

**Borland**  
INPRISE  
核心技术丛书

Steve Teixeira 著  
Xavier Pacheco  
龙劲松 王瑜 谢尚书 译

Delphi 6 Developer's Guide

# Delphi 6

## 开发人员指南

Delphi 6  
开发指南

作者：Thomas  
作者：Pavlov  
ISBN: 0-01-000000-0

Delphi 6 Development Guide

# Delphi 6

## 开发人员指南

McGraw-Hill

0-01-000000-0

Borland/Inprise 核心技术丛书

# Delphi 6 开发人员指南

(美) Steve Teixeira 著  
Xavier Pacheco

龙劲松 王 瑜 谢尚书 译



机械工业出版社  
China Machine Press

本书是讲述 Delphi 最经典的著作之一，作为 Delphi 的新版本，Delphi 6 不仅能帮助程序员高效开发 Windows 应用程序，简化 Web 服务、中间软件以及后台数据库系统的合成，还是目前惟一全面支持所有主流业界标准的开发工具和提高电子商务能力的利器。本书内容丰富、条理清晰，用深入浅出的语言阐述了 Delphi 6 的精髓，对 Delphi 或即将移植到 Delphi 的程序员来说，具有极大的参考价值。本书最后还介绍了 Delphi 在 Internet 方面的应用。在移动商务网络编程越来越热的今天，无疑会成为程序员关注的焦点。

Steve Teixeira and Xavier Pacheco: Delphi6 Developer's Guide.

Authorized translation from the English language edition published by Sams Publishing, an imprint of Macmillan Computer Publishing U.S.A.

Copyright © 2002 by Sams Publishing.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2002 by China Machine Press.

版权所有，侵权必究。

本书版权登记号：图字：01-2002-0607

### 图书在版编目（CIP）数据

Delphi 6 开发人员指南/（美）泰克塞亚（Teixeira, S.），（美）帕齐科（Pacheco, X.）著；龙劲松等译 - 北京：机械工业出版社，2002.10

（Borland/Inprise 核心技术丛书）

书名原文：Delphi 6 Developer's Guide

ISBN 7-111-10968-6

I. D… II. ①泰…②帕…③龙… III. Delphi 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2002）第 076048 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：周 睿

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2003 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·54 印张

印数：0001-3000 册

定价：118.00 元（附光盘）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

## 译者序

作为 Delphi 爱好者，我们一直关注着它的发展，2001 年 6 月 12 日，Borland 软件公司在京发布了 Windows 快速应用开发环境的新版本 Borland Delphi 6。它不仅能帮助程序员高效开发 Windows 应用程序、简化 Web 服务器、中间件以及后台数据库系统的集成。还是目前惟一全面支持所有主流业界标准（XML、SOAP、WSDL 和 XSL 等）的开发工具。与此同时，它还支持基于 Web 服务的 Microsoft.NET 和 SunONE 体系，为 Web 开发人员提供了必要的伸缩性和可靠性。可以这样说，Delphi 6 是提高电子商务能力的利器。

了解 Delphi 6 的新特性之后，我们非常高兴能承担本书的翻译工作。本书的前一版《Delphi 5 开发人员指南》是世界上最畅销的 Delphi 书籍之一，曾两次获得最佳 Delphi 图书读者选拔奖。在翻译过程中，我们的心情一直非常愉快，既能学习 Delphi 6 的新特性，又能借他山之石，丰富自己已有的编程知识。本书内容丰富，条理清晰，用深入浅出的语言阐述了 Delphi 6 的精髓，对 Delphi 或即将移植到 Delphi 的程序员来说，无疑具有指导意义。尤其值得称道的是，本书最后还详细介绍了 Delphi 在 Internet 方面的应用。在移动商务的网络编程越来越流行的今天，这无疑会成为程序员关注的焦点。

在翻译过程中，我们在保持原意的基础上，尽量保持语句流畅、行文优美，并出于爱好，对其中部分源代码进行了调试（当然，这个过程是非常惬意的）。由于时间过于仓促，疏漏之处在所难免，敬请广大读者指正，并欢迎大家与我们讨论 Delphi。

译者

2002 年春于北京

# 序 言

“Delphi 6, 历经两年精心打造, 使您开发永远高效。”

我很荣幸为 Borland 公司服务了 16 年。在 1985 年夏天, 我开始在这里工作, 成为了新一代编程工具 (UCSD Pascal 系统和命令行工具已不能满足需要) 开发者中的一分子, 我们的工作提高了程序员的编程效率, 延长了程序员的生命 (本人也包括在内)。在过去的 18 年中, 我们一直在努力创新, 同时致力于提高编程者的技术。我很荣幸成为全球范围内 Borland 大家庭中的一员。

Turbo Pascal 1.0 永远地改变了编程工具的面貌, 在 1983 年确定了它的标准。Delphi 也再次改变了编程工具的面貌。Delphi 1.0 的重点在于面向对象编程、面向 Windows 编程并让数据库开发更为简捷。Delphi 的后期版本致力于让 Internet 和分布式应用程序开发变得更为简单。虽然多年来在产品中加入了众多的特性, 编写了大量的文档和数兆的在线帮助, 但是程序员要编写出成功的项目, 还是需要了解其他方面大量的信息、知识和忠告。

你们是如何超越全球称誉的 Delphi 5 的? Delphi 5 是不是在简化了 Internet 程序开发和分步式应用程序的同时, 也提高了 Delphi 程序员的效率? Delphi 开发小组能不能够再次超越自我满足当今的需要, 并且适合明天的开发人员?

