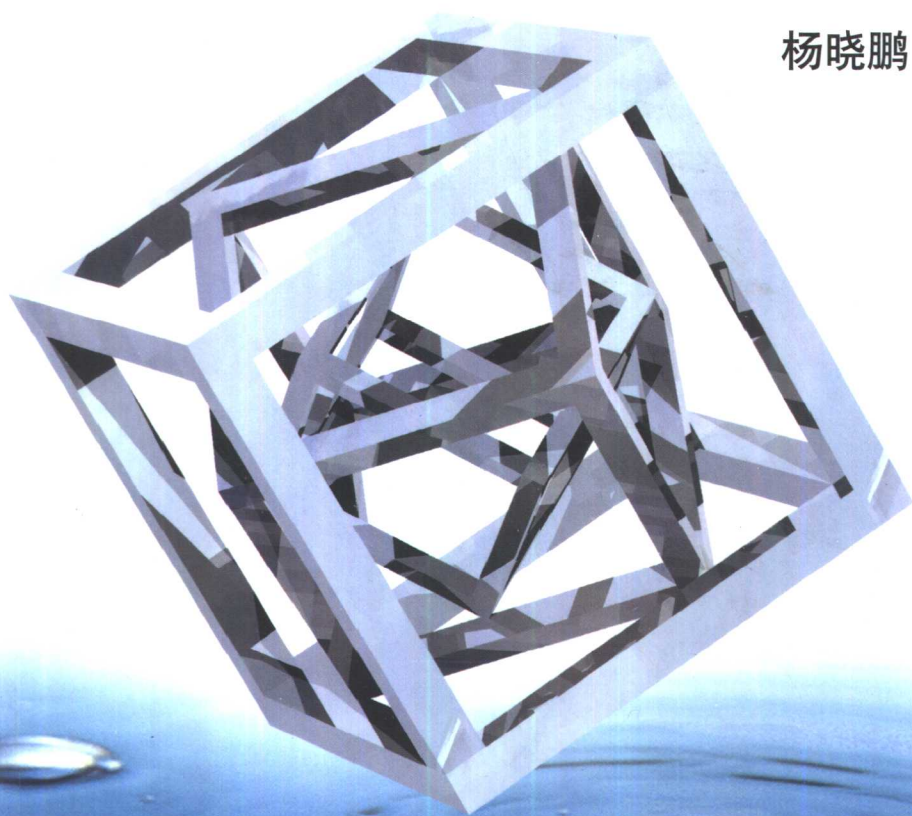


万水 计算机实用编程技术系列

Visual C++ 7.0

实用编程技术

杨晓鹏 宗明 等编著



中国水利水电出版社
www.waterpub.com.cn

万水计算机实用编程技术系列

Visual C++ 7.0 实用编程技术

杨晓鹏 宗明 等编著

中国水利水电出版社

内 容 提 要

Visual C++.NET 是微软公司最新的开发环境 Visual Studio.NET 中的一个重要开发工具。而 Visual Studio.NET 作为微软公司的 Microsoft.NET 战略中的 .NET 开发平台,是当前最热门的开发环境。Visual C++.NET 提供了对网络和 .NET 框架的完美配合,并可以通过受管扩展 (Managed Extention) 和 C# 进行 .NET 框架程序开发。通过 Visual C++.NET 您可以得到微软最新技术的全部特性,包括公共语言运行时 (Common Language Runtime) 对内存的自动管理和垃圾收集、元数据的使用等。特别重要的是,通过受管扩展,现存的大量代码可以平滑地移植到新的 .NET 框架下,既保护了用户以前的投资,又使用户有机会能享受到新框架带来的安全性和通用性。

本书从实例出发,对 Visual C++.NET 开发应用程序的各个方面做了探讨,包括对新的 C++ 受管扩展的探讨、Windows 程序开发、网络应用开发、数据库开发和组件使用等方面。最后,我们对微软最新推出的专为 .NET 战略开发的新语言 C# 做了原理性的介绍,并通过几个实例让您感性认识它的特点。

