

权威 全面 实用

Microsoft Visual C# .NET 2003
Unleashed

C# .NET 技术内幕

(美) Kevin Hoffman 著
Lonny Kruger
董梁高猛 译



SAMS

清华大学出版社

C# .NET 技术内幕

(美) Kevin Hoffman 著
Lonny Kruger 译
董梁 高猛 译

清华大学出版社

北 京

Simplified Chinese edition copyright © 2005 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Microsoft Visual C# .NET 2003 Unleashed by Kevin Hoffman, Lonny Kruger, Copyright © 2005

EISBN: 0-672-32676-0

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as **Sams Publishing**.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由培生教育出版集团授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2005-3354

版权所有,翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

C# .NET 技术内幕/(美)霍夫曼(Hoffman, K.), (美)科鲁格(Kruger, L.) 著;董梁,高猛 译.

—北京:清华大学出版社,2006.4

书名原文: Microsoft Visual C# .NET 2003 Unleashed

ISBN 7-302-12492-2

I. V… II. ①霍…②科…③董…④高… III. C 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(2006)第 006653 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客 户 服 务: 010-62776969

组稿编辑: 曹 康

文稿编辑: 于 平

封面设计: 康 博

版式设计: 康 博

印 刷 者: 北京市世界知识印刷厂

装 订 者: 三河市金元印装有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 43.25 字数: 1107 千字

版 次: 2006 年 4 月第 1 版 2006 年 4 月第 1 次印刷

书 号: ISBN 7-302-12492-2/TP · 8010

印 数: 1 ~ 4000

定 价: 78.00 元

前 言

欢迎使用 Visual C# .NET 2003

今天，人们生活在一个忙碌的时代。要花两三个星期才能得到一辆汽车的信贷申请结果的时代已经一去不复返了。现在这个时代是我们想到需要什么信息，就立即想要得到什么信息。

如今想要即时获取信息成为人们的一个共同需求。无论是在汽车上、飞机上、工作时或在家看电视的时候，我们都想即时得到信息。为了解决这种困境，IBM 为世界带来了个人电脑。那时这个发明是很伟大的，但是对于普通大众来说，想拥有一台个人电脑需要一大笔钱，于是其他公司推出了便携式电脑。不要受便携(portable)这个术语的迷惑，这种电脑根本不便于携带。不过便携式电脑最终发展得足够小以便于携带，并且在一段时期内，一切都看起来很完美了。然而，随着时间的推移，又出现了另一种风尚：打开膝上型电脑查找联系人的电话号码或地址。接着出现了移动设备。再随着时间推移，移动设备，如个人数字助理(personal digital assistants, PDA)、手机，以及带 PDA 功能的手机变得更为先进，所有我们想要的信息几乎就是唾手可得了……但还不完全是。

这个时代的伟大之处在于每个人都有他自己的想法。有些想法是伟大的，有些则不是。这就带来了问题，大部分人会偏爱他们自己的想法并认为他们的方法才是正确的。这就使得各个公司以自己的想象开发技术，完成任务，最终使不同制造商生产的相似设备之间的通信很难实现。想象一下：现在得到一个任务，要从不同的股票牌价供应商那里搜集股票价格。第一个来源要求以专用协议查询供应商数据库。第二个来源每分钟都重新生成一个文件，可以下载并解析该文件来获取所需的信息。最后，第三个来源没有一个发布接口，因而要求执行“抓屏(screen scraping)”操作来得到所需的信息。没有一个访问信息的标准方法，在这种情况下，毫无疑问，要找到需要的数据会很复杂。

.NET 的问世

幸运的是，对于这个问题，我们在 Redmond 的朋友考虑了很久，提出了一个称为 .NET 的解决方案。对于很多领域，如跨语言互操作性、系统稳定性的运行时管理、使用标准协议(如 SOAP)的 Web 服务、分布式组件的无缝通信等，Microsoft 都适时提出了一个端到端的解决方案。

编写这本书的首要目标是介绍理解 .NET 技术所必需的信息，让读者能够将这些知识运用到将来的项目中。其次，我们希望您不仅了解这项技术的工作方式，而且能知道为什么以这种方式工作。如今，知道调用函数 A 可以使系统执行某项任务后，很多程序员就满足了。但如果不知道函数 A 如何执行该任务，当在某段上下文中使用函数 A 却中断了函数 B 时，就无法得知其原因。如果不知道函数是如何运作的，也就不知道该如何使代码执行得更快、更有效率、更有扩展性以及更可靠。本书将向您揭示 .NET Framework 的技术内幕，您不仅会学习如何使

