

Visual C++ .NET
使用者的必备参考

Visual C++ .NET 编程详解

Special Edition Using Visual C++ .NET

[美] Kate Gregory 著

蔡众众 范宏宇 咸容禹 等译

que



电子工业出版社

Publishing House of Electronics Industry
<http://www.phei.com.cn>

Visual C++ .NET 编程详解

Special Edition Using Visual C++ .NET

[美] Kate Gregory 著

蔡众众 范宏宇 咸容禹 等译

电子工业出版社

Publishing House of Electronics Industry

北京·Beijing

内 容 简 介

本书详述了使用 Visual C++ 进行 Windows 和 .NET 编程的不同层次的知识。本书共分两个部分：第一部分是与传统 Windows 编程相关的内容，包括对话框/控件、消息/命令、视图/文档、打印/存储、组件对象模型、Internet 编程、数据库访问以及其他的 Windows 专题；第二部分则是对 .NET 及 .NET 环境下 Visual C++ 的使用及新功能的介绍，包括跨语言开发、托管与非托管代码、Web 服务、数据库编程、COM 互操作、安全与策略等内容。

本书从传统的 Windows 编程开始，循序渐进地讲解一些基本但是很重要的程序任务，适合不同层次读者的需求，即使是刚刚接触 Visual C++ 的新手也能顺利地进入 .NET 的世界。同时本书也是一本供使用过 Visual C++ .NET 的程序员进一步提高的有益书籍，并可作为自学参考书。

Authorized translation from the English language edition, entitled Special Edition Using Visual C++ .NET, ISBN: 0789724669 by Kate Gregory, published by Pearson Education, Inc, publishing as Que, Copyright © 2002.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Simplified Chinese language edition published by Publishing House of Electronics Industry, Copyright © 2003.

This edition is authorized for sale only in the People's Republic of China excluding Hong Kong, Macau and Taiwan.

本书中文简体专有权由 Pearson 教育集团所属的 Que 授予电子工业出版社。其原文版权及中文翻译出版权受法律保护。未经许可，不得以任何形式或手段复制或抄袭本书内容。

此版本仅限在中华人民共和国境内（不包括香港、澳门特别行政区以及台湾地区）发行与销售。

版权贸易合同登记号：图字：01-2001-4965

图书在版编目 (CIP) 数据

Visual C++ .NET 编程详解 / (美) 格雷戈里 (Gregory, K.) 著; 蔡众众等译.

-北京: 电子工业出版社, 2003.4

书名原文: Special Edition Using Visual C++ .NET

ISBN 7-5053-8643-3

I. V... II. ①格... ②蔡... III. C 语言 - 程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 025535 号

责任编辑: 史 平

印刷者: 北京东光印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787 × 1092 1/16 印张: 38 字数: 949 千字

版 次: 2003 年 4 月第 1 版 2003 年 4 月第 1 次印刷

定 价: 59.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077

译 者 序

C++ 语言功能强大、应用面广、可移植性强，而 Visual C++ 是目前国内软件开发者最广泛使用的编程工具。尤其是进入 .NET 时代之后，.NET 为 C++ 语言提供了更强大的支持，使 Visual C++ 功能更加丰富，使用更加便捷。

目前，市面上关于 Visual C++ .NET 的书籍多是专注于讲解 .NET 的新功能及特性，使许多对传统的 Visual C++ 还并不熟练的读者望而生畏。本书的特点是不仅仅讲述了 .NET 环境下的编程，而是从传统的 Windows 编程开始，循序渐进，一步步引导读者进入 .NET 的世界。并通过大量实例将所讲述的知识融会贯通，即使是只了解一些 C++ 语言知识，从未使用过 Visual C++ 的读者也能在不知不觉间了解和掌握 .NET 编程技术。

本书还为那些编写 COM 组件和已经编写了 MFC 应用程序，而又想使用 .NET 新功能并为此困惑的程序员指点迷津，告诉他们如何才能“鱼与熊掌兼得”。当然，这也正是 .NET 的特性与优势所在。

