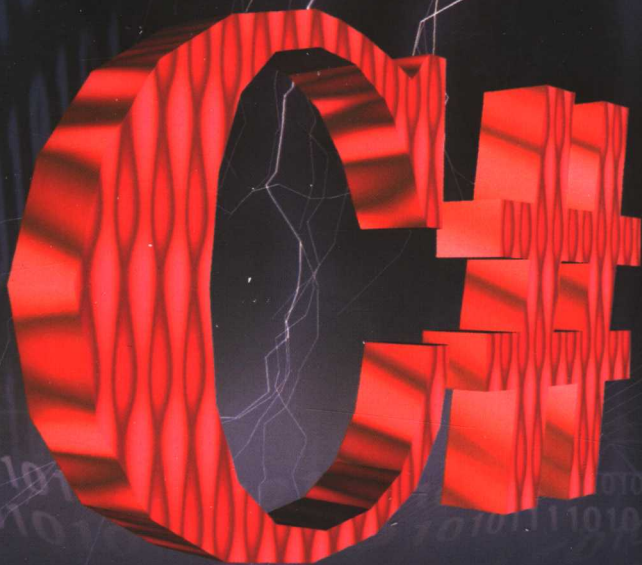


C# 高级编程



学习捷径

刘洪成 编著



清华大学出版社

C#高级编程学习捷径

刘洪成 编著

清华大学出版社

北 京

内 容 简 介

本着快速实用的目的,本书作者将其在美国几家大公司设计完成的数个大型项目分类浓缩后有机地融入本书,实现基础知识和实际开发经验的完美结合,为有一定编程经验的计算机软件开发人员提供了一条快速开发 C#程序的学习捷径。

本书的最大特点是:在介绍基础知识的过程中,结合实例强调了各知识点容易出现的编程错误及相应处理办法;实例部分则用软件工程的思想指导读者去开发一个大型应用系统,从 Windows 的前端界面到最底层数据库设计,中间经历网络服务程序和数据存取组件的开发,通过一步步地分析讲解,向读者展示 C#项目开发技术和目前美国软件生产基本流程,相当于带领读者参加了一个大型数据处理系统的研制、开发和调试的全过程。

本书适合有一定基础的读者、大专院校师生、企业技术开发人员学习参考,也适用于各类计算机培训班学员学习。

版权所有,盗版必究。

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

C#高级编程学习捷径/刘洪成编著. —北京:清华大学出版社, 2003

ISBN 7-302-06641-8

I. C#… II. 刘… III. C 语言—程序设计 IV. TP 312

中国版本图书馆 CIP 数据核字(2003)第 036551 号

出版者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.com.cn>

印刷者:北京市耀华印刷有限公司

发行者:新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 22.125 字数: 538 千字

版 次: 2003 年 7 月第 1 版 2003 年 7 月第 1 次印刷

书 号: ISBN 7-302-06641-8/TP·4969

印 数: 0001~4000

定 价: 30.00 元

献给我的妻子郭松平和儿子刘心恺、刘心靖。

——刘洪成

前 言

C#.NET 是微软在 2002 年初全力推出的最新的计算机编程语言。配合微软的 .NET 新技术平台, C#.NET 使得网络编程更快并且更容易。虽然 C# 正式推出时间极为短暂, 美国 Evans Data Corporation 在 2002 年春的一份调查已经显示, 28.6% 的商业用户正在或者打算采用 .NET 作为其网络服务 Web Service 的技术平台, 比较之下, 只有 27.7% 的用户正在或者打算采用 Sun 公司的 J2EE Java 平台 (详细调查可参见 http://www.evansdata.com/naTOC_02-1_xmp2.htm)。关于 C#.NET, 调查显示 12% 的用户正在使用, 另外 24% 的用户表示计划在 2002 年—2003 年开始使用。由此可以预见, .NET 的市场走势有多么强劲。

具体地说, C#.NET 是一门完全面向对象的编程语言, 其类型检查非常严格, 从而摒弃了以往 C++ 之类语言的类型不安全性。C#.NET 为微软 .NET 新技术平台的核心语言, .NET 平台本身的绝大部分程序都是用 C# 来编写的, 从而其编译器与 .NET 所支持的同类语言编译器相比, 已经得到了严格的测试。微软推出的 C#.NET 集成开发环境也使得用 C# 进行网络编程非常容易, 从而使得大型程序的开发周期大大缩短, 明显提高了软件开发的效率。随着 .NET 技术的迅速普及, 对 C# 软件人员的需求也自然会不断增大, 因而对于目前在 IT 行业中的大量的 C++/Java 工作人员来说, 如何能够快速地转移到 C# 这一新的领域便成为一个迫切的问题。

