PROGRAMMER TO PROGRAMMER™

Visual C++ .NET:

A primer for C++ developers

Written and tested for final release of .NET v1.0

Visual C++ .NET 编程经典

——从 C++ 到 Visual C++ .NET 快速进阶

Aravind Corera Stephen Fraser 等著 康博 译

清华大学出版社
http://www.tup.tsinghua.edu.cn

Visual C++ .NET 编程经典

——从C++到 Visual C++ .NET 快速进阶

Aravind Corera
Stephen Fraser
等著

康　博　译

清华大学出版社

(京) 新登字 158 号
北京市版权局著作权合同登记号：01-2002-0260

内 容 简 介

在微软的.NET 浪潮中，Visual C++是惟一既能编写托管代码，又能编写非托管代码的语言，因此在集成原有代码和新的.NET 代码时具有不可替代的作用。本书第一部分介绍 Visual C++的新增特性、托管的 C++代码、程序集、属性和类库等。第二部分讲述非托管的 C++代码、属性化编程、ATL 的新功能、ATL Server 和 ATL Server Web 服务等。

什么是.NET？C++有什么变化？它在.NET 中的地位和作用如何？曾经是微软组件软件开发核心的 COM 在.NET 到来之后又将扮演什么角色？是否有必要将非托管代码转换为托管代码？又该如何进行转换？您的这些问题都将在本书中找到答案。

本书适用于想了解 Visual C++的新功能，以及想在.NET 环境中开发 C++代码的 C++程序员。

图书在版编目(CIP)数据

Visual C++ .NET 编程经典/(美)科雷拉等著；康博译.—北京：清华大学出版社，2002
书名原文：Visual C++ .NET：A Primer for C++ Developers
ISBN 7-302-05715-X
Ⅰ.V… Ⅱ.①科…②康… Ⅲ.C 语言-程序设计 Ⅳ.TP312
中国版本图书馆 CIP 数据核字(2002)第 057341 号
出 版 者：清华大学出版社(北京清华大学学研大厦，邮政编码：100084)
http://www.tup.tsinghua.edu.cn
责任编辑：徐燕萍
印 刷 者：北京昌平环球印刷厂
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16 **印张**：25.75 **字数**：659 千字
版　　次：2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷
书　　号：ISBN 7-302-05715-X/TP・3372
印　　数：0001～5000
定　　价：46.00 元

出版者的话

随着国际互联网的快速崛起和迅猛发展，计算机之间的互联需求越来越迫切，而目前计算机硬件设备的不兼容性严重束缚了互联网的发展，引发了新一轮的跨平台软件的开发浪潮。软件商纷纷推出新的战略规划和解决方案，Microsoft 提出的.NET 战略就是其中的经典之作。

在经历这场浪潮的洗涤和考验过程中，全球的软件开发人员都迫切需要了解新的软件技术和开发思路。为了满足国内 IT 从业人员的需要，清华大学出版社从 Wrox 出版公司引进了若干套编程系列丛书，“入门经典”系列是其中不可或缺的入门之作。作为世界著名的编程技术图书的出版公司，Wrox 推出的这套“入门经典”系列丛书主要面向编程的初学者、需要了解.NET 策略的程序员，以及需要迅速掌握多门编程工具的程序员。该丛书依旧秉承了 Wrox 公司“由程序员为程序员而著（Programmer to Programmer）”的创作理念，每本书均由世界顶级的编程高手执笔。他们站在资深程序员的高度，循序渐进地为初学者讲述了.NET 的理念和构架、编程基本思想、编程语言基础、程序的控制和调试、Windows 应用程序的开发、对象编程技术、数据库访问技术、Web 程序开发和.NET 构架应用等最新的软件开发知识，同时辅以大量操作性强的程序为示例，为读者提供了清晰的编程思路和宝贵的编程经验。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码和获取相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前　言