新千年伊始, Delphi 开发小组耗费两年时间来倾听用户的声音、观察开发人员是如何使用该产品的以及寻找编程中的棘手之处。他们致力于从根本上简化开发下一代电子商务 Web 应用程序的过程, 其中包括基于 XML/SOAP 的 Web 服务、B2B/B2C/P2P 应用程序集成、跨平台应用程序、集成 AppServer/EJB 的分布式应用程序以及 Microsoft Windows ME/2000 和 Office 2000 的应用程序。

Steve Teixeira 和 Xavier Pacheco 再次联手合作, 为我们奉献了开发指南的新版本, 读者将从中获益匪浅, 本书从深度和广度上阐述了 Delphi 6 编程技术。

我很早就认识 Steve Teixeira (有人称他 T-Rex) 和 Xavier Pacheco (有人仅仅用 X 来称呼他), 他们是我多年的老友、同事、年会发言人和 Borland 大家庭中的成员。

他们上一个版本的开发指南曾经受到全世界的 Delphi 开发者的好评。现在他们带来了新的版本, 相信读者会喜欢此书。

有爱好才会去钻研。在此希望读者的 Delphi 项目开发能够愉快、成功并受到赞赏!

David Intersimone(David I)

客户关系部副部长

Borland 软件公司

davidi@borland.com

# 前 言

本书至今已增订到第 5 版，在 7 年的时间中，本书不断完善，为了写作本书，很多人对此付出了相当多的精力。Xavier 和 Steve 是最早的 Borland Delphi 开发小组成员，本书是他们结合 15 年的 Delphi 开发经验写成的。在本书中，我们可以感受到作者的努力精神，正是这种精神，让 Delphi 开发丛书成为世界上最畅销的 Delphi 书籍，并两次获得 Delphi 读者最佳选择奖。这本书由开发人员撰写，同时也供开发人员阅读。

本书建立在《Delphi 5 开发人员指南》基础之上，理想状况下，我们希望本书包括《Delphi 5 开发人员指南》的全部章节和所有新内容。但是《Delphi 5 开发人员指南》一书本身太厚，为了给 Delphi 6 新特性的讨论留下足够空间，我们只抽取了其中的部分章节，再加上 Delphi 6 的新特性讨论，新书就这样展现在大家面前。

本书包括了一些全新章节，许多章节是《Delphi 5 开发人员指南》对应章节的扩展。但在《Delphi 5 开发人员指南》中介绍的内容还是极其有用的。因此我们在本书配套光盘中附带了整个电子版的《Delphi 5 开发人员指南》，每个章节单独地用 PDF 格式存储。在电子版的封面中包含了《Delphi 5 开发人员指南》的目录表，读者可以在其中查找感兴趣的章节。对读者而言，可谓一举两得。

本书分成六部分。第一部分，“基础知识”，讲述了 Delphi 编程必要的基础知识。第二部分，“高级技术”，讨论了一些通用的高级开发主题，例如线程和动态链接库。第三部分，“数据库开发”，从多个层面上讨论了 Delphi 数据库访问主题。第四部分，“基于组件的开发”，讨论了关于 VCL、CLX、包、COM 和 Open Tools API 开发问题。第五部分，“企业开发”，其中讨论了开发企业级应用程序需要的实践知识，包括 COM+、COBRA、SOAP/BizSnap 和 DataSnap。最后，第六部分，“Internet 开发”，演示了使用 Delphi 开发 Internet 和无线应用程序的技术。

## 本书面向的读者

书如其名，本书是面向开发人员的。如果读者是一位开发人员，并且使用 Delphi，那就需要拥有一本这样的书。本书面向 3 类读者：

- 希望将技术提高一个层次的 Delphi 开发人员。
- 熟悉 Pascal、C/C++、Java 或 Basic，并希望开始使用 Delphi 的程序员。
- 希望最大限度利用 Delphi 来研究其中先进特性和隐含特性的程序员。

## 本书使用的约定

本书使用了下列约定：

- 代码行、命令、声明、变量、程序输出以及任何屏幕内容都用计算机打印字体显示。
- 需要读者输入的内容用粗体显示。
- 语法描述中的占位符用斜体表示。
- 技术术语在文章中第一次出现或者用于强调重点时，用斜体表示。
- 程序或过程用在其名后加上一对圆括号表示。虽然这不是标准 Pascal 风格，但是有助于和属性、变量和类区分。

各章都有一些注意、提示和警告，为了方便阅读，用特殊的字体标出。

本书配套光盘中有书中所有的源代码和项目文件，同时还有一些本书中尚未提到的源代码示例。此外,光盘中还包括一些第三方组件和工具的使用版。

## Delphi 6 开发指南网站

请访问网站 <http://www.xapware.com/ddg> 并且加入《Delphi 6 开发人员指南》会员，从而得到本书的额外信息。读者也可以加入我们新闻邮件的邮件列表，并访问我们的讨论组。

## 写作本书的缘由

有人会问，是什么驱使我们编写本书的。这很难解释，但是当看到其他开发人员仍然将标满了注解、且被翻阅得破烂不堪的《Delphi 5 开发人员指南》视为至宝时，就感到一切付出都是值得的。

完成本书的写作后，现在我们可以稍微轻松下来并愉快地编写一些 Delphi 程序。本书开始讲述的节奏比较慢，但是会迅速深入到 Delphi 的高级主题中，不过读者应该能够适应这种节奏。如果读者意识到了这一点，说明您已经通读了本书，并且具有了足够的 Delphi 方面的知识与技术，也许已经可以被称作 Delphi 专家了。

