

微软.NET程序员系列

- ◆ 欧美读者评价 ★★★★★
- ◆ 著名编程专家精心编著
- ◆ .NET核心开发组鼎力推荐
- ◆ 内容经典，示例典型
- ◆ C++程序员通往.NET世界的金钥匙

Microsoft®

Visual C++ .NET

技术内幕(第6版)

Programming with Microsoft®
Visual C++ .NET Core Reference

(美) George Shepherd, David Kruglinski 著

潘爱民 译
Visual Studio .NET产品组 审校



Microsoft
.net

清华大学出版社

微软.NET 程序员系列

Microsoft Visual C++ .NET 技术内幕 (第 6 版)

(美) George Shepherd, David Kruglinski 著
潘爱民 译

清华大学出版社
北 京

内 容 简 介

本书是经典的“Visual C++技术内幕”的最新版。Visual C++一方面是一个C++实现，它提供了一个功能强大的集成开发环境；另一方面，它提供了一些C++类库，其中最具影响力的当然是MFC。随着Visual C++的发展，MFC已经不是Visual C++中惟一的类库了，其他还有ATL、STL等。此外，Visual C++还融合了大量的开发工具，特别是与Internet相关的开发支持。最后，本书也介绍了Visual C++对.NET的支持。总体而言，本书涵盖了当前最新的Visual C++的主流技术，既有对老技术的继承和更新，也有对新技术的精辟阐述。所以，阅读本书可以全面拓宽读者的知识面。从这层意义上讲，本书相当于Visual C++的百科全书，它浓缩了Visual C++的精华。

本书面向中级Visual C++程序设计人员，并可以作为高年级计算机专业学生及软件学院学生的程序设计教程。

Microsoft Visual C++ .NET 技术内幕(第6版)

Programming with Microsoft Visual C++ .NET, Sixth Edition(Core Reference)(0-7356-1549-7)

George Shepherd with David Kruglinski

Copyright © 2003 by George Shepherd.

Original English Language Edition Copyright © 2003 by George Shepherd.

Published by arrangement with the original publisher, Microsoft Press,
a division of Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文版由 Microsoft Press 授权清华大学出版社出版。

北京市版权局著作权合同登记号 图字：01-2003-0828

版权所有，翻印必究。举报电话：010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目(CIP)数据

Microsoft Visual C++ .NET 技术内幕(第6版)/(美)谢菲德(Shepherd,G)，克鲁格林斯基(Kruglinski,D.)著；
潘爱民译.—北京：清华大学出版社，2004

(微软.NET 程序员系列)

书名原文：Programming with Microsoft Visual C++ .NET, Sixth Edition(Core Reference)

ISBN 7-302-08931-0

I. M… II. ①谢…②克…③潘… III. C语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 006838 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

责任编辑：汤斌浩

封面设计：陈刘源

印 刷 者：北京季蜂印刷有限公司

装 订 者：三河市昭亮装订有限公司

发 行 者：新华书店总店北京发行所

开 本：185×230 印张：59.75 插页：5 字数：1434 千字

版 次：2004 年 7 月第 1 版 2004 年 7 月第 1 次印刷

书 号：ISBN 7-302-08931-0/TP·6318

印 数：1~5000

定 价：106.00 元(含 1 张光盘)

本书如存在文字不清、漏印以及缺页、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)62770175-3103 或(010)62795704

《微软.NET 程序员系列》序

自 2000 年 6 月微软宣布自己的 .NET 战略以来,在不到两年的时间里,.NET 已经从战略变成现实,.NET 带来了全新的、快速而敏捷的企业计算能力,也给软件开发商和软件开发人员提供了支持未来计算的高效 Web 服务开发工具。作为微软 .NET 战略的重要组成部分——Visual Studio .NET (中文版)已于 2002 年 3 月 22 日正式在中国推出。

Visual Studio .NET 是一个功能强大、高效并且可扩展的编程环境。它充分展现了应用程序开发的潜能,并提供了生成应用程序所需的工具和技术。这些应用程序将给当今的企业、机构提供强大的支持,并推动下一代基于 XML Web 服务软件的发展。

有了 Visual Studio .NET,那些对全世界数以百万计的程序员来说曾一度极端复杂、费时费力,甚至让人望而生畏的编程任务已不再神秘。更重要的是,Visual Studio .NET 使开发人员能运用既有的技能和知识来迎接新的编程挑战。

10 年前,Visual Basic 1.0 成为数以百万计的开发人员所使用的革命性的应用程序开发语言。现在,Visual Studio .NET 为未来的 10 年做好了准备。