面对众多对微软.NET 计划的评论，如果您开始考虑花时间来学习 Visual C++是否是一项明智的投资，那将是勿庸置疑的。Visual C++是 Visual Studio 的旗舰产品(还记得那张“用 Visual C++创造”的宣传画吗？)，但在对 ASP.NET，Visual Basic .NET，特别是在 C#的介绍中几乎没有提及 C++。难道这真的意味着 C++在 Windows 下一波的发展浪潮中毫无用武之地吗？在这本书中，我们会证明这种想法并不客观。

首先很重要的一点是，当您应用 Visual C++的时候，您不必考虑.NET，甚至 Windows 编程。无论是在课堂上还是在家里，标准版对于初学标准 C++编程的人来说都是很好的选择。在 Visual C++ .NET 中，微软公司进一步改进了它的集成开发环境，并在 Visual C++ .NET 中继续沿用了一直以来采用的 ISO/ANSI 标准。

其次，COM 和 ATL 都做了较大的革新。ATL 库在开发上已经超出了 COM 编程的范围，它首先在 3.0 版本中引进了 CWindow 以及相关的功能，现在这种开发工作还在继续：出现了更多的实用类和 ATL Server —— 为在 ISAPI 框架上创建优化的 Web 应用和服务提供的一套全新的类。进一步来讲，COM 编程本身也通过引进一些新属性有了新的突破，这些新属性使您不必再管理单独的 IDL 文件，同时提供了另外的选择使您不必再编写详细而重复的代码。

最后，微软对 C++语言本身也作了扩充，使之成为.NET 中出色的一员。大多数.NET 代码将可能用 Visual Basic 或 C#来编写，还有很多老的代码需要在短期或中期内整合到这些新的应用程序中。Visual C++ .NET 提供了惟一的方法使遗留代码与新代码相兼容，而且一些新的特性就是以此为目的而开发出来的。

本书概要

这本书非常注重实际：它的目的是告诉有经验的 C++程序员他们所需要掌握的关于 Visual C++ .NET 的知识。要做到这一点，就有必要详细讨论.NET Framework 本身的知识，但这决不是本书的惟一目的。在本书中，您同时会了解到 COM，ATL 和 Web 服务的新特性，以及 C++在其中所扮演的角色。

更准确地说，本书的前半部分(第 1~6 章)将单独讨论.NET 内容以及它对 C++程序员的重要意义。第 1 章会简单介绍值得您在应用程序开发中使用的.NET 新特性，然后将介绍微软对 C++做出的更改，这使我们能访问那些新的特性。了解了一些托管 C++的知识后，我们要从 C++的角度来学习.NET 的一些关键概念，从程序集、属性、元数据开始，最后是作为类库的 Framework 与 MFC 及 STL 的对比分析。

在本书的后半部分(第 7-11 章)，我们将讨论“非托管”的 C++。首先要看老的代码如何与为.NET Framework 而编写的新代码实现互操作—— 这是其他任何编程语言都无法实现的。之后，我们还要看看属性驱动的 COM 编程、ATL(现在是 7.0 版本)的新特性、ATL Server 和 ATL

的 Web 服务。如果您要了解微软平台下 C++在软件开发中仍能起到的关键作用，本书就是为您而准备的。

客户配置

本书以及其中的代码在已经发布的 Microsoft Visual Studio .NET 中做过测试。因而本书要求的最低配置也同样为：PII-450 CPU ，Microsoft Windows NT 4.0（或更高版本），至少 128MB 内存，3GB 左右的空闲硬盘空间。书中大部分的例程都能在标准版的 Visual C++ .NET 中运行，但一些说明性的例子用到了 Visual Basic .NET 和 Visual C# .NET。

客户支持和反馈

我们很重视读者的反馈，当然很希望能知道您对本书的看法：哪些是您喜欢的？哪些是您不喜欢的？您认为下次我们如何能做得更好？您可以把意见通过用户回执卡告诉我们，也可以发信到 feedback@wrox.com. 请您在信中务必告诉我们书名和书号。

