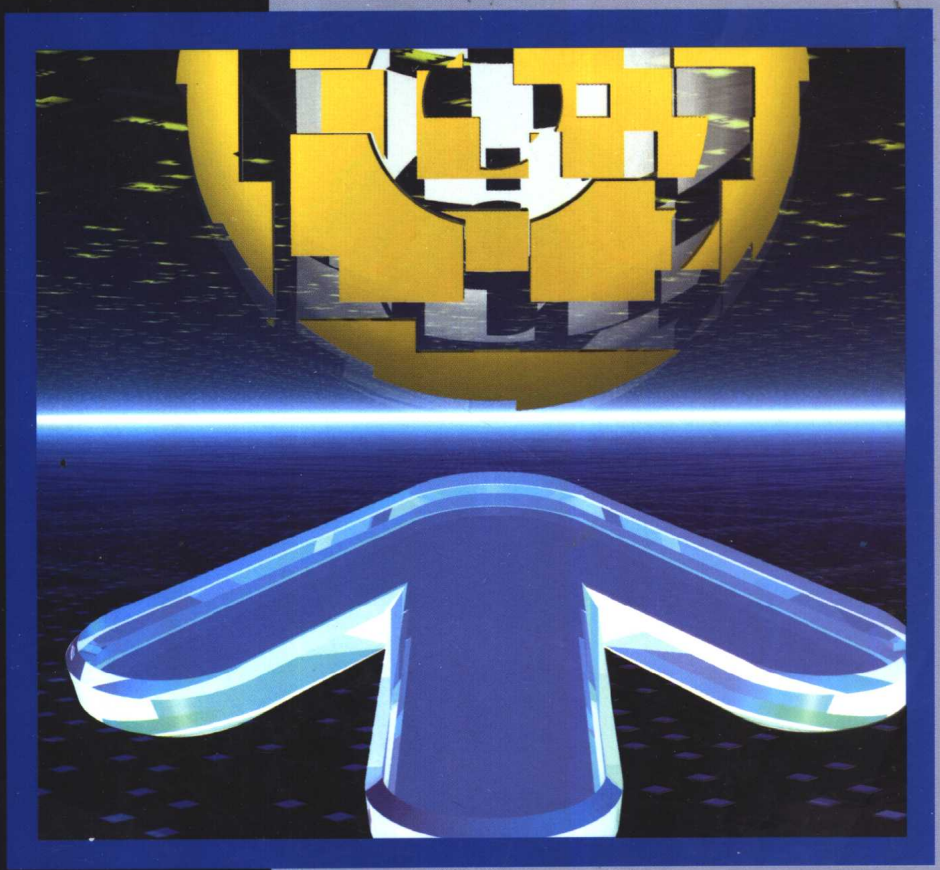
C# Network Programming



C#网络应用编程

C#简化了网络编程技术。如果把它与本书中的指导相结合，你会发现编写网络应用程序比过去任何时候都更加容易和快捷。

[美] Richard Blum 著

高春蓉 谷 宇 阎 隽 等译

李双庆 审校

SYBEX

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

C# Network Programming

C#网络应用编程

〔美〕 Richard Blum 著

高春蓉 谷 宇 阎 隽 等译

李双庆 审校

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 提 要

本书是一本全面介绍C#编程语言的书。书中详细介绍了各种C#网络类和方法，用大量的范例程序演示这些类和方法在程序中的运用，帮助读者在自己的实际工作中编写出更具专业水准的网络程序。作者采用对比的方法，分析了多种编程技术的优缺点，使读者更能体会到C#语言给编程者带来的方便和快捷。

本书特别适合对C#语言网络编程感兴趣的程序员们阅读，已经熟悉其他语言（例如C、C++或者Java）的读者，阅读本书后会感到用C#编写网络程序多么容易。没有编程经验的读者也会从本书受到启发，因为作者对网络编程基本方法和思路的叙述，会使初学者更快地成为一名C#语言的行家。