微软出版社为了配合 Visual Studio .NET 的推广以及 .NET 技术的普及,邀请 Visual Studio .NET 项目开发组的核心开发人员和计算机图书专业作家精心编写了英文版《微软.NET 程序员系列》丛书。该丛书自面市以来,在美国图书销量排行榜上一直高居前列,颇受好评,成为程序开发人员和网络开发人员了解 .NET 技术的权威工具书。尤其是《Microsoft .NET Framework 程序设计》一书,长期占据美国及欧洲此类书籍的排行榜冠军位置,成为程序开发人员必读的书籍。

清华大学出版社为了满足中国广大程序开发人员和网络开发人员学习最新技术的渴望,在微软出版社的配合下,从《微软.NET 程序员系列》这套丛书中精选了 50 余本翻译成中文,以满足国内广大读者的需要。这套丛书阵容庞大(且在不断扩充之中),几乎涵盖了 .NET 技术及其应用的各个方面;也正因为如此,翻译和编辑加工的工作量也大得惊人。但为了保持国外优秀技术图书的魅力,同时使读者领会新技术的真谛,本丛书的译者都是经过严格筛选的、具有很高的翻译水平和丰富经验的技术人员;另外,我们还聘请微软公司 Visual Studio .NET 产品组的技术专家审读每一本书,确保在技术上准确无误。

相信这套丛书定会帮助程序开发人员、网络开发人员以及那些具有一定编程基础的中、高级读者,快速、全面地掌握 .NET 技术,协助他们为技术生涯的下一个 10 年做好准备,为培养新一代软件人才、推动中国软件产业的快速发展起到积极的作用!

这套丛书分为 3 个子系列：技术内幕系列、语言参考系列和程序员系列。目前，已出版和在编的共有 36 本，已从 2002 年 6 月份起陆续和读者见面。

- 技术内幕系列

目前共有 7 本：

- ◆ 《Visual C++ .NET 技术内幕(第 6 版)》

本书是 Visual C++ 和 MFC 开发的经典著作。它秉承了第 4 版和第 5 版的风格，已根据该编程语言的最新版本 Visual C++ .NET 进行了全面更新和补充，是 .NET 时代的 C++ 程序员必读的教材。此外，本书仍由第 4 版的译者潘爱民先生翻译。

- ◆ 《Microsoft .NET Compact Framework 技术内幕》

- ◆ 《Visual Basic .NET 技术内幕》

- ◆ 《Visual C# .NET 技术内幕》

- ◆ 《ADO.NET 技术内幕》

- ◆ 《Microsoft .NET 程序设计技术内幕》

- ◆ 《Visual J# .NET 技术内幕》

- 语言参考系列

目前共有 3 本：

- ◆ 《Visual Basic .NET 语言参考手册》

- ◆ 《Visual C# .NET 语言参考手册》

- ◆ 《Visual C++ .NET 语言参考手册》

- 程序员系列

目前共有 27 本。详细书目，请参见本书彩插页。

其中，Microsoft .NET Framework 1.1 类库(1-4 卷)是 .NET System 命名空间大全，直接源自 Microsoft .NET Framework。这 4 卷书包含了 .NET Framework Class Library System 命名空间的全部详细信息。帮助开发人员充分利用 .NET Framework 来开发 Web 应用程序、客户端应用程序和 XML Web 服务。

随着技术的发展，我们将根据读者的需要，不断增加新的书目。

丛书版式特色

本丛书在风格上力求文字精炼，并采用小 5 号字编排，内容紧凑，版面清晰美观，易于阅读。此外，书中还安排了一些特色段落，提供正文之外的一些细节知识。



注意：提醒阅读和操作过程中应注意的事项，避免出现错误或问题。



提示：指点一些操作捷径或实用技巧，使您少走弯路，阅读和操作更为高效。



要点：总结关键知识点或操作细节，助您适时掌握要领。



注：提示首次出现的编程元素，以及书中涉及到该元素的其他位置以供参考。



警告：阅读和操作过程中，应特别注意的事项，如果使用不当，可能会导致严重后果。

尽管我们倾心相注，精心而为，总有疏忽纰漏，恳请广大读者不吝赐教与指正，我们定会全力改进，以期在后续工作中得以完善。

本丛书在创作过程中得到了微软(中国)公司的大力支持。本丛书能够顺利出版，更是倾注了无数幕后人员的汗水和心力。在此，对他们的辛勤劳动一并表示衷心感谢！

编者

2004 年 7 月

译者序

终于可以松一口气了，翻译一本上千页的图书实在是一个不小的工程。想起五年之前我第一次翻译计算机图书(本书的第四版)的时候，期望借这本书将自己的编程知识总结和回顾一下。时隔五年之后再翻译第六版，算是将过去的编程经验又仔细地复习了一遍，在翻译过程中，有许多地方又唤起了当初摸索 Visual C++ 和 MFC 时的那种感觉，因为那时摸索得来的经验与本书中的许多内容不谋而合，所以，每每碰到这种地方，总是感觉特别亲切，作为 C++ 程序员的那种掌控一切的豪情油然而生，也真切地希望每一个 Visual C++ 程序员都能够掌握这些技术，并享受这种因知识和技术而带来的人生体验和乐趣。当然，除了怀旧之外，本书中新增的内容又让我手心发痒，恨不得回到程序员的角色之中，利用这些新的技术(特别是 .NET 和 Internet 部分)做一个实实在在的项目。一本讲解技术的书能够将译者和读者的情绪带动起来，并且让他(们)产生亲近技术的欲望，这恐怕正是本书的魅力所在吧。