源代码和升级

本书所有的源代码都能通过 wrox.com Web 站点得到。当您打开 Web 站点http://www.wrox.com/，首先可以通过所提供的搜索引擎和书名列表来找到您要找的书。然后在书的详细页面中单击“Download Code”链接，这样您就能得到书中的所有代码。

您所下载的文件是 WinZip 的压缩文件。当您把文件存入硬盘后，需要用 WinZip 或 PKUnzip 把文件进行解压缩。解压缩的时候，请确保您的软件中“使用文件夹名” 的选项已经打开。

勘误表

我们尽一切努力使本书尽可能准确。然而我们也知道，出错是无法避免的。如果您能在书中发现错误并及时告诉我们，我们将非常感激。通过给我们指出错误，您能为下一位读者节省时间，而且还能帮助我们为读者提供更高品质的书籍。把您的建议通过 E-mail 发给我们，您提供的信息会及时得到反馈。如果您的指正是正确的，我们将把它发送到勘误页或在再版的时候作出修正。

如果您想了解别人发现的错误，请登录到 http://www.wrox.com ，通过搜索引擎找到书名。然后在该书的页面中，单击“Book Errata”链接。在该页中，您能找到已更正过的错误。您也可以通过单击“Submit Errata”链接告诉我们您所发现的错误。

技术支持

如果您需要和一位了解细节的专家讨论书中的问题，发 E-mail 到 support@wrox.com ，请在邮件主题中写上书名和书号的后 4 位。一封典型的 E-mail 应该包括以下内容：

(1) 书名，书号的后 4 位，出错内容所在的页数，行数。

(2) 您的名字，您的联系信息，以及您所发现的问题。

我们不会给您发垃圾邮件。为了节省大家的时间，我们需要知道一些具体细节。当您发出一封 E-mail 会得到如下支持：

• 客户支持：您的信息会转给我们的客户支持人员，他们将是第一个帮助您的人。他们会对您对本书提出的一般性问题或是关于我们 Web 站点的问题作出迅速的答复。

• 编辑：更深层次的问题会被反映到技术编辑那里，他们负责对这些问题作出答复。这些技术人员在某一方面很有经验，能很好解决一些细节问题。问题被解决后，我们会把解决方案公布在 Web 站点上。

• 作者：最后，可能会有一些问题编辑也无法回答，那么他会向作者寻求答案。我们尽力不让作者为一些小的问题分心，但他们和我们一样也都很乐意来处理这些特殊的问题。所有 Wrox 的作者都会对他们的书提供技术支持。他们会把问题的答案寄给客户和编辑，让所有的读者都能从中受益。

需要说明的是，Wrox 能提供的技术支持仅限于我们出版的书。如果您的问题超出一般范围，您可以通过 http://p2p.wrox.com 的公共社区来寻找答案。

p2p.wrox.com

如果您想和作者及同行进行讨论，请加入 P2P 邮件列表。我们独一无二的系统除了能为您提供一对一的 E-mail 支持外，还能通过邮件列表、论坛、新闻组使您实现与程序员之间的联系。请放心，您的问题会得到很多 Wrox 作者还有其他行业专家的帮助。在 p2p.wrox.com， 您能看到许多不同的列表，无论您是正在阅读本书，还是在开发自己的应用程序，这些信息都会对您有所帮助。

如果您想订阅邮件列表，仅需采取如下步骤：

(1) 登录到 http://p2p.wrox.com。

(2) 从左边的菜单里选择合适的类别。

(3) 单击您想加入的邮件列表。

(4) 按照订阅的提示，填入您的 E-mail 地址和密码。

(5) 回复您收到的确认邮件。

(6) 使用订阅管理器加入更多的列表，然后设置邮件的选项。

目　　录

第1章 Visual C++的新增功能

许多年以来，C++已经成为开发高度优化的应用程序和组件最流行的语言。尽管 Java 取得了巨大的成功，但要开发高效代码时，C++仍然是全球千百万程序员首选的编程语言。

