



Delphi for .NET Developer's Guide

SAMS

Borland
核心技术丛书

amazon.com 亚马逊 Delphi 类第一畅销书



曾获 Delphi Informant 杂志最佳图书

Delphi for .NET 开发人员指南

(美) Xavier Pacheco 等著
马朝晖 顾良翠 等译

“Delphi技术领域中最著名的作家Xavier Pacheco向你全面介绍
Delphi for .NET!”

—— Danny Thorpe (Borland 公司首席科学家)



机械工业出版社
China Machine Press



Delphi

Delphi for .NET 开发人员指南

(美) Xavier Pacheco 等著
马朝晖 顾良翠 等译

 机械工业出版社
China Machine Press

本书是 Delphi 经典技术图书。新版本详细讲解 Delphi 和 .NET Framework 的基本特点, 深入讨论了如何使用 Borland 最好的开发工具编写 .NET 应用程序, 如何高效地利用 .NET Framework 的实用范例、技术和原则。本书内容丰富, 包括 Delphi 语言、高级 .NET 开发概念 (如内存管理)、COM-Interop、反射、GDI+、Windows、Web Forms 组件开发等等, 还包括关于 ADO.NET 和 ASP.NET 等核心 .NET 技术的大量信息和范例。Delphi 用户通过掌握这些新技术可以将自己的开发技能提高到新的层次。Delphi for .NET 的初学者可以快速掌握使用 Delphi 进行 .NET 开发的技巧。

Xavier Pacheco, et al: Delphi for .NET Developer's Guide (ISBN 0-672-32443-1) .

Authorized translation from the English language edition entitled *Delphi for .NET Developer's Guide* by Xavier Pacheco, et al, published by Pearson Education, Inc, publishing as Sams Publishing, Copyright © 2004 by Sams Publishing.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without permission of Pearson Education, Inc.

Chinese Simplified language edition published by China Machine Press.

Copyright © 2005 by China Machine Press.

本书中文简体字版由美国 Pearson Education 培生教育出版集团授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2004-3284

图书在版编目 (CIP) 数据

Delphi for .NET 开发人员指南 / (美) 帕切科 (Pacheco, X.) 等著; 马朝晖等译. - 北京: 机械工业出版社, 2005.1

(Borland 核心技术丛书)

书名原文: Delphi for .NET Developer's Guide

ISBN 7-111-15512-2

I. D… II. ①帕… ②马… III. 计算机网络-程序设计 IV. TP393

中国版本图书馆 CIP 数据核字 (2004) 第 112312 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 华 章

北京瑞德印刷有限公司印刷 · 新华书店北京发行所发行

2005 年 1 月第 1 版第 1 次印刷

787mm × 1092mm 1/16 · 39.5 印张

印数: 0 001-5 000 册

定价: 78.00 元 (附光盘)

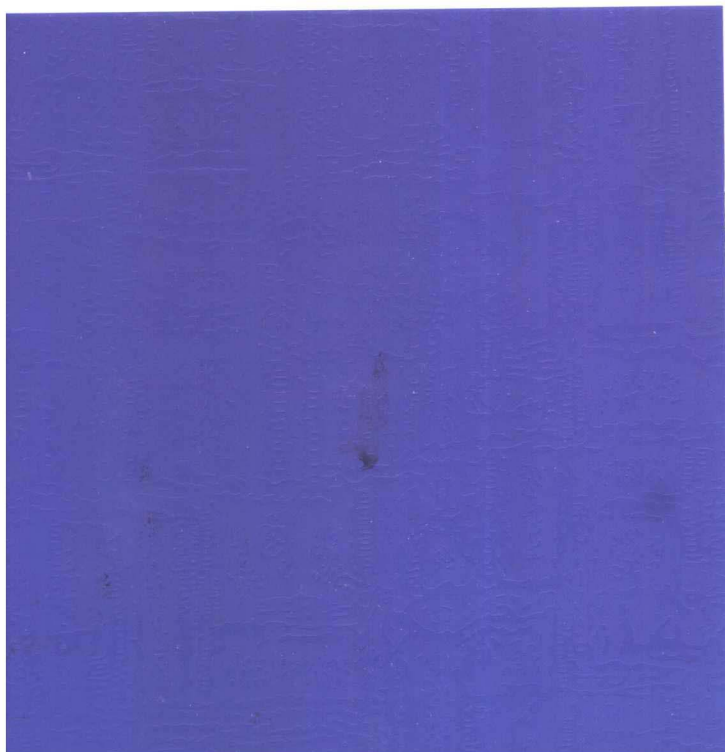
凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换
本社购书热线: (010) 68326294

作者简介



Xavier Pacheco 就是 Borland 社区大名鼎鼎的 X。他曾是 Borland 公司资深开发人员，Borland 梦之队(Team Borland)成员。1998 年创建了 Xapware Technologies 公司，担任总裁。Xavier 在开发软件解决

方案方面具有超过 16 年的专业经验，涉及分布式系统、应用程序体系结构以及过程和设计方法学等方面。Xavier 是一位国际知名的开发人员、作家、顾问和培训讲师。他编写了多本关于 Delphi 的名著，常常撰写技术论文并且受邀在主要业界会议上做演讲。