用 C# 中的属性，而且还将学习这些属性实际的运作方式。这样当某些程序中断时，您对框架的了解将能帮助您确定问题并快速想出解决方案。

Visual Studio .NET 中的新内容

第一次打开 Visual Studio .NET 的时候，您会注意到它与之前的版本有很多不同之处。Visual Studio .NET 和 .NET Framework 将会改变您查看软件和创建软件的方式。Visual Studio .NET 与其前身，比如 Visual C++ 6.0，之间的区别包括：

- **全新设计的用户界面：**Visual Studio .NET 的用户界面把 Visual C++、Visual Basic 以及 Visual InterDev 各个之前版本中最好的功能组合起来了。
- **高级帮助系统：**帮助完全是唾手可得了。有了新的 Dynamic Help System(动态帮助系统)，与帮助文档的链接识别是上下文的，也就是会根据当前的工作内容来显示帮助文档。同样现在将 MSDN(Microsoft Developers Network，微软开发人员网络)合并到集成开发环境(Integrated Development Environment，IDE)中，而不必作为一个外部的帮助应用程序来使用。
- **极大地改进了调试器：**将多种语言合并到 Visual Studio .NET 中，能够支持跨语言调试。现在，可以很容易地从 Visual Basic .NET 代码切换到 Visual C# 代码。
- **支持部署：**Visual Studio .NET 现在包含了必要的工具组，可以将完成后的应用程序部署到最终目标。使用 Microsoft Installer 技术，可以在 Visual Studio .NET 的 IDE 中创建合并模块或整个安装程序。
- **自动代码文档：**组合使用可扩展标记语言(Extensible Markup Language，XML)标记，您无需借助第三方工具，就可以为代码创建格式完善的文档。
- **.NET：**仅通过一个简短的词组是不足以概括 .NET 这项技术的。不言而喻，.NET 不仅仅是一个架构，在 .NET 这把大伞下包含了很多内容，如 Windows Forms、Web 服务，当然还有新的 Microsoft 编程语言——C#。

Visual Studio .NET 带来的变化是巨大的，而要试着了解所有这些变化也是很困难的。无论如何，在阅读完本书之后，您将能够使用 .NET Framework 的所有功能，并了解这些功能的工作方式。如果您需要创建若干个对象，并且这些对象需要通过进程甚至是通过 Internet 进行通信，您将会运用第 39 章介绍的内容，“Remoting”。如果您需要创建一个对象，并且这个对象会在程序终止后保留自身，您会用到第 7 章介绍的知识点，“文件与流 IO 以及持久对象”。总之本书将使您获益匪浅。

目 录

第 I 部分 Visual Studio .NET IDE 简介

第 1 章 Visual Studio .NET IDE	3
1.1 窗口管理和定制	3
1.1.1 用 Visual Studio .NET 的 “Developer 配置文件” 定制界面	3
1.1.2 配置快捷键	4
1.1.3 过滤 MSDN Help	5
1.1.4 最大化可视区域	5
1.1.5 使用 Visual Studio .NET 2003 的可停靠窗口	5
1.2 Visual Studio .NET 2003 的工具 窗口	8
1.3 处理文档	11
1.3.1 增强文本编辑器的功能	11
1.3.2 找到需要的帮助	15
1.3.3 编辑 HTML	17
1.3.4 编辑 XML	19
1.3.5 编辑二进制文件	20
1.4 使用外部工具	20
1.5 小结	21

第 II 部分 语言基础

第 2 章 C#简介	25
2.1 要再学另外一种编程语言	25
2.2 学习公共类型	26
2.2.1 理解值类型	26
2.2.2 理解引用类型	27
2.2.3 引用类型与值类型的区别	27
2.3 公共语言运行库	28
2.3.1 多种语言, 一个运行库	29
2.3.2 隔离性	29
2.3.3 平台调用	29

2.3.4 代码访问安全性	29
2.3.5 JIT 编译器	30
2.3.6 代码执行	30
2.3.7 COM 互操作	30
2.3.8 Rotor: Microsoft 通用语言基础 结构的共享源代码	30
2.4 垃圾回收理论	30
2.4.1 引用计数	30
2.4.2 generation(代)	31
2.4.3 回收	31
2.4.4 部分回收	31
2.4.5 不确定的终止	31
2.4.6 使用 IDisposable 创建功能 良好的对象	32
2.5 基类库简介	33
2.6 规范的“Hello World”示例	34
2.7 小结	36
第 3 章 表达式和控制结构	37
3.1 表达式和控制结构简介	37
3.2 基本表达式	37
3.3 合法的变量名	37
3.4 使用 C#运算符	38
3.5 前缀/后缀运算符	43
3.6 程序流程控制: 控制结构	43
3.6.1 程序执行路径	43
3.6.2 条件语句	43
3.6.3 if 控制结构	43
3.6.4 if/else 控制结构组合	44
3.6.5 短路计算	44
3.6.6 使用三元运算符	46
3.6.7 switch 语句	47
3.7 循环	49
3.7.1 for 循环	49
3.7.2 使用 for 语句	50
3.7.3 foreach 语句	50
3.7.4 while 循环	50