新版本的 Visual Studio 选择了与微软的.NET Framework 同时发布，.NET Framework 为创建 Web 应用程序和单机程序提供了一个强大的、语言中立的开发环境。为了跟上发展的步伐，Visual Basic 已经升级为 Visual Basic .NET，ASP 被升级成为 ASP.NET，还引进了一种全新的语言 C#。同时，微软还为众多 C++程序员推出了 Visual C++ .NET。

这些年，人们用 Visual C++ 开发了大量的程序，因而微软想要确保，无论技术如何更新，都要保持产品的向下兼容性。但同时，微软也意识到 C++ 程序员应该能够很好地利用新的平台。这一章我们有两个目标：首先我们站在一个适当的高度来评估.NET Framework，然后我们要研究一下 Visual Studio(现在叫 Visual Studio .NET)有哪些变化，这将会影响到您今后如何用 C++创建应用程序。

正如我们在前言中提到的，您不大可能用 C++来开发全新的.NET 应用程序，因为还有更好的选择。在我们阅读本书的过程中，这一点将变得越来越明确。在 .NET 中 C++的重要性在于，使.NET 出现之前开发的经过验证的代码与在.NET Framework 下开发的应用程序实现互操作。为了达到这个目的，我们首先要了解.NET 是如何工作的。

此外，并不是所有 Visual C++ .NET 的新增功能都与.NET Framework 有关，虽然它的名字叫做.NET。在本书的第 3 部分，我们将要看看对 COM 和 ATL 编程的改进，我们在本章也要接触有关这方面的内容。

本章要从 3 个不同的方面进行论述，如果您对.NET 和 Visual C++比较了解，对这些内容可以作快速的浏览。首先，我们来大致了解一下.NET Framework，概述在本书中要学到的一些关键特性。然后，我们要看看 Visual Studio 的集成开发环境都有哪些改变，其中一些只是界面上的变化，一些是基于新增功能做出的改变。最后，我们要来看一个小例子，它用到了第 1 节中提到的一些.NET 的新特性。接着还要测试一下 Visual Studio .NET 的调试能力。

1.1 .NET Framework

微软公司说.NET Framework 是“一个全新的计算平台，能够简化基于 Internet 的分布式软件的开发过程。”从应用的角度讲，这意味着每台安装了.NET Framework 的计算机在性能、外观、安全性上都有一个统一的操作环境。此外，通过采用许多机制，.NET Framework 使代码存储与执行的相对定位不再重要，而且它还简化了部署，版本化等棘手的问题。除了这些，它所提供的开发环境还能免除程序员很多传统的但很乏味的工作。

.NET Framework 由彼此独立又相关的两部分组成：公共语言运行时(CLR)和类库。大致来

说，CLR 是它为我们提供的服务，类库是它实现的功能。.NET 的大部分特性——垃圾收集，版本控制，线程管理等，都使用了 CLR 提供的服务，具体的方法我们接着会提到。另一方面，当人们说到.NET 启动了 Windows Forms 应用程序和 XML Web 服务时，他们实际上是在谈由类库所体现出来的功能。让我们先从 CLR 开始吧。

1.1.1 公共语言运行时

当您为.NET Framework 编译源代码(应用程序、库、控件等)时，您得到的目标代码并不能直接在处理器上运行。为.NET 进行编译实际是在为 CLR 进行编译，因为编译得到的代码并不是 CPU 能够识别的机器指令，而是一种叫作“微软中间语言(MSIL，或简称为 IL)”的新语言。MSIL 为一个“虚拟”CPU 定义了一套处理指令——已编译为 IL 的代码必须进一步编译为本机机器指令后才能在中央处理器上运行。

CLR 提供了一个实时编译器，用来把 IL 代码编译为本机机器代码。编译后的代码就会进行缓存，在程序中断时，这些代码可能会从系统中删除出。整个程序不会一次全部编译完成，而是把它们分成小的程序段，在需要的时候单独编译。一个正在运行的程序在其生命周期内也许不会用到所有的代码，就像在某一次运行中，用户不会用到应用程序所有的功能一样。