关于作者

Xavier Pacheco 是 1988 年 1 月建立的 Xapware Technologies 公司的总裁。该公司帮助用户通过 Xapware 的产品 Active! Focus 成功地进行软件开发，这个产品是一个基于团队的软件开发管理套件。Xavier 在开发软件解决方案方面具有超过 16 年的专业经验，涉及分布式系统、应用程序体系结构以及过程和设计方法学等方面。Xavier 是一位国际认可的开发人员、作家、顾问和培训师。他编写了几本关于 Delphi 的书，常常撰写文章并且在业界会议上做演讲。Xavier 和他的妻子 Anne 以及他们的孩子 Zachary 和 Jessica 生活在科罗拉多州的 Colorado Springs。

关于合作作者

Steven Beebe 是 Xapware Technologies 公司的首席运营官和高级顾问。Xapware 公司是 Active! Focus 的提供者，这个产品管理软件项目的许多方面的实用解决方案。Steven 从事软件开发和软件开发管理已经超过 15 年。Steven 和他的妻子 Diane 以及他们的女儿 Hannah 和 Sarah 生活在科罗拉多州的 Colorado Springs。

Alessandro Federici 是 Delphi 群体中受到高度认可和尊敬的成员。在 2002 年，他建立了 RemObjects Software 公司，这个公司已经发展成为 Delphi 远程和数据访问领域的重要公司之一。Alessandro 有超过 14 年的程序设计经验，当前擅长的领域是分布式系统和体系结构的设计与开发。意大利米兰出生的 Alessandro 现在和他的妻子生活在伊利诺斯州的 Hoffman Estates。

Nick Hodges 是 Delphi 顾问和培训师以及 Lemanix 公司的 CTO。Nick 写过许多书籍章节和杂志文章，并且经常在 Borland 会议上做演讲。他是 Borland 的 TeamB 和 Borland Conference Advisory Board 的成员。Nick 从 Naval 研究生院获得了信息技术管理硕士学位。Nick 和他的家庭生活住在明尼苏达州的 St. Paul。

Brian Long 是 Delphi、C++ Builder 和 .NET 方面的培训师和顾问，并且常常在国际会议上做演讲。除了在 1994 年编写了一本解决 Borland Pascal 问题方面的书以及参与编写了一本较早的 Delphi 书籍之外，Brian 还在《Delphi Magazine》主持 Q&A 专栏，并且为其他期刊和 <http://bdn.borland.com> 供稿。Brian 在 2000 年获得 Spirit of Delphi 奖提名并且在 2002 年 5 月被选举为 BorCon 最佳演讲人。

Rick Ross 是 PILLAR Technology Group 公司的高级顾问，他在那里帮助客户开发最适当地利用技术的解决方案。Rick 具有超过 15 年的专业经验。他喜欢编写复杂的应用程序（比如分布式系统和多线程应用程序）来解决特定的业务挑战。Rick 是《Kylix 2 Development》的合作作者并且为《Delphi Informant》杂志撰稿。他经常在世界各地的会议上做演讲。Rick 和他的家庭生活住在密歇根州的 Brighton。

Steve Teixeira 是软件开发咨询公司 Falafel Software 的 CTO。以前，Steve 在 Zone 实验室（一家业界领先的因特网安全解决方案开发商）担任产品体系结构主管，并且在其他两家软件咨询公司担任 CTO。在 Borland 担任 R&D 软件工程师期间，他参与了 Delphi 和 C++ Builder 的开发。

Steve 编写了 5 本获奖的书和许多软件开发方面的杂志文章，并且常常在世界各地的业界会议上做演讲。

关于技术编辑

Hallvard Vassbotn 是挪威奥斯陆的资深 Delphi 程序员。他是 Infront AS 的高级系统开发人员，他在那里开发通过因特网实时提供金融信息的应用程序，见 <http://www.theonlinetrader.com/>。Hallvard 为《Delphi Magazine》撰写了许多文章。在不工作的时候，他把时间花在三个可爱的女儿 Nina、Ida 和 Thea 身上。

前 言

本书是一本关于开发 .NET 程序，特别是使用 Delphi 开发 .NET 程序的书。本书并不是以前的《Delphi 开发人员指南》^①的更新版。除了第 5 章之外，其他内容都是全新的。对于本书，我还采用了与以往稍微不同的方式。在这里，你会找到较多的说明几个或单一概念的范例，而将概念组合使用的大型范例比较少。这使你不会陷入技术意义不大的代码中，能够更容易地研究相关技术。

另外，我有意尽可能避免使用 IDE 的设施。例如，在关于 ADO.NET 的章节里，你会看到我通过程序来设置属性。我这样做，你就可以看到使用的是哪些属性，而且这可以避免采用大量描述 IDE 使用过程的文字，避免打乱描述实际技术的思路。但是，因为使用 IDE 对于开发效率很重要，所以第 3 章重点讲解了某些能够帮助你提高开发效率的 IDE 功能。从本质上说，我的目标是写一本专门针对 .NET 的使用 Delphi 开发的技术书，而不是写一本关于 Delphi for .NET 的书。