3.7.5	do..while 循环	52
3.8	小结	53
第 4 章	字符串与正则表达式	54
4.1	字符串基础	54
4.1.1	理解字符串的不变性	54
4.1.2	格式化字符串	54
4.1.3	使用转义序列	58
4.1.4	定位子字符串	59
4.1.5	添加填充	59
4.1.6	截断字符	59
4.1.7	替换字符	60
4.1.8	拆分字符串	60
4.1.9	修改大小写	60
4.1.10	StringBuilder	61
4.1.11	追加字符串	61
4.1.12	使用 AppendFormat	61
4.1.13	插入字符串	61
4.1.14	替换字符串和字符	62
4.1.15	移除子字符串	62
4.2	使用正则表达式	62
4.3	如何使用匹配	62
4.3.1	用正则表达式验证数据	65
4.3.2	匹配分组	66
4.3.3	替换所匹配的字符串	67
4.4	小结	67
第 5 章	数组与集合	68
5.1	使用数组	68
5.1.1	一维数组和多维数组	68
5.1.2	锯齿数组	70
5.1.3	将数组作为参数传递	71
5.2	集合	72
5.2.1	基本集合接口	72
5.2.2	遍历集合	73
5.3	基础类库中的集合	74
5.3.1	使用 ArrayList	74
5.3.2	使用 Stack	75
5.3.3	使用 Hashtable	76
5.3.4	使用 BitArray	77
5.3.5	使用 Queue	78

5.3.6	使用 SortedList	78
5.4	创建自定义集合	79
5.4.1	实现 CollectionBase 类	80
5.4.2	创建 AddressList 集合	80
5.5	小结	81
第 6 章	对象与类	83
6.1	对象与类	83
6.1.1	类的属性	83
6.1.2	面向对象设计与编程中的操作	85
6.1.3	类	86
6.1.4	对象	86
6.1.5	对象状态维护	87
6.2	对象与类的深入讨论	88
6.2.1	继承基类	88
6.2.2	多态性	90
6.3	小结	93
第 7 章	文件与流 I/O 以及对象持久性	94
7.1	文件与流 I/O	94
7.1.1	文件与流 I/O	94
7.1.2	使用流: FileStream、 MemoryStream、StringReader 和 StringWriter	97
7.1.3	使用 FileSystemWatcher	101
7.2	对象持久性	105
7.2.1	串行化对象	105
7.2.2	扩展标准串行化	110
7.3	小结	113
第 8 章	XML 基础	114
8.1	运用 DOM	114
8.1.1	创建 XML 文档	119
8.1.2	运用 XML 节点、元素 和属性	120
8.1.3	保存 DOM	120
8.2	使用 XmlReader 类	121
8.2.1	什么是 XmlReader	121
8.2.2	使用 XmlTextReader	121
8.2.3	使用 XmlValidatingReader	123
8.2.4	使用 XmlNodeReader	123

8.2.5	使用 XmlConvert	123
8.3	XPath	124
8.3.1	语法学习	125
8.3.2	节点筛选	126
8.4	XPathDocument	127
8.4.1	XSLT	129
8.4.2	转换 XML 文档	130
8.5	串行化 XML 样式	132
8.5.1	使用基本的 XML 串行化	132
8.5.2	自定义 XML 串行化	134
8.6	小结	135
第 9 章	多线程编程	137
9.1	线程基础	137
9.1.1	主要的线程属性和线程方法	137
9.1.2	ThreadStart 委托	139
9.1.3	创建线程	139
9.1.4	运行线程	139
9.1.5	终止线程	140
9.1.6	挂起线程	142
9.1.7	通过“休眠”来暂停线程	142
9.1.8	联接线程	142
9.2	同步	143
9.3	使用 lock 关键字	144
9.3.1	引发线程事件	144
9.3.2	使用 Mutex 类	146
9.3.3	使用 Monitor 类	147
9.3.4	变量保护(Interlocked 递增/递减)	151
9.3.5	无等待读取(Reader WriterLock)	151
9.4	对非同步编程使用线程池	154
9.4.1	WaitCallback 委托	154
9.4.2	将工作项排入队列	154
9.4.3	向线程传递数据	155
9.5	小结	156
第 10 章	事件与委托	157
10.1	委托的定义	157
10.1.1	委托的类型	161
10.1.2	委托的内部机制	163