Copyright©2003 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国SYBEX公司出版，SYBEX公司已将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可，不得以任何形式和手段复制或抄袭本书内容。

版权贸易合同登记号：01-2002-6041

图书在版编目（CIP）数据

C#网络应用编程/（美）布莱姆（Blum，R.）著；高春蓉等译.—北京：电子工业出版社，2003.5
书名原文：C# Network Programming
ISBN 7-5053-8617-4

Ⅰ.C… Ⅱ.①布… ②高… Ⅲ.C语言—程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字（2003）第023159号

责任编辑：陈 宇
印 刷：北京天竺颖华印刷厂
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
北京市海淀区翠微东里甲2号 邮编：100036
经 销：各地新华书店
开 本：787×1092 1/16 印张：33.75 字数：860千字
版 次：2003年5月第1版 2003年5月第1次印刷
定 价：56.00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

谨以此书献给修女Marie Imelda，C.S.C，她曾在20世纪70年代后期竭尽全力、孜孜不倦地教会一群愚笨的高中生编写计算机程序。谢谢，Marie修女。“不要依赖自己的理解，而要用你整个的心灵相信我主，用自己的方式感谢主的恩赐，他将使你前面的道路变得笔直而平坦。”

圣经.旧约 3：5-6 (NIV)

致　谢

首先，所有的荣誉、辉煌和赞颂都属于上帝，他使一切事情都成为可能，他给了我们永恒的生命。

我要感谢Sybex公司所有优秀员工的大力帮助，感谢他们细心的指导和精深的专业技巧。感谢采集编辑Denise Santoro Lincoln给我提供了编写本书的机会。还要感谢开发编辑Carol Henry对本书的指导工作，她的帮助使书中的许多段落更有意义。

非常感谢拷贝编辑Sally Engelfried，他把我糟糕的叙述变成了完善的句子。也要感谢技术编辑Dan Frumin，他认真负责地纠正了书中的程序错误并指出我技术上的问题。还要对Waterside Productions的Carole McClendon表示谢意，她为我安排了本书的出版。

最后，要感谢我的父母，Mike Blum和Joyce Blum，对我写书的全力支持。感谢我的妻子Barbara，女儿Katie Jane和Jessica，感谢她们对我的信任、爱和理解，特别是在我写这本书的时候。

译 者 的 话

随着计算机技术的发展，特别是因特网的推广应用，网络技术已深入到我们日常生活和工作的各个方面。网络编程，这个过去只有少数高级程序员才能做到的事情，现在已经由越来越多的普通计算机用户来完成了。早期是在UNIX环境下用C语言编写网络程序的，后来是用Java语言编写的。用这些语言编写网络程序难度大、费时多，程序的运行速度慢。C#语言大大地简化了网络编程技术，通过TCP进行套接字的连接，通过UDP进行“无连接”的连接，通过使用异步套接字技术、多线程和多点传送技术，C#语言使过去困难的网络编程工作现在可以轻松容易地完成。

本书的作者Richard Blum在美国国防部做了14年网络和系统管理员的工作，他丰富的网络管理经验和编程实践给编写本书打下了坚实的基础。书中大量的程序实例使读者可以学到更多网络编程的实用知识。本书语言简炼、层次分明，全面系统地介绍了C#语言编程技术。

本书的第1章~第4章由李双庆翻译，第5章~第7章由高春蓉翻译，第8章~第12章由谷宇翻译，第13章~第17章由阎隽翻译，全书由李双庆审校。在本书的翻译过程中，刘文林、张亚宁、刘贡生、王炎、李苏云、许柏庆、马文清、彭苏鲁、卢刚、王向荣、刘杰等也参加了部分翻译、校对工作，在此对他们的支持和帮助表示诚挚的感谢。书中翻译的错误和不当之处，欢迎读者批评指正。