# 作者介绍

## 主要作者介绍

**Steve Teixeira** 是 Zone Labs 的核心技术主管, Internet 安全解决方案首席设计师。Steve 以前担任过移动无线软件公司 ThinSpace 的首席技术官, 并且在 Full Moon Interactive 担任过全职电子商务开发人员。在 Borland 担任研究开发软件工程师期间, Steve 在 Delphi 和 C++ Builder 开发中作出了巨大的贡献。Steve 是 4 本获奖畅销书的作者, 并且在软件开发杂志上发表过许多文章, 他的作品被译成了多国文字。Steve 经常在行业会议上发表演讲。

**Xavier Pacheco** 是 Xapware 技术公司的总裁, 该公司是一个软件开发咨询公司。Xavier 也经常担任行业会议发言人, 并且是 Delphi 期刊的资深作者。Xaiver 是全球公认的 Delphi 专家和 Borland 指定的 TeamB 团体志愿人员。他是 4 本获奖畅销书的作者, 其作品被译成了多种语言。Xavier 和他的妻子 Annie 以及两个孩子 Amanda 和 Zachary 住在 Colorado Springs。

## 其他作者介绍

**Bob Swart (Dr.Bob, [www.drbob42.com](http://www.drbob42.com))** 是英国 Borland 协会成员, 独立技术作者, 荷兰 Helmond 的 Delphi、Kylix、C++Builder 技术顾问和培训师。Bob 在 Delphi Magazine、Delphi Developer、UK-BUG Developer's Magazine 等杂志担任专栏作者, 同时也在 DevX、TechRepublic 和 Borland Community Web 网站上发表文章。Bob 为《The Revolutionary Guide to Delphi 2》、《Delphi 4 Unleashed》、《C++Builder 4 Unleashed》、《C++Builder 5 Developer's Guide》、《Kylix Developer's Guide》和现在的《Delphi 6 Developer's Guide》(Sams 出版社) 都撰写了部分章节。

Bob 经常在 Borland 和 Delphi/Kylix 相关研讨会上担任发言人, 并且为他自己的 Dr. Bob's Delphi 培训班(在英国和荷兰)撰写培训教材。

业余时间, Bob 喜欢和他的儿子 Eric Mark Pascal 和女儿 Natasha Louise Delphine 一起看“Star Trek Voyager”和“Deep Space Nine”的录影带。

**Dan Miser** 是 Borland DSP 小组的研究开发项目经理, 他花了大量的时间去钻研技术。Dan 在 Delphi 的研究开发小组负责 DataSnap 的开发。Dan 的主要精力在于研究分布式计算技术, 包括 MIDAS、SOAP、DCOM、RMI、J2EE、EJB、Struts 和 RDS。他也致力于 Delphi 的应用推广, 他担任技术编辑, 在杂志上撰写有关文章, 作为 TeamB 的成员参与 Borland 新闻组, 在 BorCon 会议的有关 COM 和 MIDAS 主题上担任发言人。

**David Sampson** 是 Borland RAD Tools Group 的研发工程师, 负责集成 CORBA 到 RAD 产品中。他是资深的 Pascal、Delphi 和 C++开发人员, 经常在 Borland 开发人员会议上担当发

## VIII

言人。他和妻子住在佐治亚的罗斯威尔，爱好曲棍球，有空还帮妻子遛他们的 **Basenjjs** 小狗。

**Nick Hodges** 是圣保罗市 **Lemanix** 公司的一名资深开发工程师。他是 **Borland TeamB** 小组成员，有着长期的 **Pascal** 和 **Delphi** 开发经验。他为 **Borland** 协会咨询部服务，并且多次担任会议发言人，是 **Borland** 联合网站的老作者。他和妻子带着两个孩子住在圣保罗。喜欢阅读、跑步，帮妻子照顾孩子。

**Ray Konopka** 是 **Raize Software** 公司的创始人，是 **CodeSite** 和 **Raize** 组件的主要设计师。**Ray** 是受到好评的《*Developing Custom Delphi Components*》一书的作者，是《*Visual Developer magazine*》杂志“**Delphi by Design**”的专栏作者。**Ray** 的专长是用户界面设计和 **Delphi** 组件开发，在全球的一些开发者协会中经常担当发言人。

# 目 录

|      |  |
|------|--|
| 译者序  |  |
| 序言   |  |
| 前言   |  |
| 作者介绍 |  |

## 第一部分 基 本 知 识

|                       |    |
|-----------------------|----|
| 第 1 章 Delphi 编程简介     | 1  |
| 1.1 Delphi 的产品家族介绍    | 2  |
| 1.2 认识 Delphi         | 3  |
| 1.2.1 可视化开发环境的特性      | 4  |
| 1.2.2 编译器速度和编译后代码执行效率 | 5  |
| 1.2.3 编程语言的功能和复杂性的对立  | 5  |
| 1.2.4 数据库结构的灵活性和可扩展性  | 6  |
| 1.2.5 实现机制对设计和使用模式的增强 | 7  |
| 1.3 历史回顾              | 7  |
| 1.3.1 Delphi 1        | 7  |
| 1.3.2 Delphi 2        | 8  |
| 1.3.3 Delphi 3        | 8  |
| 1.3.4 Delphi 4        | 9  |
| 1.3.5 Delphi 5        | 9  |
| 1.3.6 Delphi 6        | 10 |
| 1.4 Delphi IDE 环境     | 10 |
| 1.4.1 主窗口             | 10 |
| 1.4.2 主菜单             | 11 |
| 1.4.3 Delphi 工具栏      | 11 |
| 1.4.4 组件选项板           | 12 |
| 1.4.5 窗体设计器           | 12 |
| 1.4.6 对象观察器           | 12 |
| 1.4.7 代码编辑器           | 13 |