作者作为一名资深的 C++/Java 软件开发人员, 同时为微软认证的方案集成设计师 (MCSD) 和 Sun 认证的 Java 程序员 (SCJP), 从 2001 年当 C# 尚处在开发阶段时即已经着手对其进行试用, 历经 Beta 1, Beta 2, Release Candidate 直到后来的正式版。作者首先在美国一家互联网软件咨询公司中用 C#.NET 设计开发出了一个建筑业项目投标预评估系统, 目前又在一家金融理财公司中负责设计一套学生贷款数据处理系统, 积累了大量的 C# 使用经验。

本着快速实用的目的, 作者将其开发过的项目经过分类浓缩后有机地融入本书, 通过一步步地分析讲解, 快速地带领读者开始应用 C# 进行大型应用程序的开发和研制。作者在书中的上篇详细介绍了 C# 的基础知识, 包括 C# 数据类型、C# 类、控制语句、异常处理、输入输出、多线程操作和未操控语句等; 下篇则运用软件工程的思想去指导读者来开发一个大型的应用程序系统, 从 Windows 或者 Internet 前端界面到最底层的数据库设计, 中间历经网络服务程序和数据存取组件 (包括对事务处理的控制)。

通读本书既可以学到 C# 的基本知识, 又等于实际参加完成了一个大型数据处理系统的设计、开发和调试全过程。全书浑然一体组成一个有机数据处理系统, 同时各章又可以独立运行, 不受前后次序的影响。所有源程序既可以在 Visual C#.NET 环境下编译运行, 也可以在只有 .NET 软件程序开发包的情况下在命令行状态下去编译运行。读者可以从 Internet 上下载源程序并获得最新的勘误表 (地址: <http://www.csharponline.net> 或北京科海电子出版社网址: <http://www.KHP.com.cn>)。本书中的实例是作者在美国几家大公司中所设计完成的

几个大型项目的浓缩，可以从一个侧面为读者展示目前美国软件生产的基本流程，以供借鉴。

本书为有一定编程经验的计算机软件开发人员提供了一个快速开发 C#程序的捷径，适合有一定基础的读者、大专院校师生、企业技术开发人员学习参考，也适合各类培训班学员学习 C#.NET 程序设计技术。

本书在编写过程中得到美国波士顿微软培训中心、Edgewater 公司培训部以及 American Student Assistance 系统部的大力支持和帮助，中国科学出版集团北京科海电子出版社的夏非彼总编及编辑陈跃琴、梁问媛对本书的编排、内容提出了大量宝贵的意见和建议，国家统计局计算中心的黄德会和夏毓彦也为本书的出版联系提供了极大的帮助。

作者

2002 年 7 月于美国波士顿

目 录

上篇 C#基础知识

第 1 章 C#概述	1
1.1 .NET 简介	1
1.2 C#简介	2
1.3 本书编排体系	2
1.4 下载与安装.NET SDK	4
1.5 本书源程序下载及勘误表	5
第 2 章 编译运行一个 C#简单程序	6
2.1 设置 C#环境参数	6
2.2 设置从浏览器到命令行状态的快捷转换	7
2.3 在命令行状态下用 SDK 编译运行一个 C#程序	7
2.4 C#命令行编译器 csc 的常用格式	10
2.5 在 Visual C#.NET 集成开发环境 (IDE) 中编译运行一个 C#程序	11
第 3 章 C#数据类型	14
3.1 C#值类型	14
3.1.1 简单数据类型	14
3.1.2 结构类型	19
3.1.3 枚举类型	21
3.2 引用数据类型	22
3.2.1 字符串类型 (string)	25
3.2.2 数组数据类型 (array)	45
3.3 装箱和拆箱	60
第 4 章 C#类	62
4.1 类成员变量	62
4.2 C#构造函数和析构函数	65
4.3 类继承	68
4.4 成员函数	70
4.4.1 成员函数级别	70
4.4.2 成员函数参数	73

