万水计算机编程技术与应用系列

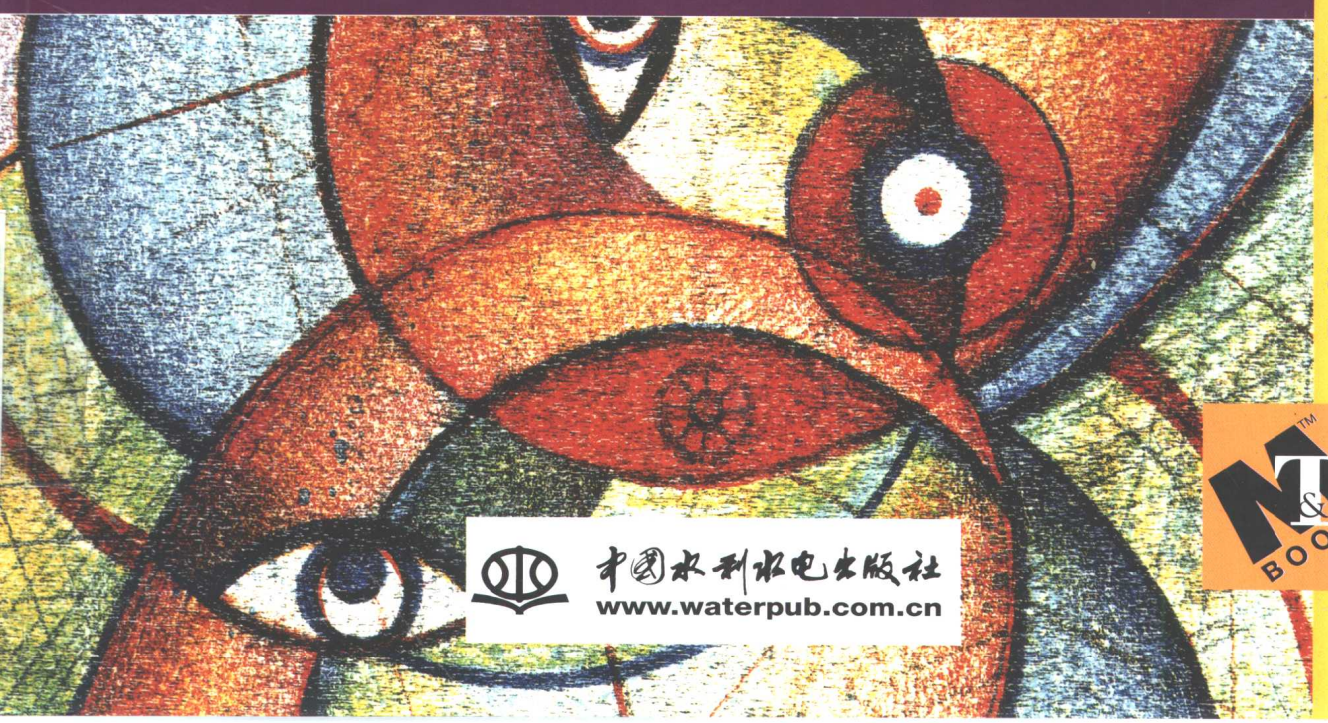
C# COM+ 编程指南

C# COM+ Programming

[美] Derek Beyer 著
龚小平 史艳辉 杜大鹏 管英强 译
杜国梁 审校

如果想要在今天的COM+服务与未来用C#语言开发出来的下一代服务之间自由地“跳华尔兹”是需要一些高超技术的，而本书可成为您的“舞蹈教练”。

——Michael Lane Thomas （.NET系列丛书编辑）



中国水利水电出版社
www.waterpub.com.cn

PROFESSIONAL MINDWARE™

M&T BOOKS™

万水计算机编程技术与应用系列

C# COM+编程指南

[美] Derek Beyer 著

龚小平 史艳辉 杜大鹏 管英强 译

杜国梁 审校

中国水利水电出版社

内 容 提 要

.NET 框架是 Microsoft 公司为适应 Internet 发展与市场形势而提出的开发平台。C#是 Microsoft 公司为.NET 框架量身定做的首选语言。本书向读者概要地介绍了.NET 框架和通用语言运行库的基本概念，进一步讲解了新的.NET 框架与已有的 COM+（组件对象模型扩展）技术的互操作方法，即如何在 COM+中使用.NET 组件以及如何在.NET 框架中使用已有的 COM+组件；如何用 C#语言创建全新的符合.NET 规范的 COM+组件，这些组件可用于事物处理、安全、事件、对象共享、事件排队以及远程处理。