谁应该阅读本书

本书是为开发人员，特别是使用 Delphi 的 .NET 开发人员而写的。具体地说，本书针对以下三类读者：

- 希望将技能提升到更高水平并且对 .NET 感兴趣的专业开发人员。
- 过去使用其他语言，现在希望学习如何使用 Delphi 开发 .NET 程序的 .NET 开发人员。
- 希望学习 .NET 和使用一种强大且直观的语言来开发 .NET 程序的初学者。

本书是如何组织的

本书分为五个部分：

- 第一部分概述 .NET 总体情况和 .NET Framework 的特点。
- 第二部分讨论 Delphi for .NET、IDE 和 Delphi 语言的特点。
- 第三部分讨论开发 .NET Framework 时的核心概念。例如，线程化、反射、流、内存管理等等。
- 第四部分研究使用 Delphi 和 ADO.NET 创建数据库应用程序的细节。
- 第五部分研究使用 ASP.NET 创建 Web 应用程序的各种技术。

配套光盘上有什么

在本书的配套光盘上你会找到所有源代码和项目文件，以及我们无法放进本书中的源代码范例。源代码目录是按章和范例命名的，比如：

```
\Code\Chapter 01\Ex01  
\Code\Chapter 01\Ex02
```

① 本书作者还写过《Delphi 6 开发人员指南》，该书已由机械工业出版社引进出版，书号是：7-111-10968-6。

```
\Code\Chapter 02\Ex01
```

```
...
```

在各章的目录下，列出了相应章的程序清单。

在光盘上，你还会找到完整的《Delphi 6 开发人员指南》(Delphi 6 Developer's Guide) 英文版，并采用每章一个 .pdf 文件的形式提供。你会在\VD6DG 文件夹下找到它。此外，还包括《Delphi 6 开发人员指南》的源代码(在\VD6DG\Code 文件夹里)。因此，你买这一本书，实际上得到了两本书。

最后，你还会找到许多第三方工具/组件的演示版，可以在开发 Delphi for .NET 应用程序时使用。

本书中使用的约定

本书有以下约定：

过程和函数用过程名或函数名后的一对圆括号来表示。尽管这不是标准的 Pascal 语法，但是这有助于使它们有别于属性、变量和类型。

在每一章中，你会遇到一些“注意”、“提示”和“警告”，这些特殊段落强调了重点并且帮助你避开陷阱。

把你的想法告诉我们！

作为本书的读者，你是我们最重要的批评家和评论家。我们重视你的意见，希望得知我们在什么方面做得对，在什么方面还需要改进，你希望我们出版哪方面的内容，以及你希望传达给我们的其他建议。

作为 Sams 出版公司的合作出版商，我欢迎你提出意见。你可以直接给我发电子邮件或写信，让我知道你喜欢或不喜欢本书的哪些内容，以及我们如何改进图书内容。

请注意，我无法帮助你解决与本书主题有关的技术问题。但是，我们有一个用户服务组，我将把与本书有关的特定技术问题转发到那里。

在你写信时，请一定要包含本书的书名和作者名以及你的姓名、电子邮件地址和电话号码。我将认真阅读你的建议并与本书的作者和编辑交流。

电子邮件：feedback@sampublishing.com

邮件地址：Michael Stephens

Associate Publisher

Sams Publishing

800 East 96th Street

Indianapolis, IN 46240 USA

要想获得关于本书和 Sams 出版公司的其他书籍的信息，请访问我们的网站 www.sampublishing.com。在 Search 框中输入书籍的 ISBN（不包括连字符）或书名，就可以找到你需要的信息。

本书主要由马朝晖、顾良翠、陈美红翻译，参与翻译、录入、审校的还有刘丽珍、王建芬、杨帆、邹辉、潘浩、楼涵、董小蕾、王悦、李军、罗伟、鲍广华、翟兰、陆明、宋丽、杨立军、李鸣、马晓云、成荣光等。

目 录

关于作者
前言

第一部分 .NET Framework 概述

第 1 章 .NET 概述	2
1.1 使用 .NET 的动机	2
1.1.1 .NET 展望	2
1.1.2 .NET Framework 组件——通用 语言运行环境和类库	4
1.1.3 .NET 应用程序的类型	4
1.1.4 VCL for .NET	5
1.2 通过 Web 服务进行分布式开发	5
1.2.1 Web 服务定义	6
1.2.2 Web 服务客户	7
1.2.3 Web 服务开发工具	7
第 2 章 .NET Framework 概况	9
2.1 从开发到执行	9
2.2 通用语言运行环境 (CLR)	10
2.2.1 托管模块	10
2.2.2 配件	10
2.2.3 托管代码和非托管代码	10
2.2.4 MSIL 和 JIT 编译和执行	11
2.3 通用类型系统 (CTS)	12
2.3.1 值类型	12
2.3.2 引用类型	13
2.4 通用语言规范 (CLS)	13
2.5 .NET Framework 和基本类库 (FCL)	13
2.5.1 名字空间	13
2.5.2 System 名字空间	13
2.5.3 主要的子系统名字空间	13