4.4.3 函数重载 (overloading)	76
4.4.4 虚拟函数改写	78
4.4.5 虚拟函数隐藏	80
4.5 类成员属性函数	83
4.6 类索引器	84
4.7 事件函数	86
4.8 抽象类和密封类	96
4.9 访问权限	98
4.10 类成员修饰符	100
第 5 章 C#流程控制语句	104
5.1 条件执行语句	104
5.2 选择执行语句	110
5.3 循环语句	112
5.3.1 for 循环	112
5.3.2 foreach 循环	114
5.3.3 while 循环	115
5.3.4 do...while 循环	116
5.4 转移语句	117
5.4.1 goto 语句	118
5.4.2 break/continue 语句	123
第 6 章 异常处理	124
6.1 捕获一个异常	125
6.2 捕获所有异常	126
6.3 finally 语句	127
6.4 主动触发一个异常	132
第 7 章 C#输入输出操作	136
7.1 字符串文件的输入输出	136
7.2 二进制文件的输入输出	139
7.3 单字符输入输出	141
7.4 加密读写文件	143
7.5 随机读写文件内容	145
7.6 在操作系统级对文件和目录的操作	148
7.6.1 创建/删除目录	148
7.6.2 创建/删除文件	150
7.6.3 显示目录	151
7.7 内存读写	153
7.8 网络端口读写	155

第 8 章 多线程处理	159
8.1 创建一个线程	159
8.2 为线程设置名字	161
8.3 Sleep 函数	164
8.4 设置线程优先级	165
8.5 线程的后端与前端运行	167
8.6 终止一个线程	169
8.7 线程的暂停与再运行	172
8.8 线程 Join 函数	174
8.9 线程异常	176
8.10 线程加锁	179
第 9 章 与未操控语句交互运作	187
9.1 C#中调用 COM 组件	187
9.1.1 将 COM 类库转换为 .NET 组件	187
9.1.2 在 C#中具体使用一个 COM 组件	191
9.1.3 编译运行一个调用 COM 组件的 C#程序	192
9.1.4 安装一个使用 COM 组件的 .NET 程序	195
9.2 在 .NET 框架外调用 .NET 组件	195
9.2.1 为非 .NET 程序创建一个 .NET 组件	195
9.2.2 为 COM 客户程序包装一个 .NET 组件	197
9.2.3 注册一个 .NET 组件为 COM 组件	198
9.2.4 在 COM 程序中具体使用一个 .NET 组件	198
9.2.5 安装一个使用 .NET 组件的 COM 客户程序	200
9.3 在 C#中使用未受操控的语句	200
9.3.1 包装一个 DLL 函数	200
9.3.2 调用一个 DLL 库中的函数	201
9.3.3 从 C#中传送一个复杂类型的变量给 DLL 函数	202
9.4 在 C#中使用指针	207
9.4.1 在 unsafe 语句中使用指针	208
9.4.2 在 fixed 语句中锁定一个 C#变量	210

下篇 高级编程实例篇

第 10 章 用 C#编写 Windows 用户界面程序 WindowsGUI	212
10.1 创建 WindowsGUI 项目文件	212
10.2 用 DataSet 和 DataTable 在内存中构造一个数据库	214
10.3 用 DataGridTableStyle 控制 DataGrid 的输出格式	217