为了读者更好地阅读本书，作者在附录中还介绍了 C#语言的要点。在本书所附的光盘上包括了本书中的所有源代码。

本书适合有志于学习 Microsoft 新的.NET 框架平台的开发人员。

北京市版权局著作权合同登记号：图字 01－2001－4560

图书在版编目（CIP）数据

C# COM+编程指南 /（美）拜尔（Beyer，D.）著；龚小平等译. —北京：中国水利水电出版社，2002

（万水计算机编程技术与应用系列）

书名原文：C# COM+ Programming

ISBN 7-5084-1005-X

Ⅰ.C… Ⅱ.①拜…②龚… Ⅲ. ①C 语言－程序设计②因特网－程序设计 Ⅳ. ①TP312②TP393.4

中国版本图书馆 CIP 数据核字（2002）第 013334 号

书　　名	C# COM+编程指南
作　　者	[美] Derek Beyer　著
译　　者	龚小平　史艳辉　杜大鹏　管英强
出版、发行	中国水利水电出版社（北京市三里河路 6 号　100044） 网址：www.waterpub.com.cn E-mail：mchannel@public3.bta.net.cn（万水） sale@waterpub.com.cn 电话：（010）68359286（万水）　63202266（总机）　68331835（发行部）
经　　售	全国各地新华书店
排　　版	北京万水电子信息有限公司
印　　刷	北京蓝空印刷厂
规　　格	787×1092 毫米　16 开本　15 印张　327 千字
版　　次	2002 年 3 月第一版　2002 年 3 月北京第一次印刷
印　　数	0001—5000 册
定　　价	30.00 元（含 1CD）

凡购买我社图书，如有缺页、倒页、脱页的，本社发行部负责调换

译 者 序

自从 2000 年以来，相信许多人都从各种媒体上听说过.NET 了。Microsoft 为.NET 给出的官方定义是".NET 是 Microsoft 用于 XML Web 服务的平台"（.NET is Microsoft's platform for XML Web services.）。这是 Microsoft 为适应 Internet 发展和市场竞争需要而提出的产品战略。.NET 框架成为 Microsoft 所有产品共享的平台。Microsoft .NET 平台包括许多建立在 XML 和 Internet 业界标准的产品族，可提供开发、管理、使用和体验 XML Web 服务的各项功能。XML Web 服务将成为目前已经在使用的 Microsoft 应用程序工具和服务器的一部分，而且将被引入到新产品中，以便满足所有的业务需求。XML Web 服务允许应用程序通过 Internet 共享数据，而不管使用的是什么操作系统和编程语言。其中共享数据的格式就是 XML（eXtended Markup Language，扩展标记语言）。C#语言是 Microsoft 在 C++和 Java 的基础上为.NET 框架特意开发的完全面向对象的语言，是.NET 平台上的首选开发语言。但本书并非讲解 C#语言的教科书，只是使用了 C#语言编写书中的所有代码。

本书的书名中还有一个缩略语 COM+，其全称是"组件对象模型"的增强版本。这也是 Microsoft 的一项较新的技术。1997 年 Microsoft 在加利福尼亚的 San Diego 召开的专业开发人员大会上宣布了 COM+计划。这是一种对组件对象模型（Component Object Model，COM）的扩展。COM+的目的是使用任何语言或任何工具来创建并使用软件组件。相对于.NET 框架来说，COM+已经是较为传统的技术。那么在.NET 框架下可以使用以前开发的 COM+组件吗？答案是肯定的，这也正是本书内容要达到的主要目的。.NET 系列丛书编辑 Michael Lane Thomas 如是说："如果想要在今天的 COM+服务与未来用 C#语言开发出来的下一代服务之间自由地'跳华尔兹'是需要一些高超技术的，而本书可成为您的'舞蹈教练'。"这句话可谓一语中的地道出了本书的目的和宗旨。