第二部分 Delphi for .NET 编程语言

第 3 章 Delphi for .NET 和新 IDE	
概述	18
3.1 Delphi for .NET——更大的画面	18

3.2 集成开发环境 (IDE) 概述	18
3.2.1 欢迎页面	19
3.2.2 设计器	19
3.2.3 窗体	20
3.2.4 工具面板/代码片段	21
3.2.5 对象观察器	22
3.2.6 代码编辑器	22
3.2.7 项目管理器	24
3.2.8 模型视图	24
3.2.9 数据浏览器	25
3.2.10 对象信息库	25
3.2.11 代码浏览器	25
3.2.12 To-Do 列表	25
第 4 章 程序、单元和名字空间	27
4.1 托管模块结构	27
4.1.1 程序结构	27
4.1.2 单元结构	28
4.1.3 uses 子句的语法	29
4.1.4 循环单元引用	30
4.2 名字空间	30
4.2.1 名字空间声明	30
4.2.2 使用名字空间	31
4.2.3 namespaces 子句	32
4.2.4 解析一般名字空间	32
4.2.5 单元别名	32
第 5 章 Delphi 语言	33
5.1 一切都通过 .NET	33
5.2 注释	34
5.3 过程和函数	34
5.3.1 调用中的圆括号	34
5.3.2 重载	35
5.3.3 默认值参数	35
5.4 变量	36
5.5 常量	37
5.6 操作符	38
5.6.1 赋值操作符	38

5.6.2	比较操作符	38
5.6.3	逻辑操作符	39
5.6.4	算术操作符	39
5.6.5	位操作符	40
5.6.6	递增和递减过程	40
5.6.7	运算 - 赋值操作符	41
5.7	Delphi 语言类型	41
5.7.1	无处不是对象	41
5.7.2	类型的对比	41
5.7.3	字符	41
5.7.4	变体类型	42
5.8	用户定义的类型	45
5.8.1	数组	45
5.8.2	动态数组	46
5.8.3	记录	47
5.8.4	集	48
5.8.5	不安全的代码	49
5.8.6	指针	50
5.8.7	类和对象	52
5.8.8	类型别名	53
5.9	强制类型转换和类型转换	54
5.10	字符串资源	54
5.11	测试条件	55
5.11.1	if 语句	55
5.11.2	case 语句	56
5.12	循环	56
5.12.1	for 循环	56
5.12.2	while 循环	57
5.12.3	repeat... until	57
5.12.4	Break 语句	58
5.12.5	Continue 语句	58
5.13	再论过程与函数	58
5.14	有效范围	61
5.15	单元和名字空间	61
5.15.1	uses 子句	62
5.15.2	循环单元引用	63
5.16	包和配件	63
5.17	面向对象的程序设计	63
5.18	使用 Delphi 对象	64
5.18.1	声明和实例化	64
5.18.2	析构	65
5.18.3	对象的祖先	65

5.18.4	字段	65
5.18.5	方法	66
5.18.6	方法类型	66
5.18.7	类引用	68
5.18.8	属性	69
5.18.9	事件	70
5.18.10	可见性指示符	72
5.18.11	“友元”类	73
5.18.12	类辅助器	73
5.18.13	嵌套的类型	74
5.18.14	操作符重载	74
5.18.15	特性	75
5.18.16	接口	75
5.19	结构化异常处理	78
5.19.1	异常类	80
5.19.2	执行流	81
5.19.3	重新引发异常	82

第三部分 .NET Framework 类库开发

第 6 章	配件——库和包	84
6.1	核心配件	84
6.2	查看配件内容和依赖性	84
6.3	GAC	85
6.4	建立配件	86
6.4.1	为什么要使用配件	86
6.4.2	使用包建立配件	86
6.4.3	使用库建立配件	90
6.5	在 Delphi 中使用配件	93
6.6	在 C# 中使用 Delphi 配件	93
6.7	将包安装进 Delphi IDE	94
6.8	强命名的配件	94
6.9	动态装载配件	94
第 7 章	GDI+ 编程——.NET 中的 绘图	96
7.1	基本概念	96
7.1.1	GDI+ 名字空间	96
7.1.2	Graphics 类	96
7.1.3	Windows 坐标系统	96
7.2	绘制直线	98
7.2.1	Pen 和 Brush 类	98
7.2.2	绘制直线	98
7.2.3	直线端点样式	100