此为试读，需要完整PDF请访问：www.ertongbook.com

简　介

在过去的20年中，网络（和网络编程）已经走过了一段很长的路程。在早期（20世纪80年代）的网络计算机技术中，网络编程是高级程序员的事情，他一般（大多数情况下）是在UNIX环境中用C程序设计语言编写应用程序。可是现在，从大型公司到家庭用户，网络无处不在。这么多的计算机通过网络连接在一起，网络组件应用程序是必不可少的。现有的应用程序必须结合网络功能才能在市场中拥有竞争力，必须要给应用程序加上网络通信能力。从孩子们的游戏程序到先进的公司数据库系统，处处都要用到网络程序。

网络编程技术一直是微软Windows操作系统的重要特点。不过很遗憾，要想利用Windows中网络编程的特点必须要了解高级的C或C++编程的概念。可是现在，.NET框架语言简化了把网络特性加到应用程序中的工作。.NET库提供了许多可以集成网络编程技术的网络类。

我作为一名网络管理员，曾经用C和Java语言为Windows和UNIX平台编写过许多网络程序。当前，网络管理和安全性方面的要求使得与网络设备通信和跟踪网络上的工作站变得非常重要。在C套接字API结构中（特别是在WinSock中）工作时，要想快速地写出网络代码是很困难的。由于Java应用程序处理的速度太慢且不受Windows支持，运行Java应用程序通常是很痛苦的。

C#语言解决了许多网络编程的难题，因为我们可以用C#类迅速地生成和开发出网络应用程序。将编写图形代码的C# Forms库与编写网络代码的C# Socket库相结合，使得创建专业水平的网络应用程序变得十分简单。有了C#网络类，过去需要一天时间编写的程序现在最多只需要一个小时就完成了。

本书适合哪些读者

很显然，本书对于那些对网络编程感兴趣的C#程序员是特别有用的，书中详细介绍了各种网络C#类，并加上了许多范例程序，帮助读者在自己的工作中实现这些类。如果读者过去从来没有写过程序，我也在书中叙述了网络编程最基本的概念和思路，还说明了在网络设备之间传送数据的最常用的方法。

读者也许已经熟悉用其他语言（例如C、C++或者Java）编写网络程序，在这种情况下，你会高兴地看到用C#语言编写网络程序是多么容易。

如果读者对C#语言感到陌生，本书在第1章中讲述了创建和编译C#程序的基本方法。你也许想跳过本书第一部分的其他各章，这些章节介绍了网络编程的基本方法，深入探讨了各种特殊的C#网络编程类。

本书内容是怎样安排的

本书分为三个部分，涉及到网络编程的各个方面。

第一部分：网络编程基础

前四章是为那些刚刚开始学习网络编程的程序员们写的，他们可能希望了解网络编程技术发展的历史背景，想知道网络编程需要了解哪些信息。

第1章“C#语言”给初学C#语言的读者提供了一些C#的基础知识，例如，开发工作使用哪种C#包和怎样编译C#程序等。

第2章“IP编程基础”介绍了网络编程技术是怎样通过WinSock接口从UNIX领域发展到Windows领域的，并说明.NET是怎样跨网络资源利用WinSock接口的。

第3章“C#网络编程的类”简要介绍了整个C#网络库，并展示了类的几种基本格式。

第4章“DNS和C#”是介绍性的一章，告诉网络新手们DNS是怎样解决主机地址问题的，并介绍怎样使用C# DNS类。

第二部分：网络层编程

第二部分的几章是本书的核心，讲述了网络编程技术的重要课题。每一章都讨论一个创建C#网络程序的主要课题。

第5章“面向连接的套接字”一开始讨论了利用TCP的流编程技术，除了讨论流编程技术中使用的标准C# Socket类之外，还讨论了存在的共同缺陷，以帮助读者建立可以在实际网络中工作的流程序。

第6章“无连接套接字”讨论了怎样用Socket类建立UDP应用程序。这一章除了告诉读者怎样建立UDP应用程序之外，还讨论了UDP程序设计方法的缺点和不足，并用创建的应用程序范例说明在真实网络环境中可能遇到的问题。