如果读者有以上知识背景或是想要了解以上知识背景，则本书意义就突现出来。译者们对以上知识并不很精通，对.NET 和 C#等新知识有极大的好奇心。翻译的过程也就是学习的过程。错误在所难免，敬请广大读者（特别是行家）提出宝贵意见。

本书是多人努力的成果。龚小平、史艳辉、杜大鹏、管英强参加了翻译工作，其中，龚小平翻译了前言和第 1 章到第 5 章，史艳辉翻译了第 6 章到第 8 章，杜大鹏翻译了第 9 章到第 10 章，管英强翻译了附录。全书由杜国梁审校并统稿。参加本书其他工作（录入、打印、校对等）的人有魏天超、梁国珍、任建畅、马相生、刘发来、董明、迟春和杨天华等。在此对所有对本书作出贡献的人表示感谢。

译者

2001 年 11 月 20 日

于防化指挥工程学院

作者简介

Derek Beyer 现在作为 Web 开发专家工作于密歇根州的 Grand Rapids 市的 Meijer Stores 公司。Derek 经常指导其他开发人员程序设计和开发技巧。他也负责实现并维护核心基础组件，如 Web 和应用程序服务器等。Derek 为公司的开发人员开发并宣讲有关 MTS、COM+、Visual Basic 和 Active Server Pages 领域的开发指南。

Derek 也在以芝加哥为基地的咨询公司 March First 作顾问工作。他所从事的项目范围涉及从开发主要基于 Internet 的消费者站点的应用程序到 SAP R/3 应用程序的 Web 集成。Derek 也在 COM+和.NET 主题的用户组会议上演讲。

闲暇时间，Derek 通常在健身房参加体育锻炼，或者是享受诸如狩猎和钓鱼这样的户外活动。

致　　谢

真的很感谢为本书辛苦而勤奋工作的审稿和编辑团队。虽然本人的名字出现在封面上，但本书其实是团队努力的结果。Matt Lusher 和 Eric Newman 担当了本项目的项目编辑并提供了很大的帮助。Matt 以其专家的身份和丰富的幽默感使紧张的时光变得更轻松。Chris Jones 努力查找出在本人犯困和睡眼朦胧时所犯的语法错误。一个好的组稿编辑室能把全书结合成一个整体并使每个人都保持快乐，而 Sharon Cox 这方面的能力就很令人惊叹。Sharon 无疑使本人通常要处理的问题减少了许多。谢谢你，Sharon！我对 Hungry Minds 的产品部也欠了许多人情，这些人士都受累于本人的绘图和屏幕抓图中的错误。我还要感谢 Rolf Crozier，本书早期的组稿编辑。Rolf 向 Hungry Minds 提出了出版本书的主意并使之启动。

在人们所喜爱的领域内遇到的最好的事情就是具有共同观点并能向其学习的人们。Steve Schofield 是我开始学习新技术时曾经遇到的最热心的人士。他对.NET 的兴趣很具感染力。Steve 也在本人需要将本书变成现实时向本人提供了与 Hungry Minds 内有关人员的联系方法。Nick McCollum 是本书的令人敬畏的技术编辑。他使我保持诚实并帮助本人使许多主题更贴近读者。还要感谢许多关键的 Microsoft 雇员，如 Mike Swanson 和 Shannon Paul。Mike 总是能提供帮助和本人所需要的一些东西。他也以微笑和点头来面对本人对许多技术问题的抱怨。Shannon 给本人提供了关于 COM+事件的关键信息。他也使作者在进入一个主题时能顺利地写作下去。谢谢你，Shannon。