7.2.4 连接直线——GraphicsPath 类	101
7.3 绘制曲线	102
7.3.1 基数样条	102
7.3.2 贝济埃样条	102
7.4 绘制形状	105
7.4.1 绘制矩形	105
7.4.2 绘制椭圆	105
7.4.3 绘制多边形	106
7.4.4 绘制饼图	106
7.4.5 再论 LinearGradientBrush	107
7.5 GraphicsPath 和 Region	107
7.5.1 用 GraphicsPath 类进行绘制	107
7.5.2 Region 类	108
7.5.3 修剪 Region	110
7.6 使用图像	111
7.6.1 图像类	111
7.6.2 装载和创建位图	112
7.6.3 修改图像的分辨率	112
7.6.4 绘制图像	113
7.6.5 插值	113
7.6.6 产生镜像效果	115
7.6.7 使用变换方法	115
7.6.8 创建缩略图	118
7.7 再论坐标系统	118
7.8 动画范例	119
第 8 章 Mono——一个跨平台 .NET 项目	125
8.1 Mono 的特点	125
8.2 Mono 的历史	126
8.3 为什么要学习 Mono	126
8.4 Mono 的发展计划	127
8.4.1 Mono 1.0 目标	127
8.4.2 Mono 1.2 目标	127
8.4.3 Mono 1.4 目标	128
8.5 安装/设置	128
8.6 创建你的第一个 Mono 程序	130
8.7 在 Mono 下运行 Delphi 产生的 配件 (在 Linux 上)	131
8.8 跨平台 ASP.NET	134
8.8.1 将 ASP.NET 部署到 Mono	135
8.8.2 XSP 配置	135
8.8.3 XSP 运行参数	135

8.8.4 对范例的一些警告和一个 次要的扩展	136
8.9 ADO.NET 与 Mono	136
8.10 Mono 和 Apache	139
8.11 Mono 和 System.Windows.Forms	140
第 9 章 内存管理和垃圾回收	141
9.1 垃圾回收如何起作用	141
9.1.1 代式垃圾回收	141
9.1.2 调用垃圾回收器	142
9.2 构造器	143
9.3 终结	144
9.4 dispose 模式——IDisposable	145
9.4.1 IDisposable 范例	145
9.4.2 自动实现 IDisposable	146
9.5 与终结有关的性能问题	147
第 10 章 集合	148
10.1 System.Collections 接口	148
10.1.1 IEnumerable 接口	149
10.1.2 ICollection 接口	149
10.1.3 IList 接口	149
10.1.4 IDictionary 接口	149
10.1.5 IEnumerator 接口	149
10.2 System.Collections 类	149
10.2.1 Stack 集合	150
10.2.2 Queue 类	152
10.2.3 ArrayList 类	154
10.2.4 HashTable 类	156
10.3 创建强类型的集合	159
10.3.1 从 CollectionBase 进行派生	159
10.3.2 使用强类型集合	162
10.4 创建强类型词典	162
10.4.1 从 DictionaryBase 进行派生	162
10.4.2 使用强类型词典	165
第 11 章 使用 String 和 StringBuilder 类	166
11.1 System.String 类型	166
11.1.1 .NET 中的字符串不可变性	167
11.1.2 字符串操作	168
11.2 StringBuilder 类	171
11.2.1 StringBuilder 方法	171
11.2.2 StringBuilder 的使用方法	172

11.3 字符串格式化	172	13.3 用户绘制: PlayingCard 控件	226
11.4 格式指示符	173	第 14 章 Delphi for .NET 中的线程化	233
11.4.1 数值格式指示符	173	14.1 进程	233
11.4.2 日期和时间格式指示符	175	14.2 线程化	234
11.4.3 枚举格式指示符	178	14.3 线程化 .NET 风格	234
第 12 章 文件和流操作	179	14.4 AppDomain	235
12.1 System.IO 名字空间类	179	14.5 System.Threading 名字空间	236
12.2 操作目录系统	180	14.5.1 System.Threading.Thread 类	236
12.2.1 创建和删除目录	180	14.5.2 System.Threading.ThreadPriority	240
12.2.2 移动和拷贝目录	181	14.5.3 System.Threading.ThreadState	240
12.2.3 查看目录信息	182	14.5.4 System.Threading.ApartmentState 枚举	241
12.3 操作文件	183	14.5.5 System.Threading.ThreadPool 类	241
12.3.1 创建和删除文件	183	14.5.6 System.Threading.Timer 类	243
12.3.2 移动和拷贝文件	183	14.5.7 代理	243
12.3.3 查看文件信息	183	14.6 编写线程安全代码的 .NET 风格	245
12.4 流	184	14.6.1 锁机制	245
12.4.1 操作文本文件流	184	14.6.2 事件	249
12.4.2 操作二进制流	186	14.6.3 线程局部存储	250
12.5 异步流访问	188	14.6.4 Win32 进程间通信	250
12.6 监视目录活动	190	14.6.5 线程安全的 .NET Framework 类 和方法	251
12.7 串行化	192	14.7 用户界面问题	251
12.7.1 串行化如何工作	192	14.7.1 System.Windows.Forms.Control .Invoke() 方法	252
12.7.2 格式化器	193	14.7.2 System.Windows.Forms.Control .InvokeRequired 属性	252
12.7.3 串行化/反串行化范例	193	14.7.3 System.Windows.Forms.Control .BeginInvoke() 方法	253
第 13 章 开发定制的 WinForms 控件	196	14.7.4 System.Windows.Forms.Control .EndInvoke() 方法	253
13.1 组件构造基础知识	196	14.7.5 System.Windows.Forms.Control .CreateGraphics() 方法	254
13.1.1 什么时候需要编写组件	196	14.8 线程化异常	256
13.1.2 组件的编写步骤	197	14.8.1 System.Threading.ThreadAbort- Exception	256
13.1.3 决定祖先类	197	14.8.2 System.Threading.ThreadInter- ruptedException	258
13.1.4 创建组件单元	198	14.8.3 System.Threading.ThreadState- Exception	258
13.1.5 创建属性	200	14.8.4 System.Threading.Synchroniza-	
13.1.6 创建事件	208		
13.1.7 创建方法	213		
13.1.8 构造器和析构器	213		
13.1.9 设计时行为	214		
13.1.10 测试组件	214		
13.1.11 提供组件图标	214		
13.2 组件范例	215		
13.2.1 ExplorerViewer: UserControl 范例	215		
13.2.2 SimpleStatusBars: 使用扩展器 提供者	223		