|                        |    |
|------------------------|----|
| 1.4.8 代码浏览器            | 13 |
| 1.4.9 对象层次图            | 14 |
| 1.5 项目源文件概述            | 14 |
| 1.6 应用程序简介             | 16 |
| 1.7 事件处理机制的优势          | 17 |
| 1.8 快速原型化              | 18 |
| 1.9 可扩展的组件和环境          | 19 |
| 1.10 必须掌握的 10 种 IDE 功能 | 19 |
| 1.11 小结                | 22 |
| 第 2 章 Object Pascal 语言 | 23 |
| 2.1 注释                 | 24 |
| 2.2 扩展过程和函数特性          | 24 |
| 2.2.1 函数调用中的圆括号        | 24 |
| 2.2.2 重载               | 24 |
| 2.2.3 缺省参数值            | 25 |
| 2.3 变量                 | 26 |
| 2.4 常量                 | 27 |
| 2.5 运算符                | 28 |
| 2.5.1 赋值运算符            | 29 |
| 2.5.2 比较运算符            | 29 |
| 2.5.3 逻辑运算符            | 29 |
| 2.5.4 数学运算符            | 30 |
| 2.5.5 按位运算符            | 31 |
| 2.5.6 增/减量运算           | 31 |
| 2.5.7 “计算-赋值”运算符       | 32 |
| 2.6 Object Pascal 类型   | 32 |
| 2.6.1 类型比较             | 32 |
| 2.6.2 字符类型             | 33 |
| 2.6.3 字符串              | 34 |
| 2.6.4 变体类型             | 43 |
| 2.6.5 货币类型             | 52 |

|                               |    |
|-------------------------------|----|
| 2.7 用户自定义类型 .....             | 53 |
| 2.7.1 数组 .....                | 53 |
| 2.7.2 动态数组 .....              | 54 |
| 2.7.3 记录 .....                | 55 |
| 2.7.4 集合 .....                | 56 |
| 2.7.5 对象 .....                | 58 |
| 2.7.6 指针 .....                | 59 |
| 2.7.7 类型别名 .....              | 60 |
| 2.8 类型转换和强制类型转换 .....         | 61 |
| 2.9 字符串资源 .....               | 62 |
| 2.10 条件判断语句 .....             | 62 |
| 2.10.1 if 语句 .....            | 63 |
| 2.10.2 case 语句 .....          | 63 |
| 2.11 循环 .....                 | 64 |
| 2.11.1 for 循环 .....           | 64 |
| 2.11.2 while 循环 .....         | 65 |
| 2.11.3 repeat..until 循环 ..... | 65 |
| 2.11.4 Break()过程 .....        | 65 |
| 2.11.5 Continue()过程 .....     | 66 |
| 2.12 过程和函数 .....              | 66 |
| 2.13 作用域 .....                | 70 |
| 2.14 单元 .....                 | 71 |
| 2.14.1 uses 子句 .....          | 72 |
| 2.14.2 单元循环引用 .....           | 72 |
| 2.15 包 .....                  | 73 |
| 2.15.1 使用 Delphi 包 .....      | 73 |
| 2.15.2 包的语法格式 .....           | 73 |
| 2.16 面向对象编程 .....             | 74 |
| 2.17 使用 Delphi 对象 .....       | 76 |
| 2.17.1 声明和实例化 .....           | 76 |
| 2.17.2 析构 .....               | 76 |
| 2.17.3 方法 .....               | 77 |
| 2.17.4 方法的类型 .....            | 78 |
| 2.17.5 属性 .....               | 80 |
| 2.17.6 可见区域说明符 .....          | 80 |
| 2.17.7 友元类 .....              | 81 |

|                                                    |     |
|----------------------------------------------------|-----|
| 2.17.8 对象的秘密 .....                                 | 82  |
| 2.17.9 TObject: 所有对象的祖先 .....                      | 82  |
| 2.17.10 接口 .....                                   | 83  |
| 2.18 结构化异常处理 .....                                 | 86  |
| 2.18.1 异常类 .....                                   | 88  |
| 2.18.2 异常执行流程 .....                                | 90  |
| 2.18.3 异常的再激活 .....                                | 91  |
| 2.19 运行时信息 .....                                   | 92  |
| 2.20 小结 .....                                      | 93  |
| 第 3 章 理解 Windows 消息 .....                          | 94  |
| 3.1 什么是消息 .....                                    | 94  |
| 3.2 消息的类型 .....                                    | 95  |
| 3.3 Windows 消息系统的工作方式 .....                        | 95  |
| 3.4 Delphi 的消息系统 .....                             | 96  |
| 3.5 消息的处理 .....                                    | 98  |
| 3.5.1 消息处理: 并非无约定 .....                            | 100 |
| 3.5.2 为 Result 消息值赋值 .....                         | 100 |
| 3.5.3 TApplication 类型的 OnMessage<br>事件 .....       | 101 |
| 3.6 发送自己的消息 .....                                  | 101 |
| 3.6.1 Perform()方法 .....                            | 102 |
| 3.6.2 SendMessage()和 PostMessage() API<br>函数 ..... | 102 |
| 3.7 非标准消息 .....                                    | 102 |
| 3.7.1 通知消息 .....                                   | 103 |
| 3.7.2 内部 VCL 消息 .....                              | 104 |
| 3.7.3 用户定义消息 .....                                 | 104 |
| 3.8 剖析 VCL 的消息系统 .....                             | 106 |
| 3.9 消息和事件之间的关系 .....                               | 112 |
| 3.10 小结 .....                                      | 112 |