无论这个过程如何被优化，都没有比直接把源代码编译为机器指令的效率更高。事实就是如此，因而微软需要一些好的理由来说服我们去使用 CLR。当然，有很多理由已经做出了暗示，我们将在以后的章节中进行具体的描述。

1. 垃圾收集

C++程序员经常会因为内存泄漏而感到苦恼。泄漏是由于没能及时将不用的资源返还给系统造成的。如果您太急于返还资源，也可能会让情况变得更糟—— 因为可能会访问到无效的内存区域。现在，.NET Framework 的 CLR 为所有这些问题提供了解决方案。

CLR 提供了垃圾收集的功能(可以扩充为对整个生命周期的管理)。它负责跟踪有效的引用，并对引用进行计数。它会定时对所有的对象引用进行检测，如果检测到当前未被引用的对象，CLR 就会把它从内存中释放掉。当垃圾收集器完成了释放对象的工作后，它会通过压缩堆的方式整理垃圾，然后用连续的内存空间满足新的分配请求，这样的方式会更加高效。

2. 代码的可移植性

从上文的描述可以看出，CLR 能够使代码变得可移植：因为.NET 应用程序的的源代码必须被编译为 IL 代码，这些代码能够运行在任何提供.NET CLR 的平台上。今天的 CLR 还只能在 Windows 平台上实现，但这种情况很可能会改变——正在进行中的 Mono 计划会把.NET 引入 Linux 平台。

3. 语言的互操作性

为了与.NET 兼容，并确保各种应用程序之间的互操作性，所有.NET 语言都必须遵从微软的技术规范要求。.NET 语言必须是面向对象的，必须使用由“公共类型系统”(CTS)定义的一套标准数据类型。不仅微软已经把自己的语言做出修改，许多第三方公司也纷纷修改它们的语言来支持.NET Framework，其中包括 COBOL, Eiffel, Perl, Python 和 SmallTalk。

因而，从 CLR 的角度来看，所有的语言都是平等的，只要有一个能生成 IL 代码的的编译器就行。同样，一旦应用程序或库被编译为 IL 代码，它就与源代码的语言无关了。这样的话，我们就可以用这一种语言轻松地调用另一种语言编写的库函数，调用过程中传递的参数和返回值也都是兼容的。

CTS 与 IL 的结合还能够改进调试过程—— 不同.NET 语言的代码不仅可以出现在同一个应用程序中，而且可以使用统一的接口，在同一次会话中调试多种语言编写的代码。

4. 代码的安全性

CLR 根据不同的标准(其中比较重要的标准之一是代码的来源——本地机、局域网或是 Internet)对 IL 代码会有不同的信任度。在访问实际执行应用程序的计算机上的可用资源时，这里的信任度对各程序段所能访问到的资源作出了限制。

IL 代码被编译以后，CLR 会在真正的 CPU 上运行该代码前对其进行校验。根据用户现在的权限，这段程序可能有权使用第三方或其他开发小组成员编写的代码，也可能不行。CLR 确保了程序运行时实现应用级别的安全性。

5. 访问.NET Framework 类库

最后，为 CLR 编写的 C++程序会自动让您快速简单地访问.NET Framework 类库。.NET Framework 类库的内容非常丰富，ATL 甚至是 MFC 的功能和它相比也显得渺小。库中有各种类型，涉及到了 Windows 编程的方方面面，我们稍后会对其中的一些影响加以介绍。在这之前，我们先花一些时间来看看这些针对 CLR 的代码都有哪些特点，从而能为我们的工作带来那么多好处。

1.1.2　托管代码的属性

为 CLR 而编写以及使用 CLR 服务的代码叫“托管代码”，同样，那些未使用 CLR 服务的代码(也就是多年以来您一直编写的代码)叫“非托管代码”。正如我们先前讨论的，编译过的 CLR 代码实际是一种中间语言代码。不过我们还未讨论这种代码是如何分布的。