这几年来，关于 Visual C++ 的书籍多如牛毛，但这本书一直占据着最重要的地位，我有幸翻译了本书的两个重要版本，所以借第六版中文版出版之际，谈谈我对本书的一些体会，算是译序吧。

本书与 Visual C++ 的发展历程

了解本书历史的读者都知道，本书的开创者 David Kruglinski 在完成了第四版之后因为事故身亡，以后的版本是别人续写的。其中，第六版是由 George Shepherd (*MFC Internals* 的作者)续写的，以 George 对于 MFC 和 Visual C++ 的理解深度，以及他的写作经验和教学经验，新版本只会增色而不会逊色。相信读者在读过这本书之后能够体会到这一点。

在我手上，除了英文第六版之外，还有本书的第二版(中文版)、第四版(中文版和英文版)。书很沉，很有分量，拿在手里沉甸甸的，当然更重的是内容。关于本书具体的内容请读者参看本书的文件夹或者引言部分。这里我想说明我对本书技术倾向的看法。

Visual C++ 开发环境包含哪些内容呢？这是被很多人误解的一个问题。有人把 MFC 和 Visual C++ 等同起来，也有人把 Visual C++ 和 C++ 混淆起来，实际上，它们都有明确的分界线，在软件开发过程中，它们有各自的含义和用途。

Visual C++ 一方面是一个 C++ 实现，它提供了一个功能强大的集成开发环境；另一方面，它提供了一些 C++ 类库，其中最具影响力的当然是 MFC。随着 Visual C++ 的发展，MFC 已经不是 Visual C++ 中惟一的类库了，其他还有 ATL、STL 等。此外，Visual C++ 还融合了大量的开发工具，特别是与

Internet 相关的开发支持。本书几乎涵盖了所有这些内容,所以,阅读本书可以拓宽知识面。从这层意义上讲,本书相当于 Visual C++ 的百科全书,它浓缩了 Visual C++ 的精华。

Microsoft Visual C++ 的版本更新基本上反映了这 10 年来 Windows 平台上软件技术的发展历程。在 Visual C++ 5.0 版本之前,一切变化的发生都伴随着 MFC 的成熟与壮大,在此期间 Visual C++ 主要用于开发桌面应用程序。到 5.0 版本之后,MFC 已经不再是发展的重点,Visual C++ 整体的发展呈多元化趋势,先是 COM 技术的普及,然后是 Internet 和 Web 开发技术的不断深入,最新版本中又引入了 .NET 框架的支持。

本书的发展历程也反映了 Visual C++ 的技术路线。在本书第四版(对应于 Visual C++ 5.0)出版的时候,桌面应用技术已经非常成熟,MFC 的结构完全定型,同时 COM 技术在 Internet 应用中大行其道,由此产生了一批 Active 技术,包括 ActiveX 控件、ASP、ADO、Active Document 等。在第四版中我们可以看到关于这些技术的讲解。(第五版又引入了 ATL 相关的内容。)

现在呢,最大的变化,也是最激动人心的变化应该是 .NET 架构的引入,当然也随之带来了一大批新的技术。从长远的发展来看,.NET 必将取代现有的大量应用技术,但不是所有,另一方面,新的技术和老的技术将会共存相当长的时间。

本书第六版基本上反映了现在的技术发展状况,MFC 仍然存在,但是一些新兴的用户界面元素也被引入 MFC 中。并且,本书也全面地讨论了 COM 和相关的开发支持,包括 MFC 和 ATL 对于 COM 的支持,这是目前仍然非常实用的两项技术。本书还讨论了针对 Internet 的程序设计技术,包括 MFC 对于 Winsock 和 Wininet 的支持,以及 DHTML 和 ATL Server 等。最后,本书也介绍了 Visual C++ 对于 .NET 的支持,其中包括托管的 C++ 以及如何用托管的 C++ 来开发 Windows Forms、ASP.NET 和 ADO.NET 应用程序。本书所有这些内容涵盖了当前的主流技术,既有对老技术的继承和更新,又有对新技术的精髓阐述,所以,通过这本书你可以全面地学习到这些实用开发技术。

