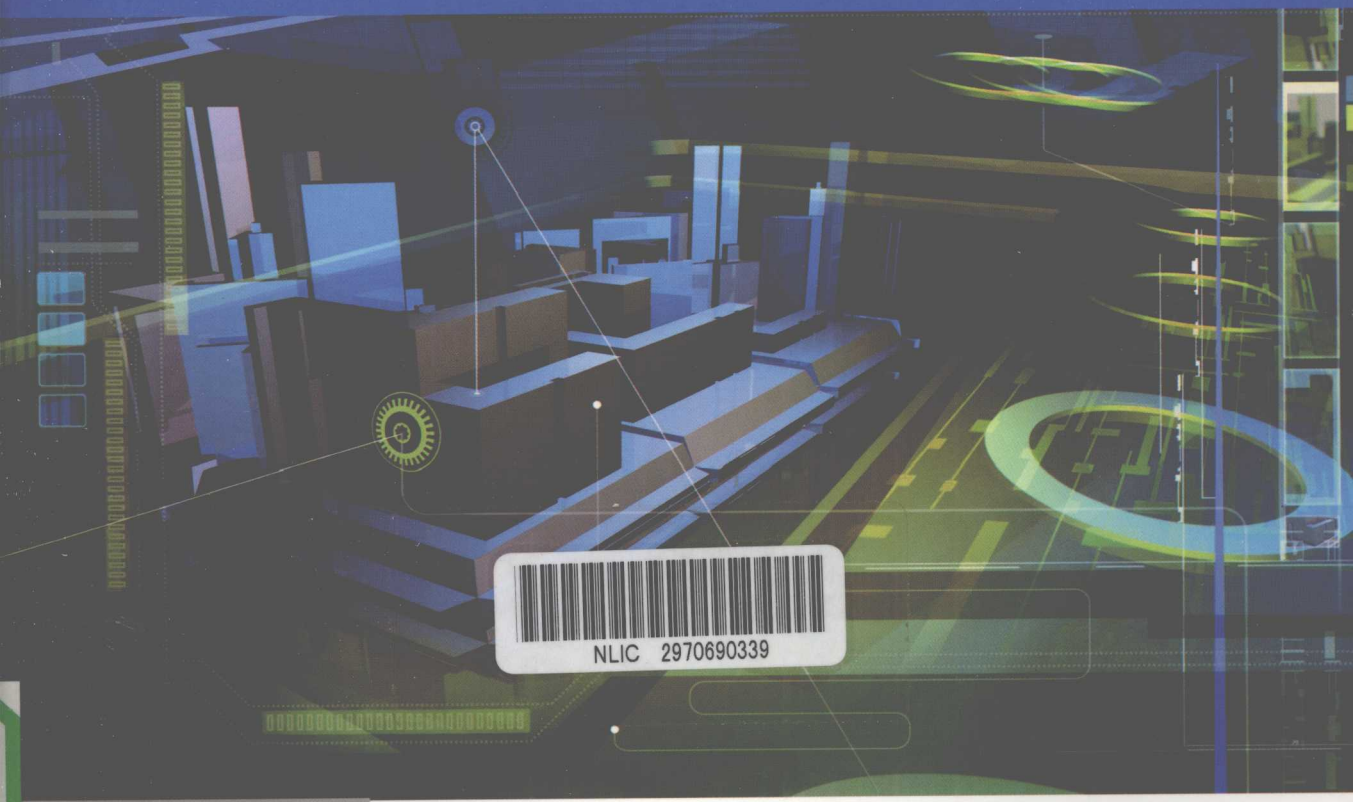
普通高等教育“十一五”国家级规划教材

# 工程化程序设计（VC++.NET）

孙连云　顾夏辉　主　编





高等教育出版社
HIGHER EDUCATION PRESS

普通高等教育“十一五”国家级规划教材

# 工程化程序设计（VC++.NET）

Gongchenghua Chengxu Sheji (VC++.NET)

孙连云　顾夏辉　主编




高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

## 内容提要

本书是普通高等教育“十一五”国家级规划教材。

本书基本涵盖了 Visual C++.NET 的主要内容，讲解循序渐进，结构严谨，对各个主题的知识介绍都与具体实战环节相结合，可操作性强。本书和其他同类书籍的最大不同体现在编写思路上，本书采用“边用边学，以用促学”，而不是“先学后用，用是为学”的撰写思路。这样，可以不断挖掘读者学习中遇到的问题，进而探究问题并最终引导读者解决问题。

本书共 13 章，主要内容包括：.NET 概念与 Windows 编程基础，面向对象编程基础，绘图与文本编程，设计应用程序外观，使用对话框和控件，动态链接库，多线程编程，COM 组件编程和 ATL 编程，网络编程，编写数据库程序，VC++.NET，Windows Forms 编程，64 位编程。

本书可作为培养应用性、技能型人才的计算机相关专业的教学用书，也可作为计算机培训教材以及计算机从业人员和爱好者的自学教材。

**图书在版编目（CIP）数据**

工程化程序设计：VC++.NET/孙连云，顾夏辉主编．
—北京：高等教育出版社，2010.11
ISBN 978-7-04-025026-8

Ⅰ.①工… Ⅱ.①孙…②顾… Ⅲ.①计算机网络-程序设计-高等学校-教材②C 语言-程序设计-高等学校-教材 Ⅳ.①TP393.09②TP312

中国版本图书馆 CIP 数据核字（2010）第 182938 号

**策划编辑** 杜 冰 **责任编辑** 许 可 **封面设计** 张志奇 **责任绘图** 尹 莉
**版式设计** 张 岚 **责任校对** 杨凤玲 **责任印制** 陈伟光

**出版发行** 高等教育出版社
**社 址** 北京市西城区德外大街 4 号
**邮政编码** 100120

**经 销** 蓝色畅想图书发行有限公司
**印 刷** 北京印刷一厂

**开 本** 787×1092 1/16
**印 张** 20.25
**字 数** 500 000

**购书热线** 010-58581118
**咨询电话** 400-810-0598
**网 址** http://www.hep.edu.cn
http://www.hep.com.cn
**网上订购** http://www.landraco.com
http://www.landraco.com.cn
**畅想教育** http://www.widedu.com

