常的创建和管理，异常的恢复。
- **第11章** 事件和委托密切相关，它们也支持后联编方法（late-bound）调用。该章首先讲述委托对象，然后介绍如何与事件类成员一块使用委托对象。
- **第12章** 命名空间主要是在本书的前面部分介绍的。但是，我们在这一章还要进行更详细的讨论。其中包括命名空间的声明，怎样使用命名空间来组织代码，以及怎样避免标识符名称的冲突。
- **第13章** 结构是另外一种C#的对象类型。结构在机制上类似于类，但是拥有不同的语义，因为结构包含了与类相同的许多成员。该章着重讲解结构的独特性能以及它与类的类型的差别。
- **第14章** 基于接口的编程在为对象暴露公用合同方面是很优秀的。该章介绍了在C#的程序中怎样实现接口的全面和详细的信息。
- **第15章** C#是一种强类型语言。该章重点介绍怎样维护类型的安全性，而同时又能够在用户定义的类型之间转换。

第三部分 在C#中使用类库

第三部分介绍了C#可以使用的几个类库。

- **第16章** 虽然Web服务是一个新的事物，但桌面图形用户界面（GUI）应用程序仍然有大量的用户。该章在着重讲解如何使用Windows窗体控件创建简单的用户界面的同时，还讲解了Windows窗体库。
- **第17章** 该章是专门用来讲解文件的输入/输出（I/O）的，说明如何使用C#写入和读取文件。另外，该章还讨论了数据流。
- **第18章** XML是集成在基类库里的，从而奠定了它在未来应用程序中的重要性。该章的例子着重介绍使用C#现有的库来操纵XML数据。
- **第19章** C#的程序数据库主要是通过ADO.NET类来实现的。该章的例子展示了怎样使用ADO.NET类来进行传统的和以Web为基础的访问。
- **第20章** ASP.NET的讨论重点介绍了Web软件的开发。ASP.NET的网页可以使用C#编写的代码来进行开发。该章的例子包括怎样使用C#代码来编写程序控件和Web程序。
- **第21章** 远程化是可扩展的分布式对象编程技术。C#非常适合于使用远程化技术来开发程序。该章不仅展示了如何对远程对象进行编程，同时也展示了怎样使用该技术的扩展特性。
- **第22章** 用C#建立Web服务特别方便。该章的例子使用了现有的ASP.NET基础结构和.NET

实用程序来开发Web服务，从而展示了这种开发的简单性。

第四部分 C#高级主题

第四部分的几个高级主题主要针对提高性能和企业级编程项目讲述。

- **第23章** 现代编程语言都支持多线程。C#也不例外。该章介绍怎样创建多线程程序及怎样实现线程同步。
- **第24章** 基本类库中包括有支持Internet通信的类。该章讲解了通过TCP/IP套接字（socket）进行通信的方法以及怎样使用实现HTTP协议的类。该章中还有关于HTTP的Web和SMTP电子邮件编程的例子。
- **第25章** 在编程中进行管理的大部分信息都能够处理文本数据。该章专门用来讲解如何应用C#进行字符串管理，包括String和StringBuilder类，字符串格式化和正则表达式等。
- **第26章** 当数组类型不能够完全满足应用程序数据结构的需求时，可以考虑用集合。该章的例子说明了怎样使用基类库的几个集合，以及怎样使用代码来创建自定义的集合，以便实现需要的接口。
- **第27章** 属性是用于在程序中添加声明性的功能的C#语言要素。该章说明了属性的用途以及如何创建自定义的属性。
- **第28章** C#有一种叫做反射（reflection）的能力。该能力使程序能够检查自身的信息。该章的例子展示了怎样执行反射，以及怎样动态地建立能够在存储器中运行或保存到可执行文件的程序。
- **第29章** 本地化（localization）是使程序为不同的文化表示信息的过程。该章中提供了如何为多种文化实现本地化的一些例子。
- **第30章** 实际上会有许多项目需要重用现有的本机库（native library）、与操作系统代码交互并执行低级的操作。该章讲述了怎样在C#中完成这些操作。
- **第31章** 复杂的系统应该有检测运行时刻问题的机制。该章的例子中展示了如何应用基类库的相关元素对运行时刻进行调试和跟踪。
- **第32章** 另外一个重要的运行时刻任务是捕获应用程序性能数据的能力。性能计数器用于收集统计数据，这些数据能够帮助分析特定情况下应用程序在运行时刻的执行情况。
- **第33章** 目前有大量的COM和COM+程序员。该章中的程序展示了如何从C#调用COM以及如何从COM调用C#。基类库中包括有使C#程序能够使用COM+服务的企业服务类。该章中的例子展示了怎样使用COM+服务来完成任务，其中包括事务处理（transaction）、对象池（pool）等等。