10.4 增加、删除和修改 DataGrid 中的记录.....	219
10.5 恢复 DataGrid 中修改前的数据.....	220
10.6 刷新 DataGrid 中的数据.....	221
10.7 保存 DataGrid 的当前数据到数据库中.....	221
10.8 调试 WindowsGUI 程序.....	222
10.9 WindowsGUI 源程序清单及 DOS 提示符下的编译方法.....	226
10.9.1 Windows GUI 源程序清单.....	226
10.9.2 DOS 提示符下的编译方法.....	234
第 11 章 用 C#编写组件程序 MyDAO	235
11.1 .NET 组件简介.....	236
11.2 XML 简介.....	236
11.3 创建.NET 组件 MyDAO.....	237
11.3.1 GetList 函数.....	238
11.3.2 Add 函数.....	241
11.3.3 Update 函数.....	242
11.3.4 Delete 函数.....	244
11.3.5 Execute 函数.....	244
11.4 从 WindowsGUI 客户端程序去调用.NET 组件 MyDAO.....	249
11.4.1 在 LoadData 函数中调用.NET 组件 MyDAO.....	249
11.4.2 在 SaveData 函数中调用.NET 组件 MyDAO.....	253
11.5 调试.NET 组件 MyDAO 程序.....	255
11.6 MyDAO 源程序清单及 DOS 提示符下的编译方法.....	255
11.6.1 MyDAO 源程序清单.....	255
11.6.2 DOS 提示符下的编译方法.....	259
第 12 章 网络服务 (Web Service)	261
12.1 网络服务简介.....	261
12.2 一个简单的网络服务 HelloService.....	262
12.3 从 IIS 上去调用/测试网络服务 HelloService.....	265
12.4 创建网络服务 ProductService 并从中调用我们的.NET 组件 MyDAO.....	267
12.5 从 Windows 应用程序 WindowsGUI 去调用网络服务 ProductService.....	270
12.5.1 在 LoadData 函数中调用网络服务 ProductService.....	271
12.5.2 在 SaveData()函数中调用网络服务 ProductService.....	272
12.6 调试网络服务程序.....	272
12.7 网络服务 (MyService) 源程序清单及 DOS 提示符下的编译方法.....	274
第 13 章 在 C#中调用数据库.....	276
13.1 准备一个简单的数据库.....	276
13.2 从 MyDAO 组件去查询数据.....	278

13.3 从 MyDAO 组件去增加/删除和修改数据	281
13.4 .NET 组件 MyDAO 源程序清单及 DOS 提示符下的编译方法	283
13.4.1 .NET 组件 MyDAO 源程序清单	283
13.4.2 DOS 提示符下的编译方法	298
第 14 章 将.NET 组件转换为事务处理 COM+组件	299
14.1 事务处理控制简介	299
14.2 对 Product.cs 的修改	300
14.3 将 MyDAO.dll 注册为一个 COM+ 程序	301
14.4 将 MyDAO.dll 安装到 GAC 中	302
14.5 实地测试新版 MyDAO.dll 对事务处理的控制	302
14.6 COM+ 组件 MyDAO 源程序清单及 DOS 提示符下的编译方法	303
14.6.1 COM+组件 MyDAO 源程序清单	303
14.6.2 DOS 提示符下的编译方法	307
第 15 章 用 C#编写网站界面程序 WebGUI	309
15.1 ASP.NET 简介	309
15.2 创建 WebGUI 项目文件	309
15.3 设置 WebGUI 界面	310
15.4 定义类变量	311
15.5 设置 DataGrid 的初始状态	311
15.6 LoadData()函数	313
15.7 绑定 DataGrid 与 DataSet	315
15.8 在 PageLoad()函数中调用 InitControls()、LoadData()和 BindGrid()函数	316
15.9 编辑/删除一行数据	317
15.10 增加一行数据	319
15.11 刷新、恢复和保存数据	320
15.12 调试 WebGUI 程序	321
15.13 WebGUI 源程序清单及 DOS 提示符下的编译方法	321
15.13.1 WebGUI 源程序清单	321
15.13.2 DOS 提示符下的编译方法	327
附录 A C#语法程序示例	328
附录 B C#关键字一览表	335
附录 C C#与 Java	338

上篇 C#基础知识

第1章 C# 概述

1.1 .NET 简介

.NET 是微软刚刚推出的用以快速开发网络应用程序的集成环境。.NET 体系包括.NET 工作运行平台和.NET 产品开发工具。.NET 平台提供了一个运行网络服务程序 (Web Service) 和其他各种应用程序的环境, 包括公用语言运行环境 (CLR, Common Language Runtime) 和内置基类库, 例如 ASP.NET、ADO.NET 等。CLR 是.NET 的核心部分, 因为它提供了.NET 的公用语言运行环境, 任何.NET 程序都必须在此环境下才可以运行。并且从理论上讲, 不管是什么操作系统, 只要安装了这套 CLR, 任何.NET 程序都可以在此操作系统上运行。

.NET 是微软为适应互联网的迅猛发展而推出的一套快速编写 Web 应用程序的系统, 但这并不是说.NET 只能用于 Web 程序的开发, 相反, .NET 也可以用来开发所有现在在 Windows 系统下可以开发的程序, 并且开发速度更快, 产品质量更高。所以准确地说, .NET 是现有 Windows 程序开发系统的一个扩充或者革新。