本书特色和新增内容

虽然这是一本再版的书,并且有差不多三分之二的内容与以前的版本(这里主要指第四版)相同,但是,它仍然有自己的特色以及丰富的新增内容。

本书的叙述风格基本上没有变化,仍然按照以前的模式,既有原理性的说明,也有指导性的介绍,还有很好的例子来帮助理解一些技术细节。并且,本书在讲解的时候也很简洁,与以前的版本相比,本书的内容增加了不少,但是篇幅基本上没有明显的增加。作者对于前面的传统内容部分有所精简,以便为后面的内容腾出空间。

本书新增的内容可以分几个方面来看:

1. 本书的前三部分内容主要针对桌面应用程序的开发,这既是 MFC 的传统优势,也是本书的传统优势。我们应该可以感受得到,这几年桌面应用技术还是有了一些变化的,特别是随着 Windows 2000/XP 的推出;尽管 MFC 的基本框架保持不变,但在许多细微之处还是新增了许多界面元素,例如,除了 SDI 和 MDI 之外,Windows 2000 又引入了一种新的界面风格:多顶级窗口界面(MTI)。而且 Internet Explorer 的新版本更是引入了许多新的 UI 元素,包括一些扩展的控件,以及 Rebar 等。Visual C++ .NET 为这些新的特性提供了支持,本书也介绍了相应的编程技术。
2. 本书第四部分介绍了 COM、自动化、ActiveX 和 OLE。COM 是一项底层的组件技术,这部分介绍了 COM 的原理,以及 MFC 和 ATL 提供的最基本的支持。要想在不到 300 页的篇幅中全面地介绍所有这些技术是不可能的,但是本书这一部分中,不仅成功地叙述了这些基本知识,同时还介绍了几项关键应用技术,包括自动化、统一数据传输、用 ATL 来开发 ActiveX 控件,以及用 OLE DB 模板来开发 OLE DB 使用者和 OLE DB 提供者。
值得一提的是,除了传统的 C++编程之外,Visual C++ .NET 还引入了一种新的编程模式,被称为属性化的编程,用于支持 COM 组件的开发,它的基本思想是,允许 C++类直接引用 COM 类和 COM 接口的一些属性,由编译器来解释这些属性并生成必要的代码,从而减轻了程序员在开发 COM 组件过程中一些不必要的负担。
3. Internet 开发是一个不可错过的部分。本书第五部分在介绍 MFC 对 Winsock 和 WinInet 支持的基础上,又介绍了动态 HTML 和 ATL Server。请读者不要将 ATL Server 与 COM 联系起来,实际上,ATL Server 主要是针对 IIS 的一个开发工具,它被用来支持对于 IIS 服务器的扩展。
4. 最后一部分介绍 Microsoft .NET 程序设计。这是本书完全新增的内容,虽然 .NET 的标准语言应该是 C#,但是 Visual C++ .NET 也提供了全面的支持,它是通过扩展标准 C++而实现的,被称为托管的 C++。被托管的代码将运行在 .NET 的公共语言运行库之上,这是对组件技术的新发展,它使得应用程序运行在一个更加完善的管理环境之中,底层大量的设施可以被直接使用,并且软件的发布和协作将更加理想。对于开发人员呢,负担减轻了,他们可以更加关注于自己的应用,而无需为一些琐碎的细节操心。这一部分首先介绍了 .NET 中最为核心的公共语言运行库,然后介绍了托管 C++扩展,以及它的编程示例。最后用三章篇幅分别介绍了如何用托管 C++来开发 Windows Forms、ASP.NET 和 ADO.NET 应用程序。

如何使用本书

面对一本涉及面如此广阔的计算机程序设计图书,我们该如何来使用呢?要想掌握书中每一项技术,不仅需要大量的时间,还需要扎实的基本功,但是,掌握这些技术对于把握现代软件设计环境又非常有帮助。

按照这本书的内容结构,读者当然可以有所取舍,但是最好的做法是,首先按顺序阅读一遍,如果有可能的话,对于其中一些关键例子最好配合在机器上进行实战练习。然后,对于感兴趣的话题可以仔细钻研,或者把这本书当作参考书,以后随时翻阅。

这本书讲解的深度属于中等程度,读者最好有一些 C++ 和 Windows 应用开发方面的基础。对于书中讲述的每一项技术,本书也只是起到基本的引导作用,如果读者希望进一步深入钻研的话,需要阅读系统提供的源代码或者查阅 MSDN Library 中的详细讲解。但是,通过这本书,你可以快速地领会每一项技术。

因为这本书既有原理性的讲解,也有示例说明,甚至还有分步讲解的实战过程。所以,本书也适合于用作研究生或者高年级本科生的程序设计课程的教材。我在本书英文版刚刚出版之际,曾用本书作为北京大学软件学院研究生课程“程序开发环境分析与实践”的教材,取得了不错的效果。