第五部分 C#的环境

第五部分深入讨论C#的环境。

- **第34章** C#程序的内存是由高性能的垃圾收集器管理的。该章讨论了垃圾收集器的工作原理及其为什么它如此有效。该章中提供了如何与垃圾收集器交互的例子，及帮助对象控制清除内存处理的不同方法。

- **第35章** 支持跨语言编程的C#的操作环境。该章的例子演示了如何应用不同语言创建代码并将它们编译到同一个程序中。关于通用语言规范（CLS）的介绍中阐明了影响C#程序的CLS的特定区域。
- **第36章** 要了解C#程序怎样运行以及管理代码的意义，你必须对CLR内幕的理论有所了解。其中大部分的信息已经作为其他C#语言元素的一部分进行了说明。但是，该章的材料确能帮助巩固读者已经掌握的内容。
- **第37章** C#是以配件（assembly）形式部署的。因此，了解什么是配件以及怎样使用它们是很重要的。该章介绍了配件的组成并介绍了定义它们的各种属性。
- **第38章** 该章介绍了基于代码的安全性方法，以及它们是怎样交互来保护系统不受代码影响的。其他部分介绍了怎样使用代码和属性来实现基于代码和基于角色的安全性。

第六部分 附录

第六部分由编译器的补充材料、.NET类库组件的概述和一些有用的其他资源组成。

- **附录A** 在这个附录中提供了各种编译器选项。
- **附录B** 这个附录提供了.NET库的概况。
- **附录C** 这个选取的C#和.NET的Web站点的索引是很有帮助的。

原书书名：C# Unleashed (ISBN: 0-672-32122-x)
原出版社网址：www.samspublishing.com

参加本书翻译工作的人员除封面署名外还有：尹军琪、侯凌燕、文兵、王京、谢炜、罗霄岚、冯小龙、王文彬等。

目　录

第二部分 用C#编写对象和组件

第三部分 在C#中使用类库

第四部分 C#高级主题

第五部分 C#的环境

第六部分 附 录

第一部分 C#的基础知识

第1章 C#的环境

本章概括地介绍了C#的操作环境。为什么熟悉它的操作环境很重要呢？有两个原因。原因之一是，在学习C#时常常会重复出现类似的问题。例如，库中有哪些功能，语言中内置了哪些功能等。本章中所讲述的内容有助于评估哪些功能可以满足需求。

了解C#环境的另外一个原因与标准化有关。一些软件的工程师需要了解C#的环境以便开发跨平台的应用程序。充分了解C#的操作环境所要求的元素将有利于标准安装，并且可以更顺利地进行跨平台的软件开发。

1.1 通用语言基础结构

通用语言基础结构（Common Language Infrastructure，CLI）主要用于分布式组件及服务的创建和执行。它通过以下方法来达到此目的：使用不同语言编写的程序的共同操作，而且给予程序自我描述的功能并提供支持多平台的执行环境。CLI是由以下四个主要部分构成的：