本人现在已经明白写书是一项很大的工程。没有人能作出这么大的努力而没有来自家庭或朋友的支持。本人幸运地有一个很优秀的支持系统。系统的基础是我的父母。父亲向我举例说明真正的工作道德是什么。父亲是我曾经看到的最刻苦工作的人。很感激这些已经从我脑海消失的工作道德。母亲为我提供了无条件的支持和鼓励。必须感谢她能理解在我潜心写作本书时几个月很难看到我。最后但并非不重要，我必须感谢 Jacque。Jacque 是很特别的朋友，在编写本书的过程中忍受了本人偏执的冲动。她能够以她的同情和积极的能量在我低沉时使我振作起来。谢谢。

前　言

欢迎阅读《C# COM+编程指南》一书。如果已经购买本书或正打算购买，您可能有许多希望本书能回答的问题。最常见的问题是“COM+消亡了吗？”及“COM+在.NET 应用程序中的角色？”。第一个问题的答案很明确，就是“没有”！Microsoft 包含在 Windows 2000 中的 COM+技术对.NET 程序员仍然可以使用。实际上，一些在早期只有 C++程序员能用的 COM+技术现在对 Visual Basic .NET 和 C#程序员也都可以用。第二个问题总是有点难以回答。从作者这里得到的典型回答是“视具体情况而定”。COM+中的技术如分布式处理和列队组件只能在 COM+中找到。需要确定是否应当使用特定 COM+服务时要自问的问题是“我的应用程序中需要这种服务吗？”。如果答案是肯定的，就可以自由使用 COM+。如果答案是否定的，COM+就不适合于该应用程序。

本书使用的所有代码都使用了新的编程语言 C#。C#是特意为.NET 开发的面向对象的编程语言。实际上，.NET 应用程序是用 C#能编写的惟一的一种应用程序。贯穿全书作者都会指出有助于编写更好的 COM+组件的 C#语言的特征。虽然所有的代码都是在 C#中编写的，但如果喜欢的话这些示例也可以用 C++编写。

本书的读者对象

COM+不是适合编程新手的主题。如果读者以前从来没有开发过应用程序，本书就可能不太合适。说到 COM+，话题就总是牵涉到分布式计算。如果已经开发过应用程序，特别是分布式 Web 应用程序，本书所讨论的主题对读者就很有意义。

如果读者是.NET 编程或 COM+编程的新手也不要害怕。本书的第一部分介绍了.NET 的基础知识及其与 COM 组件的交互。第一部分提供了理解.NET 应用程序如何工作及如何与传统的 COM 组件交互所需的环境。作者强烈建议读者在阅读任何其他章节前先阅读第 1 章。第 1 章向读者介绍.NET 环境。如果不理解该环境如何工作，本书的其余部分对读者就没有太大的意义。

对那些 C#的新手来说，附录 C 提供了对该语言的介绍。附录 C 介绍该语言的基本功能如数据类型、循环、流程控制语句以及在本书其余部分所用到的特定的语言特色。

本书假设读者并不熟悉 COM+编程。每章都介绍每种 COM+服务的基本功能和问题。读者在用本书学习如何开发 COM+组件时不必是一个有经验的 COM+开发人员。

本书内容

本书分成三个部分。每部分都提供理解下一部分所需的信息。本书的组织提供在.NET 中积累 COM+编程的技能和理解 COM+编程所需的逻辑发展。

第一部分　与 COM 的互操作

第一部分介绍名为通用语言运行库（Common Language Runtime）的基本的.NET 运行库环境。因为每个.NET 应用程序都运行于通用语言运行库中，如果要用 C#开发 COM+组件则理解这种环境是极其重要的。第一部分的内容包括了与 COM 世界进行互操作的方法。说明了如何从 C#应用程序中使用传统的 COM 组件。也说明了如何编写 COM 客户可使用的 C#组件。如何开发使用 COM 组件或从 COM 组件中使用的分布式应用程序，理解 COM 与.NET 的互操作是很重要的。