## 第二部分 高级技术

|                     |     |
|---------------------|-----|
| 第 4 章 编写可移植代码 ..... | 113 |
| 4.1 版本兼容的共性 .....   | 113 |
| 4.1.1 版本判断 .....    | 114 |
| 4.1.2 单元、组件和包 ..... | 115 |

|                                     |     |                                  |     |
|-------------------------------------|-----|----------------------------------|-----|
| 4.1.3 关于 IDE 的讨论 .....              | 115 | 5.1.3 线程的滥用 .....                | 126 |
| 4.2 Delphi 和 Kylix 兼容性 .....        | 116 | 5.2 TThread 对象 .....             | 127 |
| 4.2.1 Linux 中不具备的特性 .....           | 116 | 5.2.1 TThread 基础 .....           | 127 |
| 4.2.2 编译器和语言特性 .....                | 117 | 5.2.2 线程实例 .....                 | 130 |
| 4.2.3 平台相关特点 .....                  | 117 | 5.2.3 线程终止 .....                 | 130 |
| 4.3 Delphi 6 的新特性 .....             | 118 | 5.2.4 和 VCL 同步 .....             | 131 |
| 4.3.1 变体变量 .....                    | 118 | 5.2.5 应用程序 Demo .....            | 134 |
| 4.3.2 枚举类型 .....                    | 118 | 5.2.6 优先级和调度 .....               | 135 |
| 4.3.3 \$IF 指示符 .....                | 118 | 5.2.7 线程的挂起和恢复 .....             | 137 |
| 4.3.4 潜在的二进制 DFM 的不兼容 .....         | 118 | 5.2.8 线程中的计时 .....               | 137 |
| 4.4 从 Delphi 5 升级 .....             | 118 | 5.3 管理多线程 .....                  | 139 |
| 4.4.1 可写的类型常量 .....                 | 119 | 5.3.1 线程局部存储 .....               | 139 |
| 4.4.2 Cardinal 一元取反操作 .....         | 119 | 5.3.2 线程同步 .....                 | 142 |
| 4.5 从 Delphi 4 升级 .....             | 119 | 5.4 多线程应用程序范例 .....              | 152 |
| 4.5.1 RTL 问题 .....                  | 119 | 5.4.1 用户界面 .....                 | 152 |
| 4.5.2 VCL 问题 .....                  | 119 | 5.4.2 搜索线程 .....                 | 158 |
| 4.5.3 Internet 开发主题 .....           | 120 | 5.4.3 调整优先级 .....                | 162 |
| 4.5.4 数据库问题 .....                   | 120 | 5.5 多线程访问 BDE .....              | 164 |
| 4.6 从 Delphi 3 升级 .....             | 120 | 5.6 多线程图形操作 .....                | 168 |
| 4.6.1 无符号 32 位整数 .....              | 120 | 5.7 纤程 .....                     | 172 |
| 4.6.2 64 位整数 .....                  | 122 | 5.8 小结 .....                     | 177 |
| 4.6.3 Real 类型 .....                 | 122 | 第 6 章 动态链接库 .....                | 178 |
| 4.7 从 Delphi 2 升级 .....             | 122 | 6.1 何谓 DLL .....                 | 178 |
| 4.7.1 Boolean 类型的改变 .....           | 122 | 6.2 静态链接和动态链接 .....              | 180 |
| 4.7.2 ResourceString .....          | 123 | 6.3 为何要用 DLL .....               | 181 |
| 4.7.3 RTL 的改变 .....                 | 123 | 6.3.1 多个应用程序共享代码、资源和<br>数据 ..... | 181 |
| 4.7.4 TCustomForm .....             | 123 | 6.3.2 隐藏实施细节 .....               | 182 |
| 4.7.5 GetChildren() .....           | 124 | 6.4 创建和使用 DLL .....              | 182 |
| 4.7.6 自动化服务器 .....                  | 124 | 6.4.1 数美分 (简单的 DLL) .....        | 182 |
| 4.8 从 Delphi 1 升级 .....             | 124 | 6.4.2 通过 DLL 显示模式窗体 .....        | 185 |
| 4.9 小结 .....                        | 124 | 6.5 通过 DLL 显示无模式窗体 .....         | 187 |
| 第 5 章 多线程技术 .....                   | 125 | 6.6 在 Delphi 应用程序中使用 DLL .....   | 188 |
| 5.1 阐述线程 .....                      | 125 | 6.7 显式调用 DLL .....               | 190 |
| 5.1.1 多任务的类型 .....                  | 125 | 6.8 动态链接库的入口/出口函数 .....          | 192 |
| 5.1.2 在 Delphi 应用程序中使用多<br>线程 ..... | 126 | 6.8.1 进程和线程的初始化和结束标准             |     |