**版 次** 2010 年 11 月第 1 版
**印 次** 2010 年 11 月第 1 次印刷
**定 价** 27.70 元


**物料号 25026-00**

# 前　言

Visual C++是近年来在国内外得到广泛应用的可视化、面向对象的编程语言之一。由于利用 Visual C++开发的应用程序具有与 Windows 操作系统结合紧密、可执行代码效率高等特点，因此它一直都是专业人员开发 Windows 应用程序的首选工具。

为了配合新的.NET 战略，微软公司推出了最新的开发工具 Visual Studio.NET。Visual C++.NET 是 Visual Studio.NET 系列中重要的组成部分之一。Visual C++.NET 的程序库进行了许多更新和扩充，包括 ATL 服务器、MFC、C 的动态链接库、OLE DB 模板、共享类、标准的 C++库等，并增加了 Web Service 开发，新的控件和宏等。为了便于开发者使用，Visual C++.NET 提供了许多方便实用的功能，这些在本书中都有所体现。

在编写本书时，作者充分考虑初学者学习程序语言的特点，对基本知识的讲解力求做到深入细致，并结合大量实例，通过详细的操作步骤，帮助读者培养实际编程能力。

本书基本上涵盖了 Visual C++.NET 的主要内容，讲解循序渐进，结构严谨，对各个主题的知识介绍都与具体的实例相结合。书中每章都设有实战演练环节，以加深读者对知识点的理解和把握，从而帮助读者增强分析问题和解决问题的能力。

本书和其他同类书籍的最大不同体现在编写思路上：本书采用“边用边学，以用促学”，而不是“先学后用，用是为学”的撰写思路。这样，可以不断挖掘读者学习中遇到的问题，进而探究问题并最终引导读者解决问题。

本书共 13 章：

第 1 章简要介绍.NET 概念和 Visual C++.NET 开发环境，之后阐述了 Windows 编程原理，目的是帮助读者初步了解.NET 程序开发，熟悉 Visual C++.NET 开发环境，并通过实战演练环节对 Visual C++.NET 编程有一个直观的认识。

第 2 章介绍面向对象编程的基础内容，目的是帮助没有 C++基础的读者快速学习 C++的精髓，本章的实战演练——一个小型公司的人员信息管理系统浓缩了 C++的重要知识点。

第 3 章介绍绘图与文本编程方面的内容，目的是学习如何使用画笔、画刷和位图等绘图工具为应用程序添加丰富多彩的图形，以及如何使 VC++程序像 Word 软件一样具有文本编辑功能。实战演练——设计一个绘制程序背景的程序凝聚了本章各个知识点。

第 4 章介绍设计应用程序外观方面的内容，目的是学习如何设计应用程序的菜单、工具栏和状态栏等界面元素，从而使程序界面更友好。实战演练——设计一个时钟程序向读者呈现了一个完整的外观设计。

第 5 章介绍对话框和控件的操作，目的是学习对话框作为 Windows 程序中常用的与用户交互的窗口是如何通过放置其上面的控件来完成显示和获取用户输入。实战演练——设计一个单位换算程序呈现了使用控件的具体方法和步骤。

第 6 章介绍动态链接库，目的是学习动态链接库的基础知识。通过实战演练，读者可以学

会使用 MFC 创建和使用扩展动态链接库及规则动态链接库，并在客户程序中使用 DLL。

第 7 章介绍多线程，首先详细讲解了多线程的理论基础，包括基本的线程调度、线程互斥、线程间通信等，最后通过实例介绍了如何利用主流操作系统中的多线程技术来开发稳定的应用程序。

第 8 章介绍 COM 组件编程和 ATL 编程，目的是了解 COM 组件的功能、MFC 对 COM 编程的支持，学会使用 ATL 创建 COM 组件以及如何使用由 ATL 创建的组件。通过实例读者可以学会编写 ATL Server 程序和客户端程序。

第 9 章介绍网络编程方面的知识，由于内容较多，考虑到读者学习难度较大，本章除了介绍基本理论知识外，还通过实战演练环节详细介绍了如何利用 CHtmlView 创建一个简单的 Web 浏览器，编写 FTP 客户程序，使用 CSocket 自制聊天工具。

第 10 章介绍编写数据库程序，目的是学习使用 VC++所提供的数据库访问技术 ADO 编写数据库程序。本章的实战演练——考试系统的开发与设计综合应用了本书前面章节中所学知识，如 COM 技术、程序外观、界面元素等。

第 11 章介绍 VC++.NET 在.NET 框架下的语言特性和功能特点，C++/CLI 如何从 ISO C++ 语言标准发展而来，Visual Studio 2008 集成开发环境提供的 VC++.NET 工程类型，最后创建了一个 VC++.NET 的应用程序。

第 12 章介绍.NET 的 Windows Forms 编程以及 Windows Forms 编程和 MFC 编程之间的区别，创建了一个 Windows Forms 应用程序和一个通过 Windows Forms 进行绘图的应用程序。在 Windows Forms 绘图一节中还练习了如何实现在 Form 中进行鼠标事件和绘图的互操作。

第 13 章介绍 64 位系统下进行编程的特点和容易出现的错误，由于目前处理器已经进入 64 位时代，64 位编程将会代替 32 位编程成为主要的编程平台，最后介绍 32 位应用程序和 64 位应用程序进行互操作的解决方案。