曾经听人提到过,说这本书只是教会你怎么用向导工具来生成一个应用程序,离开了向导工具你还是什么也不会。这种观点并不正确,我说过这本书是原理和实战的结合,向导是提高开发效率的好帮手,但是如果你明白了原理,难道还会离不开向导吗?难道会看不懂代码吗?我很赞成使用 Visual C++ 提供的向导,但是一定要知道这些向导帮你做了哪些事情,这很重要,否则当向导不工作的时候,你就失控了。

最后,我想说明一点,在翻译本书时,本书中的屏幕拷贝图以及 Visual C++ .NET 集成环境中的一些称谓(比如各种向导的叫法)我都保留了英文版本,而没有将它们转译过来。国内的大多数程序员都会使用中文版的 Visual Studio .NET, Visual Studio .NET 中文版几乎彻底地进行了汉化,所以,读者在阅读本书的时候也许会感到些微的不便。但是,本书在适当的地方作了相关的说明,相信这些中英文差异不会影响你的阅读和理解。

在本书的翻译过程中,我参照了第四版(中文版)的内容,发现第四版中有不少错误,所幸的是,我这次还有机会把这些错误一一改正过来。同时也顺带着把原书中的一些错误也改正了。当然,翻译过程中必定会带来一些语言表达上的偏差,希望这些偏差不会影响您的阅读。

潘爱民

2004 年 2 月 14 日

于北大燕园

致 谢

写致谢这一部分总是最愉快的——每一位为书稿付出努力的人现在手头的工作都已经接近尾声，最后要做的事情是向大家表示谢意。因为只有作者的名字出现在封面上，所以有时候很容易会忘记那些为本书做出贡献的其他人，尤其对于像本书这么大的工程。你们为本书付出了大量的时间和精力，我要谢谢你们。

感谢 Sandy Daston 和 Ted Shepherd——我的家人，谢谢你们在我写作期间给予的支持。

感谢 Denise Bankaitis，作为本书的责任编辑，您一直提醒我本书的重要性（作为 Visual C++ .NET 的关键参考书），并且协调我们这个团队中的其他成员（包括 Julie Xiao、Ina Chang、Danielle Bird、Juliana Aldous、Joel Panchot、Carl Diltz 和 Gina Cassill）。

感谢 Julie Xiao，您使本书的内容更加精确。

感谢 Ina Chang，您使本书的语句更为通顺。

感谢 Danielle Bird 和 Juliana Aldous，作为本书的组稿编辑，你们推动了这本书的项目，并使它保持正常的运行。

感谢 Joel Panchot，您的艺术设计使本书看起来更精美。

感谢 Carl Diltz 和 Gina Cassill，你们的排版使本书看起来更棒。

我还要感谢 DevelopMentor 的伙伴们，你们提供了一个良好的环境，以及一个适合于思考和学习现代计算技术的团体。你们是最棒的。

引 言

Microsoft Visual Studio .NET(特别是 Visual C++ .NET)加强了 Microsoft 对于 Internet 技术的关注,而 Internet 技术正是 Microsoft .NET 体系结构的核心所在。除提供了对于 .NET 的率先支持以外, Visual C++ .NET 保留了以前您所熟悉的所有高工作效率特性,比如“编辑并继续”(Edit and Continue)、IntelliSense、AutoComplete 以及代码提示等特性。Visual C++ .NET 还包括了许多新的特性,比如针对 .NET 程序设计的托管代码扩展、对于属性化代码的支持,以及一个更为统一的开发环境。这些特性把 Visual C++ .NET 提升到了一个新的层次上。本书将帮助您快速掌握 Visual C++ 中引入的最新技术。

.NET、MFC 和 ATL

如今我们面临的技术更新总是日新月异。我们的办公室桌面从没有计算机,到 20 世纪 80 年代几乎每个人都有 一台运行 MS-DOS 的计算机,再到 90 年代中期几乎每台机器都在运行 Microsoft Windows 系统。技术的车轮又一次向前滚动了。在 90 年代后期,人人都在手工开发 Web 网站,使用的工具有原始的 HTML(Hypertext Markup Language,超文本标记语言)、CGI(Common Gateway Interface,公共网关接口)、ISAPI(Internet Server Application Programming Interface) DLL、Java 和 ASP(Active Server Page,活动服务器页面)。2000 年 7 月,Microsoft 向全球宣布,它将把公司未来的发展压在一个新的、名为 .NET 的技术方向上,从而改变 Web 网站的开发方式。

