数据库高级开发与专业应用系列

Delphi 5.0 数据库开发与专业应用

敬铮　主编

杨锋　编著

国防工业出版社

·北京·

需要全本请访问 www.ertongbook.com

前　言

Delphi 以它基于窗口和面向对象的编程方法，与 Windows 操作系统紧密地结合、强大的数据库技术支持、迅捷的编译速度，同时兼备 Visual C++功能强大和 Visual Basic 易学灵活的特点，一直为程序员所偏爱。尤其重要的是，Delphi 非常擅长数据库编程，因为它适应于多种数据库结构，从客户机/服务器模式到多层数据库模式，另外大量的控件给开发数据库程序提供了极大的便捷。Delphi 5.0 不仅在 Delphi4.0 的基础上提供了更强大的功能，而且还支持 ADO 技术，因此使用 Delphi 5.0 作为数据库应用程序开发的工具是编程人员最好的选择之一。

鉴于 Delphi 的优良特性和顺应广大用户的要求，我们在积累大量开发经验的基础上编写了这本书。本书包括三部分内容，即 Delphi 基础、数据库应用的快速开发和高级开发篇。

基础篇的主要内容是讲述使用 Delphi 所必须了解的的概念、Delphi 的语法基础，以及 Delphi 的各种常用控件的用法。

数据库快速开发篇的主要内容是讲述 Delphi 数据库开发原理和方法、数据库访问和控制部件的编程和如何在 Delphi 数据库应用中使用 SQL 语言，以及报表的制作等等。通过本篇的学习，读者将掌握 Delphi 数据库开发的基本原理、各种数据库部件的综合应用方法。

数据库高级开发应用篇的主要内容是讲述了 ADO 对象与控件、多层应用程序的创建、BDE 会话期、连接数据库、字段对象、数据库的存储过程、TclientDataSet 的开发、并且对几个 MIDAS 程序的剖析。通过对上述几个方面的学习，读者将大大提高自己的 Delphi 数据库的能力和并增加数据库开发的认识。

在本书的编著过程中，我们尽量注意减少冗长无味的说明，代之以具体实用的例题演示。通过例题，引导读者把握 Delphi 的精髓所在。本书注重开发实例、开发经验、开发技巧和 Windows 高级特性开发，适合于各个层次的 Delphi 用户。对初学者来说，可迅速加入 Delphi 高级用户的行列。对有一定使用经验的读者，也可通过本书掌握 Delphi 深层次的开发方法，学会用更巧妙的办法开发出高水平的 Delphi 应用。

全书由敬铮主编，杨锋编著，邹忠望、刘朋、阎永军、刘敏、耿永兵、徐宏龙、李明、邓刚、王筑军、邴安大、胡小超、姚俊波、谢石、王小伟、颜林灿等同志参与了部分章节内容的编写和程序调试工作。由于时间仓促，作者水平有限，恳请广大专家读者不吝赐教。

目　　录

第 1 篇　Delphi 基础篇

第 1 章　Delphi 概述 1
1.1　Delphi 是什么 1
1.2　Delphi 5.0 的新功能 1
1.3　本章小结 5
第 2 章　Pascal 的语法简介 7
2.1　标记、常量和变量 7
2.2　类型简介 8
2.3　运算符与表达式 9
2.4　语句 10
2.5　过程和函数 12
2.6　指针 13
2.7　文件和文件类型 14
2.7.1　文本文件 14
2.7.2　记录文件 16
2.7.3　无类型文件 16
2.7.4　Delphi 5.0 中常用的文件处理函数 16
2.8　本章小结 17
第 3 章　常用控件的使用 19
3.1　Delphi 5.0 快速设计 19
3.1.1　进入 Delphi 的可视化编程环境 19
3.1.2　设计简单的用户界面 21
3.1.3　改变对象的属性 24
3.1.4　编写事件处理过程 26
3.1.5　使用联机帮助 Help 27
3.2　文本的输入与输出控件 28
3.2.1　Label 控件 28
3.2.2　Edit、MaskEdit 和 Memo 控件 28
3.2.3　List Box 和 Combo Box 控件 29
3.3　按钮和组控件 29
3.3.1　Button 和 BitBtn 控件 29
3.3.2　Speed Button 控件 30