作者在每一章的最后,都总结了一些编程技巧和经验,希望对读者能有所裨益。

本书的内容涵盖范围很广,对每部分内容都从进阶开始到深入探讨,并且提供了大量的实例和源代码。无论是初学者还是有一定经验的 VC 程序员,都能从本书中找到适合的部分。本书力图为读者展现程序设计的全貌,使初学者读完后能自己动手写程序,而有经验的程序员则可从中学悟语言使用的技巧。本书可以作为开发时的参考,但对特定话题更详细的资料,应该参考微软的 MSDN。

书中所有示例,其源代码都可在随书光盘中找到。

图书在版编目 (CIP) 数据

Visual C++ 7.0 实用编程技术/杨晓鹏等编著. —北京:中国水利水电出版社, 2002

(万水计算机实用编程技术系列)

ISBN 7-5084-0943-4

I. V… II. 杨… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 092447 号

书 名	Visual C++ 7.0 实用编程技术
编 著	杨晓鹏 宗明 等
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	北京市天竺颖华印刷厂
规 格	787×1092 毫米 16 开本 39.75 印张 876 千字
版 次	2002 年 1 月第一版 2002 年 1 月北京第一次印刷
印 数	0001—5000 册
定 价	65.00 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
版权所有·侵权必究

前 言

毫无疑问, Visual C++一直是最优秀的开发工具之一。即使有着快速开发工具(RAD)的竞争,但由于 Visual C++强大的功能,灵活的特性,对底层的控制能力,它一直是真正的程序员所选用的编程语言。

随着整个计算机业发展转向网络, Microsoft 提出了.NET 战略。Microsoft .NET 的策略是将互联网本身作为构建新一代操作系统的基础,对互联网和操作系统的设计思想进行合理延伸。这样,开发人员必将创建出摆脱设备硬件束缚的应用程序,以便轻松实现互联网连接。Microsoft .NET 无疑是当今计算机技术通向网络时代的一个非常重要的里程碑。

.NET 对开发人员来说也十分重要,因为它不但会改变开发人员的开发应用程序的方式,而且使得开发人员能创建出全新的各种应用程序。新型开发范例的核心是 Web 服务这个概念的引入。Web 服务是一种通过简单对象访问协议(SOAP),在互联网上展露其功能性的、极为公开的服务。SOAP 是一种基于可扩展标记语言(XML)制定的协议。

从理论上讲,开发人员可通过调用 Web 应用编程接口(API),将 Web 服务集成到应用程序中。其调用方法与调用本地服务类似,不同的是 Web API 调用可通过互联网发送给位于远程系统中的某一服务。例如,Microsoft.Passport (Passport) 服务使得开发人员能够对应用程序进行认证。通过对 Passport 服务编程,开发人员可以充分利用 Passport 的基本结构,通过运行 Passport 来维护用户数据库,以确保其正常运行、定期备份等等。

.NET 正是根据这种 Web 服务原则而创建的,微软目前正着手提供这个基本结构,以便通过.NET 平台的每一部分来实现这种新型的 Web 服务。而 Visual Studio.NET、.NET 框架、Windows.NET 和.NET 企业服务器,正是为进行基于 Web 服务模型的应用程序开发而度身定做的新一代开发工具和基本结构。.NET 构建块服务、新增的.NET 设备支持以及即将到来的.NET 用户体验,将为人们彻底攻克这一难题画上一个圆满的句号,使人们能够充分利用 Web 服务模型,如愿以偿地开发出新一代应用程序。

Visual Studio.NET 是为.NET 准备的开发工具。作为 Visual Studio.NET 的主要成员,Visual C++.NET 提供了对网络和.NET 框架的完美配合,并可以通过受管扩展(Managed Extension)和 C#进行.NET 框架程序开发。通过 Visual C++.NET 您可以得到微软最新技术的全部特性,包括公共语言运行时(Common Language Runtime)对内存的自动管理和垃圾收集,元数据的使用等。特别重要的是,通过受管扩展,现存的大量代码可以平滑地移植到新的.NET 框架下,既保护了用户以前的投资,又使用户有机会能享受到新框架带来的安全性和通用性。

本书主要从实例出发,对新的 Visual C++.NET 进行探讨。由于历史和继承的关系,以及对灵活性的要求,Visual C++.NET 对同一个目的往往提供了重复的解决方案。不同的类库对