书中由浅入深地给出了大量实例，并附有源代码。经常使用 Visual C++ .NET 的老手也能从中受益匪浅。本书可供不同层次有志于使用 Visual C++ .NET 开发应用程序的读者阅读参考。

参加本书翻译的人员有：蔡众众、范宏宇、咸容禹、杨志明、贾光宇、骆团结、祝孔强、刘春兰、刘文琴、赵榕娟、庄姝、练书武、张延军、庞宇丽等。

由于译者水平有限，难免出现差错，敬请读者批评指正。

序 言

Visual C++ .NET 是一种以 C++ 为基础发展而来的语言，可以使用 Visual Studio .NET 来生成 Visual C++ 应用程序。实际上，Visual Studio .NET 这个强大的综合性工具可以用来生成两种应用程序：基于各种不同 Windows 平台（95/NT/2000/XP 等）的 32 位应用程序和运行于 .NET 运行时中的 .NET 应用程序。

Visual Studio .NET 是一个神奇的工具，使用它的代码生成向导，可以在几秒钟内生成一个 Windows 或者 .NET 应用程序的外壳。它还带有一个称为微软基础类（MFC, Microsoft Foundation Class）的类库，这个类库已成为在各类 C++ 编辑器中进行 Windows 软件开发的业界标准类库。支持 .NET 的类的基类库（Base Class Library）为用户提供更强大更易用的支持，其可视化工具可以快捷地设计出菜单和对话框的布局。在你建立第一个 Windows 或 .NET 编程项目时会发现，花在学习使用该产品上的时间绝对值得。

本书适合的读者对象

本书讲解如何使用 Visual C++ 生成 32 位 Windows 应用程序，包括数据库应用程序、Internet 应用程序以及使 COM 发挥强大功能的应用程序。书中还讨论各种与 .NET 相关的内容以及如何用 Visual C++ 编写那些使用数据库、Internet 等的 .NET 应用程序，甚至可以学到如何将 COM 和 .NET 技术融合在一起。

如果符合以下前提之一，就应该阅读本书：

- 了解一些 C++ 和 Windows 编程技术，但是对 Visual C++ 还是个新手。如果只是想编写程序，你会学得比计划的要快得多。
- 曾经使用过早期版本的 Visual C++。很多时候，用户学会使用一种方法后就不注重学习新的产品功能了。
- 想要创建 .NET 应用程序，但是不想使用 C# 或 Visual Basic。你会在本书中学到 .NET 的基础知识以及如何在 C++ 中使用它。
- 已经用 Visual C++ .NET 工作了一段时间并且逐渐发现工作起来很困难。也许确实如此。
- 经常使用 Visual C++ .NET，并且需要给自己的产品增加一些新功能。对诸如帮助、打印和线程调度等 Windows 任务，或者数据库访问、安全等 .NET 任务，你会找到一个“hand up”并从那里开始。

阅读本书前的工作

用户需要安装 Visual C++ .NET。安装过程十分简便，本书不讲解这些内容。

在购买 Visual C++ .NET 之前，需要具备最新的 Windows 操作系统：Windows 2000 或 Windows XP。要创建的应用程序可以在较早的 Windows 版本上运行，但如果想最大限度地发挥 .NET 的功能，则应该使用 Windows XP 或 Windows 2000。至于显示屏则越大越好。在 <http://www.usingvisualc.net> 上可以找到所有的范例代码，所以如果能够访问互联网就更方便了。

你必须向自己保证阅读时始终跟随本书 Visual C++ 的思路进行, 敲击、键入并将每一步尝试到底。也可以不键入所有代码: 可在网站上找到它们, 但是要提前做好打开文件查看代码的准备。

本书的内容

使用 Visual C++ 进行 Windows 和 .NET 编程是本书介绍的主要内容。本书包含 25 章和 4 个参考附录 (附录 A 到附录 D)。要确保看清了这些附录的题目, 这样在对所做的任何事情不确定的时候都可以求助它们。这些附录提供了非常有价值的参考, 如下:

- 附录 A “Windows 编程回顾与 CWnd 研究”, 包括那些被诸如 CWnd 的 MFC 类隐藏起来的 Windows 编程细节。
- 附录 B “XML 回顾”, 对 XML 中的一些重要概念进行了概述。
- 附录 C “Visual Studio 的用户界面、菜单与 Visual Studio”, 对屏幕上所有菜单、工具栏、编辑区域以及快捷方式等进行了解释。所有这些在用户和 Visual Studio 之间构架了一个高度综合性的、强有力的界面。
- 附录 D “从 Visual C++ 6 升级”, 这是为知道要做什么, 但找不到正确菜单项的人准备的。

背景知识和个人意愿决定你如何使用菜单和联机帮助。也许你只是撇上一眼这些附录就再也不看了, 也许你会在其中布满书签和标记, 即使不彻底地通读范例程序, 你仍然会从中学到很多。

本书的主要内容是从第 1 章到第 24 章, 每章都会讲解一个重要的程序任务, 有时可能是两个很接近的相关任务, 如生成一个任务栏、给应用程序加上帮助, 或者生成一个 XML Web 服务。每一章中的详细说明都会讲解如何生成一个或几个工作程序。

本书的第 1 章解释了 .NET 是什么以及各相关内容。如果只是对 .NET 感兴趣, 可跳过随后的 14 章, 因为它们都是关于 Windows 编程的内容。之后关于 .NET 的 10 章才是向 .NET 进军需要了解的。

下面概述本书的主要内容及其所在位置。

Windows 应用程序

Windows 应用程序是 .NET 出现以前用 Visual C++ 建立的那些应用程序。它们在各种不同版本的 Windows 上运行, 包括 Windows 95 和 Windows 98。很多 C++ 程序员需要维护这些程序, 而且不想立即将它们移植到 .NET。事实上在 Visual C++ 中也不必这样做, 可以使用 Visual Studio .NET 继续开发这些应用程序。

对话框和控件

什么样的 Windows 程序没有对话框、编辑框、按钮? 对话框和控件对 Windows 用户界面来说是至关重要的。所有界面, 甚至包括简单的按钮或一小段静态文本都是窗口。Win32 通用控件能够节省用户花在其他程序上的学习时间和开发人员花在操作系统上的时间, 它们能提供与别人一样的 File Open 对话框、分级树控件等。在第 3 章和第 7 章中, 可以了解到更多的控件知识。

消息与命令

消息是 Windows 编程的核心。在 Windows 中, 无论发生了什么, 只要用户敲击鼠标或按键, 都会触发一个消息并发送给一个或多个窗口, 并得到相应处理。Visual C++ 使捕捉并处理消息的代码编写工作变得更容易。第 3 章中解释了消息的概念以及 MFC 和 Visual C++ 中的其他部分

如何处理这些消息。

视图/文档范例

范例是一个样式，一种查看事物的方法。MFC的设计者假设每个程序都有些东西要保存在文件中，并在此基础上选择设计框架。信息的集合称为文档（document），视图（view）是一种查看文档的方法。将视图和文档剥离开来有很多好处，进一步的解释参见第4章。MFC提供了很多继承自文档类和视图类的类，这使实现如滚动条这样的通用编程任务不再成为问题。

无论Windows程序有多聪明，如果不能在屏幕上显示文字和图片告诉用户发生了什么，就没人会知道程序做了什么。视图类在这方面做了大量的工作（采用文档/视图范例的好处之一）。但是有时也得自己绘制，在第4章中将介绍如何在屏幕、设备环境和滚动条等之上进行绘制。

打印与存盘

为程序添加打印功能有时是世界上最简单的事情，因为用来实现屏幕绘制的代码可以重用来实现纸上绘制。但是，如果信息超过一页纸，事情还是会变得有些麻烦。第5章中讲解了这些以及映射模式、页眉与页脚等内容。