第7章“C#套接字的助手类”讨论了C# TcpClient、TcpListener和UdpClient类，这些都是.NET中特殊的类，可以帮助程序员用最小的努力创建网络程序。这一章还讨论了通过网络发送不同类型数据的基本方法。

第8章“异步套接字编程”讨论了在网络编程领域中使用的异步编程技术（在Windows程序中流行的）。

第9章“使用线程”提供了在网络程序中使用多线程应用程序技术所需要的信息。该技术一般用于必须同时为多个客户机提供服务的服务器应用程序。

第10章“IP组播”描述了怎样用广播和组播的方法把信息包发送到多个客户机，以便削减网络带宽。

第三部分：应用层编程实例

本书的最后一部分描述了一些特殊的网络应用程序，并说明怎样用C#网络类实现这些应用程序。

第11章“ICMP”说明怎样用C# raw套接字实现特定协议的应用程序。在这一章中，向读者展示了在C#网络程序上下文中普通ping和traceroute程序的作用。

第12章“SNMP”介绍了怎样用C#写网络管理应用程序。SNMP（简单网络管理协议）允许用户与网络上的许多设备通信，以便搜寻和获取网络统计数据。在这一章列举了一些读取销售商的MIB（管理数据库）表格的范例，可以创建一个C#应用程序，从网络设备获得

MIB数据。

第13章“SMTP”介绍了C#电子邮件类，列举了用这些类把使用SMTP（简单邮件传输协议）的邮件传送到远程邮件服务器的范例。本章还举了一个用其他邮件协议（例如POP3）的例子。

第14章“HTTP”介绍了C#的web类，讲述了怎样用这些类创建具有Web功能的C#应用程序。还介绍了.NET的网络服务和讨论了怎样用网络服务把用户的应用程序宿主在一个IIS服务器上。

第15章“活动目录”介绍了与微软活动目录服务器有关的各种C#类。举了一些例子说明怎样在活动目录中查询、修改、添加和删除条目。

第16章“远程技术”讨论了.NET的远程概念，它允许应用程序与网络上的客户机共享方法。本章举例演示了创建远程服务器和客户机的方法。

第17章“网络安全”描述了.NET框架是怎样处理网络安全问题的，介绍怎样在应用程序中使用加密技术保护网络数据的安全。

不断更新软件版本

本书所有的范例都是用.NET框架1.0版软件包创建和编译的。每个例子用任何一个微软Visual Studio软件包（包括Visual C#）都很容易进行编译。

编写本书的时候，.NET框架的版本是1.0版和Service Pack 1版。微软公司.NET框架Web站点的网址是http://www.microsoft.com/netframework，读者可以在该网站找到各种关于.NET的消息和通告。

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第一部分　网络编程基础

第1章　C#语言

- .NET基础
- 安装C#开发环境
- C#运行时环境
- C#编程基础
- C#的特点

在很短的时间中，微软的.NET技术就已经迅速成为一种开发微软Windows工作站和服务器应用程序的流行编程平台。虽然大多数媒体的注意力都集中在.NET网络应用能力方面，但它还是有许多对于Windows编程者非常有用的特点。

专门用于开发.NET的新的C#编程语言就是这些特点中的一个。C#正在成为一种编程平台，成为需要网件和Windows系统独立应用程序的程序设计师们广泛使用的编程语言。这种语言提供了许多资源，可以帮助建立健壮的基于Windows的应用程序。为了获得这些资源的优势，现在许多程序设计师正在转移到C#语言上来。

在学习C#网络编程的基础知识之前，读者首先应当了解C#的编程环境、.NET的基础以及怎样创建和分配C#应用程序。本章将向读者介绍怎样在自己的系统上建立C#的开发环境，并介绍怎样保证用户建立的C#程序能够在其他的Windows工作站和服务器上运行。本章的最后将简要介绍C#语言，并提出一些与网络编程有关的C#程序设计课题。本章介绍的所有概念将帮助读者做好网络编程的一切准备。

.NET基础

.NET编程语言组与以前版本的Windows编程语言的区别是，在Windows系统中建立程序和运行程序的方法不同。如果读者还不熟悉C#程序操作方法的话，本节将简要叙述为了在.NET技术的基础上展开应用程序所应当了解的基础知识。