10.1.3	组合委托	164
10.1.4	事件	164
10.2	小结	168
第 11 章	反射与代码属性	170
11.1	使用反射	170
11.1.1	反射	170
11.1.2	反射如何工作	170
11.1.3	在运行时查看类型信息	171
11.2	使用代码属性	175
11.2.1	代码属性	175
11.2.2	使用代码属性	175
11.2.3	创建自定义属性	176
11.2.4	查询自定义属性	178
11.3	小结	180
第 12 章	程序集与 AppDomain	181
12.1	程序集	181
12.2	简介	181
12.3	程序集构造块	182
12.3.1	程序集清单	183
12.3.2	元数据	184
12.3.3	深入程序集——MSIL 代码	184
12.3.4	程序集资源	185
12.4	创建程序集	185
12.5	在程序集中嵌入资源和内容	188
12.6	本地化与附属程序集	190
12.7	AppDomain	193
12.7.1	简介	194
12.7.2	AppDomain 编程	194
12.8	组合使用程序集与 AppDomain 的实例	198
12.9	小结	204
第 13 章	COM 与 Windows 互操作性	205
13.1	使用 .NET 代码与 COM 交互	205
13.1.1	COM Interop	205
13.1.2	运行时可调用包装程序	206
13.1.3	.NET 到 COM 的编组	206

13.1.4	代码互操作性示例: .NET 代码调用 COM 代码	207
13.2	COM 到 .NET	210
13.2.1	COM 可调用包装程序	210
13.2.2	用于 COM Interop 编程的 .NET 代码属性	210
13.2.3	从 COM 将数据编组到 .NET	211
13.2.4	Interop 编程示例: COM 代 码使用 .NET 组件这个组件	211
13.2.5	何时使用 Interop	213
13.3	主 Interop 程序集	213
13.3.1	主 Interop 程序集概述	213
13.3.2	使用 PIA	214
13.3.3	创建并部署 PIA	214
13.4	平台调用(P/Invoke)	215
13.4.1	简介	215
13.4.2	使用非托管 DLL	215
13.4.3	平台调用——数据编组	215
13.4.4	平台调用示例—— Win32 API	217
13.4.5	何时使用平台调用	218
13.5	小结	218
第 14 章	高性能编程	219
14.1	垃圾回收	219
14.1.1	垃圾回收内部机制—— generation(代)	219
14.1.2	思考垃圾回收器的代码 编写	220
14.1.3	非确定性终结与析构	221
14.2	CLR 中内存与类的管理	222
14.2.1	装箱与取消装箱	223
14.2.2	集合与装箱	224
14.2.3	使用 StringBuilder 类	225
14.3	高性能代码: 最优方法	226
14.3.1	使用异常	226
14.3.2	大块 API 调用	227
14.3.3	值类型与引用类型	227
14.3.4	对集合使用 AddRange	228

14.3.5	锯齿数组与矩形数组	228
14.3.6	for 与 foreach	228
14.3.7	利用异步 I/O	230
14.4	小结	230

第 III 部分 Windows 应用程序

第 15 章	Windows Forms	233
15.1	Windows Forms 基础知识	233
15.1.1	Main 方法	233
15.1.2	理解 Forms Designer	234
15.2	Hello World	236
15.2.1	利用 Windows Application Wizard 创建应用程序	236
15.2.2	设置属性	238
15.2.3	编译并运行应用程序	239
15.2.4	响应按钮的单击	240
15.3	小结	243
第 16 章	Windows Forms 用户界面 控件	244
16.1	标准 Windows Forms 用户 界面控件	244
16.1.1	执行控件操作	244
16.1.2	用控件来存储值和更改值	246
16.1.3	利用控件维护列表	249
16.2	用户控件	257
16.3	小结	258
第 17 章	创建优秀的 Windows Forms 应用程序	259
17.1	可视继承	259
17.1.1	使用继承的窗体	259
17.1.2	使用 WinForms 可视继承	260
17.1.3	可视继承的最优方法	265
17.2	创建并使用动态上下文菜单	265
17.2.1	上下文相关的、自适应的 用户界面	265
17.2.2	使用简单的动态上下文 菜单	266
17.3	绘制自定义列表元素	268