同一个要求也提供了不同的类。在内容选取上，我们尽量选取最好的，最有前途的方法。每章最后都总结出我们认为值得注意的一些小技巧。最后，我们对微软最新推出的专为.NET 战略开发的新语言 C#做了原理性的介绍，并通过几个实例让您感受到它的魅力。

本书的作者具有多年的开发经验，对于 Visual C++的掌握驾轻就熟。在实践中，我们感觉到所缺乏的不是对开发工具的描述性教程，而是好的技巧总结。因此，我们在本书中只是简单的讲述原理，而提供了很多实例和代码。他山之石，可以攻玉，注释详尽的代码，好过反复的原理性灌输。很多好的编程习惯和代码风格都是从读代码中学到的。我们希望读者在阅读本书时，一定不要忽略了各个实例的源代码。

本书由邵谦谦总策划，由杨晓鹏、宗明、唐凡、高亮和张弋共同执笔编写。雷臻、钟普、王一超、周卫、赵中伟、张忠东、李晓裳、范智育、王宏生、李毅、王瑾、吴浩、李炎、刘伟、刘华刚、朱峰、赵晓燕、李晓、马苍、郝春容、韦勇、成美华、萧峰、李菊、张浩然、李欣、张浩、李想、朱大成、杨竞锐、王贵新、张诚华等同志在整理材料方面给予了作者很大的帮助。在此向他们致以感谢！

由于时间仓促，加之编者的水平有限，缺点和错误在所难免，恳请专家和广大读者不吝赐教，批评指正。

编者

2001年8月

目 录

前言

第 1 章 .NET 和受管扩展	1
1.1 什么是 Microsoft.NET	1
1.1.1 .NET 的核心组件	1
1.1.2 .NET 的重大意义	1
1.1.3 Web 服务崭露头角	2
1.1.4 .NET 体验	3
1.2 在 VC++.NET 中进行.NET 编程	3
1.2.1 COM 与软件可复用性	3
1.2.2 什么是.NET 框架	4
1.2.3 .NET 框架与 COM 及 COM+的关系	5
1.3 受管扩展：在 VC++中进行.NET 框架程序开发	6
1.3.1 什么时候用受管代码编程	6
1.3.2 添加 C++ 受管扩展支持	7
1.3.3 受管类型介绍	8
1.3.4 受管数组	14
1.3.5 受管字符串	17
1.3.6 委派	18
1.3.7 全局属性（Global Attributes）	19
1.3.8 受管类型和 MFC	20
1.3.9 C++ 受管扩展开发实例	21
1.4 本章技巧小结	31
第 2 章 Windows 程序基本概念	35
2.1 消息机制	35
2.2 程序的入口点 WinMain	36
2.3 消息循环	37
2.4 窗口函数	37
2.5 消息映像（Message Map）的雏形	38
2.6 Windows 程序的创建与销毁	40
2.7 闲置时间的处理：OnIdle	41

第 3 章 对话框与常用控件	43
3.1 MFC 与对话框	43
3.2 实现非模式对话框	44
3.3 对话框及常用控件的用法	45
3.3.1 用于 Windows 常用控件的通知	45
3.3.2 Button 控件及示例	47
3.3.3 Edit 控件及示例	63
3.4 其他控件简介与应用技巧	76
3.4.1 CHeadCtrl	76
3.4.2 CImageList	78
3.4.3 CList	81
3.4.4 CProgress	83
3.4.5 CSlider	85
3.4.6 CSpin	87
3.4.7 CTree	89
3.4.8 CTab	92
3.5 DDX/DDV 技术	95
3.5.1 数据交换 (DDX) 函数	95
3.5.2 数据有效性 (DDV) 函数	97
3.5.3 创建定制的有效性函数	98
3.6 本章技巧小结	99
第 4 章 文档与视图结构	105
4.1 菜单	105
4.1.1 菜单命令的处理	106
4.1.2 增加上下文相关菜单	108
4.1.3 创建和访问菜单对象	110
4.2 工具条和状态条	114
4.2.1 定制自己的工具条	114
4.2.2 定制状态条	128
4.3 单文档应用程序(SDI)	134
4.3.1 Windows 应用程序对象	134
4.3.2 SDI 文档模板	135
4.3.3 使用文档和视图	139
4.4 多文档应用程序 (MDI)	166
4.4.1 MDI 应用程序的类	167