本书由孙连云、顾夏辉担任主编，王金库、靳广斌担任副主编，孙连云编写第 1、2、3、4、5、9 章，王金库编写第 6 章，马骋、靳广斌编写第 7、8 章，第 10 章由孙连云、王铮、张晓暖、马冰冰共同编写，顾夏辉编写第 11、12、13 章。

全书由微软（中国）有限公司张广军、北华航天工业学院安志远审稿，在此表示衷心的感谢。

本书所有实例都在 Visual Studio .NET 上调试通过。

由于时间仓促，书中难免有不妥之处，祈望读者批评指正。

编　者

2010 年 9 月

# 目　录

# 第 1 章
# .NET 概念与 Windows 编程基础

C++作为成功的编程语言，从它面世以来，一直受到众多程序员的青睐，很多里程碑式的程序设计技术都是基于这一经典的编程语言。而 VC++作为一款成功的编程工具和框架，为 Windows 编程提供了许多便捷、强大的支持。本章介绍在 VC++之后，VC++.NET 出现的原因以及一些基础概念，并且学习通过 VC++进行 Windows 编程的基础。

本章主要完成以下任务：

- 了解 VC++.NET 框架出现的原因。
- 学习 VC++.NET 的编译工具。
- 学习 Windows 下使用 VC++进行程序设计的一些概念和基础。

## ▶ 1.1 .NET 基础

### ▶▶ 1.1.1 从 COM 技术说起

COM 技术是一种通过接口方式进行组件间协作的模型。由于 COM 仅定义了组件间协作的规范，所以它提供了一种平台无关性并且不依赖于特定语言的规范。

COM 技术因为其语言无关的特性，近年来在 Windows 系统下得到了迅速发展。其优点包括：

- 通过交互式数据语言（Interactive Data Language，IDL）定义各个组件的接口，从而使得用户可以调用和具体实现解耦，并提供了语言无关性的特点。
- 每个 COM 对象和 COM 接口都有特定的 GUID 与之对应。通过 GUID，用户可以通过 COM 库提供的一些接口加载并获得对应接口指针。而且在 Windows 操作系统中，COM 组件可以把所包含的 COM 对象和 COM 接口的 GUID 注册到 Windows 注册表中，从而使得用户不必了解 COM 组件文件路径即可将其加载。COM 技术提供了一种加载 COM 组件的机制。
- COM 技术通过存根/代理(stub/proxy)的方式将 COM 组件的载体透明化。在 Windows 操作系统中，COM 组件是以 DLL 和 EXE 为载体的。以 DLL 为载体的 COM 组件是进程内加载，以 EXE 为载体的 COM 组件将以进程外加载的形式进行。由于 COM 技术实现了存根/代理的机制，所以用户在加载一个 COM 组件的时候并不用考虑进程内和进程外加载的不同。

COM 技术具有诸多优点，但是在各种技术和应用层出不穷的今天，它依然暴露出了一些

缺点，例如：

- 尽管 COM 技术并没有限制组件的实现方式和编程语言，但是它并没有做到跨平台应用。在 Windows 操作系统下实现的 COM 组件，在其他的操作系统下是无法运行的。开发团队不得不在组件的开发过程中考虑代码的可移植性，或者干脆为每个目标操作系统开发一个版本，并对各个版本进行繁重的维护工作。
- COM 技术可以跨语言使用，如用 C++编写的 COM 组件可以由 C#调用。但由于 C++中可以使用指针，而在 C#中没有指针类型，为了可以实现跨语言调用，开发人员需要小心地选择可使用的数据类型。

COM 技术的这些问题，限制了这种组件模型的应用。在当今越来越复杂的应用程序和网络技术应用中，使用新的模型势在必行。

### ▶▶ 1.1.2 .NET 基本概念

一直以来，计算机应用程序的开发与运行都是针对于特定的操作系统平台的。某个应用程序若想在不同的操作系统平台下运行，就必须针对某一特定的操作系统进行修改，这无疑削弱了应用程序的可移植性，也增加了应用程序开发的复杂性。为了改变这种状况，业内提出了“让应用程序的运行独立于操作系统”的构想；也就是说，让程序员开发的应用程序可以方便地跨平台（操作系统）运行，做到“一次开发，多处使用”。

微软公司作为计算机工业界的领头企业，其倡导的 Windows 程序开发思想与技术已取得了空前的成功，它自然希望能通过 Windows 程序开发的理念去实现上述构想。由此，.NET 平台技术概念应运而生。

### ▶▶ 1.1.3 .NET 框架

将 Windows 操作系统配置为.NET 平台环境，其核心是在 Windows 操作系统上安装.NET 框架（.NET Framework）。.NET 框架以 Windows 操作系统内部组件的方式实现下列功能。

- 提供一个一致的面向对象的编程环境，无论对象代码是在本地存储和执行，还是在本地执行、在 Internet 上分布，或者是在远程执行。
- 提供一个软件部署和版本控制冲突最小化的代码执行环境。
- 提供一个可提高代码（包括由未知的或不完全受信任的第三方创建的代码）执行安全性的代码执行环境。
- 提供一个可消除脚本环境或解释环境性能问题的代码执行环境。
- 使开发人员的经验在面对类型大不相同的应用程序（如基于 Windows 的应用程序和基于 Web 的应用程序）时保持一致。
- 按照工业标准生成所有通信，以确保基于.NET 框架的代码可与其他代码集成。

.NET 框架由两个主要部分构成：公共语言运行时（Common Language Runtime，CLR）和 .NET 框架类库（.NET Framework Class Library）。

公共语言运行时是.NET 框架的基础，也是.NET 程序运行的环境。它的功能是在应用程序执行时进行代码管理。代码管理包括内存管理、线程管理、代码执行、代码安全验证、编译以及其他系统服务。