有些很不错的东西被设计成只是临时性的，如计算器的显示或者在线聊天窗口。但是，大多数程序都能将其文档保存到一个文件，并可以打开和加载这一文件来重新创建一个文档。通过使用存档并扩展流输入/输出运算符>>和<<，MFC能够简化这一工作。关于对文件的读写操作，在第5章中会讲到。

组件对象模型

.NET出现以前，COM是构建组件的最好方法，而且也没有必要弃COM而投奔.NET——一个COM组件就是一个.NET对象，一个.NET对象也就是一个COM组件。在第9章中可以学到如何构建一个COM组件。还可以在第23章中了解如何融合.NET和COM。

Internet

由两台或更多计算机分担工作的分布式计算变得越来越普遍。程序之间需要相互对话，人们需要在LAN乃至全世界范围内传递消息，而MFC中有支持这些通信的类。尽管.NET将强大的Internet支持带入了每个应用程序，但是仍然可以发挥Windows应用程序自身的强大威力。第10章中讲述了如何使用WinInet类，或者也可以自己进行套接字（socket）编程。

数据库访问

数据库编程更加容易。通过ODBC，可以调用API函数访问众多种类的数据库文件，包括Oracle、dBase、Excel电子表格、纯文本文件、使用SQL的老式主机系统等，几乎无所不包。调用一个标准的名称函数及数据库厂商或第三方提供的API即可处理转换工作。详见第11章。

高级Windows专题

对那些已经掌握基础内容的程序员来说，本书提供了一些高级的专题，可使你的编程技巧进一步提高。你将学会如何使用第12章讲述的技术，以防止内存泄漏、发现瓶颈和定位代码中的错误。第13章则描述了如何像一位专家那样发现和纠正错误。

当用户要求软件既高性能又多任务时，程序员必须全面地学习新技术以提供更强大的、具

备很短反应时间的应用程序。对许多程序员来说，编写多线程的应用程序是一项关键的技术。在第14章中可以学到线程方面的知识，第15章则介绍了控制台应用程序以及如何使用和生成自己的DLL和Unicode。

.NET 编程

.NET是什么？它意味着太多的东西：代码可以在其上运行的平台；功能强大、十分有用的类库；一项跨语言的开发技术；用于Web服务的一系列协议以及更多的内容。本书的后半部分将带你进入.NET的世界。

开始

第1章解释了.NET并将其与传统的Windows程序进行了对比，这是此次.NET旅程的出发点。下一站是第16章，这里讲述了.NET程序如何运行及其提供的强大功能。在第17章中，你可以学习如何开发属于自己的.NET代码。

跨语言开发

想像一下在C++中调用VB代码，或者在C#中调用C++代码。想像一下不需要特别做什么就可以突破语言的限制，写一个继承自C++类的Visual Basic类将会怎样。而且调试时也是跨语言的，这是.NET承诺的一部分。看看在第18章和第19章中这一切是怎么发生的。

托管代码和非托管代码

在完成了一些.NET开发之后，你可能希望了解对于代码和数据而言，“托管（managed）”意味着什么。第20章为你解释正在发生的事情以及为什么这些与你（程序员）有关。

Web 服务

XML Web服务设计的初衷是跨厂商的，并且有可能在十年之内改变软件开发的方式。要想知道它到底是什么，利用XML Web服务与Visual C++进行开发有多么简单，读一读第21章。

数据库编程

数据库编程在传统的Windows应用程序中已经很容易了。从.NET的角度看，ADO.NET使其迈向了一个新的高度。它可以很容易地处理非连接编程并且以超越数据的处理方式进行全面控制。在第22章中对此进行了详述。

COM 互操作性

大多数开发工作室投入大量的时间和精力在现有的应用程序上，并且许多这类应用程序的功能是通过COM来实现的。当然，你想在新的.NET应用程序中使用旧的COM对象，你可能还想要你的COM应用程序使用.NET对象。没问题！实际上，在.NET应用程序中使用COM对象比在一个传统的COM C++应用程序中使用它还要简单。第23章将告诉你这一切有多容易。