4.4.2	MDI 文档模板	168
4.4.3	开发 MDI 应用程序	171
4.5	创建多视图	174
4.6	本章技巧小结	176
第 5 章	设备环境、矢量图、位图和字体	191
5.1	设备环境	191
5.2	图形设备接口	191
5.3	MFC 设备环境类	192
5.3.1	基类 CDC	192
5.3.2	CPaintDC	195
5.3.3	CClientDC	196
5.3.4	CWindowDC	196
5.4	GDI 坐标	197
5.4.1	映射模式	197
5.4.2	坐标转换	199
5.5	GDI 图形对象	199
5.5.1	画笔: CPen 类	200
5.5.2	画刷: CBrush 类	202
5.5.3	字体: CFont 类	203
5.5.4	位图: CBitmap 类	203
5.5.5	调色板: CPalette 类	203
5.6	字体与文本	204
5.6.1	字体特征	204
5.6.2	显示简单的文本	204
5.6.3	TEXTMETRIC 结构	206
5.6.4	LOGFONT 结构	208
5.6.5	创建字体	210
5.7	矢量图形	212
5.7.1	绘图模式	212
5.7.2	点	213
5.7.3	直线	213
5.7.4	折线	215
5.7.5	矩形	215
5.7.6	区域	218
5.7.7	多边形	218

5.7.8	椭圆	218
5.7.9	贝塞尔曲线	218
5.8	位图	219
5.8.1	两种位图	219
5.8.2	DDB	220
5.8.3	DIB	222
5.9	综合示例：画板	223
5.10	本章技巧小结	268
第 6 章	多任务编程	281
6.1	使用线程	281
6.1.1	创建 Threads 程序	282
6.1.2	启动新线程	284
6.1.3	编写新线程的线程过程	285
6.1.4	从线程返回结果	287
6.1.5	通知程序线程已经结束	287
6.1.6	建立新 Windows 消息	288
6.1.7	报告线程结果	289
6.2	协调多线程	296
6.2.1	使用事件协调线程	296
6.2.2	创建 DoubleThreads 程序	298
6.2.3	启动线程	299
6.2.4	设置和重置事件	300
6.2.5	编写线程过程	301
6.2.6	编写 MinThread 进程	302
6.2.7	处理 Windows 消息	303
6.3	本章技巧小结	311
第 7 章	OLE 与 ActiveX 技术	317
7.1	OLE 与 Active 的有关概念	317
7.1.1	OLE 文档	317
7.1.2	链接与嵌入	317
7.1.3	OLE 容器与 OLE 服务器	318
7.1.4	就地激活（可视编辑）	318
7.1.5	自动化（Automation）	318
7.1.6	活动文档	318
7.1.7	ActiveX 控件	319

7.2	OLE 技术应用	319
7.2.1	OLE 文档服务程序	319
7.2.2	OLE 容器	357
7.2.3	活动文档服务程序	397
7.2.4	自动化	400
7.2.5	自动化程序实例	401
7.3	ActiveX 控件	445
7.3.1	ActiveX 简介	445
7.3.2	使用 ActiveX 控件	446
7.3.3	创建 ActiveX 控件	446
7.3.4	用 MFC 创建 ActiveX 控件	447
7.4	本章技巧小节	454
第 8 章	32 位内存处理	460
8.1	内存概述	460
8.2	功能强大的内存函数	461
8.3	使用虚拟内存	463
8.3.1	分配内存	463
8.3.2	提交内存	464
8.3.3	释放内存	464
8.4	扫描所有内存	465
8.5	内存映射文件	474
8.5.1	获得需要传递的数据	474
8.5.2	编写内存映射文件	475
8.5.3	创建内存文件视图	476
8.5.4	读内存映射文件	477
8.6	本章技巧小结	485
第 9 章	数据库编程	489
9.1	ADO 概述	489
9.2	在 Visual C++ 中使用 ADO	489
9.2.1	引入 ADO 库文件	489
9.2.2	初始化 OLE/COM 库环境	490
9.2.3	ADO 接口简介	490
9.2.4	创建连接	491
9.2.5	执行 SQL 命令并取得结果记录集	493
9.2.6	记录集的遍历、更新	497