.NET 框架的另一个主要组成部分是.NET 框架类库，它是一个综合性的面向对象的可重用类型集合。开发.NET 应用程序必须使用.NET 框架类库。它实际上是对 MFC（Microsoft Foundation Classes，微软基本类库；将会在 1.3 节中着重介绍）的扩充。

图 1-1 说明了.NET 框架与 Windows 操作系统之间的关系。

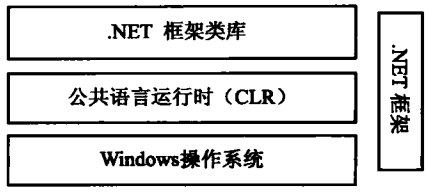



图 1-1　将 Windows 操作系统升级配置为.NET 环境

2001 年，微软公司推出了.NET 框架 1.0 版本。接下来的几年内，微软公司不断地对其进行升级，到目前为止，已经是 4.0 版本了。在 Windows Server 2003、Windows Vista 以及后续版本的 Windows 系统中都默认集成了.NET 框架。之前版本的 Windows 操作系统需要下载并安装.NET 框架，微软公司的网站提供了最新版本的.NET 框架安装包的免费下载。

### ▶▶ 1.1.4　公共语言运行时

公共语言运行时在继承了 COM 技术优点的同时也解决了 COM 技术带来的各种问题。

1．公共类型系统

公共语言运行时建立了公共类型系统（Common Type System），运行在公共语言运行时中的组件都需要使用公共类型系统。通过公共类型系统的支持，各种语言可以实现所有类型通用。

2．System.Object

公共语言运行时要求所有的对象都由 System.Object 派生。System.Object 提供了一些方法以保证公共语言运行时更好地管理对象。System.Object 有以下方法：Equals、GetType、ToString、MemberwiseClone 和 Finalize 等。Equals 用来比较两个对象内容是否相同；GetType 用来获得运行时对象类型；ToString 用来返回对象类型；MemberwiseClone 用来进行对象的深层复制，它会调用每个聚合对象的 MwmberwiseClone 方法，从而实现对该对象聚合的所有对象的复制；Finalize 方法用来执行对象被销毁前自定义的一些清理操作。

3．程序集

.NET 框架的最小单元是组件模块，组件模块包括的内容如表 1-1 所示。

**表 1-1　组件模块所包含内容**

| 内　容 | 描　述 |
| --- | --- |
| PE 表头 | 指定该程序集类型是可执行文件或者动态链接库；除此之外，PE 表头还包括了创建时间 |
| CLR 表头 | 该表头包括组件模块所运行的 CLR 版本、组件入口函数的元数据 |
| 元数据 | 元数据分为两种，一种用于描述源代码中定义的类型和成员，另一种用来描述源代码中所引用的类型和成员 |
| 中间语言代码 | 编译器在编译过程中生成的代码，在组件运行时会被编译成 CPU 指令 |

组件模块并不会直接运行在公共语言运行时中，它需要一个载体才能运行。组件模块的载体和 COM 组件类似，都是 DLL 和 EXE。不同的是，在 DLL 和 EXE 中保存的不是 CPU 指令代码，而是.NET 中间语言代码。.NET 的组件单元又被称为程序集(Assembly)，每个程序集都包括一个程序集清单，以及包含在程序集之中的组件模块。程序集清单描述了程序集使用公共运行时的版本信息，以及这个程序集和其他程序集的依赖关系信息。一个程序集可以包含多个组件模块。

4．启动公共语言时

程序集需要在公共运行时中才可以执行。公共语言运行时是如何被启动的呢？在程序集被运行的时候，会调用 Mscoree.dll 方法来加载公共运行时。MScoree.dll 是一个非托管 DLL。这样在开始执行程序集代码之前，公共运行时已经完成了初始化。

5．.NET 中间语言和即时编译

程序集中保存的并不是 CPU 指令代码，而是.NET 中间语言代码。目前的 CPU 并不能直接运行这种代码，所以在 CPU 执行相应的任务之前，需要先将.NET 中间语言代码转换为 CPU 指令代码。这个过程是由公共运行时的即时编译器完成的。程序集中的方法在第一次运行时，公共语言运行时会通过 Mscoree.dll 将.NET 中间语言转化为 CPU 指令代码。

程序集为什么不直接保存 CPU 指令代码，而保存.NET 中间语言代码呢？这是由于通过.NET 中间语言代码可以使.NET 消除语言的边界。在.NET 框架之中，不管是用哪种语言进行编程，只要可以将所使用的语言编译成.NET 中间语言，就可以在公共语言运行时中运行，而.NET 中间语言会在程序运行的时候被即时编译（Just-in-time Compiling）成 CPU 指令。

6．托管服务

所有可以运行在公共语言运行时中的语言都可以认为是托管语言，因为公共语言运行时会帮助这些语言做一些琐碎、繁重的运行和维护工作。这些工作包括内存管理服务、跨语言集成管理服务、代码访问安全性服务以及对象生存周期管理服务。通过这些服务，运行的语言实现由公共语言运行时托管。

前面已经介绍过，通过中间语言和即时编译可以做到跨语言集成管理。另外，即时编译过程会对中间语言进行代码验证过程，从而保证代码访问的安全性。代码验证的过程是一个烦琐的工作，它包括内存非法访问检查、方法的参数数量和类型检查、类型安全检查等。

C++和 C 语言可以直接对指针和堆进行操作，这就需要程序员小心地管理对象的生存周期，在正确的时机销毁指针或者删除对象，否则就会造成内存非法访问或者内存泄露。只有经验极为丰富的程序员才能够驾驭大型系统众多对象的生存周期。公共语言运行时将对象分配在自己申请的一块内存空间上，该空间以堆的形式对内存进行管理，并记录内存的占用状态。这个堆被称为托管堆，通过这个托管堆，公共语言运行时可以进行自动的内存管理。