公用语言运行时

微软.NET技术的核心是公用语言运行时（Common Language Runtime，CLR）环境。

该环境使编程者能够用多种编程语言创建程序，并在任何一个支持CLR的平台上运行这些程序。CLR的中心思想就是提供一个多种应用程序接口（Application Program Interface，API）的中间层，该中间层作用于低级的Windows Win32 API功能和应用程序代码之间。用提供一个公用的中间层的方法，微软使大量的应用程序语言有权使用核心的Windows技术（例如网络支持）。

应用程序在CLR环境中的运行方案如图1.1所示。用各种.NET语言（例如Visual Basic .NET、Visual C++.NET、Visual J#.NET，当然还有Visual C#.NET）编写的高级应用程序被编译成一种特殊的叫做微软中间语言（Microsoft Intermediate Language，MSIL）的特殊语言，当程序在主机操作系统上运行的时候，CLR再把MSIL代码翻译成普通的可执行程序。当然，不使用CLR的传统程序仍然可以像以前一样直接访问低级的Windows Win32 API。

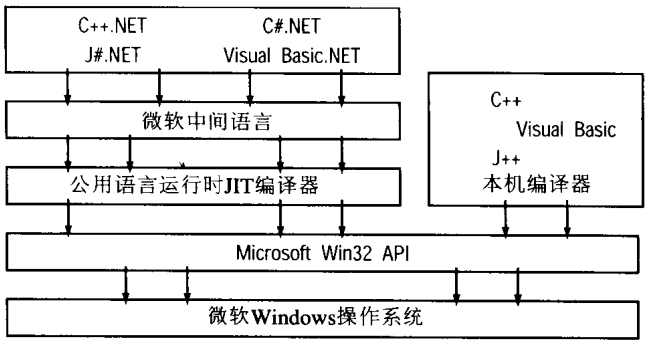


图1.1 公用语言运行时（CLR）环境

这种CLR模式也适合于其他操作系统。由于CLR被移植到其他操作系统，所以.NET程序不需要重新编译就可以在新的主机系统中运行。当前，微软支持共享源公用语言接口（CLI）方案（别名为Rotor），该方案把CLR环境转换到FreeBSD操作系统。Rotor（转子）方案也将扩展到其他操作系统，在编写本书的时候，读者可以在下列Web站点获得更多有关Rotor的信息：

```
http://msdn.microsoft.com/downloads/default.asp?URL=/downloads/sample.asp?url=
/msdn-files/027/001/901/msdncompositedoc.xml
```

为了使程序在CLR上运行，程序必须要编译成MSIL格式。.NET C#编译器用来把C#语言的程序转换成在CLR环境中运行的MSIL格式。下一节将介绍MSIL代码。

MSIL代码

当编译C#程序的时候，会产生一个可执行文件。不过，这个可执行文件与读者过去见过的其他Windows编译器所产生的可执行文件不同。它不是一个可直接在Windows中运行的低级的汇编程序，这种可执行文件包括两个组成部分：

- 启动CLR编译器的存根汇编语言程序
- 编译后应用程序的MSIL代码

存根程序启动CLR即时（Just-In-Time，JIT）编译器，JIT编译器则把MSIL程序代码编译成可以在系统中运行的本地Win32代码。与原来的Windows应用程序不同的是，原Windows应用程序直接影响低级的Win32 API系统，而.NET应用程序要依靠.NET框架CLR来运行。如果在没有安装.NET框架的系统中运行.NET应用程序，就会产生如图1.2所示的错误信息。对任何.NET应用程序来说（不论是在Windows工作站上还是在服务器上运行），.NET框架都是至关紧要的，没有它，MSIL代码就无法运行。任何打算运行.NET程序的Windows工作站或服务器都必须安装.NET框架。