9.2.7	关闭记录集与连接	498
9.2.8	绑定	498
9.2.9	事务处理	504
9.3	ADO 事件	505
9.3.1	ADO 事件概述	505
9.3.2	ADO 事件类型及其参数	506
9.3.3	事件处理程序如何共同工作	508
9.3.4	Visual C++ ADO 事件实例	510
9.3.5	ADO 事件小结	512
9.4	应用实例剖析	512
9.4.1	实例一：人事管理数据库	512
9.4.2	实例二：应用 ActiveX 控件开发应用程序	527
9.4.3	实例三：存储二进制大对象	532
9.5	本章技巧小节	554
第 10 章	Internet 编程	557
10.1	Internet 基础	557
10.1.1	分层	557
10.1.2	TCP/IP	558
10.1.3	HTTP 基础	564
10.1.4	FTP 基础	566
10.1.5	测试内部网—ping 程序	566
10.2	CSocket 程序设计	567
10.2.1	Winsock 和 MFC	567
10.2.2	建立联网的应用程序实例	582
10.2.3	两个利用 WinSock 进行通信的例子	591
10.3	本章技巧小结	603
第 11 章	在 Visual C++ 7.0 中编写 C#程序	606
11.1	C#简介	606
11.2	NGWS runtime 技术基础	609
11.2.1	NGWS Runtime	609
11.2.2	中间语言和元数据	610
11.2.3	即时编译器 (JITters)	611
11.3	虚拟对象系统 (VOS)	613
11.3.1	VOS 类型系统	613
11.3.2	元数据	614

11.3.3	通用语言规范	614
11.3.4	虚拟执行系统 (VES)	617
11.4	C#实例	617
11.4.1	实例一: Hello World.....	618
11.4.2	实例二: 使用 Windows 窗口的 Hello World	619
11.5	本章技巧小结	621

第 1 章 .NET 和受管扩展

1.1 什么是 Microsoft.NET

Microsoft.NET 是 Microsoft 公司的新战略，所有 Microsoft 产品都将围绕这个战略开发。简而言之，Microsoft.NET 能使用户通过 Web 与众多的智能设备交互，同时确保用户而不是应用程序控制这个交互。Microsoft.NET 能使得用户对应用程序、服务、个性化设备的体验简单、一致而安全。

Microsoft .NET 的策略是将互联网本身作为构建新一代操作系统的基础，对互联网和操作系统的设计思想进行合理延伸。这样，开发人员必将创建出摆脱设备硬件束缚的应用程序，以便轻松实现互联网连接。Microsoft .NET 无疑是当今计算机技术通向计算时代的一个非常重要的里程碑。

1.1.1 .NET 的核心组件

.NET 的核心组件包括：

- 一组用于创建互联网操作系统的构建块，其中包括 Passport.NET（用于用户认证）以及用于文件存储的服务、用户首选项管理、日历管理以及众多的其他任务。
- 构建和管理新一代服务的基本结构和工具，包括 Visual Studio.NET、.NET 企业服务器、.NET 框架和 Windows.NET。
- 能够启用新型智能互联网设备的.NET 设备软件。
- .NET 用户体验。

1.1.2 .NET 的重大意义

.NET 对最终用户来说非常重要，因为计算机的功能将会得到大幅度提升，同时计算机操作也会变得非常简单。特别地，用户将完全摆脱人为的硬件束缚：用户可以自由冲浪于互联网的多维时空，而不是束缚在便携式电脑的方寸空间——可通过任何桌面系统、任何便携式电脑、任何移动电话或 PDA 进行访问，并可对其进行跨应用程序的集成。

.NET 可使用户轻松进行互联网连接，并轻松完成那些在当今看来十分费时而且费力的事务，它们往往要求用户进行数据重输入并需运行几个小时才能完成。通过将多项安全数据流合并到单一的用户界面（或者甚至是可编程决策引擎），.NET 架构将用户从充斥于当今 Web 的数据陷阱的束缚中解脱出来。用户可以自由访问、自由查看、自由使用他们的数据。