同时，公共语言运行时为托管语言提供了垃圾回收机制。垃圾回收机制自动对对象的生存周期进行管理，会在适当的时刻对内存进行清理，从而保证程序可以正常地运行。

运行在公共语言运行时中的程序，会将对象分配在一个托管堆上，所有的对象根据引用关系组成一个图。公共语言运行时存储着图的根。通过根，垃圾回收机制就可以利用算法遍历所有被引用的对象。当执行垃圾回收时，垃圾回收机制首先会挂起应用程序中的所有线程，然后通过根遍历整个对象图。没有在图中出现的对象，都需要从托管堆中清理。由此可见，垃圾回

收过程是一个复杂并占用时间的过程。程序员并不能真正决定垃圾回收何时会被执行。垃圾回收会通过以下几种方法触发：应用程序运行退出，显式调用 System.GC 的 Collect 静态方法，其他执行时间点。

由于垃圾回收机制会自动地管理对象的生存周期，所有对象都可以被正常地销毁，因此公共语言运行时并不支持析构函数。如果想要在对象被销毁之前得到通知，可以重载 System.Object 的 Finalize 方法。当垃圾回收准备销毁一个对象时，首先会调用该方法。但重写 Finalize 方法可能会影响垃圾回收算法，所以只有在对象拥有非托管资源时，才重写 Finalize 方法以对非托管资源进行处理。Finalize 方法会在以下情况下被调用，包括托管堆占用已满，显式调用 System.GC 的 Collect 静态方法，公共语言运行时准备卸载应用程序域（有关应用程序域稍后会介绍），公共语言运行时被终止。正确理解垃圾回收机制并合理地对其进行应用能够有效地提高程序的效率和质量。

7．应用程序域

在 Windows 操作系统中，进程是最大粒度的执行边界。同一个进程内的各个资源可以共享和使用同样的地址空间，但是进程之间都有着互相独立的地址空间，每个进程的地址空间都是相对安全的。在公共语言运行时中执行的应用程序，不仅进程之间的地址空间是独立的，而且公共语言运行时还可以将一个进程划分出多个应用程序域（AppDomain），每个应用程序域都有着自己安全执行的边界。通过应用程序域，可以使代码在各个域中独立运行，互不影响，在需要通信的时候又不需要像跨进程访问那样占用更多的资源和开销，常见的 IIS 服务就是一个典型的应用程序域的例子。IIS 为每个访问网页服务器的客户端创建一个应用程序域，这样各个客户端在访问网页服务器的时候信息是相互独立的。

8．公共语言规范

各种编程语言只要能够编译出符合规范的.NET 中间语言就可以运行在公共语言运行时中，但并不是所有编程语言都可以运行在公共语言运行时中的，只有符合公共语言规范(CLS)的编程语言才能运行在公共语言运行时中。公共语言规范规定了编程语言需要遵守的公共类型系统和其他规范，使得运行在公共语言运行时中的各种语言可以正常地运行并能够互相通信。

## ▶ 1.2 Visual C++.NET 编程工具

### ▶▶ 1.2.1 为何要选择 Visual C++.NET

Visual C++.NET 由 Visual C++ 6.0 演化而来。Visual C++ 6.0 是微软公司推出的使用 C++语言开发 Windows 应用程序最成熟的可视化集成开发环境。Visual C++.NET 与 Visual C++ 6.0 相比，增加了.NET 框架软件开发工具包（.NET Framework Software Development Kit，.NET Framework SDK）。这个工具包中包括生成、测试和部署.NET 应用程序时所需要的编译器、命令行工具、文档以及示例等。由于 Visual C++6.0 的巨大成功，Visual C++.NET 延续了所有 Visual C++ 6.0 的程序开发思想和开发工具（如 MFC）。

另一方面，.NET 框架以 Windows 操作系统的基础组件形式实现，是用 C++语言编写而成的，因此，使用 Visual C++.NET 进行程序开发是深刻理解.NET 框架的基础。

## ▶▶ 1.2.2 Visual C++.NET 的安装

本书选择 Visual Studio.NET Team System 版本作为演示示例。

第一步，下载 Visual Studio.NET Team System 并运行其中的安装文件。出现图 1-2 所示的页面后，选择“Install Visual Studio 2008”继续安装。