3.3.3 Check Box 与 Radio Button 控件 30
3.3.4 分组、分界控件 30
3.4 图形、图像处理控件 31
3.4.1 Image 控件 31
3.4.2 Shape 控件 31
3.4.3 PaintBox 控件 31
3.5 关系图、文件列表控件 31
3.5.1 OutLine 控件 31
3.5.2 目录访问控件 31
3.6 滚动控件 32
3.6.1 ScrollBar 控件 32
3.6.2 ScrollBox 控件 32
3.6.3 几个进度显示控件 32
3.7 网格、表格控件 33
3.8 多媒体(MultiMedia)和 OLE 控件 33
3.9 使用非可视控件 34
3.9.1 使用菜单控件 34
3.9.2 使用计时器控件 Timer 36
3.9.3 使用公用对话框控件 37
3.10 本章小结 38

第 2 篇 数据库应用的快速开发

第 4 章 Delphi 开发数据库应用程序概述 41
4.1 数据库应用简介 41
4.1.1 数据库管理系统 41
4.1.2 数据库应用程序 42
4.2 Delphi 的数据库特性及功能简介 43
4.2.1 Delphi 的数据库特性 44
4.2.2 Delphi 可以访问的数据源（DataSource） 45
4.3 Delphi 数据库应用程序的开发方法和步骤 46
4.3.1 概述 46
4.3.2 数据库应用程序的开发步骤 47
4.3.3 交付数据库应用程序 48
4.3.4 安装 BDE 48
4.3.5 安装 SQL Link 50
4.4 本章小结 52
第 5 章 简单数据库应用程序的创建 54
5.1 简单的基于单表的数据库应用 54
5.1.1 选择相关的控件 54
5.1.2 设置控件的属性 55

5.1.3 运行程序 55
5.2 利用 TDBNavigator 控件创建存取程序 56
5.2.1 创建应用程序窗体 57
5.2.2 使用 TDBNavigator 控件移动记录指针 58
5.2.3 定制 TDBNavigator 控件 58
5.3 创建主要—明细数据库应用 59
5.3.1 一对多关系的主要—明细型数据库应用程序 59
5.3.2 一对多—多关系的数据库应用 61
5.4 字段对象的使用 62
5.4.1 字段对象的类型 63
5.4.2 创建永久性的字段对象 63
5.4.3 字段对象的属性设置 64
5.4.4 字段对象的访问 66
5.4.5 设定字段对象的显示格式 70
5.4.6 自定义字段以及计算字段对象的创建 71
5.5 查询数据库中的记录 73
5.5.1 使用 GotoKey 方法查找数据库中的记录 73
5.5.2 使用 FindKey 方法查找数据库中的记录 75
5.5.3 利用 GotoNearest 和 FindNearest 执行不精确查找 76
5.6 修改数据库中的记录 78
5.6.1 Edit 方法和 Post 方法 79
5.6.2 实现异常保护的 TRY...FINALLY 语句 80
5.7 插入和删除记录 83
5.7.1 逐步插入方法 84
5.7.2 调用 InsertRecord 插入记录 84
5.8 输入数据的有效性验证 87
5.9 本章小结 90
第 6 章 Delphi 5.0 常用数据库控件的使用 94
6.1 Delphi 数据库的体系结构 94
6.2 数据访问控件的应用及编程 95
6.2.1 Delphi 数据访问控件的层次结构 95
6.2.2 TSession 控件及其应用 96
6.2.3 数据集控件 TDataSet 及其应用 101
6.2.4 TTable 控件及应用 107
6.2.5 TDataSource 控件及其应用 113
6.2.6 字段控件和字段编辑器的使用 115
6.2.7 TReport 控件及其应用 124
6.2.8 应用举例：多个窗体显示同一个数据库表 125
6.3 数据浏览控件的应用及编程 125
6.3.1 数据浏览控件的基本特性 126

需要全本请访问 www.ertongbook.com

6.3.2 使用 TDBText 控件显示表中的数据 127
6.3.3 使用 TDBEdit 控件显示和编辑表中的数据 128
6.3.4 用 TDBGrid 控件显示和编辑表中的数据 130
6.3.5 TDBNavigator 控件及其应用 134
6.3.6 TDBMemo 控件及其应用 135
6.3.7 TDBImage 控件及其应用 136
6.3.8 数据浏览控件中的列表框和组合框 136
6.3.9 TDBComboBox 控件 137
6.3.10 TDBListBox 控件 138
6.4 辅助决策控件的使用 138
6.4.1 Teechart 统计图表控件的应用 138
6.4.2 在 QuickReport 报表上创建图表的一般步骤 145
6.4.3 创建决策图表的一般步骤 145
6.5 本章小结 146
第 7 章 SQL 编程及查询控件的使用 147
7.1 SQL 语言简介 147
7.1.1 SQL 的历史 147
7.1.2 SQL 的优点 147
7.2 TQuery 控件在 SQL 编程中的运用 148
7.2.1 TQuery 控件的使用 149
7.2.2 在 TQuery 控件中编写简单的 SQL 查询命令 150
7.3 SQL 语言编程概述 152
7.3.1 SQL 命令文本的编写 152
7.3.2 SQL 程序的执行 153
7.3.3 通过 TQuery 控件如何获得活动的数据 154
7.4 动态 SQL 语句的编程 156
7.4.1 使用 Params 属性为参数赋值 157
7.4.2 使用 ParamByName 方法为参数赋值 157
7.4.3 使用 Datasource 属性为参数赋值 157
7.4.4 Prepare 方法的使用 159
7.5 SQL 编程实例 159
7.5.1 设计简单的 SQL 程序编辑器 159
7.5.2 设计一个数据库查询器 161
7.6 本章小结 171
第 8 章 Delphi 5.0 的报表制作 174
8.1 QuickReport 的特点 174
8.2 报表控件的介绍 174
8.2.1 TQuickRep 控件 174
8.2.2 TQRSubDetail 控件 177
8.2.3 TQRBand 控件 178