安全

在.NET中，代码可以来自任何地方，包括硬盘、局域网或者Internet。你可能都没有留意，代码就下载并运行了。但是，如果代码是恶意的，你不想留意一下它的运行安全吗？为保护系

统,.NET框架会阻止任何远程代码对本地资源的访问。第24章中解释了它是如何工作的以及如何使机器尽在掌控中。

本书使用的惯例

本书中包含大量的代码,有时仅有1、2行,这种情况下代码与文本混在一起编排,如:

```
int SomeFunction(int x , int y );
{
    return x+y
}
```

有时一段代码太长不能与文本混编,如下面的程序清单0.1。

程序清单 0.1

```
CHostDialog dialog(m_pMainWnd);
    if (dialog.DoModal() == IDOK)
    {
        AppSocket = new CSocket();
        if (AppSocket->Connect(dialog.m_hostname,119))
        {
            while (AppSocket->GetStatus() == CONNECTING)
            {
                YieldControl();
            }
            if (AppSocket->GetStatus() == CONNECTED)
            {
                CString response = AppSocket->GetLine();
                SocketAvailable = TRUE;
            }
        }
        if (!SocketAvailable)
        {
            AfxMessageBox("Can't connect to server. Please
            ➤quit.",MB_OK|MB_ICONSTOP);
        }
    }
}
```

倒数第2行的符号(➤)叫做代码延续符,它指出在此处代码行为适应纸张宽度而转行,在实际中并非如此。如果你依照本书键入代码,那么请不要在此转行,而应延续该行。如果正在阅读 Visual C++ 产生的代码,也不必为代码未在此转行而迷惑。

记住,书中的代码是为了帮助你理解正在讲述的内容,并不需要实际键入,所有的代码在合作网站上都可以找到。有时在程序开发过程中,你会看到一个代码块的几个不同版本。最终的版本都放在 Web 站点上,在 <http://www.quepublishing.com> 或 <http://www.usingvisualc.net> 上都可以找到。

小结

现在可以为序言做总结了。我们了解了开始阅读前都需要做什么,包括对书中通篇都在使用的某些标识的预先提示,是时候正式开始阅读了。让我们学习用 MFC 编写 Windows 应用程序,或者用 .NET 框架编写 .NET 应用程序。

目 录

第 1 章 .NET 的背景	1
1.1 .NET 是什么	1
1.2 为什么要编写 XML Web 服务	4
1.3 如何处理已有的 MFC 应用程序	5
1.4 COM 程序员的任务	6
1.5 C++ 语言是否适于开发.NET	7
1.6 开发.NET 程序的新手须知	9
1.7 本章小结	9
第 2 章 生成第一个 Windows 应用程序	11
2.1 创建 Windows 应用程序	11
2.2 文件名和类名	18
2.3 亲自尝试	19
2.4 创建基于对话框的应用程序	21
2.5 创建 DLL、控制台应用程序及其他	23
2.6 更改 MFC 应用程序向导的结果	25
2.7 理解 MFC 应用程序向导生成的代码	26
2.8 本章小结	34
第 3 章 与应用程序交互	35
3.1 了解对话框	35
3.2 创建对话框资源	35
3.3 了解消息传递的路线	47
3.4 使用 MFC 类向导捕捉消息	53
3.5 了解命令	56
3.6 本章小结	59
第 4 章 显示信息	60
4.1 文档类简介	60
4.2 了解视图类	62
4.3 创建 Rectangles 应用程序	64
4.4 其他视图类	67
4.5 文档模板、视图和框架窗口	69
4.6 了解设备环境	70
4.7 介绍 Paint1 应用程序	71