第二部分　COM+的核心服务

第二部分介绍 COM+的核心服务。所有的核心服务如分布式处理、基于角色的安全性、松散耦合事件、列队组件及其他都是在第二部分介绍的。这部分的各章顺序是按从较为简单的服务到较高级的服务（尽可能好地）组织的。

第三部分　高级 COM+计算

本书的最后一部分即第三部分介绍 COM+较高级的一些主题。第三部分介绍.NET 远程处理框架。.NET 远程处理框架为开发者提供通过网络调用组件方法的途径。正如读者将会看到的，用 C#编写的 COM+组件可通过类层次插入到远程处理框架中。第三部分也讨论了现在的 Windows XP 所拥有的 COM+、Internet 信息服务器（Internet Information Server）和 Microsoft 消息队列（Microsoft Message Queue）的新功能（所有这些技术都在本书中使用了）。许多 COM+新功能都把重点放在为 COM+组件提供更稳定的环境上。

本书所使用的约定

任何书籍都使用一些约定帮助读者更好地理解原文。本书也不例外。作者在本书中使用了排版和编码约定以帮助读者更清晰地理解原文。

排版约定

因为这是一本有关编程的书籍，作者介绍了许多代码示例。作者几乎原样复制了每个代码示例（较长的代码则有自己的清单表号）。解释特定代码示例的段落经常引用示例中的代码。

如果引用示例中的代码，总是使用等宽字体。下面是第5章中的一个示例。

```
using System;
using Microsoft.ComServices;
[assembly: ApplicationAccessControl(
    AccessChecksLevel = AccessChecksLevelOption.ApplicationComponent
  )
]
public class SecuredComponent {
    // some method implementations
}
```

注意在属性标签内使用了关键词 assembly。这会告诉 C#编译器该属性是装配级的。在属性的声明中，通过使用 AccessChecksLevelOption 枚举把 AccessChecksLevel 属性设置成应用程序和组件。

上面的代码示例（从 using System 开始的行）全部设置成等宽字体。上面一段解释了代码示例。在这一段中从代码示例中引用了关键词，如 assembly、AccessChecksLevel 和 AccessChecksLevelOption。不管在段落中的哪个地方看到等宽字体，都肯定是前面的或将要出现的代码示例中所用的关键词。

代码书写约定

.NET 框架使用 Pascal 惯例命名其类、方法参数、枚举等。本书所使用的代码示例也遵守这个惯例。Pascal 惯例在名称中将每个单词的首字母大写。例如，如果编写一个访问顾客订单信息的类，就可能将它命名为 CustomerOrders。因为使用 Pascal 惯例，必须将 Customer 中的 C 和 Orders 中的 O 大写。使用这个约定有助于使代码示例更具可读性。

本书使用的图标

本书介绍的许多主题都有相关的主题。很多时候如果要理解所讨论的中心主题则理解这些相关主题是很重要的。然而，如果离主题太远则很容易失去读者。为了既介绍这些重要信息又不失去读者，把这些主题放到了“注意”中。例如：

“注意”解释相关主题。它们也用来提醒读者 C#有助于编写优秀的 COM+组件的特殊功能。

目　　录

第二部分 COM+的核心服务

第三部分 高级 COM+计算

第一部分　与 COM 的互操作

第 1 章　理解.NET 结构

本章内容包括:

* 在通用语言运行库（Common Language Runtime）内载入和执行代码
* 自动内存管理
* 装配件
* 应用程序域
* 通用类型系统（Common Type System）

.NET 框架（.NET Framework）试图解决 Microsoft Windows 环境中许多与应用程序开发和部署有历史联系的问题。例如使用 Visual Studio 6 及其早期版本在 C++中不能编写类而且在 Visual Basic 内不能直接使用类。COM 已经通过允许已编译组件之间经二进制协议进行对话来试图减轻这种不便。然而，COM 也有其缺陷。COM 没有提供运行时发现组件所提供的服务的明白易懂的方法。.NET 框架则通过所谓的“映像”（reflection）概念提供了解决这一问题的机制。错误处理是该框架解决的另一个问题。根据所作出的 API 调用，此 API 调用可能产生一个错误或返回一个错误码。如果返回错误码，程序员必须具有可能返回的常见错误方面的知识。该架构通过为所有错误产生一个异常简化了错误处理。Framework 库提供了对传统上属于 C++程序员领域的低级特性的访问。Windows 服务、COM+ Object Pooling（COM+对象共享）以及对如 HTTP、SMTP 和 FTP 之类的 Internet 协议的访问现在已处于 Visual Basic .NET 或 C#开发者的牢牢掌握之中。