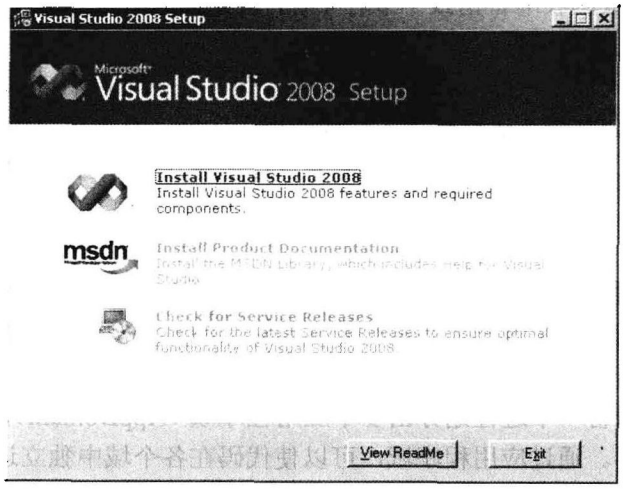



图 1-2 安装 Visual Studio.NET

第二步，选择“I have read and accept the license terms”（同意软件授权及安装协议），如图 1-3 所示。

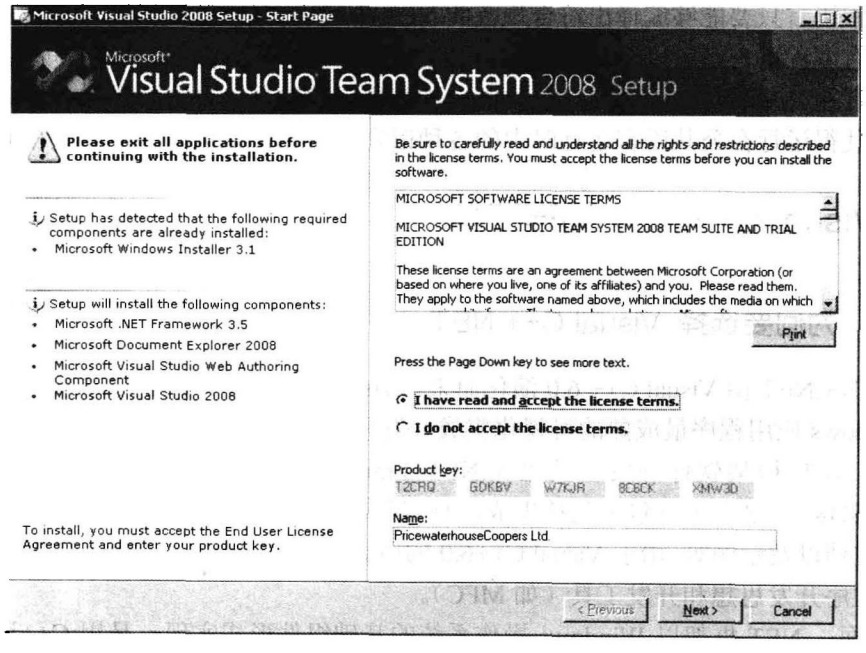



图 1-3 同意软件授权及安装协议

第三步，选择“Custom”（自定义安装），并设置安装路径，如图 1-4 所示。

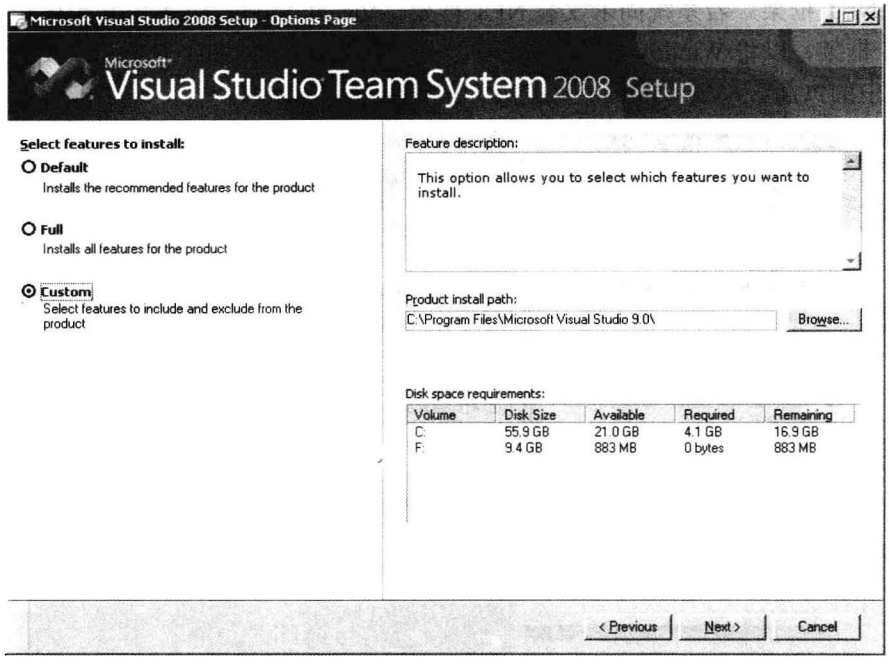



图 1-4　选择“自定义安装”，并设置安装路径

第四步，由于本书只涉及使用 Visual C++.NET 进行程序开发的内容，所以在编程语言选项中选择仅安装 Visual C++.NET，如图 1-5 所示。

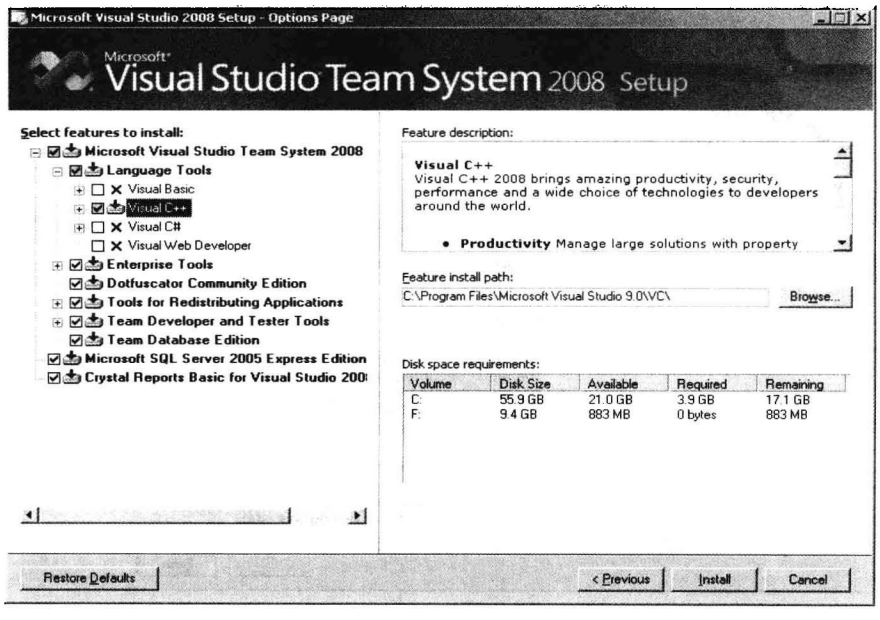



图 1-5　仅安装 Visual C++.NET

第五步，安装继续并直至安装完毕。Visual Studio.NET 在安装过程中会自动检测操作系统是否已安装.NET 框架。若系统尚未安装.NET 框架，Visual Studio.NET 安装程序会自动在系统中安装.NET 框架，以将 Windows 操作系统升级到.NET 环境，如图 1-6 所示。安装成功后，会出现图 1-7 中的页面，单击“Finish”（完成）按钮便可完成安装。