4.8 生成 Paint1 应用程序	71
4.9 使用字体	74
4.10 调整和定位窗口	77
4.11 使用画笔	78
4.12 使用笔刷	80
4.13 滚动窗口	82
4.14 生成 Scroller 应用程序	83
4.15 添加递增文本行的代码	85
4.16 添加递减文本行的代码	86
4.17 本章小结	88
第 5 章 打印与存储	89
5.1 了解 MFC 的基本打印和打印预览	89
5.2 缩放比例	91
5.3 多页打印	92
5.4 设置原点	97
5.5 MFC 与打印	98
5.6 了解存储与持续	101
5.7 创建可持续类	106
5.8 直接读写文件	111
5.9 创建自己的 CArchive 对象	113
5.10 使用注册表	114
5.11 本章小结	117
第 6 章 生成完整的应用程序：ShowString	118
6.1 生成一个显示字符串的应用程序	118
6.2 生成 ShowString 的菜单	121
6.3 生成 ShowString 的对话框	124
6.4 使菜单发挥作用	127
6.5 使对话框发挥作用	130
6.6 在 Options 对话框中加入外观选项	131
6.7 本章小结	136
第 7 章 状态栏、工具栏和通用控件	137
7.1 建立工具栏	137
7.2 建立状态栏	144
7.3 创建一个新的命令 ID	146
7.4 创建默认字符串	147
7.5 在指示器数组中添加 ID	147
7.6 创建窗格的命令更新处理器	148
7.7 设置状态栏外观	149

7.8	建立 Rebar	151
7.9	通用控件	154
7.10	本章小结	185
第 8 章	帮助、属性页和向导	186
8.1	联机帮助	186
8.2	不同种类的帮助	186
8.3	帮助系统的组件	189
8.4	应用程序向导对帮助的支持	190
8.5	计划实现帮助的方法	190
8.6	编写命令帮助	191
8.7	编写关联帮助	193
8.8	编写帮助文本	196
8.9	调整目录	204
8.10	为应用程序加入属性单	213
8.11	将属性单改为向导	213
8.12	本章小结	217
第 9 章	用 ATL 生成 COM+ 组件	219
9.1	COM 的基本概念	219
9.2	ATL——活动模板库	222
9.3	COM+ 中的加号	232
9.4	本章小结	234
第 10 章	Internet 编程	235
10.1	使用 Windows 套接字	235
10.2	MFC 中的 Winsock	236
10.3	使用消息 API (MAPI)	238
10.4	在应用程序中添加对 MAPI 的支持	239
10.5	使用 Internet 服务器 API (ISAPI) 类	242
10.6	使用 WinInet 类	245
10.7	ATL 服务器	256
10.8	本章小结	265
第 11 章	数据库编程	266
11.1	Visual C++ 中的 Windows 数据库编程	266
11.2	了解数据库概念	266
11.3	创建 ODBC 数据库程序	269
11.4	排序与过滤	281
11.5	在 ODBC 和 ADO 间选择	287
11.6	了解 SQL	292

11.7	存储过程	302
11.8	本章小结	310
第 12 章	提升应用程序性能	311
12.1	用 ASSERT 和 TRACE 防止错误发生	311
12.2	加入 Debug-Only 功能	313
12.3	防止内存泄漏	314
12.4	使用 Optimization 创建高效代码	318
12.5	本章小结	319
第 13 章	调试	320
13.1	调试的重要性	320
13.2	调试词汇：断点及相关概念	320
13.3	调试命令与窗口	321
13.4	使用 ATL/MFC Trace Tool	329
13.5	定义 Dump 成员函数	330
13.6	故障排除	331
13.7	本章小结	333
第 14 章	用 Windows 线程实现多任务	334
14.1	了解简单线程	334
14.2	了解线程通信	338
14.3	使用事件对象实现通信	341
14.4	使用线程同步	343
14.5	本章小结	352
第 15 章	特殊的 Win32 应用程序类型	353
15.1	高级 Win32 编程专题	353
15.2	创建控制台应用程序	353
15.3	创建和使用 32 位动态链接库	357
15.4	发送消息和命令	369
15.5	国际软件开发状况	370
15.6	创建 Windows 服务	371
15.7	本章小结	377
第 16 章	通用语言运行时	378
16.1	.NET 的结构	378
16.2	系统名称空间	379
16.3	友好.NET 编码的技巧	386
16.4	查找从前的家族成员	386
16.5	本章小结	388