Microsoft 当前推动的是真正的 .NET。几年前,我们就可以在某个地方安放一台服务器,并申请一个 IP 地址,然后在服务器上放上一些内容,这样就算建成了一个 Web 网站。任何人只要用该网站的 URL 就可以在上面冲浪,并访问其上的内容。商务性质的企业已经充分使用了这类 Web 的优势,他们把有用的信息通过 Web 网站告知顾客。Web 也已经成为一个无价的研究工具和一个有效的新闻传播媒介。

未来的计算世界将紧密地与 Web 联系起来。然而,Web 网站不再局限于想方设法吸引人们的眼球,而是要吸引计算机本身。也就是说,通过 Web 服务,Web 站点将是可编程的。.NET 同时也将推动“由服务器来提供丰富的用户界面”的趋势。

随着重心转移到“Web 服务”和“基于服务器的用户界面”上来,看起来似乎独立的应用系统和基于客户端的用户界面程序将会逐渐消亡,而这正是 MFC(Microsoft Foundation Class)库的主要应用领域。但是,人们对于“丰富的客户端用户界面”的需求不可能一下子消失。许多人认为,PC 机

和分布式技术的出现将是“大型机和小型机上集中式处理时代”的终结。它表明 PC 和分布式技术只是增加了可用的计算资源。Web 服务的 .NET 架构，以及服务器提供的丰富的用户界面只是为软件开发人员增加了可用的选择而已。丰富的客户端用户界面对于许多类型的应用程序而言仍将非常重要。在一个系统中，既需要这些使用客户端用户界面的应用程序，同样也需要使用其他类型的用户界面（比如服务器生成的用户界面）的应用程序。

MFC 是一个非常成熟又比较容易理解的技术，它拥有大量的第三方扩展。至少在相当长一段时间内，MFC 仍将是编写功能全面的独立应用程序的最有效途径。本书的一部分内容将集中在 MFC 风格的开发上，但是我们将涉及 Windows Forms 的开发，这是编写客户端用户界面的 .NET 途径。

当然，接下去的问题是：COM 的地位何在？COM 已经解决了许多与分布式处理有关的问题，但是它有一些严重的缺陷——其中最根本的缺点都与组件的版本和类型信息有关。Microsoft 的 .NET 架构是以公共语言运行库为基础的。该运行库作为 .NET 中的互操作性标准，取代了 COM 的地位。我们将在本书的第六部分介绍 .NET 和公共语言运行库。

COM 和公共语言运行库分别代表了实现组件架构的不同途径，但是 Microsoft 已经将它们有机地融合起来，使之能够共存。在绝大多数的情况下，COM 和运行库之间的互操作性是非常平滑的。在 .NET 的世界中，您可能不会发现自己正在使用 COM 作为组件架构。然而，您可能会发现自己正在使用 ATL (Active Template Library, 活动模板库) 服务程序，这是一种高性能的编写 Web 站点的方法。

在本书的这一版中，我已经更新了有关 ATL 和 MFC 的内容，因为您将会发现，它们仍然非常有利用。更重要的是，我将向您演示，在迁移到 .NET 世界中的时候，如何调整您的传统代码（这听起来比“遗留代码”要好听，是不是？）。

托管的 C++ 和 C#

.NET 平台新引入了一种类似 C++ 的语言，称为 C#。C# 是一种 curly-brace-oriented (面向括弧) 的语言，但没有 C++ 的所有头痛之处。C# 之所以如此有吸引力，很大程度上在于它去掉了 C++ 中一些有问题的语言元素（比如原始的指针管理），同时又保留了 C++ 有用的特性（比如虚函数）。C# 编译器最终会产生托管的代码——运行在公共语言运行库之上的一种代码。

不过，整个世界不会在一夜之间完全转换到 C# 上来。仍然会有大量的 C++ 代码还不能转换。而且对于开发人员而言，他们也需要花一定的时间才能完全适应 C#。同时，.NET 也引入了对于 C++ 的扩展，允许产生托管的代码（即运行在公共语言运行库之上的代码）。C++ 的托管扩展将有助于减轻开发 .NET 平台上的软件的负担，因为这些扩展使得您可以快速地更新现有的代码，以便在 .NET 环境下工作。为了利用 C++ 来开发托管的代码，这意味着您需要在自己的代码中加上各种关键字。最后，当编译器的编译工作完成之后，C# 和托管的 C++ 可以被混编到同样的可执行代码中。在 .NET