tionLockException	258	16.5.3 参数类型和遍组	295
14.9 垃圾回收和线程化	258	16.5.4 错误处理	297
第 15 章 反射 API	259	16.5.5 Win32 错误码	297
15.1 对配件进行反射	259	16.5.6 HRESULT 错误码	299
15.2 对模块进行反射	261	16.5.7 性能问题	300
15.3 对类型进行反射	263	16.6 在 Win32 代码中使用 .NET 例程	304
15.4 对类型的成员的运行时调用 (晚绑定)	264	16.6.1 传统的 Delphi 语法	304
15.4.1 为了效率而调用成员的类型	267	16.6.2 参数类型和遍组	305
15.4.2 成员调用的另一个范例	267		
15.5 通过反射产生 MSIL	269	第四部分 用 ADO.NET 进行 数据库开发	
15.5.1 用于产生 MSIL 的工具/类	270	第 17 章 ADO.NET 概况	310
15.5.2 产生过程	270	17.1 设计原则	310
15.5.3 System.Reflection.Emit 范例	270	17.1.1 不连接的数据体系结构	310
第 16 章 互操作性——COM Interop 和平台调用服务	273	17.1.2 与 XML 集成	310
16.1 为什么需要互操作性	273	17.1.3 通用数据表示	311
16.2 常见的互操作性问题	274	17.1.4 在 .NET 框架上进行构建	311
16.3 在 .NET 代码中使用 COM 对象	274	17.1.5 利用现有技术手段	311
16.3.1 晚绑定 Automation	274	17.2 ADO.NET 对象	311
16.3.2 值、引用和可选参数	276	17.2.1 连接的类	312
16.3.3 早绑定 COM	278	17.2.2 不连接的类	312
16.3.4 Interop 配件	279	17.3 .NET 数据提供者	313
16.3.5 创建 Interop 配件	280	第 18 章 使用 Connection 对象	314
16.3.6 Interop 配件中有什么	281	18.1 连接功能	314
16.3.7 使用 COM 事件	281	18.2 设置 ConnectionString 属性	314
16.3.8 COM 生存周期控制	283	18.2.1 指定 SqlConnection.Connection- String	315
16.3.9 错误处理	283	18.2.2 指定 OleDbConnection.Connection- String	315
16.3.10 主 Interop 配件	283	18.2.3 指定 OdbcConnection.Connection- String	315
16.3.11 定制 Interop 配件和 PIA	284	18.2.4 指定 OracleConnection.Connection- String	315
16.4 在 COM 代码中使用 .NET 对象	285	18.3 打开和关闭连接	316
16.4.1 为 Automation 注册 .NET 配件	286	18.4 Connection 事件	316
16.4.2 晚绑定 Automation	287	18.5 连接池	318
16.4.3 Interop 类型库	287	第 19 章 使用 Command 和 DataReader 对象	319
16.4.4 Interop 类型库中有什么	288	19.1 执行命令	319
16.4.5 实现接口	288	19.2 非查询命令	320
16.4.6 参数类型和遍组	290	19.3 获得单一值	321
16.4.7 错误处理	292		
16.5 在 .NET 代码中使用 Win32 DLL 导出	292		
16.5.1 传统 Delphi 语法	293		
16.5.2 定制特性语法	293		