.NET 平台目前可以运行于 Windows 98 之后的所有微软操作系统上, 包括 Windows 98/Me/2000、Windows NT 和 Windows XP。即将推出的 Windows.NET 顾名思义会内置.NET 作为其基础运行平台。微软同时声称.NET 简化版将可以运行于手机、电视等电子设备上, 并且不久还会支持 Windows CE。.NET 支持的语言包括 Microsoft C#.NET、C++.NET、VB.NET、JScript.NET、COBOL、Eiffel、Perl、Python、Smalltalk 等等。但是如果要用.NET 去开发应用程序, 也就是说, 如果要想安装运行.NET 软件程序开发包 (.NET Framework Software Development Kit, SDK), 则必须是在 Windows NT 4.0 (SP 6a)、Windows 2000 或者 Windows XP 系统下, Windows 95/98/Me 系统下是不能开发.NET 程序的。

.NET 最初在微软内部被称为“下一代视窗服务 (NGWS)”, 最终确定为.NET。微软于 2001 年相继推出了 Beta1、Beta2、RC, 直到 2002 年 3 月才推出了.NET 正式版。本书所有程序都是基于正式版。

1.2 C# 简介

C#是微软全力推出的一门全新语言，用以快速开发各种运行于.NET 平台上的程序。C#是在 C、C++基础上发展起来的，具有简单、现代、类型安全及完全面向对象等特点。由 C#编写的程序经编译后产生一种中间代码，微软称其为可移植、可执行的 PE (Portable & Executable) 代码。这种代码是一种受操控的代码，运行时由 CLR 按照一定的操控规范对其进行编译从而产生出目标机器上的本地码。正如 Java JVM 中大部分模块是用 Java 本身编写的一样，.NET CLR 中的大部分模块是用 C#来编写的，所以作为一种全新的语言，C#在还没有投向市场之前就已经经过了充分的考验。C#继承了 C++的大量语言要素和编程模式，同时又吸收了许多已有语言尤其是 Java 的优点，所以对于一个熟悉 C++或者 Java 的程序员，可以很容易地使用 C#进行程序设计。但是 C#在 C/C++的基础上又作了很大的革新，即便与 Java 相比也是增加了很多内容，例如属性、Delegate、Web Service 等，所以对于这部分程序员来说，尤其需要一本 C#方面的快速应用教程，以引领他们尽快进入 C#的世界。本书接下来便要通过对 C#最常用最核心的部分进行深入浅出的解析，从而带领读者快速开始应用 C#编写程序。

1.3 本书编排体系

本书上篇详细介绍了 C#的基础知识，包括如何编译运行一个 C#简单程序、C#数据类型、C#类定义、C#控制语句、C#异常处理控制、C#输入输出操作、多线程操作和与未操控语句交互运作等。几乎每一小节中都会列举可编译运行的小例子，并且所有例子都可以在命令行状态下编译运行，从而为读者提供了多种练习机会。有经验的读者可以跳过本篇而直接进入 C#高级编程实例篇，但是作者并不建议这样做，这是因为本篇包含了 C#中最基本的内容，例如事件处理、delegate 类型、类成员属性等。所有这些都是 C#的基础，在开始 C#程序编写前是一定要掌握的，最少应该了解其基本含义。所以即使是一个很有经验的读者也应该在开始下一篇前先快速通读本篇。对一个不是很熟悉编程的读者，通读本篇则会从 C#的高度上开始学习时下最先进的编程技术，自然会受益匪浅。

本书下篇是向读者展示如何用 C#进行高级编程。下篇一开始先介绍了如何去编写一个 Windows 应用程序，紧接着就进行 .NET 组件的编写。然后转入介绍 Web Service 程序的编写。并从第 10 章介绍的 Windows 应用程序，通过刚刚编写的 Web Service，在互联网上调用第 11 章写好的 .NET 组件。从 .NET 组件，到居于中间部位的 Web Service，到位于最前端的 Windows 用户界面程序之间，所有的数据传输一律采用当今最流行、最具有发展前途的 XML 格式，所以中间还会穿插介绍 XML 语言。然后在第 13 章介绍了如何从 .NET 组件去真正连接数据库并实现了对事务处理的控制，从而为大型应用程序的开发提供了充分的环境。最后介绍了如何用 C#编写一个网站程序，并把前边已经成功编译运行的 Windows