|                                 |     |                                 |     |
|---------------------------------|-----|---------------------------------|-----|
| 函数 .....                        | 192 | 7.5.9 使用数据模块 .....              | 243 |
| 6.8.2 DLL 入口 / 出口示例 .....       | 193 | 7.5.10 查找、范围及过滤器演示 .....        | 243 |
| 6.9 DLL 中的异常 .....              | 196 | 7.5.11 书签 .....                 | 250 |
| 6.9.1 在 16 位 Delphi 中捕捉异常 ..... | 196 | 7.6 小结 .....                    | 251 |
| 6.9.2 异常和 Safecall 指示符 .....    | 197 | 第 8 章 用 dbExpress 进行数据库         |     |
| 6.10 回调函数 .....                 | 197 | 开发 .....                        | 252 |
| 6.10.1 使用回调函数 .....             | 199 | 8.1 使用 dbExpress .....          | 252 |
| 6.10.2 绘制 Owner-Draw 列表框 .....  | 200 | 8.1.1 单向、只读数据集 .....            | 252 |
| 6.11 通过 DLL 调用回调函数 .....        | 200 | 8.1.2 dbExpress 与 BDE 比较 .....  | 253 |
| 6.12 共享不同进程中的 DLL 数据 .....      | 202 | 8.1.3 dbExpress 支持跨平台开发 .....   | 253 |
| 6.12.1 创建共享内存的 DLL .....        | 203 | 8.2 dbExpress 组件 .....          | 253 |
| 6.12.2 使用共享内存的 DLL .....        | 206 | 8.2.1 TSQLConnection .....      | 253 |
| 6.13 从 DLL 中输出对象 .....          | 208 | 8.2.2 TSQLDataset .....         | 256 |
| 6.14 小结 .....                   | 212 | 8.2.3 显示查询结果 .....              | 256 |
| 第三部分 数据库开发                      |     | 8.2.4 向后兼容组件 .....              | 259 |
| 第 7 章 Delphi 数据库体系结构 .....      | 213 | 8.2.5 TSQLMonitor .....         | 259 |
| 7.1 数据库类型 .....                 | 213 | 8.3 设计可编辑的 dbExpress 应用程序 ..... | 259 |
| 7.2 数据库体系结构 .....               | 214 | 8.4 部署 dbExpress 应用程序 .....     | 260 |
| 7.3 连接到数据库服务器 .....             | 214 | 8.5 小结 .....                    | 261 |
| 7.3.1 数据库连接概述 .....             | 214 | 第 9 章 用 dbGo for ADO 进行数        |     |
| 7.3.2 建立数据库连接 .....             | 215 | 据库开发 .....                      | 262 |
| 7.4 操作数据集 .....                 | 215 | 9.1 dbGo 介绍 .....               | 262 |
| 7.4.1 打开及关闭数据集 .....            | 216 | 9.2 Microsoft 的统一数据访问策略         |     |
| 7.4.2 导航数据集 .....               | 219 | 概述 .....                        | 262 |
| 7.4.3 管理数据集 .....               | 223 | 9.3 OLE DB、ADO、ODBC 概述 .....    | 262 |
| 7.4.4 数据集状态 .....               | 226 | 9.4 使用 dbGo for ADO .....       | 263 |
| 7.5 操作字段 .....                  | 226 | 9.4.1 为 ODBC 建立 OLE DB          |     |
| 7.5.1 字段值 .....                 | 226 | 提供者 .....                       | 263 |
| 7.5.2 字段数据类型 .....              | 227 | 9.4.2 Access 数据库 .....          | 265 |
| 7.5.3 字段名称及号码 .....             | 228 | 9.5 dbGo for ADO 组件 .....       | 265 |
| 7.5.4 管理字段数据 .....              | 228 | 9.5.1 TADOConnection .....      | 265 |
| 7.5.5 操作 BLOB 字段 .....          | 233 | 9.5.2 建立数据库连接 .....             | 265 |
| 7.5.6 过滤数据 .....                | 238 | 9.5.3 回避/替代登录提示 .....           | 267 |
| 7.5.7 搜索数据集 .....               | 239 | 9.5.4 TADOCommand .....         | 269 |
| 7.5.8 关键字查找 .....               | 240 | 9.5.5 TADODataset .....         | 269 |
|                                 |     | 9.5.6 类似 BDE 的数据集组件 .....       | 270 |