世界中，您可能会发现，您将用 C# 编写新的组件，同时利用托管的 C++ 在原有 C++ 代码的基础上增加 .NET 特性。

.NET 和 Java 平台

最近几年，我们看到 Java 程序设计语言 and 平台已经引起了业界广泛的关注。Java 提供了一种有用的方法用来分发客户端的用户界面（通过 Java applet [小程序]），并且通过 Java Enterprise Edition 来提供企业解决方案，所以 Java 为 Internet 开发人员提供了极大的便利。现在，.NET 已经成为目前最好的 Internet 开发平台。Java 平台要求您使用 Java 语言编写所有的代码，.NET 则与此不同，它允许您使用多种语言实现同样的机器指令集。您可以使用 C++（这是本书的焦点所在）和它的托管扩展、Visual Basic .NET、C#，甚至大量第三方的 .NET 语言来编写您的程序。一旦您编写了源代码，它就被编译成中间语言，并最终在运行之前变成机器代码。因为 .NET 代码是由一个运行库托管的，所以您可以由此获得许多好处，比如垃圾回收和更好的代码安全性。

本书的适用对象

Visual C++ .NET 具有复杂的应用程序框架 (application framework) 以及对于 .NET 的支持，它是为专业程序员准备的，本书也是如此。我假定您已经精通了 C 语言——不用参考手册就可以写出一条 if 语句。我还假定您已经接触过 C++ 语言——至少学过这门课程或者读过 C++ 的书籍，但可能还没写过太多的代码。您可以将 C++ 语言的学习和法语的学习做比较。您可以在学校里学习法语，但您不可能说得很流利，除非您去一个说法语的国家，并开始与那里的人进行交谈。

Visual C++ 中的向导 (wizard) 可以为您节约时间，并且保证准确性，但是程序员必须要理解这些向导生成的代码，并且最终他们必须理解 MFC 和 ATL 库的结构、Windows 操作系统的内部工作机制以及 .NET 的工作方式。然而，我并不假定您已经了解 Windows 和 .NET 程序设计。我知道，精通 C 的程序员能够以 MFC 的方式和 .NET 的方式学习 Windows。此外，虽然了解 C++ 比了解 Win32 API 更重要，但您还是应该知道如何运行 Windows 以及基于 Windows 的应用程序。

即使您已经对 Win32 API 或者 MFC 库有经验了，这本书仍然可供您学习。首先，您可以学到一些新的特性，比如多顶级窗口界面 (MTI, Multiple Top-Level Interface) 和 Visual C++ .NET 向导。如果您尚未领会组件对象模型 (Component Object Model, 简称 COM)，则本书介绍的一些重要理论会让您理解 ActiveX 控件。您也可以学习 ATL 服务程序和 OLE DB 模板，而且还可以学到针对 Internet 的 C++ 程序设计 (包括动态 HTML)。最后，本书也包括了您在别处难以找到的新的有关托管 C++ 扩展的内容。

本书没有包括的内容

要在本书里涵盖 Windows 和 .NET 编程的所有方面是不可能的, 因此我把一些依赖于特定用途硬件和软件的主题排除了, 比如 MAPI、TAPI 和通讯口的访问等。我将介绍如何在一个应用中使用 ActiveX 控件, 以及如何使用 ATL 编写 ActiveX 控件, 但是, 有关的深入讨论我则推荐 Adam Denning 和他的书 *ActiveX Controls Inside Out* (Microsoft Press, 1997)。我建议您从 32 位内存管理、DLL 理论、多线程编程技术和 .NET 编程开始, 但如果您对这些内容缺乏了解的话, 您还需要参阅 Jeffrey Richter 编写的 *Programming Applications for Microsoft Windows*, 3rd Edition (Microsoft Press, 1997)。另本有用的书是 George Shepherd 和 Scot Wingo 的 *MFC Internals* (Addison-Wesley, 1996)。我也将带领您进入 .NET 空间, 但是关于运行库的核心内容我则推荐 Jeffrey Richter 的 *Applied .NET Programming* (Microsoft Press, 2002)。

怎样使用本书

当您开始学习使用 Visual C++ .NET 的时候, 最好首先从头到尾阅读本书, 它会给您提供最好的学习指导。以后, 您可以将此书用作一本参考手册, 通过目录来查阅一些内容。由于应用程序框架的许多元素之间是彼此紧密相关的, 而且许多概念也难以按章节完全分离开来, 因此本书不可能像百科全书那样包罗万象。在使用本书时, 最好打开联机帮助 (online help), 以备查找类和成员函数。

如果您对 Visual C++ 的早期版本很有经验的话, 则可以浏览一下第一部分以使对新特性有个大致的了解, 然后跳过第二部分中关于 MFC 基础的部分, 但是请阅读高级话题部分。同时, 也请阅读关于 .NET 的部分。软件开发团队中的大多数工作都在朝着这个方向努力, 并且 Visual C++ .NET 全面支持 .NET 编程模型。