用户界面程序 WindowsGUI 轻松修改后移植到网页中变为一个动态网站程序 WebGUI。各章节既有机地融合在一起，组成一套可借鉴、可利用的大型应用程序蓝本；又可以独立运行，保证了读者在学习时的连贯性和分步性。下篇整个系统结构可以用 FastCSharp 整体结构图（图 1-1）来表示。

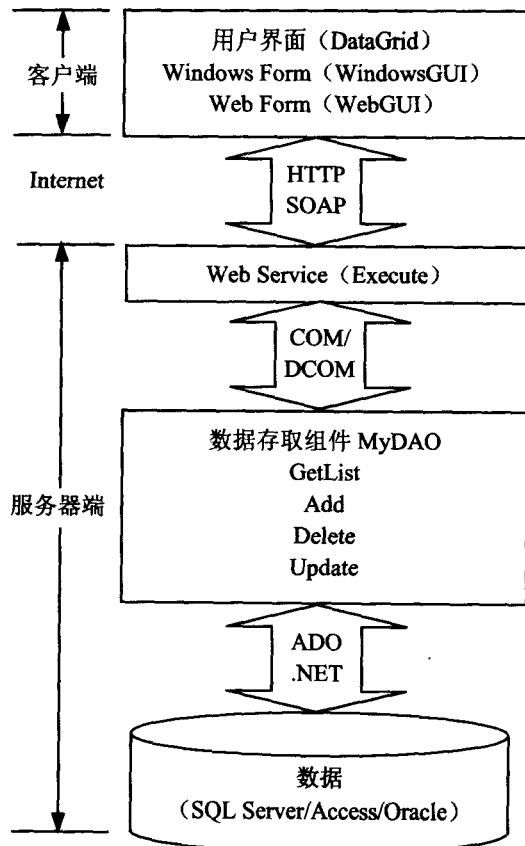


图 1-1 FastCSharp 整体结构图

从图 1-1 可以看出，客户端程序事实上通过互联网与服务器端紧密地结合在了一起。因为中间数据传输协议是 HTTP/SOAP，从而巧妙地绕过了服务器端的公司防火墙，而这正是 Web Service 的精妙之处。当然在开发初期所有这些程序完全可以运行于一台机器上，直到最后，项目完成时才安装到不同的机器上，从而便于开发工作。本篇将分步介绍这套系统的开发过程，使读者意识到用 C# 来开发这么一套大型软件系统并不难。通读本篇后，读者会很快领会 C# 的关键开发要诀，并且将本书实例稍微改装后即可开发出适合自己需要的程序。

1.4 下载与安装.NET SDK

要想自己开发.NET 程序，首先要确认是在 Windows NT 4.0、Windows 2000 或者 Windows XP 系统下，然后再去下载安装.NET 软件程序开发包（.NET SDK）。这套软件程序开发包大约 134 兆。不过如果只是要运行.NET 程序，那么就只需要安装.NET 公用语言运行环境，因为所有在.NET 上开发的程序都要通过这个环境才能运行。可以到 MSDN 网站上去下载.NET 软件程序开发包或者公用语言运行环境，其地址是：<http://msdn.microsoft.com/netframework/downloads/howtoget.asp>。

有了.NET SDK 后即可开始用 C# 去开发程序了，但是只可以在命令行状态下编译运行 C# 程序。具体地讲，就是先用 Notepad 或者其他任何一种 Windows 编辑器编写 C# 源程序，之后用 csc 命令去编译此源文件，再在.NET 下去运行该编译过的程序。不过要想快速开发 C# 程序，还是要在 Visual C#.NET 环境下，因为 Visual C#.NET 提供了一个 C# 集成开发环境，在这个环境下可以创建项目文件（组），编写、运行并调试程序，大大加快了开发速度。MSDN 订户可以到 MSDN 网站（<http://msdn.microsoft.com/vstudio/>）上直接下载 Visual Studio.NET，其中包括 C#.NET、C++.NET、VB.NET、JScript.NET 等。一般用户可以直接联系微软或者其代理商。下载的专业版文件大约 1.62 GB，解压缩后大约 2.16 GB。需要说明的是 Visual Studio.NET 中包含了.NET SDK，所以记住不要重复下载。