17.4	使用 DrawMode 属性	268
17.4.1	创建自定义 ListBox	268
17.4.2	创建自定义菜单项	270
17.5	形状窗体	271
17.5.1	简介	271
17.5.2	创建一个形状窗体示例	272
17.6	小结	273
第 18 章	使用 Web 服务	274
18.1	添加 Web 引用	274
18.1.1	在 Visual Studio .NET 中 添加引用	274
18.1.2	使用 WSDL.EXE	278
18.2	动态 URL 支持	278
18.2.1	在 app.config 中存储 URL	279
18.2.2	在独立存储中保存 URL	279
18.2.3	使用 UDDI 实现动态 URL	282
18.3	异步使用 Web 服务	282
18.3.1	多线程服务使用示例	283
18.3.2	Web 服务客户端可靠性	284
18.3.3	网络连接测试	284
18.3.4	处理 Web 服务错误	285
18.3.5	支持离线操作	285
18.4	小结	285
第 19 章	智能客户端	287
19.1	智能客户端	287
19.1.1	理解智能客户端	287
19.1.2	用集中服务器部署和更新 智能客户端	288
19.1.3	在智能客户端后端支持中 使用 Web 服务	291
19.1.4	根据效率决定是在服务器 端处理, 还是在客户端 处理	294
19.1.5	使用在线和离线功能	294
19.2	小结	294
第 20 章	部署 Windows 应用程序	296
20.1	安装 .NET 应用程序	296

20.1.1	程序集部署	296
20.1.2	将程序集放到全局程序集 缓存中	297
20.1.3	私有安装	301
20.2	Web 安装	302
20.2.1	从 URL 部署	302
20.2.2	部署智能客户端应用程序	303
20.3	MSI/项目安装	305
	创建一个安装项目	306
20.4	小结	310

第IV部分 Web 应用程序

第 21 章	Web Forms 和 ASP.NET	313
21.1	Web Forms Designer	313
21.2	创建 ASP.NET “Hello World” 应用程序	314
21.3	基本事件处理	319
21.4	小结	323
第 22 章	Web UI 控件	324
22.1	服务器控件	324
22.2	小结	335
第 23 章	ASP.NET 中的状态管理	336
23.1	客户端状态管理	336
23.1.1	视图状态	336
23.1.2	使用隐藏表单字段	341
23.1.3	cookie	344
23.1.4	查询字符串	347
23.1.5	在窗体之间传送服务器 控件值	350
23.2	服务器端状态管理	354
23.2.1	应用程序状态	354
23.2.2	会话状态	355
23.3	小结	356
第 24 章	缓存	358
24.1	缓存	358
24.1.1	介绍 ASP.NET 的缓存	358
24.1.2	OutputCache 指令	359

24.1.3	使用 HttpCachePolicy	361
24.1.4	使用 Cache 对象	363
24.2	小结	366
第 25 章	高级 ASP.NET	368
25.1	创建 Web farm 中的 ASP.NET 应用程序	368
25.1.1	Web farm 中的 ViewState	369
25.1.2	Web farm 中的会话状态维护	369
25.1.3	Web farm 中的应用程序状态	371
25.1.4	Web farm 的配置和部署	372
25.1.5	Web farm 的最佳习惯、建议和告诫	372
25.2	ASP.NET 的本地化和全球化	373
25.2.1	使用本地化的资源	373
25.2.2	显示本地化的内容	374
25.2.3	其他的本地化功能	376
25.3	创建自定义的 HttpModule	378
25.3.1	了解 ASP.NET 应用程序事件	378
25.3.2	创建自定义的 HttpModule	378
25.4	创建自定义的 HttpHandler	380
25.5	小结	382
第 26 章	部署 ASP.NET 应用程序	383
26.1	手工部署 ASP.NET 应用程序	383
26.1.1	通过“Copy Project”部署	383
26.1.2	XCopy 部署	384
26.1.3	何时 XCopy	384
26.2	自动化部署	384
26.2.1	创建安装项目	385
26.2.2	部署安装项目	389
26.3	高级 ASP.NET 部署	389
26.3.1	Web farm 考虑事项	389
26.3.2	防火墙、DMZ、路由器和安全约束	390
26.3.3	宿主环境考虑事项	391
26.4	小结	392

第 V 部分 数据访问

第 27 章	使用 .NET 数据提供程序	395
27.1	SQL Server 数据提供程序	395
27.1.1	SQL Server 数据提供程序	395
27.1.2	使用 SqlConnection 类	395
27.1.3	使用 SqlConnection 的数据库连接字符串	396
27.1.4	SqlCommand 类	397
27.1.5	SqlDataReader 类	399
27.1.6	SqlDataAdapter 类	399
27.2	处理 OLEDB 数据提供程序	401
27.2.1	OLEDB 数据提供程序	401
27.2.2	使用 OleDbConnection	401
27.2.3	使用 OleDbCommand	401
27.2.4	使用 OleDbDataReader	402
27.2.5	使用 OleDbDataAdapter	402
27.3	其他数据提供程序	403
27.3.1	Oracle 的 .NET 数据提供程序(ODP.NET)	403
27.3.2	Microsoft 为 Oracle 提供的 .NET 数据提供程序	403
27.3.3	.NET ODBC 数据提供程序	403
27.3.4	MySQL .NET 数据提供程序	403
27.4	DataSet 和 DataAdapter 绑定	404
27.4.1	DataSet	404
27.4.2	将 DataSet 与 DataAdapter 关联	404
27.4.3	示例: 将 DataSet 关联到动态数据源	404
27.5	小结	406
第 28 章	创建自定义的 ADO.NET 数据提供程序	408
28.1	自定义数据提供程序	408
28.1.1	何时创建数据提供程序	408
28.1.2	实现自定义数据提供程序的步骤	409