8.2.4 TQRChildBand 控件 179
8.2.5 TQRGroup 控件 179
8.2.6 TQRLabel 控件 179
8.2.7 TQRDBText 控件 179
8.2.8 TQRExpr 控件 180
8.2.9 TQRSysData 控件 180
8.2.10 TQRMemo 控件 180
8.2.11 TQRRichText 控件 181
8.2.12 TQRDBRichText 控件 181
8.2.13 TQRShape 控件 181
8.2.14 TQRImage 控件 181
8.2.15 TQRDBImage 控件 181
8.2.16 TQRCompositeReport 控件 181
8.2.17 TQRPreview 控件 182
8.2.18 TQRChart 控件 182
8.3 一个比较复杂的例子 182
8.4 本章小结 192

第 3 篇 Delphi 5.0 数据库的高级编程

第 9 章 ADO 对象和控件以及数据库的连接 195
9.1 ADO 简介 195
9.2 ADO 控件的使用 196
9.2.1 原生 ADO 的架构 196
9.2.2 Delphi 5.0 中的 ADOExpress 组件 196
9.3 一个使用 ADO 的例子 198
9.4 连接数据库 203
9.4.1 永久和临时的 TDatabase 构件 204
9.4.2 控制连接 205
9.4.3 遍历一个数据库的所有数据集 208
9.4.4 TDatabase 与 TSession 的关系 208
9.5 本章小结 209
第 10 章 创建多层应用程序 211
10.1 多层体系结构的概述 211
10.1.1 多层体系结构的优势 211
10.1.2 MIDAS 技术 211
10.1.3 MIDAS 应用程序是怎样工作的 212
10.1.4 客户程序的结构 212
10.1.5 应用服务器的结构 212
10.1.6 MTS 213
10.1.7 IDataBroker 接口和 IProvider 接口 213

10.2 选择连接方式 214
10.3 创建应用服务器的一般步骤 215
10.4 远程数据模块 216
10.4.1 TRemoteDataModule 216
10.4.2 TMTSDataModule 217
10.4.3 TCORBADataModule 218
10.5 Provider 219
10.5.1 控制数据包中的字段 219
10.5.2 Options 属性 219
10.5.3 在数据包中加入自定义的信息 220
10.5.4 响应客户的数据请求 220
10.5.5 响应客户的更新请求 221
10.5.6 在更新数据库之前编辑 Delta 数据包 221
10.5.7 怎样定位记录 222
10.5.8 在服务器端纠错 223
10.6 创建客户程序的一般步骤 223
10.7 与应用服务器连接 224
10.7.1 用 DCOM 来连接 224
10.7.2 用 TCP/IP 连接 224
10.7.3 用 OLEnterprise 连接 224
10.7.4 用 CORBA 连接 225
10.7.5 标识服务器 225
10.7.6 TSimpleObjectBroker 225
10.7.7 开始连接 226
10.7.8 断开连接 226
10.8 调用服务器上的接口 226
10.9 在客户端纠错 227
10.10 更新数据 227
10.10.1 更新数据的一般步骤 228
10.10.2 ApplyUpdates 函数 228
10.10.3 核对出错的记录 228
10.10.4 刷新记录 229
10.10.5 从应用服务器获取参数 229
10.11 自定义应用服务器 229
10.12 多层体系结构下的事务 232
10.13 把客户程序设计为 ActiveForm 232
10.14 本章小结 233
第 11 章 BDE 会话期 237
11.1 TSession 237
11.1.1 默认的 BDE 会话期对象 237