- 通用类型系统（Common Type System，CTS）。
- 通用语言规范（Common Language Specification，CLS）。
- 元数据（Metadata）。
- 虚拟执行系统（Virtual Execution System，VES）。

CTS支持来自不同编程语言的通用数据类型。它为许多语言使用CLI打开了方便之门。除此之外，CTS还支持类型安全性的概念。利用这个概念就能够使程序具有更强的生命力和更高的安全性。

CLS是CTS的子规范。它可以增强与其他语言编写的程序之间的通信。在程序或类与CLS兼容时，也就说明它能够可靠地用于跨语言的环境。CLI具有一组叫做基类库（Base Class Library，BCL）的库。整个BCL都是兼容于CLS的。

元数据为代码提供了自我描述功能。这是一个强大的功能，它使配件能够展示它们使用各种工具的能力。元数据还使程序的功能可以通过运行时方法的调用来动态地实现。这项功能可用于后联编的需求和软件开发工具。

通过使用CTS、CLS和元数据，CLI还能够支持功能强大的虚拟执行系统。VES所提供的服务包括管理代码的执行和安全性。VES可在由兼容于CLI的编译器产生的通用中间语言（Common Intermediate Language，CIL）代码上运行。

注意 托管（managed）代码是指VES处理存储器分配和代码执行的方式。为了提供可靠的安

全性，VES对托管代码有完全的控制能力。托管代码被编译成中间语言（CIL），在CIL中能够通过VES进行校验。传统的计算机语言，例如C和C++，可以看作是非托管（unmanaged）语言，因为它们被直接编译成机器代码，并且只有程序员在代码中建立的控件。所有的C#代码都是托管代码。

经常容易混淆的概念是托管代码和不安全代码。C#允许程序员使用某种被认为是不安全的语言元素。这些语言元素包括指针和堆栈上的存储器分配。不安全的代码的主要优点之一是能与API直接接口。通常API的参数或结构要求使用指针。虽然存在在不安全代码中进行不安全操作的可能，不安全代码仍是托管的。

1. 基类库

CLI基类库是一套完整的库，它是为支持多种语言而设计的。CLI兼容性的级别是由所实现的库的子集来定义的。迄今为止有三种符合ECMA标准的BCL配置库：核心配置库、精简配置库和完全配置库。实现产品中会有更多，以支持正式的兼容性认证。图1-1所示的就是这些库之间的关系。中心是核心配置库。精简配置库除了核心配置库之外，还包括自己的全部库。同样，完全配置库也包括它自己和所有其他的库。

注意　.NET中带有四个语言。这些语言是Visual Basic.NET、Visual C++.NET、Visual JScript.NET及Visual C#.NET。目前大约有50个其他的独立软件开发商在将语言移植到.NET平台上。

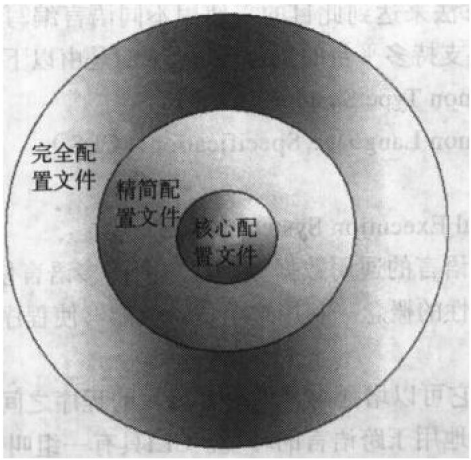

图1-1　CLI配置库

2. 核心配置库

核心配置库是一种实现产品能自称与CLI兼容性所必需支持的最小的库。这是能够支持C#和VES的最低要求。

3. 精简配置库

精简配置库主要是为嵌入式设备等小型实现所指定的。它通常用在资源紧缺的环境中。

4. 完全配置库

完全配置库包括25个CLI程序包。这是一个相当完整的库，它支持各种范围的服务。这个配