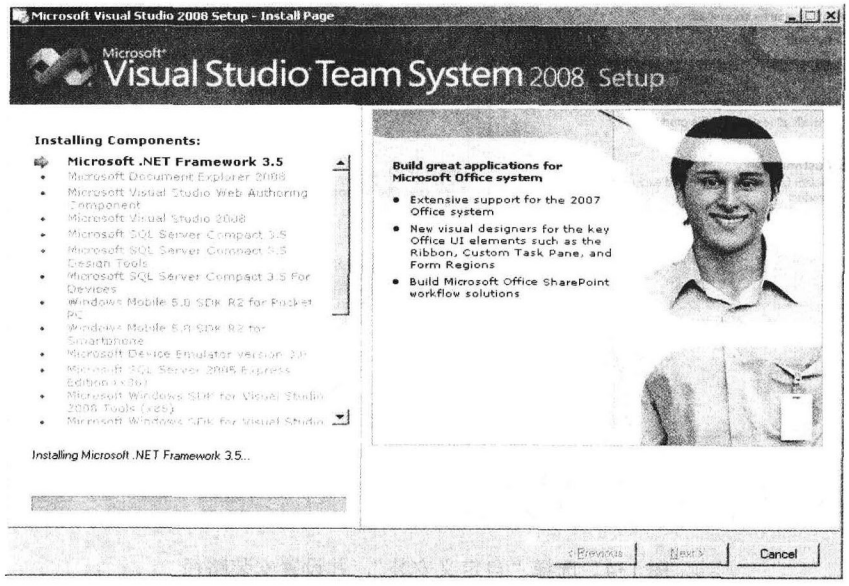



图 1-6　开始安装 Visual C++.NET 和.NET 框架

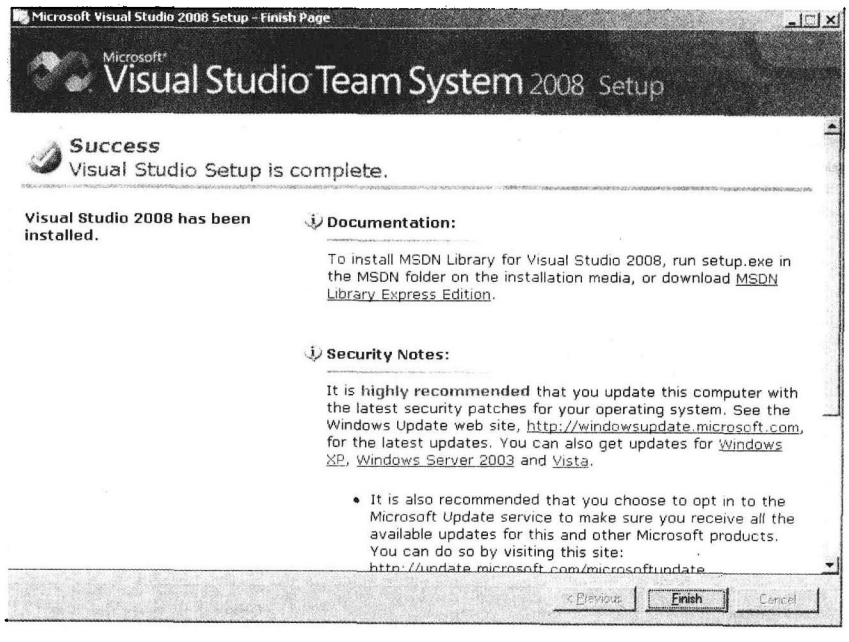



图 1-7　Visual C++.NET 安装完毕

第六步，安装成功后，第一次运行 Visual Studio.NET，会出现“Choose Default Environment Settings”（选择默认开发环境）的对话框。选择“Visual C++ Development Settings”（Visual C++开发环境）选项，如图 1-8 所示。

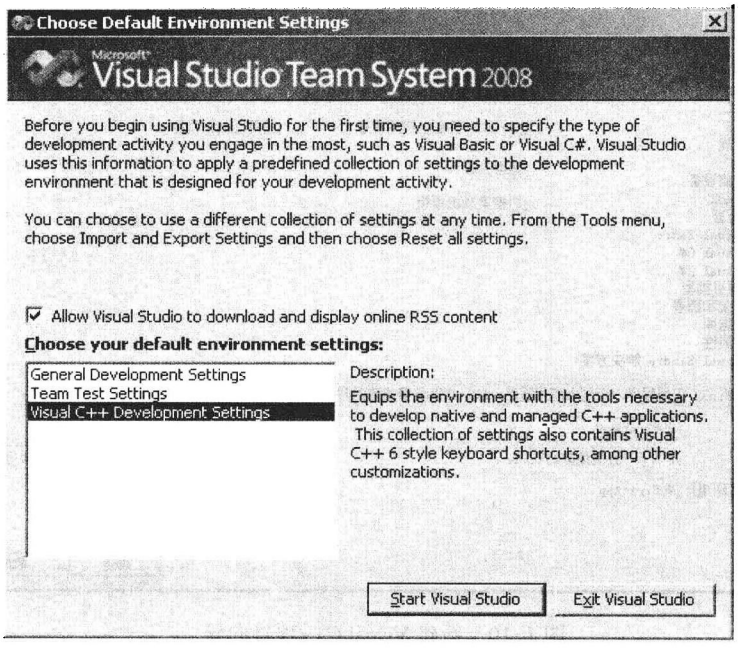



图 1-8　配置 Visual Studio.NET 默认开发环境

## ▶▶ 1.2.3　Visual C++.NET 开发界面

成功安装 Visual C++.NET 后，通过“开始→程序→Microsoft Visual Studio 2008→Microsoft Visual Studio 2008”命令进入 Visual C++.NET 开发环境，如图 1-9 所示。