11.1.2 创建另外的 BDE 会话期对象 238
11.1.3 给 BDE 会话期对象命名 238
11.1.4 激活 BDE 会话期对象 239
11.1.5 KeepConnections 属性 239
11.1.6 打开和断开连接 240
11.2 检索有关 BDE 会话期的信息 241
11.3 管理 BDE 别名 241
11.3.1 指定别名的可见性 241
11.3.2 创建、修改和删除别名 242
11.4 遍历所有的 TDatabase 构件 243
11.5 访问 Paradox 表 244
11.6 口令 244
11.6.1 AddPassword 244
11.6.2 RemovePassword 和 RemoveAllPasswords 245
11.6.3 OnPassword 和 GetPassword 245
11.7 管理多个 BDE 会话期对象 246
11.8 本章小结 247
第 12 章 TClientDataSet 250
12.1 浏览和编辑数据 250
12.1.1 浏览数据 250
12.1.2 CanModify 属性 250
12.1.3 取消修改 251
12.1.4 合并修改 251
12.1.5 纠错 252
12.2 索引 252
12.2.1 创建一个新的索引 252
12.2.2 删除和切换索引 253
12.2.3 用索引把数据分组 253
12.3 计算字段 253
12.4 统计值 254
12.4.1 指定统计方式 254
12.4.2 指定分组 255
12.4.3 怎样获取统计值 255
12.5 数据包 256
12.5.1 直接对 Data 属性赋值 256
12.5.2 在数据包中加入自定义的信息 256
12.5.3 克隆另一个数据集 257
12.6 与应用服务器通信 257
12.6.1 怎样在客户端获得 IProvider 接口 257
12.6.2 向应用服务器传递参数 257

12.6.3 怎样向应用服务器请求数据 258
12.6.4 更新数据库 258
12.7 在文件中存取数据 258
12.8 本章小结 259
第 13 章 剖析几个 MIDAS 示范程序 264
13.1 一个 ActiveForm 的例子 264
13.2 一个动态传递 SQL 语句的示范程序 266
13.3 一个计算统计值的示范程序 271
13.4 一个全面演示 TClientDataSet 功能的示范程序 272
13.5 本章小结 290
附录 A 控件 TQueryStock 292
附录 B 编译错误信息 299
附录 C InterBase 概述 324
附录 D 配置数据库引擎 BDE (Borland DataBase Engine) 329

第 1 篇　Delphi 基础篇

第 1 章　Delphi 概述

1.1　Delphi 是什么

Delphi是著名的Borland（现在已和Inprise合并）公司开发完全导向的的可视化软件开发工具。“真正的程序员用C，聪明的程序员用Delphi”，这句话是对Delphi最经典、最实在的描述。Delphi被称为第四代编程语言，它具有简单、高效、功能强大的特点。和VC相比，Delphi更简单、更易于掌握，而在功能上却丝毫不逊色；和VB相比，Delphi则功能更强大、更实用。可以说Delphi同时兼备了VC功能强大和VB简单易学的特点。它一直是程序员至爱的编程工具。

Delphi具有以下的特性：基于窗体和面向对象的方法、高速的编译器、强大的数据库支持、与Windows编程紧密结合、强大而成熟的组件技术。但最重要的还是Object Pascal语言，它才是一切的根本。Object Pascal语言是在Pascal语言的基础上发展起来的，简单易学。

Delphi软件开发工具是一个完全导向的可视化系统开发工具，具有功能强大、运行速度快、易于使用以及开发迅速等特点，一推出就受到广大用户的喜爱。Delphi结合了可视化技术、面向对象编程、数据库和分布式应用技术等先进的软件编程技术和思想，并使用了全特征的代码编辑器，使其成为创建功能丰富、界面友好的Windows应用软件的工具之一。

Delphi提供了各种开发工具，包括集成环境、图像编辑（Image Editor），以及各种开发数据库的应用程序，如DesktopDataBase Expert等。除此之外，还允许用户挂接其他的应用程序开发工具，如Borland公司的资源编辑器（Resourse Workshop）。

在Delphi众多的优势当中，它在数据库方面的特长显得尤为突出：适用于多种数据库结构，从客户机／服务机模式到多层数据结构模式；高效率的数据库管理系统和新一代更先进的数据库引擎；最新的数据分析手段和提供大量的企业组件。

1.2　Delphi 5.0 的新功能

从1995年至今，Delphi已经发展到5.0版本。Delphi 5.0采用面向对象的程序设置、组件化的编程方式、快速Pascal编译器、OLE自动化、ActiveX编程、数据库及Internet/Intranet编