28.1.3	数据提供程序情景示例	409	29.3.1	在 Visual Studio .NET 中 创建类型化的 DataSet	448
28.1.4	远程数据提供程序概述	410	29.3.2	使用 XSD.EXE 构造类型 化的 DataSet	448
28.2	实现 IDataParameter 和 IDataParameterCollection	411	29.3.3	使用类型化的 DataSet	448
28.2.1	IDataParameter 接口	411	29.3.4	注解类型化的 DataSet	449
28.2.2	IDataParameterCollection 接口	411	29.4	小结	451
28.2.3	RDPPParameter 类	411	第 30 章	Windows Forms 数据绑定	452
28.2.4	RDPPParameterCollection 类	414	30.1	数据绑定	452
28.3	实现自定义 Connection	416	30.1.1	Windows Forms 数据绑定	452
28.3.1	IDbConnection 接口	416	30.1.2	简单数据绑定	452
28.3.2	RDPCConnection 类	417	30.1.3	复杂数据绑定	454
28.4	实现自定义 Command	419	30.1.4	单向和双向数据绑定	455
28.4.1	IDbCommand 接口	419	30.2	数据绑定结构	455
28.4.2	RDPCCommand 类	420	30.2.1	BindingContext 类	455
28.5	实现自定义 DataReader	424	30.2.2	CurrencyManager 类	455
28.5.1	IDataReader 接口	424	30.2.3	PropertyManager 类	458
28.5.2	RDPDDataReader 类	424	30.3	数据绑定示例	458
28.6	实现自定义 DataAdapter	429	30.3.1	简单绑定	458
28.6.1	IDbDataAdapter 接口	429	30.3.2	绑定到 ComboBox	459
28.6.2	RDPDDataAdapter 类	429	30.3.3	DataGrid 绑定	459
28.7	小结	433	30.4	高级数据绑定示例	461
第 29 章	类型化的 DataSet 和 XSD	434	30.4.1	主/从窗体	461
29.1	XML 模式定义	434	30.4.2	层叠式主/从	463
29.1.1	XSD	434	30.5	小结	466
29.1.2	XSD 中的基本数据类型	436	第 31 章	Web Forms 数据绑定	467
29.1.3	派生的数据类型	436	31.1	数据绑定	467
29.1.4	复杂数据类型	437	31.1.1	Web Forms 数据绑定	467
29.1.5	分组元素	437	31.1.2	<%# %>绑定语法	468
29.1.6	注解 XML 模式	438	31.1.3	简单数据绑定	468
29.1.7	XML 模式剖面	438	31.1.4	复杂数据绑定	470
29.1.8	设计 XML 模式—— XmlSchema 类	439	31.1.5	DataBind()方法	470
29.2	使用模式构造 DataSet	442	31.2	数据绑定结构	470
29.2.1	使用 XML 模式定义 表和列	442	31.2.1	Container.DataItem	470
29.2.2	使用 XML 模式定义 DataSet 键和约束	443	31.2.2	DataBinder.Eval	470
29.3	类型化的 DataSet	445	31.2.3	ItemDataBound 事件	471
			31.3	数据绑定示例	471
			31.3.1	简单绑定	472
			31.3.2	Repeater 绑定	472

31.3.3	DataList 绑定	475
31.3.4	DataGrid 绑定	477
31.4	高级绑定示例	477
31.4.1	主从窗体	477
31.4.2	层叠式主从	481
31.5	小结	484

第VI部分 Web 服务

第 32 章	Web 服务介绍	489
32.1	Web 服务	489
32.1.1	定义 Web 服务	489
32.1.2	SOAP	489
32.1.3	WSDL	491
32.2	创建 Web 服务	491
32.2.1	Hello World	491
32.2.2	复杂串行化	494
32.2.3	使用 Web 服务的事务	497
32.2.4	维护 Web 服务的状态	497
32.2.5	Web 服务的合同优先编程	498
32.3	回顾 Web 服务的使用	498
32.3.1	为 Web 服务创建客户端 代理	498
32.3.2	同步调用	498
32.3.3	异步调用	498
32.4	小结	499
第 33 章	WSE 2.0	501
33.1	WSE 2.0	501
33.1.1	GXA	501
33.1.2	WSE 的发展	501
33.2	TCP 消息传送	502
33.2.1	TCP 承载的 SOAP	502
33.2.2	SoapSender 和 SoapReceiver	502
33.2.3	SoapService 类	505
33.2.4	SoapClient 类	506
33.3	安全性	508
33.3.1	WSE 安全性	508
33.3.2	UsernameTokens	508