19.4 执行数据定义语言 (Data Definition Language、DDL) 命令	322	窗体	362
19.5 使用 IDbParameter 指定参数	323	第 22 章 将数据存入数据源	371
19.6 执行存储过程	325	22.1 使用 SqlCommandBuilder 更新数据源	371
19.7 派生参数	327	22.2 使用定制的更新逻辑更新数据源	373
19.8 使用 DataReader 查询 Resultset	328	22.2.1 使用 Command 类	373
19.9 查询 Resultset	328	22.2.2 使用 SqlDataAdapter 类	379
19.10 使用 DataReader 查询多个 Resultset	329	22.2.3 使用存储过程进行更新	383
19.11 使用 DataReader 获得 BLOB 数据	330	22.2.4 处理并发	389
19.12 使用 DataReader 获得计划信息	331	22.2.5 在更新数据后刷新数据	391
第 20 章 DataAdapter 和 DataSet	333	第 23 章 使用事务和强类型的 DataSet	394
20.1 DataAdapter	333	23.1 事务处理	394
20.1.1 DataAdapter 构成	333	23.1.1 简单的事务处理范例	394
20.1.2 创建 DataAdapter	333	23.1.2 使用 DataAdapter 时的事务	397
20.1.3 获得查询结果	335	23.1.3 隔离级别	397
20.1.4 映射查询结果	337	23.1.4 保存点	398
20.2 使用 DataSet	339	23.1.5 嵌套事务	399
20.2.1 DataSet 构成	339	23.2 强类型的 DataSet	399
20.2.2 DataSet 操作	340	23.2.1 优点/缺点	399
20.3 使用 DataTable	341	23.2.2 创建强类型的 DataSet	400
20.3.1 定义列	341	23.2.3 查看强类型 DataSet 的 .pas 文件	401
20.3.2 定义主键	342	23.2.4 使用强类型的 DataSet	407
20.3.3 使用约束	343	第 24 章 Borland 数据提供器	408
20.3.4 使用 DataRelation	345	24.1 体系结构概述	408
20.3.5 操作数据——使用 DataRow	347	24.2 Borland 数据提供器类	409
20.3.6 搜索、排序和过滤数据	349	24.2.1 BdpConnection	409
第 21 章 使用 WinForm——DataView 和数据绑定	351	24.2.2 BdpCommand	410
21.1 使用 DataView 和 DataViewManager 显示数据	351	24.2.3 BdpDataReader	411
21.1.1 DataView 类	351	24.2.4 BdpDataAdapter	411
21.1.2 DataViewManager 类	352	24.2.5 BdpParameter/BdpParameter-Collection	412
21.1.3 使用 DataView 和 DataViewManager 的项目范例	352	24.2.6 BdpTransaction	413
21.2 数据绑定	361	24.3 IDE 内的设计器	414
21.2.1 数据绑定接口	361	24.3.1 连接编辑器	414
21.2.2 简单绑定和复杂绑定	361	24.3.2 命令文本编辑器	414
21.2.3 WinForm 数据绑定类	361	24.3.3 参数集合编辑器	414
21.2.4 构建数据绑定的 Windows 对话框		24.3.4 Data Adapter Configuration 对话框	414

第五部分 用 ASP.NET 进行 因特网开发

第 25 章 ASP.NET 基础	418	26.5 在 Web 窗体之间导航	445
25.1 Web 技术	418	26.5.1 通过 POST 传递数据	445
25.1.1 HTTP 协议概述	418	26.5.2 使用 Response.Redirect() 和 QueryString	445
25.1.2 HTTP 请求包	419	26.5.3 使用 Server.Transfer() 方法	446
25.1.3 HTTP 响应包	419	26.5.4 使用会话变量	447
25.2 ASP.NET	420	26.6 提示和技巧	448
25.2.1 简单的 Web 应用程序	420	26.6.1 使用面板控件进行多窗体模拟	448
25.2.2 ASP.NET 页面结构	421	26.6.2 从客户上传文件	449
25.2.3 事件驱动的通信	423	26.6.3 从窗体发送电子邮件响应	450
25.2.4 VIEWSTATE 和状态维护	424	26.6.4 显示图像	452
25.2.5 CodeBehind	424	26.6.5 动态地添加控件——基于 缩略图的图像浏览器	453
25.3 ASP.NET 类	425	第 27 章 构造数据库驱动的 ASP. NET 应用程序	455
25.3.1 HTTPResponse 类	425	27.1 数据绑定	455
25.3.2 HTTPRequest 类	428	27.1.1 简单数据绑定	455
25.3.3 HTTPCookie 类	429	27.1.2 复杂数据绑定	459
25.3.4 处理回送事件	430	27.2 数据绑定列表控件	459
第 26 章 构造 ASP.NET Web 页面	431	27.2.1 CheckBoxList 控件	459
26.1 使用 ASP.NET 控件构造 Web 页面	431	27.2.2 DropDownList 控件	461
26.1.1 下载请求窗体范例	432	27.2.3 ListBox 控件	462
26.1.2 页面布局	432	27.2.4 RadioButtonList 控件	463
26.1.3 创建窗体	433	27.3 数据绑定重复控件	465
26.1.4 处理装载事件	433	27.3.1 Repeater 控件	465
26.1.5 从 ASP.NET 应用程序内保存 文件	434	27.3.2 DataList 控件	468
26.1.6 Web 窗体的事件处理顺序	435	27.4 使用数据网格	472
26.2 预填充列表框控件	436	27.4.1 数据网格分页	473
26.3 执行 Web 窗体检验	437	27.4.2 编辑数据网格	475
26.3.1 客户端和服务端端的检验	437	27.4.3 把条目添加到数据网格	480
26.3.2 BaseValidator 类	437	27.4.4 数据网格排序	480
26.3.3 RequiredFieldValidator	438	27.5 数据库驱动的下载请求窗体 和管理程序	481
26.3.4 CompareValidator	438	第 28 章 建立 Web 服务	486
26.3.5 RegularExpressionValidator	440	28.1 Web 服务的相关术语	486
26.3.6 RangeValidator	441	28.2 构造 Web 服务	486
26.3.7 CustomValidator	441	28.2.1 [WebService] 特性	490
26.3.8 ValidationSummary	442	28.2.2 从 Web 服务返回数据	490
26.4 Web 窗体格式化	442	28.2.3 [WebMethod] 特性讲解	491
26.4.1 WebControl 的强类型属性	442	28.3 使用 Web 服务	493
26.4.2 层叠式样式表	443	28.3.1 发现过程	493
26.4.3 使用 Style 类	444	28.3.2 构建代理类	493