.NET 对开发人员来说也十分重要，因为它不但会改变开发人员开发应用程序的方式，而且使得开发人员能创建出全新的各种应用程序。新型开发范例的核心是 Web 服务这个概念的引入。Web 服务是一种通过简单对象访问协议（SOAP），在互联网上展露其功能性的、极为公开的服务。SOAP 是一种基于可扩展标记语言（XML）制定的协议。

1.1.3 Web 服务崭露头角

在过去，开发人员通过集成本地系统服务来构建应用程序。在这种模型下，开发人员可以访问丰富的开发资源并能严格控制应用程序的行为。

如今，开发人员已在很大程度上挣脱了这种模型的束缚，致力于构建具有复杂结构的 n 层化系统，这种系统能将网络上众多的应用程序一并进行集成，大大提升了应用程序的价值。这样，开发人员便可把精力集中在充分挖掘软件独特的商业价值，而不是构建基本结构上。可喜的局面将应运而生：软件投放市场的时间大大缩短、开发人员的编程效率明显提高，最终把质量上乘的软件呈现给用户。

我们正在进入一个崭新的计算时代，一个由互联网（尤其是 Internet 核心技术 XML）实现的年代。利用 XML，能够创建出可供任何人从任何地方使用的、功能非常强大的应用程序。它极大地拓展了应用程序的功能，并实现了软件的动态提供。在这种情况下，软件已不完全指那些从光盘进行安装的程序，而是演变成了一种服务——类似于 ID 调用程序或按收看次数进行收费的电视——人们可通过通信媒体订购的服务。

n 层计算技术具有能够大幅度提高生产力、紧密耦合的特点，而 Web 概念具有面向消息、松散耦合的特点，我们将两者有机地糅合在一起，实现了上述构想。我们将这种计算风格称为 Web 服务，它的出现标志着人类已经迈入应用程序开发技术的新纪元。Web 服务是一种应用程序，它可以通过编程并使用标准的 Internet 协议，像超文本传输协议（HTTP）和 XML，将功能展示在互联网和企业内部网上。还可将 Web 服务视作 Web 上的组件编程。

从理论上讲，开发人员可通过调用 Web 应用编程接口（API），将 Web 服务集成到应用程序中。其调用方法与调用本地服务类似，不同的是 Web API 调用可通过互联网发送给位于远程系统中的某一服务。例如，Microsoft Passport（Passport）服务使得开发人员能够对应用程序进行认证。通过对 Passport 服务编程，开发人员可以充分利用 Passport 的基本结构，通过运行 Passport 来维护用户数据库，以确保其正常运行、定期备份等等。

.NET 正是根据这种 Web 服务原则而创建的，微软目前正着手提供这个基本结构，以便通过 .NET 平台的每一部分来实现这种新型的 Web 服务。而 Visual Studio.NET、NET 框架、Windows.NET 和 .NET 企业服务器，正是为进行基于 Web 服务模型的应用程序开发而度身定做的新一代开发工具和基本结构。.NET 构建块服务、新增的 .NET 设备支持以及即将到来的 .NET 用户体验，将为人们彻底攻克这一难题划上一个圆满的句号，使人们能够充分利用 Web 服务模型，如愿以偿地开发出新一代应用程序。

1.1.4 .NET 体验

.NET 体验是最终用户与 XML Web 服务交互的一种手段。

一次.NET 体验类似于一个用户与现行单个的应用程序交互。假设你想打开暖气，而你又不在家，这时你就可以进行一次.NET 体验，即通过智能设备控制你的家用电器。如果此刻你在办公室里，你可以用你的桌上型电脑；如果你正汽车上，你可以用你的智能电话；如果你是在商场买菜，你则可以用多媒体信息亭。