除了 Visual C#.NET 以外，市场上另外还有一些 C#.NET 集成开发环境，像 Antechinus C# Editor（<http://www.c-point.com/csharp.htm>）、SharpDevelop（<http://www.icsharpcode.net>，免费）、UltraEdit（<http://www.ultraedit.com/>，共享软件）等等。如果没有 Visual C#.NET，不妨试一试这几个工具。

安装.NET 很容易，只要按照提示操作就是了。惟一要强调的是，如果先前安装了.NET Beta 版或者.NET RC 版，一定要先删除旧版，再安装新版，并且记住在安装.NET 前一定要先安装 Windows Update（包括在 Visual Studio.NET 中，是一个与 Visual Studio.NET 并列的选项），否则会对以后的开发造成很大的困扰。从.NET 平台到 Visual Studio.NET 的整个安装过程在一台典型的奔腾 II Windows 2000 下要花数个小时，所以一定要有耐心，中间不要运行 Windows 应用程序，尤其是 Internet Explorer。最好是在下午要下班时或者晚上开始安装，这样整个安装过程不会受到干扰。.NET Beta/RC 版会提示安装过程剩余 1 小时（Time Remaining: 1 hour），.NET 正式版则提示剩余 40 分钟，不过千万不要相信它，实际上这个“剩余 1 小时/40 分钟”的提示会持续显示数小时之久。另外.NET 开发环境与 Visual Studio 6.0 可以共存，没有任何冲突，所以在安装.NET 前无需备份任何以前的 Visual Studio 环境文件。

1.5 本书源程序下载及勘误表

本书所有源程序读者也可以从网上直接下载，地址是 <http://www.csharponline.net>。除了源程序外，读者也可以从此网站上发现最新的勘误表及作者联系方法。

第 2 章 编译运行一个 C#简单程序

有两种方法可以编译运行一个 C#应用程序，即在命令行状态（Command Line）下用 SDK 编译运行 C#程序和 Visual C#.NET 集成开发环境（IDE）中编译运行 C#程序。下面我们便来用这两种方法分别编译运行一个最简单的 C#程序，也就是几乎学习任何一种语言必讲的 HelloWorld Console 程序。

2.1 设置 C#环境参数

要在命令行状态下编译运行一个 C#程序，必须设置其系统环境参数，包括 Path、Lib、Include 等。所有这些参数的值可能如下所示：

```
Set Path=d:\Program Files\Microsoft Visual Studio.NET\FrameworkSDK\Bin\;  
C:\WINNT\Microsoft.NET\Framework\v1.0.3705\;d:\Program Files\Microsoft Visual  
Studio.NET\Vc7\bin\;d:\Program Files\Microsoft Visual Studio.NET\Common7\IDE\  
%PATH%  
Set LIB=d:\Program Files\Microsoft Visual Studio .NET\Vc7\lib\;"d:\Program  
Files\Microsoft Visual Studio .NET\FrameworkSDK\Lib\";%LIB%  
Set INCLUDE=d:\Program Files\Microsoft Visual Studio .NET\Vc7\include\  
"d:\Program Files\Microsoft Visual Studio .NET\FrameworkSDK\include\  
%INCLUDE%  
Set NetSamplePath=d:\PROGRA~1\MICROS~1.NET\FRAMEW~1\Samples\
```

所有这些参数设置实际上都存放在一个文件中，这就是 X:\Program Files\Microsoft Visual Studio .NET\frameworksdk\bin\covars.bat，其中 X 代表 Microsoft Visual Studio .NET 的安装盘号。

安装 SDK 时，系统会自动设置所有这些环境参数，以保证在命令行状态下可以编译运行一个 C#程序。

注意：如果 .NET SDK 是在安装 Visual Studio.NET 时由 Visual Studio 安装程序安装的，那么所有这些参数是不会被自动加到系统参数中的，要想在命令行状态下编译一个 C#程序，每次进入命令行状态后都要先运行这个批处理文件。或者可以将这些参数设置到系统参数中，也就是在 Windows2000: || Start | Settings | Control Panel | System | Advanced | Environment Variables 上加上这些参数。注意，要加到系统参数（system variables）而不是用户参数（user variables）中。当然，如前所述，如果 .NET SDK 是独立安装而不是与 Visual Studio.NET 套装在一起，便没有