28.3.3 使用代理类	494	30.2 跟踪消息	522
28.3.4 使用来自 Web 服务的数据集	497	30.3 分析 SOAP 包	524
28.3.5 调用异步的 Web 服务方法	499	30.4 客户激活	525
28.4 Web 服务的安全	500	30.4.1 工厂模式	526
第 29 章 .NET Remoting 和 Delphi	503	30.4.2 运行时的范例	531
29.1 现在可用的 Remoting 技术	503	30.4.3 CAO 的问题	531
29.1.1 套接字	503	30.5 生存周期管理	532
29.1.2 RPC	503	30.6 租用更新失败	534
29.1.3 Java RMI	504	30.7 配置文件	534
29.1.4 CORBA	504	30.7.1 服务器配置	536
29.1.5 XML-RPC	504	30.7.2 客户配置	537
29.1.6 DCOM	504	30.8 从 HTTP 切换到 TCP 通信	542
29.1.7 COM-Interop	504	30.9 从 SOAP 切换到二进制 Remoting	542
29.1.8 SOAP	504	30.10 SOAP 和二进制编码之间的区别	543
29.1.9 .NET Remoting	505	第 31 章 保护 ASP.NET 应用程序的	
29.2 分布式体系结构	505	安全	545
29.2.1 客户/服务器	505	31.1 ASP.NET 安全方法	545
29.2.2 对等	506	31.2 验证	545
29.2.3 多层	506	31.2.1 配置 ASP.NET 的验证模型	545
29.3 多层应用程序开发的优势	507	31.2.2 Windows 验证	545
29.3.1 可伸缩性和容错性	507	31.2.3 基于窗体的验证	547
29.3.2 开发和部署	508	31.2.4 护照验证	553
29.3.3 安全	508	31.3 授权	553
29.4 .NET Remoting 基础	508	31.3.1 文件授权	553
29.4.1 体系结构概述	508	31.3.2 URL 授权——< authentication >	
29.4.2 应用程序域	508	部分	554
29.4.3 System.Runtime.Remoting 名字		31.3.3 基于角色的授权	554
空间	509	31.3.4 冒名	556
29.4.4 可远程化的对象	510	31.4 注销	557
29.4.5 对象激活	511	第 32 章 ASP.NET 的部署和配置	558
29.4.6 租用和发起者	512	32.1 部署 ASP.NET 应用程序	558
29.4.7 代理	512	32.1.1 简单部署的考虑因素	558
29.4.8 通道	512	32.1.2 XCOPY 部署	560
29.5 第一个 .NET Remoting 应用程序	513	32.2 配置设置	561
29.5.1 建立项目	513	32.2.1 machine.config 文件	561
29.5.2 添加引用	514	32.2.2 web.config 文件	561
29.5.3 BankPackage.dll: 客户和服务		32.3 配置技巧	566
器之间的合约	514	32.3.1 处理错误重定向	566
29.5.4 实现服务器	516	32.3.2 工作者进程重启	566
29.5.5 实现客户	519	32.3.3 为性能进行输出缓冲	567
第 30 章 使用 .NET Remoting	522	32.3.4 监视 ASP.NET 进程	568
30.1 模板项目	522	32.3.5 跟踪应用程序	569

32.4	添加/获得定制的配置设置	571	33.2.7	应用程序状态管理	590
32.4.1	添加和读取 < appSettings >	571	33.2.8	使用 Cache 和应用程序状态的 比较	591
32.4.2	添加和读取定制的配置部分	571	第 34 章	开发定制的 ASP.NET 服务器 控件	592
第 33 章	在 ASP.NET 应用程序中 进行缓存和管理状态	573	34.1	用户控件	592
33.1	缓存 ASP.NET 应用程序	573	34.1.1	一个非常简单的用户控件	593
33.1.1	页面缓存	573	34.1.2	查看简单的控件	594
33.1.2	页面片段缓存	576	34.1.3	用户登录用户控件	596
33.1.3	数据缓存	577	34.2	Web 控件	597
33.1.4	缓存文件依赖性	579	34.2.1	构建一个非常简单的 Web 控件	597
33.1.5	扩展文件依赖性以便在 SQL Server 中使用	580	34.2.2	持久化值	599
33.1.6	缓存回调方法	581	34.2.3	添加一些定制的显示	601
33.2	ASP.NET 应用程序中的状态管理	583	34.2.4	确定 HTML 块类型	603
33.2.1	使用 cookie 管理状态	583	34.2.5	处理回送数据	604
33.2.2	使用 ViewState	585	34.2.6	TPostBackInputWebControl	605
33.2.3	会话状态管理	586	34.2.7	合成控件	608
33.2.4	在会话状态服务器中存储会话 数据	588	34.2.8	实现合成控件——TNewUser- InfoControl	608
33.2.5	在 SQL Server 中存储会话	588			
33.2.6	会话事件	589			