正如读者所看到的，.NET 框架为运行于其环境内的应用程序提供了许多瞄准执行领域的服务。所有为.NET 编写的程序（包括用 C#编写的 COM+组件）都运行在称为通用语言运行库（Common Language Runtime，CLR）的环境内。为运行于 CLR 内编写的应用程序被看作是托管代码。托管代码可利用 CLR 提供的服务。某些这类服务，如垃圾收集（Garbage

Collection），是自动提供的。其他服务，如对软件的版本编号，则要求程序员干预。

本章包括由CLR提供的服务。对CLR的理解能提供用C#开发COM+组件所需的适当基础。

1.1 在通用语言运行库内载入和执行代码

正如前面所提到的，CLR提供许多简化开发与部署应用程序的服务。CLR能提供这些服务的部分原因是所有应用程序都运行在称为虚拟执行系统（Virtual Execution System，VES）的相同执行引擎上。实际上，它是编译器支持和某种允许CLR提供其服务的规则的运行库执行的组合。这一节讨论应用程序可用的运行库支持以及提供这些服务所需的编译器和VES支持。在整个这一章里，用术语class（类）和dll（动态链接库）来说明这些概念，因为它们直接应用于COM+编程模型。这些概念适用于所有类型与文件格式（exe文件和dll文件）。

1.1.1 Microsoft中间语言和元数据

编译C#应用程序时，不会得到所期望的典型文件，而是包含描述组件的Microsoft中间语言（Microsoft Intermediate Language，MSIL）代码和元数据的可移植的可执行文件（Portable Executable，PE）。MSIL是CLR解释的指令集。MSIL告诉CLR如何载入与初始化类，如何调用对象上的方法，以及如何处理逻辑与算法运算。在运行时，CLR的一个组件，即即时编译器（Just In Time Compiler，JIT），将MSIL指令集转换成操作系统可以运行的代码。

MSIL指令集对任何硬件或操作系统来说都没有什么特殊之处。Microsoft已经把基础设置成允许将MSIL代码移植到支持CLR的其他平台。虽然Visual Studio .NET和Windows 2000提供了运行CLR的惟一工具和平台组合，但可以想像，CLR是可以移植到其他平台的。如果真是这样，MSIL代码就可直接移植到其他平台。当然，使用如COM+所提供的特定平台的服务会使应用程序较难移植到其他平台。

C#代码：真可移植？

如果应用程序使用了COM+服务或其他只针对对Microsoft或另外的软件商的特定的服务，就有机会碰到那些服务在其他平台上不可用的情况。另一方面，如果应用程序使用了System .Net Sockets名称空间提供的如TCP/IP所支持的服务，应用程序就可能相对来说易于移植。TCP/IP是一种大多数平台可能支持的支持度和通用性较好的服务。只要这种支持在平台之间没有很大的差别，这类代码就有很高的可移植性。这里要明白的一点是MSIL和CLR为不同的软件商提供了一组相容的标准。虽然为CLR编写真正可移植的代码还不是现实，但不久就会实现。

正如前面所提到的，在dll（Dynamic Link Library，动态链接库）和MSIL中都存在元数据。元数据在整个CLR中广泛应用，如果要理解.NET框架是如何操作的，它就是一个要掌握的重要概念。元数据提供CLR进行注册（进入COM+目录）、调试、内存管理和安全所需的关于应用程序的信息。对COM+组件来说，元数据告诉CLR和COM+运行库诸如类应当使用的事物处理级别和共享组件的最大和最小共享池容量方面的信息，提到只是一小部分。