当您用 Visual C++ .NET 创建托管的 C++应用程序时，得到的是一个 EXE 文件；而当您在创建托管的 C++库的时候，您得到的是 DLL 文件。但现在的情况不一样了，您得到的文件将会以 PE(portable executeable)格式出现，其中不仅包含了 MSIL 代码，也包含了能描述一切的元数据：所使用的类型，所包含的类(包括类的方法和属性)，还有用到的其他文件类型。这些 PE 文件被称为“程序集(assembly)”，我们会在第 3 章进行更详尽的描述。尽管我们对 PE 文件只作了很少的介绍，但这些知识就能解释 CLR 是如何为我们提供上述服务的。

1. 部署

在.NET Framework 中，应用程序的部署非常简单。程序集中含有完整的功能描述，您不必在系统的其他位置注册同样的信息。安装.NET 应用程序也非常简单，只需把文件从一个地方拖到另一个地方。如果您要安装的库与现有的库不兼容，不用担心，下面我们就告诉您该如何做。

2. 版本化

.NET 一个关键的卖点是它解决了普遍存在的“DLL Hell”问题，“DLL Hell”会使一个版本的库重写另外一个版本的库(通常是新版重写旧版)，结果是瞬间很多库就不能使用了。在.NET 中，程序集在元数据中含有自身的版本号，CLR 发现它们后，如果有必要，同一个库的不同版本将会同时被加载，为新老客户程序同时服务。

我们将在第 2，3，4 章详述托管 C++的这些特性。接着让我们来看看.NET Framework 的第 2 部分：类库。

1.1.3 .NET Framework 类库

.NET Framework 类库为开发各种类型的应用程序定义了一套可重用的类。库中的类提供了诸如数据库访问、联网、故障诊断、安全性等标准服务。由于它对联网提供支持，所以能够开发单机的、分布式的以及基于 Web 的应用程序。同时，它们还支持 XML 和 WSDL(Web Service Description Language，Web 服务描述语言)，使我们能够开发基于标准协议的应用程序和服务。

所有.NET 应用程序共享同一套类，像其他优秀的类库一样，.NET 的类库也把它所包含的所有类都放在同一个命名空间里。C++ STL 提供了 std 命名空间(ATL 现在也提供了 ATL，您稍后就会发现这一点)，而.NET Framework 提供了 System 命名空间。这个命名空间依次包含了许多 2 级和 3 级的命名空间，下面列出了其中一些命名空间以及它们的功能：

- System::Data 定义了访问和管理数据源的 ADO.NET 体系结构
- System::Xml 定义了对 XML 的支持
- System::Diagnostics 用来调试、跟踪、创建日志和监控系统性能
- System::DirectoryServices 定义了访问 Active Directories 的类
- System::Net 定义了支持联网的类
- System::Drawing 定义了对 GDI+图形的访问
- System::Windows::Forms 定义了用来创建 Windows 应用程序的类
- System::Security 提供 CLR 安全系统，包括安全策略分析，权限，栈查看(stack walk)的类
- System::Security::Cryptography 定义提供加密服务的类，服务包括编码，解码，散列法，产生随机数，消息鉴别，数字签名
- System::Web 提供 ASP.NET 和 Web Forms 支持的核心结构
- System::Web::Services 提供基于 SOAP 的 Web 服务

看了上述列表，好像我们一直是在讨论.NET Framework 所能实现的巨大功能，但却没有详细地研究它们。OK，下面我们就.NET 开发中最常用的 4 个领域进行更进一步的探讨。要记住的一点是，大部分的开发都不需要用 C++来完成，我们有其他的方法来实现同样的功能。您可能会发现，其中有一种应用程序类型需要采用 C++的代码，这就是它的作用所在。

1. ASP.NET 应用程序

微软对 ASP.NET 模型做出了修改。ASP.NET 不仅可以用来创建传统的基于 Web 的 ASP 2.0 和 3.0 应用程序——现在是使用 Web Forms 来创建——还可以创建最新的 Web 服务。我们马上