33.3.3	X.509 认证	509
33.3.4	签名消息	509
33.4	使用 WSE 2.0 传递附件消息	510
33.4.1	DIME	510
33.4.2	通过 WSE 传输文件	510
33.5	小结	510

第VII部分 保障应用程序的安全

第 34 章	代码访问安全	515
34.1	CAS	515
34.1.1	使用代码访问安全权限	516
34.1.2	代码访问权限	516
34.1.3	身份权限	517
34.1.4	基于角色的安全权限	517
34.2	CAS 管理	518
34.2.1	修改 CAS 策略	518
34.2.2	策略管理工具	518
34.2.3	提高程序集的信任度	519
34.2.4	调整区域安全	519
34.2.5	评估程序集	520
34.2.6	创建部署包	521
34.3	编写识别 CAS 的代码	522
34.3.1	使用强制性安全句法	522
34.3.2	使用声明性安全句法	522
34.3.3	封闭不必要的客户端	523
34.4	小结	525
第 35 章	敏感数据安全	526
35.1	密钥(对称)加密	526
35.1.1	DESCryptoServiceProvider	526
35.1.2	RC2CryptoServiceProvider	527
35.1.3	RijndaelManaged	527
35.1.4	TripleDESCryptoService Provider	528
35.2	使用散列保护数据	530
35.2.1	MACTripleDES	530
35.2.2	SHA1Managed	531
35.2.3	MD5CryptoServiceProvider	532
35.3	公钥加密和签名	533

35.3.1	DSACryptoServiceProvider	533
35.3.2	RSACryptoServiceProvider	535
35.4	Windows Data Protection API	535
35.4.1	使用 DPAPI	536
35.4.2	创建 DPAPI 包装器	537
35.4.3	使用 DPAPI 保护.NET 中 的数据	542
35.5	小结	543
第 36 章	ASP.NET Web 应用程序 的安全	544
36.1	用户安全	544
36.1.1	用户验证	544
36.1.2	用户角色授权	548
36.1.3	实现 IIdentity 和 IPrincipal	548
36.2	ASP.NET 应用程序中的 数据安全	553
36.2.1	保护连接字符串和 Web.config 数据	554
36.2.2	保护用户口令	554
36.2.3	决定何时使用 SSL	555
36.2.4	使用 ViewState 加密的 数据安全	556
36.3	小结	556
第 37 章	授权和知识产权	557
37.1	授权	557
37.1.1	授权定义	557
37.1.2	何时授权和保护	557
37.1.3	授权和验证的类型	558
37.2	实现自定义授权	558
37.2.1	许可提供者和许可管理员	558
37.2.2	创建许可	559
37.2.3	创建许可提供者	560
37.2.4	建立授权控件	563
37.2.5	授权 Web 控件和 Windows Forms 控件	566
37.3	授权实现策略	566
37.3.1	决定授权部署方法	567
37.3.2	决定授权验证方法	567
37.3.3	决定许可购买方法	567

37.3.4	决定授权方法	567
37.4	保护知识产权	567
37.4.1	通过隐藏授权算法保护 知识产权	567
37.4.2	通过混淆法保护知识产权	568
37.4.3	使用备用后端保护知识 产权	568
37.5	小结	568