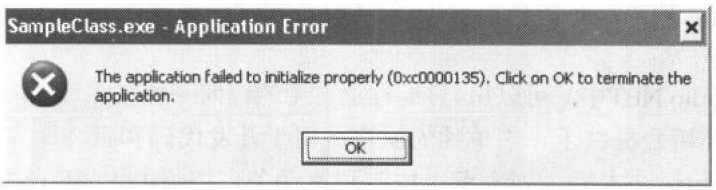


图1.2 没有.NET框架的情况下试图运行.NET应用程序

微软公司保证在它今后所有版本的Windows操作系统中都会装有.NET框架，不过，旧版本的Windows必须手工安装.NET框架。下一节将介绍在一个开发者的环境中安装.NET框架的方法和步骤，以便使用户可以创建、编译和运行.NET应用程序。

安装C#开发环境

在用C#语言编程之前，必须要有一个C#开发环境，即一个用来创建、编译和调试C#程序的系统。很遗憾，对于.NET应用程序开发最苛刻的需求就是指定用于开发的OS平台，当前，.NET要求用以下系统中的一个进行完全的C#程序开发：

- Windows NT 4 Workstation或Server（带有Service Pack 6a）
- Windows 2000 Professional或Server（带有Service Pack 2）
- Windows XP Home Edition或Professional Edition

无法使用以上系统的程序员就不能开发C#程序，但对此也有一个解决的办法，将在本章的“C#运行时环境”一节中进行介绍。

C#开发选项

微软为C#开发者提供了三种开发环境，每种开发环境都有它自己的优缺点，本节将向读者介绍这些C#开发环境和它们之间的区别。

- Visual Studio.NET
- Visual C#.NET
- .NET框架软件开发套件（SDK）

说明： 本书所有的范例程序都可以在任何.NET开发环境中进行编译。为了使叙述简单化，书中显示的是使用.NET框架命令行编译器的范例，这样就保证了不管使用哪种开发环境，任何人都可以使用这些范例程序。

Visual Studio.NET

Visual Studio.NET软件包是Microsoft.NET最重要的开发产品。这个综合开发环境（Integrated Development Environment，IDE）提供了许多功能部件，可帮助用户进行Windows应用程序的程序设计。微软把Visual Studio软件包形容为“一个快速的应用程序开发（RAD）工具，使编程者能快速地编码和调试.NET应用程序”。它有一个完整的图形环境，可以创建Windows的窗体、键入代码和调试程序。除了一个优良的开发环境之外，Visual Studio还支持所有的.NET编程语言——Visual Basic.NET、Visual C++.NET、Visual J#.NET和Visual C#.NET。如果用户想用所有的.NET语言进行程序设计的话，Visual Studio软件包正是值得考虑的对象。

在Visual Studio.NET中，可以用4种编程语言中的任何一种在图形环境中创建应用程序。IDE（综合开发环境）提供了一些单独的窗口，用于开发代码和直观地安排Windows的各种应用程序控件，包括文本框、列表框、按钮和滚动条。Visual Studio.NET提供了一种容易的方法来创建、检测和调试.NET应用程序，不管这些应用程序是独立的Windows应用程序还是ASP.NET Web页面。

根据用户的开发需要（和财务状况），Visual Studio.NET软件包分好几种等级，越高的等级包括的功能越多，当然售价也越高。

Visual C#.NET

如果用户只对用C#语言编程感兴趣，他就不需要购买全部的Visual Studio.NET软件包。微软提供的Visual C#.NET软件包具有与Visual Studio.NET相同的功能，但是只支持C#语言。这对于C#编程者是一种节省得多的方法。Visual C#软件包与Visual Studio一样，也有多种软件包等级，从最基本的学生版本到全功能的专业开发者版本都有。但是在购买这个版本之前一定要搞清楚，这种版本的软件包不具备Visual Studio.NET的某些高级功能，例如自动数据库支持。

不管是Visual Studio还是C#的开发选项，都要求用户购买微软的商业软件开发软件包。这两种软件都是极好的软件开发工具，当用户创建Windows应用程序和调试程序的时候，可以节约用户大量的时间。但是，许多初学编程的和具有某些特殊爱好的编程者可能会感到这些IDE软件的价格太高。