|                                |     |                                   |     |
|--------------------------------|-----|-----------------------------------|-----|
| 9.6 事务处理 .....                 | 271 | 10.7 小结 .....                     | 308 |
| 9.7 小结 .....                   | 272 | 第 11 章 编写 VCL 组件 .....            | 309 |
| <b>第四部分 基于组件的开发</b>            |     | 11.1 组件设计基础 .....                 | 309 |
| 第 10 章 组件体系: VCL 和             |     | 11.1.1 决定是否应编写组件 .....            | 309 |
| CLX .....                      | 273 | 11.1.2 编写组件的步骤 .....              | 310 |
| 10.1 关于新的 CLX .....            | 274 | 11.1.3 确定祖先类 .....                | 311 |
| 10.2 何谓组件 .....                | 274 | 11.1.4 创建组件单元 .....               | 311 |
| 10.3 组件的层次结构 .....             | 275 | 11.1.5 创建属性 .....                 | 313 |
| 10.3.1 非可视组件 .....             | 276 | 11.1.6 创建事件 .....                 | 321 |
| 10.3.2 可视组件 .....              | 276 | 11.1.7 创建自定义方法 .....              | 325 |
| 10.4 组件结构 .....                | 278 | 11.1.8 构造函数和析构函数 .....            | 326 |
| 10.4.1 属性 .....                | 278 | 11.1.9 注册组件 .....                 | 328 |
| 10.4.2 属性的类型 .....             | 279 | 11.1.10 测试组件 .....                | 329 |
| 10.4.3 方法 .....                | 280 | 11.1.11 提供组件图标 .....              | 331 |
| 10.4.4 事件 .....                | 280 | 11.2 组件示例 .....                   | 331 |
| 10.4.5 流式属性 .....              | 281 | 11.2.1 扩展 Win32 组件包装器的            |     |
| 10.4.6 所有权 .....               | 282 | 性能 .....                          | 331 |
| 10.4.7 父子关系 .....              | 282 | 11.2.2 TddgRunButton: 创建属性 .....  | 340 |
| 10.5 可视组件的继承关系 .....           | 283 | 11.3 TddgButtonEdit: 一个容器组件 ..... | 345 |
| 10.5.1 TPersistent 类 .....     | 283 | 11.3.1 设计决策 .....                 | 345 |
| 10.5.2 TPersistent 方法 .....    | 283 | 11.3.2 显露属性 .....                 | 346 |
| 10.5.3 TComponent 类 .....      | 284 | 11.3.3 显露事件 .....                 | 346 |
| 10.5.4 TControl 类 .....        | 285 | 11.3.4 TddgDigitalClock: 创建组件     |     |
| 10.5.5 TWinControl 和 TWidget-  |     | 事件 .....                          | 348 |
| Control .....                  | 285 | 11.3.5 把窗体添加到组件面板 .....           | 351 |
| 10.5.6 TGraphicControl 类 ..... | 287 | 11.4 小结 .....                     | 353 |
| 10.5.7 TCustomControl 类 .....  | 287 | 第 12 章 高级 VCL 组件构造技术 .....        | 354 |
| 10.5.8 其他类 .....               | 287 | 12.1 伪可视组件 .....                  | 354 |
| 10.6 运行期类型信息 .....             | 290 | 12.1.1 扩展提示功能 .....               | 354 |
| 10.6.1 TypInfo.pas 单元: 运行类型信   |     | 12.1.2 创建一个 THintWindow 派         |     |
| 息的定义者 .....                    | 291 | 生类 .....                          | 354 |
| 10.6.2 获取类型信息 .....            | 293 | 12.1.3 椭圆型窗口 .....                | 356 |
| 10.6.3 获得方法指针的类型信息 .....       | 299 | 12.1.4 激活 THintWindow 派生窗口 .....  | 357 |
| 10.6.4 获取有序类型的类型信息 .....       | 302 | 12.1.5 运用 TDDGHintWindow .....    | 357 |
| 10.6.5 通过 RTTI 为属性赋值 .....     | 306 | 12.2 动态组件 .....                   | 357 |
|                                |     | 12.2.1 滚动字幕组件 .....               | 357 |

|                                                              |     |                                    |     |
|--------------------------------------------------------------|-----|------------------------------------|-----|
| 12.2.2 编写组件 .....                                            | 358 | 第 13 章 CLX 组件开发 .....              | 409 |
| 12.2.3 在屏幕以外的位图上绘制 .....                                     | 358 | 13.1 何谓 CLX .....                  | 409 |
| 12.2.4 描绘组件 .....                                            | 359 | 13.2 CLX 的体系结构 .....               | 410 |
| 12.2.5 让组件“动起来” .....                                        | 360 | 13.3 移植问题 .....                    | 412 |
| 12.2.6 测试 TddgMarquee 组件 .....                               | 368 | 13.4 组件示例 .....                    | 414 |
| 12.3 编写属性编辑器 .....                                           | 369 | 13.4.1 TddgSpinner 组件 .....        | 414 |
| 12.3.1 创建一个继承的属性编辑器<br>对象 .....                              | 370 | 13.4.2 设计期增强工具 .....               | 425 |
| 12.3.2 将属性作为文本进行编辑 .....                                     | 371 | 13.4.3 组件引用和图像列表 .....             | 430 |
| 12.3.3 注册新的属性编辑器 .....                                       | 374 | 13.4.4 CLX 数据感知组件 .....            | 435 |
| 12.3.4 利用对话框将属性作为整体进<br>行编辑 .....                            | 375 | 13.5 CLX 设计编辑器 .....               | 443 |
| 12.4 组件编辑器 .....                                             | 378 | 13.6 包 .....                       | 446 |
| 12.4.1 TComponentEditor .....                                | 378 | 13.6.1 命名约定 .....                  | 446 |
| 12.4.2 一个简单的组件 .....                                         | 379 | 13.6.2 运行期包 .....                  | 448 |
| 12.4.3 一个简单的组件编辑器 .....                                      | 380 | 13.6.3 设计期包 .....                  | 450 |
| 12.4.4 注册组件编辑器 .....                                         | 380 | 13.6.4 单元注册 .....                  | 452 |
| 12.5 对非公开的组件数据执行流操作 ..                                       | 382 | 13.6.5 组件位图 .....                  | 453 |
| 12.5.1 定义属性 .....                                            | 382 | 13.7 小结 .....                      | 454 |
| 12.5.2 DefineProperty()的例子 .....                             | 383 | 第 14 章 充分发挥包的作用 .....              | 455 |
| 12.5.3 TddgWaveFile: DefineBinary<br>Property()用法示例 .....    | 384 | 14.1 为何要用包 .....                   | 455 |
| 12.6 属性类别 .....                                              | 390 | 14.1.1 精简代码 .....                  | 456 |
| 12.6.1 类别类 .....                                             | 391 | 14.1.2 发布更小的应用程序——应用<br>程序分割 ..... | 456 |
| 12.6.2 自定义类别 .....                                           | 391 | 14.1.3 组件容器 .....                  | 456 |
| 12.7 组件列表: TCollection 和<br>TCollectionItem .....            | 394 | 14.2 为何不用包 .....                   | 456 |
| 12.7.1 定义 TCollectionItem 类:<br>TRunBtnItem .....            | 396 | 14.3 包的类型 .....                    | 456 |
| 12.7.2 定义 TCollection 类:<br>TRunButtons .....                | 396 | 14.4 包文件 .....                     | 457 |
| 12.7.3 实现 TddgLaunchPad、TRun-<br>BtnItem 和 TRunButtons ..... | 397 | 14.5 使用运行期包 .....                  | 457 |
| 12.7.4 用对话框属性编辑器编辑<br>TCollectionItem 组件的列表 ..               | 403 | 14.6 把包安装到 Delphi IDE 中 .....      | 457 |
| 12.8 小结 .....                                                | 408 | 14.7 创建包 .....                     | 458 |
|                                                              |     | 14.7.1 包编辑器 .....                  | 458 |
|                                                              |     | 14.7.2 包设计方案 .....                 | 459 |
|                                                              |     | 14.8 包的版本化 .....                   | 462 |
|                                                              |     | 14.9 包编译器指示符 .....                 | 462 |
|                                                              |     | 14.10 包的命名约定 .....                 | 463 |
|                                                              |     | 14.11 使用运行期(插件)包<br>的可扩展应用程序 ..... | 463 |