第 17 章 .NET 初步	389
17.1 第一个.NET 应用程序	389
17.2 第一个.NET 组件	391
17.3 设计一个有用的.NET 组件	395
17.4 本章小结	402
第 18 章 与 Visual Basic 集成	403
18.1 在 Visual Basic 中使用 C++ 组件	403
18.2 在 Visual Basic 中扩展 C++ 组件	407
18.3 在 Visual C++ 中使用 Visual Basic 组件	410
18.4 本章小结	412
第 19 章 与 C# 集成	413
19.1 在 C# 中使用 C++ 组件	413
19.2 在 ASP.NET 页面中使用 C++ 组件	417
19.3 在 C# 中扩展 C++ 组件	419
19.4 在 C++ 中使用 C# 组件	423
19.5 本章小结	424
第 20 章 托管与非托管 C++	425
20.1 C++ 支持托管与非托管编程	425
20.2 什么是托管代码	425
20.3 是否所有类都可以是碎片收集类	429
20.4 值类型	431
20.5 托管接口	433
20.6 属性	437
20.7 混合托管与非托管代码	438
20.8 本章小结	439
第 21 章 创建 XML Web 服务	440
21.1 XML Web 服务	440
21.2 XML Web 服务应该提供什么	440
21.3 设计 Web 服务范例	441
21.4 测试 Web 服务范例	444
21.5 在代码中使用 Web 服务	451
21.6 DISCO 和 UDDI	454
21.7 本章小结	454
第 22 章 用 ADO.NET 实现数据库访问	456
22.1 了解 ADO.NET	456
22.2 数据访问组件	457

22.3	使用 DataRelation	464
22.4	从数据库中获得 XML	468
22.5	从 XML 填写数据库	470
22.6	本章小结	472
第 23 章	COM 互操作性	473
23.1	在 .NET 应用程序中使用 COM 组件	473
23.2	在 .NET 中扩展 COM 组件	476
23.3	将 .NET 对象作为 COM 组件使用	478
23.4	本章小结	482
第 24 章	安全与策略	483
24.1	让应用程序更安全	483
24.2	.NET 安全模型是相容的	483
24.3	独立与共享程序集	486
24.4	基于用户和基于角色的安全	490
24.5	来自编译器的安全保证	491
24.6	本章小结	492
附录 A	Windows 编程回顾与 CWnd 研究	493
附录 B	XML 回顾	502
附录 C	Visual Studio 的用户界面、菜单和 Visual Studio	524
附录 D	从 Visual C++ 6 升级	579

第1章 .NET的背景

本章要点：

- .NET是什么
- 为什么要编写 XML Web 服务
- 如何处理已有的 MFC 应用程序
- COM 程序员的任务
- C++ 语言是否适于开发 .NET
- 开发 .NET 程序的新手须知
- 本章小结

1.1 .NET 是什么

无论你是熟练使用 Visual C++ 的 Windows 程序员，正在观察技术动向的 Internet 程序员，还是连不同编程平台都分不清楚的新手，在准备使用 Visual Studio .NET 时都会问自己这样一个问题，那就是 .NET 到底是什么？

怎样读它？如何在网上搜索它？

“net”前面的字符读做“dot”，所以在“.”字符与所处的环境没有关联的情况下（如在邮件列表名中和文件名开头中），人们通常将其读做“dot-net”。如果你想在网搜索一下与“.NET”相关的新闻，用“dotnet”比用“.NET”运气会好些。毕竟，“net”这个词在某些与.NET这个新名词毫无关系的网页上大量出现。