本书的组织

正如本书目录中所列出的那样, 本书主要包括六大部分和附录。

第一部分: Windows、Visual C++ .NET 和应用程序框架基础

在这部分中, 既有抽象理论的阐述, 又有实际的应用程序, 我力求使这两者达到一种平衡。在这部分中, 我快速地回顾了 Win32 和 Visual C++ .NET 的基本组成, 进而逐步将 MFC 应用程序框架以及文档-视图结构 (document-view architecture) 展现在读者面前。同时, 在这部分中还给出了一个通

过 MFC 类库中的类来构造的简单的“Hello, world!”程序,该程序只需 30 行代码。

第二部分: MFC 的本质

在假定您已经熟悉基本的 Windows API 的前提下, MFC 库文档简捷而又完整地介绍了应用程序框架的所有元素。在本书的第二部分中, 我将把读者的注意力集中在应用程序框架的主要构成元素之一——“视图”(view)上, “视图”实际上是一个窗口。在这里, 您将从 C++ 和 MFC 库中的类的角度来学习和掌握熟练的 Windows 程序员早已熟知的一些知识。您还将使用 Visual C++ .NET 工具, 这些工具避免了以前 Windows 程序员不得不忍受的一些繁杂的编码工作。

第二部分介绍了很多内容, 包括用位图进行图形编程、对话框数据交换、ActiveX 控件的使用、32 位内存管理, 以及多线程编程。其中的练习将帮助您编写比较复杂的基于 Windows 的程序, 但这些程序并没有充分利用应用程序框架的高级特性。

第三部分: MFC 的文档-视图结构

这一部分介绍了应用程序框架的核心——文档-视图结构。在这部分中, 您将会了解什么是“文档”(document) (这里的“文档”并不是文字处理工作中通常所说的文档, 它有更进一步泛化的含义), 也将学习如何将文档同第二部分中所介绍的视图联系起来。一旦您动手编写出了一个自己的文档类, 就会禁不住对 MFC 库简化文件 I/O 及打印的方式感到非常惊讶。

另外, 在这部分中也会接触到命令消息处理(command message processing)、工具栏(toolbar)、状态栏(status bar)、切分框架(splitter frame)以及上下文相关帮助(context-sensitive help)。同时, 这一部分中还将介绍单文档界面(SDI)、多文档界面(MDI)以及多项级窗口界面(MTI), 多项级窗口界面是当前基于 Windows 的应用程序(比如 Microsoft Word)的标准。

第三部分还讨论了如何用 MFC 库编写动态链接库(DLL)。您将了解扩展 DLL(extension DLL)和常规 DLL(regular DLL)之间的区别。

第四部分: COM、自动化、ActiveX 和 OLE

要介绍 COM 不是一本书能够完成的。第四部分从 MFC 的角度出发, 使您开始学习基本的 COM 理论。接下来您将学习 Automation(自动化), 这是连接 C++ 和 Visual Basic for Applications (VBA) 的纽带。您还将熟悉统一数据传输(uniform data transfer), 还要学习复合文档(compound document)和

嵌入对象(embedded object)的基础知识。您还将学习ATL类库对于OLE DB的支持。

第五部分：Internet 程序设计

这部分从 Internet 的技术指南开始，涉及了 TCP/IP 协议和 Internet 程序设计的基础。您将学到如何利用 ATL 服务程序开发服务器端程序，还将学习如何编写动态 HTML。

第六部分：.NET 和远景

Internet 作为软件开发的新领域，目前正在不断发展。Internet 不再仅限于构建一些 Web 网站供人们浏览，而是允许人们对 Web 网站进行编程。网络连接早已存在，但是在 XML 出现之前，人们还无法对“在 Internet 上发送方法调用”达成共识。.NET 中两个最主要的推动力是 Web 服务和基于服务器的用户界面。.NET 完全支持这些概念，并且还支持一种新的编写客户端用户界面的方法：Windows Forms。第六部分介绍了 .NET 的内容，以及您在 .NET 平台上可以做哪些事情。这部分中包含的章节涉及到公共语言运行库和托管的代码，以及利用 C++、ASP.NET 和 ADO.NET 来编写托管的组件。

附录

附录 A 给出了消息映射宏和对应的消息控制函数原型的列表。Class View 中的 code wizards(代码向导)通常会为您生成这些代码，但有时您必须手工添加一些条目。

附录 B 描述了 MFC 应用程序框架的运行时类信息和动态创建系统。这是独立于 RTTI(runtime type information, 运行时类型信息)的特征，而 RTTI 现在已经成了 ANSI C++ 的一部分。

Win32 和 Win16

有一些老的计算机仍然在使用 Windows 3.1。然而，再花费钱去为老的平台编写新程序有些不太值得。这一版的《Visual C++ .NET 技术内幕》是关于 32 位编程的，针对的是 Microsoft Windows 98/Me 和 Windows NT/2000/XP 平台上的 Win32 API。如果您实在需要进行 16 位编程的话，我建议您去看本书的第二版。