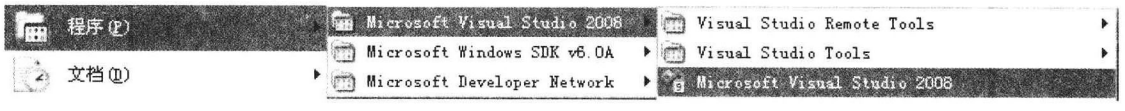



图 1-9　进入 Visual C++.NET 开发环境

Visual Studio.NET 提供了开发各种 Visual C++.NET 应用程序的模板，如图 1-10 所示。模板已经生成了应用程序的主干框架，这样可以大大提高应用程序的开发效率。

图 1-10 中，在“项目类型”下选择并展开 Visual C++，可以看到 ATL、CLR、常规、MFC、智能设备、Win32 项目类型，可以根据实际需要选择不同的项目类型。此处选择 Win32，窗口右侧“模板”中会列出 Visual Studio 已安装的模板：Win32 控制台应用程序(第 2 章讲述 C++语言基础所创建的程序均为控制台应用程序)和 Win32 项目。选择 Win32 项目，然后在项目名称文本框中输入 MyFirstApp，并为该项目选择一个合适的路径，选中“创建解决方案的目录”复选框，单击“确定”按钮，会打开如图 1-11 所示的“Win32 应用程序向导-MyFirstApp”对话

框，该向导提供了概述和应用程序设置（不同类型的项目向导内容是不同的）两部分内容，完成所有设置后单击“完成”按钮，就会生成 Win32 应用程序框架，同时会打开如图 1-12 所示的应用程序开发界面。

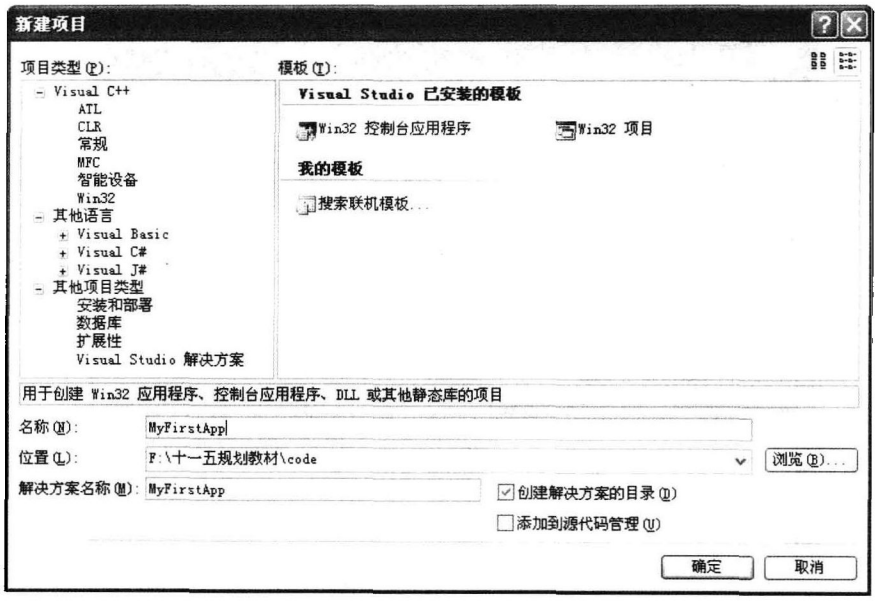



图 1-10　新建 Visual C++项目向导

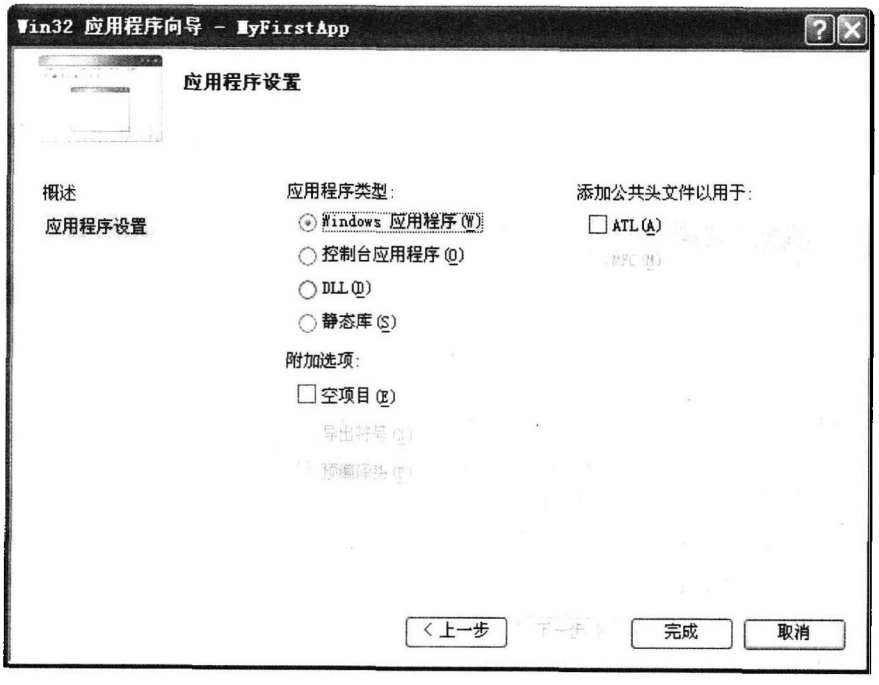



图 1-11　Win32 应用程序向导