这一实例中.NET 的独特性在于：

- 每一个设备都为.NET 体验产生一个不同的界面——桌上型电脑可能会显示一个地形图并接受鼠标输入，智能手机会给你语音选择并要求你按键选择，多媒体信息亭可能会有一个触摸屏和一组温度控制选项。你可用同一.NET 体验完成同一任务，但是你看到的是适合于不同设备的截然不同的界面。
- .NET 体验是位置无关的——利用定位安装和基于 Web 应用程序两者各自的优势，.NET 体验可同时在线和脱机状态下进行，在前面的实例中，不管你在哪里：办公室、车里或商场，都可使用.NET 体验。
- .NET 体验是个性化的——.NET 体验读取存储在低层基础中的用户参数和配置信息，并自动将用户信息和参数结合在一起呈现出来。在前面的实例中，当你打开暖气，.NET 体验根据你的登录 ID 已经知道你住在哪里，你不需要输入地址。

1.2 在 VC++.NET 中进行.NET 编程

1.2.1 COM 与软件可复用性

可复用软件不是一个新概念。八年来，人们一直在使用各种形式的组件对象模型 (COM)。事实证明，它是最为成功的可复用软件模型。COM 引进了“组件”的概念——它是可复用的代码块，可以将多个独立函数的功能进行组合，从而扩充成诸如 Microsoft Word 这样的应用程序。

大多数开发人员使用 OLE 时深刻体验了 COM 功能。OLE 是基于 COM 形成的一组功能，使得用户能将一种文档嵌入到另一种文档中。这个功能本身似乎不太引人入胜，但它的作用却不同凡响：当用户将一个 Excel 文档粘贴到 Word 文档中后，单击嵌入的 Excel 文档时，OLE 将会把 Word 的工具栏和菜单转换成 Excel 的工具栏和菜单。

从开发人员的角度看，COM 通过引进几个明确定义的接口（诸如 IUnknown）便可提供代码复用功能，开发其工具可通过这些接口来查询一个组件的功能，并能把这些功能添加到工具中。这就像 Visual Basic 开发系统工具箱中的控件能够被拖到某个窗体中一样。实际上，每个控件都代表上百行甚至上千行的代码，可以容易地封装在“黑匣子”中，开发人员只需

直接调用其功能即可。

开发人员在使用 COM 时感到不便的一个问题是，他们必须编写附加代码来将业务逻辑程序转换成可复用的组件，同时还必须实现许多接口才能进行这种转换。最重要的是，COM 要求开发人员必须手动处理复杂问题，比如：清空不再使用的组件曾占用的内存、计算组件的使用次数、建立或撤消线程和进程以及处理版本控制问题等。

有人可能认为，让开发人员亲自执行这类工作的主意非常好，但这也有几个弊端。首先，要开发人员一一执行上述所有工作非常困难，往往容易出错：导致应用程序错误、系统崩溃以及可怕的“DLL Hell”。另外，严格地写出所有这种附加代码，会降低开发人员的工作效率，导致延期上市。

这对使用 Visual C++ 开发系统的开发人员来说，尤其如此。而对使用 Visual Basic 的开发人员来说，这种情况不是很严重。Visual Basic 抽象并简化了 COM 的许多概念，是全世界最具生产力、最流行的开发环境之一，但它的局限性在于：为了实现这种高生产力而不得不向开发人员屏蔽了 COM 的一些功能。

1.2.2 什么是 .NET 框架

微软在 2000 年的专业开发人员大会（PDC）上引进的 .NET 框架，能自动在软件编写过程中进行“智能拼接”，使得开发人员可以集中精力编写业务逻辑，而不必编写 COM 基本结构。

.NET 框架是一个多语言组件开发和执行环境，它由以下三个主要部分组成：

- 公共语言运行时（Common Language Runtime）

此名称不能准确反映它的全部功能。实际上，公共语言运行时在组件的开发及运行过程中，都扮演着非常重要的角色。在组件运行过程中，运行时负责管理内存分配、启动或删除线程和进程、实施安全性策略，同时满足当前组件对其他组件的需求。在开发阶段，运行时的作用有些变化：与现今的 COM 相比，运行时的自动化程度大为提高（比如可自动执行内存管理），因而开发人员的工作变得非常轻松。尤其是映射功能将锐减开发人员将业务逻辑程序转化成可复用组件的代码编写量。对编程语言而言，运行时这个概念并不新奇：实际上每种编程语言都有自己的运行时。Visual Basic 开发系统具有最为明显的运行时（名为 VBRUN），Visual C++ 跟 Visual FoxPro、JScript、SmallTalk、Perl、Python 和 Java 一样有一个运行时，即 MSVCRT。 .NET 框架的关键作用在于，它提供了一个跨编程语言的统一编程环境，这也是它能独树一帜的根本原因。在本书后面部分，我们把公共语言运行时简写为 CLR。