第VIII部分 企业应用与相关的应用

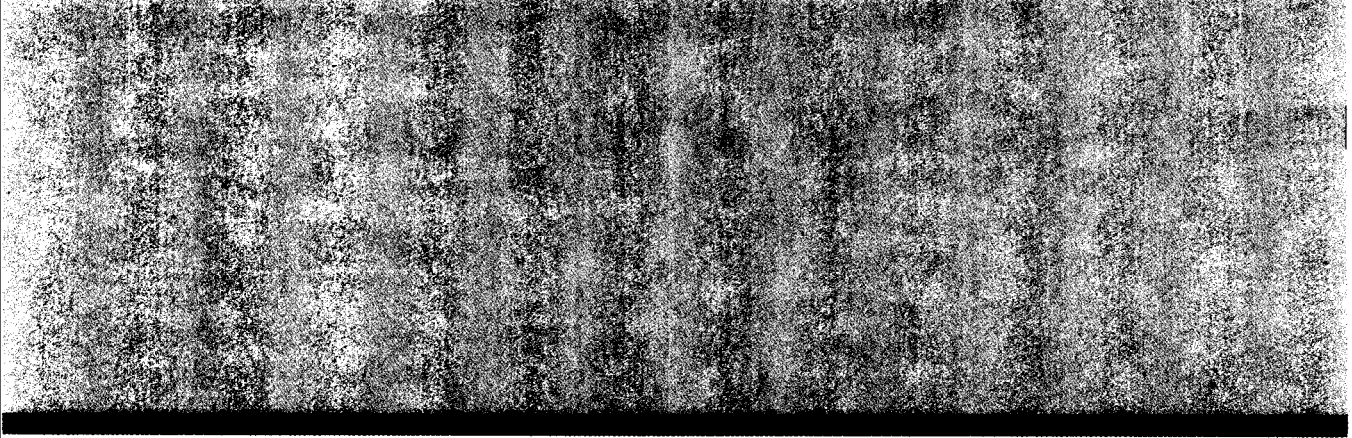
第 38 章	接口编程	573
38.1	接口编程	573
38.1.1	了解接口	573
38.1.2	隐式声明接口	576
38.1.3	显式声明接口	580
38.1.4	映射接口	582
38.1.5	继承接口	584
38.2	小结	585
第 39 章	Remoting	586
39.1	Remoting 体系结构	586
39.1.1	Remoting	586
39.1.2	应用程序域	587
39.1.3	上下文	588
39.1.4	选择信道	589
39.1.5	远程对象的生存与死亡	590
39.1.6	构建 Remoting 服务器 应用程序	595
39.1.7	构建客户端	596
39.2	IIS 和 Remoting	597
39.3	小结	598
第 40 章	COM+企业服务	599
40.1	COM+	599
40.1.1	事务	599
40.1.2	COM+中的 JIT 激活	600
40.1.3	对象入池	600
40.1.4	构造字符串	601
40.1.5	基于角色的安全性	601
40.1.6	排队的组件	601

40.1.7	事件	601
40.2	构建 COM+组件	602
40.2.1	事务	602
40.2.2	构造字符串	604
40.2.3	JIT 激活示例	605
40.2.4	对象入池	605
40.2.5	共享的属性	606
40.3	COM+中的安全性	607
40.3.1	对象和安全性上下文	608
40.3.2	基于角色的安全性	608
40.4	高级 COM+	610
40.4.1	事件	610
40.4.2	排队的组件	612
40.5	小结	612
第 41 章	企业模板	613
41.1	企业模板	613
41.1.1	企业模板介绍	613
41.1.2	静态和动态内容	614
41.1.3	静态原型	615
41.1.4	子项目向导	615
41.1.5	自定义向导	615
41.2	策略文件	615
41.3	示例教学	617
41.3.1	设置先决条件	618
41.3.2	布置模板	618
41.3.3	创建模板结构	618
41.3.4	将策略分配给模板	622
41.3.5	建立模板	622
41.3.6	建立用户可用的模板	624
41.3.7	测试模板	625
41.4	小结	627

第IX部分 调试和测试

第 42 章	调试应用程序	631
42.1	Visual Studio .NET 调试环境	631
42.1.1	设置调试的应用程序	631
42.1.2	理解语法和错误消息	633
42.1.3	理解调试工具窗口	634

42.1.4	导航应用程序	637
42.1.5	设置并使用断点	638
42.2	使用 Visual Studio .NET 调试	639
第 43 章	监控应用程序	647
43.1	调试和跟踪语句	647
43.1.1	跟踪监听器	650
43.1.2	跟踪开关	654
43.1.3	自定义跟踪监听器	657
43.2	小结	658
第 44 章	检测应用程序	659
44.1	检测应用程序	659
44.1.1	检测应用程序介绍	659
44.1.2	检测方法	660
44.1.3	Windows 事件日志	660
44.1.4	检查 Debug 和 Trace 类	662
44.1.5	Windows 管理检测	664
44.2	企业检测框架	668
44.2.1	EIF 简介	668
44.2.2	企业检测框架的需求	668
44.2.3	EIF 的元素	668
44.2.4	请求跟踪	670
44.2.5	配置 EIF	671
44.3	小结	671
第 45 章	C#的前景	672
45.1	C#的前景	672
45.1.1	泛性(Generics)	672
45.1.2	匿名方法	675
45.1.3	最终的可空类型	675
45.1.4	带有迭代器的列表管理	676
45.1.5	不完全类型	676
45.1.6	静态类	677
45.2	小结	678



第 I 部分

Visual Studio .NET IDE 简介

第 1 章 Visual Studio .NET IDE