遗憾的是，关于这个问题没有惟一的答案。微软目前在不止一种产品上使用.NET这个标识，但是它们都基于同一个初衷和策略，所以.NET是：

- 一个代码运行的平台。
- 一个可以用于任何语言的代码类库。
- 诸如 Visual Studio 这样的新版本开发工具。
- 一些新版本开发语言。
- 一套促进.NET发展并使其更便于使用的服务器端产品。
- 一个设计和创建应用程序并使其通过组件甚至 Internet 分担工作的新方法。

.NET不能执行那些可在 Windows 和其他平台执行的程序，而是由通用语言运行时（CLR，Common Language Runtime）执行那些已被编译为微软中间语言（MSIL，Microsoft Intermediate Language，也叫做 Intermediate Language 或 IL）的.NET应用程序。这个“执行引擎”将 MSIL 转换为本机代码，但是它也提供其他许多服务，包括内存管理、安全、程序间的交互等。例如，

当代码使用一个 COM 组件或调用一个来自 DLL 的代码时。目前 .NET 平台只能在各种 Windows 环境中实现，但是它本身会给程序员提供许多便利。

Visual Studio .NET 还附有一个 .NET 软件开发工具包 (SDK, Software Development Kit), 包括 .NET 类, 这些类可以用于多种 .NET 语言并且给开发者提供强大的功能。有一些也提供给使用 MFC 的纯 Windows 程序员, 其中一些将你可能在 COM 和 COM+ 中一直在使用的服务打包了。但是, 很多功能 (如用于 XML 或检验安全认证的功能) 都只能在 .NET 中使用。

Visual Studio .NET 是 Visual Studio 的最新版本, 并且与以前的版本有很大不同。过去, Visual Studio 一直是一些相关工具的集成, 每个工具都有自己的用户界面和各自的优缺点。现在, Visual Studio 是一个可以用多种语言编写代码的单一产品, 它是将代码与按钮和敲击联系起来的优秀的 Visual Basic 方法。不同于做法相同的 Visual C++, 现在用到的是纯粹的 Visual Studio 方法。当然, 这意味着一些程序员又有东西要学了。你可能想阅读一下附录 C, 甚至想彻底了解上一个版本的 Visual Studio。附录 D 可能对有经验的 Visual C++ 程序员有帮助。

Visual Studio .NET 中的一些语言为适应 .NET 而有所改变。Visual C++ 的改变最小, 其中加入了少数关键字使 C++ 程序员可以在代码中明确请求 .NET 功能。已有的 Visual C++ 项目应该在 Visual Studio 中打开并生成而无须任何更改。如果你听到过很多关于 .NET 更改了开发语言的传言, 那可能是指 Visual Basic, 它确实改变了很多, 而 Visual C++ 的程序员在这点上则不必担心。

图 1.1 显示了不同的 .NET 语言、XML Web 服务、XML、.NET 类库以及 CLR 是如何协同工作以创建一个 .NET 应用程序的。

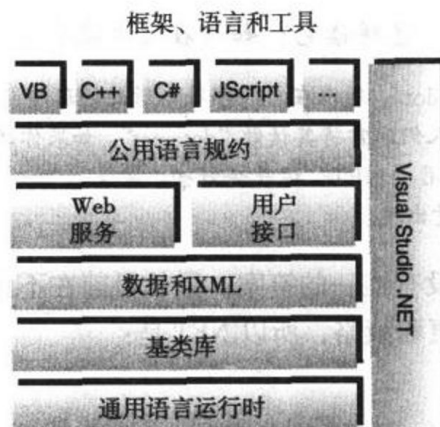


图 1.1 .NET 的创意绝不仅仅是一个新版本的编译器

微软在许多服务器端产品上使用了 .NET 标识, 它们是:

- Microsoft Application Center 2000
- Microsoft BizTalk Server 2000
- Microsoft Commerce Server 2000
- Microsoft Exchange 2000
- Microsoft Host Integration Server 2000
- Microsoft Internet Security and Acceleration Server 2000
- Microsoft Mobile Information 2001 Server