- 统一的编程类

.NET 框架为开发人员提供了一个统一、面向对象、层次化、可扩展的类库集（API）。现今，C++ 开发人员使用的是 Microsoft 基类库，Java 开发人员使用的是 Windows 基类库，而 Visual Basic 用户使用的是 Visual Basic API 集。只是简单地一用，.NET 框架就统一了微软当前的各种不同类框架。这样，开发人员无需学习多种框架就能顺利编程。远不止于此的是，通过创建跨编程语言的公共 API 集，.NET 框架可实现跨语言继承性、错误处理功能和调试

功能。实际上,从 JScript 到 C++ 的所有编程语言,都是相互等同的,开发人员可以自由选择理想的编程语言。在 .NET 框架的 System 名字空间中,提供了一百多个类,你可以用它来完成大多数应用。

- 活动服务器页面 (ASP+)

ASP+ 是使用 .NET 框架提供的编程类库构建而成的,它提供了 Web 应用程序模型,该模型由一组控件和一个基本结构组成。有了它,Web 应用程序的构建变得非常容易。开发人员可以直接使用 ASP+ 控件集,该控件集封装了公共的、用于超文本标识语言 (HTML) 用户界面的各种小器件 (诸如文本框、下拉菜单等等)。实际上,这些控件运行在 Web 服务器上,它们将用户界面转换成 HTML 格式后再发送给浏览器。在服务器上,控件负责将面向对象的编程模型提供给 Web 开发人员,这种编程模型能提供面向对象编程技术的丰富功能。ASP+ 还提供一些基本结构服务 (诸如会话状态管理和进程重启服务),这些服务大大减少了开发人员要编写的代码量,并使应用程序的可靠性得到大幅度提高。ASP+ 还允许开发人员将软件作为一项服务来提供。通过使用 ASP+ Web 服务功能,ASP+ 开发人员只需进行简单的业务逻辑编程,而由 ASP+ 基本结构负责通过简单对象访问协议 (SOAP) 来提供服务。

1.2.3 .NET 框架与 COM 及 COM+ 的关系

.NET 框架的一个主要目的是使 COM 开发变得更加容易。COM 开发过程中最难的一件事是处理 COM 基本结构。因此,为了简化 COM 开发,.NET 框架实际上已自动处理了所有在开发人员看来是与“COM”紧密相关的任务,包括引用计算、接口描述以及注册。

必须认识到,这并不意味着 .NET 框架组件不是 COM 组件。事实上,使用 Visual Studio 6.0 的 COM 开发人员可以调用 .NET 框架组件,并且在他们看来,后者更像是拥有 IUnknown 数据的 COM 组件。相反,使用 Visual Studio.NET 的 .NET 框架开发人员则将 COM 组件视作 .NET 框架组件。

为了避免引起误解,这里需对这种关系加以特别说明:COM 开发人员必须手动去做大多数 .NET 框架开发人员可以在运行时自动执行的事情。例如,必须手写 COM 组件的安全性模块,且无法自动管理模块占用的内存,而在安装 COM 组件时,注册条目必须放进 Windows 注册表中。对 .NET 框架而言,运行时实现了这些功能的自动化。例如,组件本身是自我描述型的,因而无需注册到 Windows 注册表中便能安装。

当把 COM 与 Microsoft 事务服务器 (MTS) 和分布式 COM (DCOM) 结合在一起时,就变成了 COM+。COM+ 提供了一组面向中间层的服务。特别是 COM+ 提供了进程管理功能和数据库与对象连接池处理功能。在将来的版本中,它还将提供一种称为分区的功能——专门为应用程序服务提供商设计的更强大的进程隔离功能。

COM+ 服务主要面向中间层应用程序开发,并主要为大型分布式应用程序提供可靠性和可扩展性。这些服务是对 .NET 框架所提供服务的补充;通过 .NET 框架类,可以直接访问这些服务。