.NET框架SDK

如果想找一种省钱的办法学习C#编程方法的话，可以选择使用.NET框架SDK。SDK是微软公司免费提供的，它包含编译和调试.NET程序的命令行工具，包括C#。这个软件包使得用户可以不用花钱购买昂贵的IDE开发环境就可以获得开发C#应用程序的感受。用户可以通过从微软的.NET框架站点下载完整的软件包来获得.NET框架SDK，或者花一点钱直接从微软购买一张光碟来获得这个软件包（有关Web站点的信息，参见下一节）。

如果用户对于C#还完全是个新手的话，最好先从网上下载.NET框架SDK并试着用它编程。人们经常说，学习编程语言的最好方法就是动手编写程序代码，经历代码的逻辑推理过程——我认为，网络编程尤其如此。随着用户C#编程技巧的不断提高，就可以转移到用Visual

C#.NET软件包创建Windows窗体和调试复杂的应用程序了。为了帮助用户起步，下一节将叙述下载和安装.NET框架SDK的方法。

下载.NET框架SDK

在编写本书的时候，.NET框架SDK的当前版本是第1版，此时可以从微软站点免费下载或者购买一张光碟。如果读者选择从微软下载，有两种方法可供选择。因为SDK很大（131MB），所以既可以把它下载成一块，也可以分为10个小的（13.1MB）软件包来下载，下载之后再组合在一起。两种方法需要下载的数据量是一样的，但是连接速度慢的用户可能想一次下载一小块数据。

当前.NET框架站点的URL是www.microsoft.com/netframework/。因为它是一个Web上的公用站点，所以在读者见到这本书的时候，它的网址已经变了。在这种情况下，读者可以直接进入Microsoft的主页（www.microsoft.com），查找有关.NET的内容即可。

.NET框架站点含有大量关于SDK的信息，其中包括一个到单独的软件下载页面的链接。下载页面上有各种下载该软件的选项。单文件下载是一个名为setup.exe的文件，它可以下载到用户的工作站或服务器上进行安装。如果用户选择多部分下载的选项，那就必须把所有分开的SDK文件和一个setup.bat文件下载到一个临时的目录中。在下载完所有的文件之后，必须运行setup.bat文件，这个文件会从所有的SDK文件建立一个主控setup.exe文件。

不管用两种方法中的哪一种，最终都形成一个setup.exe文件，必须运行这个文件来安装.NET框架SDK软件包。下一节将介绍安装的过程。

安装.NET框架SDK

在获得了setup.exe文件之后，不管是用一次下载的方式还是用分成多部分下载的方式，或者是从光碟上获得的，我们就可以安装.NET框架SDK了。在DOS命令提示下运行setup.exe文件或者在Windows资源管理器中双击它都可以。

在安装开始之后，对话框问用户是否想安装.NET框架SDK。单击Yes按钮之后就启动安装过程。

.NET安装程序首先把操作安装过程的文件放到TEMP环境变量指向的临时目录下，如果用户的磁盘空间比较紧张的话，可以把临时操作文件提取出来放在与系统驱动器（通常指C:\）不同的某个驱动器上。在取出操作文件之后，安装程序会更新工作站的Windows安装器软件包，然后用.NET框架安装程序启动Windows安装器。

在开始屏幕和一个版权协议屏幕之后，程序会问用户想安装哪些SDK软件包的组件，如图1.3所示。

如果用户的磁盘空间紧张，可以取消对SDK samples复选框的选择，不把SDK样本加载到工作站。在Install Options（安装选项）屏幕之后，程序会问是否要安装SDK组件。这只适用于某些组件，例如DLL（动态链接库）和必须安装在系统驱动器（通常是C:\）上的可执行文件。在用户选择了安装目的地之后，安装便开始进行。在安装完成之后就可以编译和运行C#程序了。

此为试读，需要完整PDF请访问：www.ertongbook.com