|                                              |     |                                  |     |
|----------------------------------------------|-----|----------------------------------|-----|
| 14.12 从包中导出函数 .....                          | 469 | 15.7.1 Automation 事件 .....       | 512 |
| 14.13 获取包的信息 .....                           | 471 | 15.7.2 Automation 集合 .....       | 521 |
| 14.14 小结 .....                               | 474 | 15.7.3 类型库新增接口类型 .....           | 529 |
| 第 15 章 COM 开发 .....                          | 475 | 15.7.4 二进制数据的交换 .....            | 530 |
| 15.1 COM 基础 .....                            | 475 | 15.7.5 后台: COM 的语言支持 .....       | 533 |
| 15.1.1 COM: 组件对象模型 .....                     | 475 | 15.8 TOleContainer .....         | 537 |
| 15.1.2 COM、ActiveX 和 OLE 之间的对比 .....         | 476 | 15.8.1 一个简单的示例程序 .....           | 537 |
| 15.1.3 术语 .....                              | 476 | 15.8.2 一个较为复杂的示例程序 .....         | 539 |
| 15.1.4 ActiveX 的优势 .....                     | 477 | 15.9 小结 .....                    | 547 |
| 15.1.5 OLE 1 与 OLE 2 的对比 .....               | 477 | 第 16 章 Windows 外壳编程 .....        | 548 |
| 15.1.6 结构化存储 .....                           | 477 | 16.1 托盘图标组件 .....                | 548 |
| 15.1.7 统一数据传输 .....                          | 478 | 16.1.1 API .....                 | 548 |
| 15.1.8 线程模式 .....                            | 478 | 16.1.2 处理消息 .....                | 550 |
| 15.1.9 COM+ .....                            | 478 | 16.1.3 图标及提示 .....               | 551 |
| 15.2 COM 与 Object Pascal .....               | 479 | 16.1.4 鼠标点击 .....                | 552 |
| 15.2.1 接口 .....                              | 479 | 16.1.5 隐藏应用程序 .....              | 553 |
| 15.2.2 使用接口 .....                            | 481 | 16.1.6 托盘图标应用程序示例 .....          | 559 |
| 15.2.3 HRESULT 返回类型 .....                    | 485 | 16.2 应用程序桌面工具栏 .....             | 560 |
| 15.3 COM 对象和类工厂 .....                        | 485 | 16.2.1 API .....                 | 561 |
| 15.3.1 TComObject 和 TComObject-Factory ..... | 486 | 16.2.2 TAppBar: AppBar 的窗体 ..... | 562 |
| 15.3.2 in-process COM 服务器 .....              | 487 | 16.2.3 使用 TAppBar .....          | 569 |
| 15.3.3 创建 in-proc COM 服务器实例 .....            | 488 | 16.3 外壳链接 .....                  | 572 |
| 15.3.4 out-of-process COM 服务器 .....          | 489 | 16.3.1 获取 IShellLink 实例 .....    | 573 |
| 15.4 聚合 .....                                | 489 | 16.3.2 使用 IShellLink .....       | 573 |
| 15.5 分布式 COM .....                           | 490 | 16.3.3 创建一个外壳链接 .....            | 575 |
| 15.6 Automation .....                        | 490 | 16.3.4 获取及设置链接信息 .....           | 576 |
| 15.6.1 IDispatch .....                       | 491 | 16.3.5 示例程序 .....                | 580 |
| 15.6.2 类型信息 .....                            | 492 | 16.4 外壳扩展 .....                  | 586 |
| 15.6.3 后期绑定与早期绑定的对比 .....                    | 492 | 16.4.1 COM 对象向导 .....            | 588 |
| 15.6.4 注册 .....                              | 493 | 16.4.2 复制钩子处理器 .....             | 588 |
| 15.6.5 创建 Automation 服务器 .....               | 493 | 16.4.3 环境菜单处理器 .....             | 592 |
| 15.6.6 创建 Automation 控制器 .....               | 505 | 16.4.4 图标处理器 .....               | 600 |
| 15.7 高级 Automation 技术 .....                  | 512 | 16.4.5 信息提示处理器 .....             | 607 |
|                                              |     | 16.5 小结 .....                    | 612 |
|                                              |     | 第 17 章 使用 Open Tools API .....   | 613 |
|                                              |     | 17.1 Open Tools 接口 .....         | 613 |