需要全本请访问 www.ertongbook.com

程支持，从而使得 Delphi 5.0开发功能更加强大，同时更加易于学习和使用。该版本的主要特征之一就是支持Web，它能有效地帮助开发者个人或其开发团队快速建立Windows应用程序，并将其扩展到Internet中；快速简化Windows与浏览器客户、Web服务器、中间件以及后台数据库系统的集成；支持HTML4、MXL及建立Internet数据分配与交换应用程序的工业标准。此外，Delphi 5.0还加入了一些人性化的东西，比如颜色的设置、对象属性的控制、更多的集成组件、多语言支持工具（包括中文）等。Delphi 5.0在开发Internet应用、数据库系统等方面的性能有了很大提高，同时在提高开发人员的效率、方便开发人员的使用方面也做了很多改进。

下面分类说明一下Delphi 5.0的新增功能：

1．Object Pascal 语言扩展

Delphi 5.0扩展了Object Pascal语言的一些功能，包括动态数组定义、方法重载和缺省参数设置等。

2．工程管理功能

Delphi 5.0提供了一个新的工程管理器。该工程管理器允许开发人员将多个工程组合在一个工程组中进行处理，这种功能就可以帮助开发人员组织和操作多个相互关联的工程(如多列应用的单列处理或DLLs，以及可执行文件等)一起工作。

3．类实现代码探测器、模块导向和代码浏览器

代码探测器(Code Explorer)使创建类变得非常容易，其通过多个步骤自动创建一个或多个类，并在接口部分定义方法原型和在实现部分创建类实现框架代码。导向功能通过单元文件可以在接口和实现部分之间实现快速定位。使用Tooltip符号暗含功能浏览任何标识符的说明信息，然后使用代码浏览器跳到其说明位置。

4．可连接的工具窗

Delphi 5.0提供的IDE是一个多配置的工具窗，它可以连接成代码编辑器。可以使用拖放工具窗到任何的地方。代码探测器和工程管理器也可以进行连接与分解。

5．改进的调试器

Delphi 5.0提供的集成调试器具有许多新特征，包括远程和多处理器调试、CPU浏览与监视、断点加强、调试器子菜单以及可分离窗等。Delphi 5.0的代码调试器包含以下一些新特点：

- 各种调试窗口和编辑器之间紧密集成，拥有一致的用户界面，支持剪贴板和拖动操作。
- 增加了断点提示条(Breakpoint ToolTips)，以加快浏览和定位源代码。
- 新增的断点动作(Breakpoint Actions)功能加强了对调试进程的控制。
- 断点集合(Breakpoint Groups)方便了对多条断点的控制。
- 在DCUs和调试符号查找路径的帮助下，调试工作变得更简单。
- FPU调试窗口支持MMX指令，从而增强了底层调试的能力。
- 进行多进程调试时，增加了临时进程选项，并支持调试子进程。
- 支持跨越进程边界的调试。
- 能够挂接到正在运行的进程上进行调用。

6．MTS 支持

Delphi 5.0完全支持MTS，MTS支持多数据库应用。另外，一个新的Wizard将使创建MTS

服务器对象变得更容易。

7．增强的 ActiveX

Delphi 5.0提供扩展的ActiveX和ActiveForm支持。

8．增强 VCL 功能

Delphi 5.0加强了Delphi的对象层次功能，包括面向NT服务应用的新组件。另外，新的列表组件(在Standard组件板页)允许开发人员对菜单和工具条按钮命令进行居中操作。VCL控件加上了支持拖放功能，并提供对窗体布局的附加控制等。在Delphi 5.0中，用户可以利用面向对象设计的强大功能开发出稳定、可靠、高效的程序，可以利用现有的面向对象的控件创建自己的控件。Delphi 5.0企业版的VCL中一共包含有200多个控件，利用这些控件，用户可以奇迹般地加快应用程序从开始开发到推向市场的速度。无论是开发Windows程序还是Internet应用，Delphi 5.0基于控件的开发模式都能大大降低开发任务的难度。

9．RTL 支持 2000 年表示

TwoDigitYearCenturyWindow变量用于StrToDate和StrToDateTime函数中，控制在进行日期转换时的两位数字的解释，可以避免2000年问题。

10．支持 CORBA

Delphi 5.0的客户机与服务器和Enterprise版本均包含对CORBA客户机和服务器应用的支持。Delphi 5.0提供的向导使创建一个CORBA服务器变得非常容易，而且动态请求接口(Dynamic Invocation Interface，简称DII)允许开发人员为存在的CORBA服务器写客户机。CORBA也内建在多数据库支持中，用户可以建立同时处理COM和CORBA客户机的服务器。

11．增强客户机数据集

Delphi 5.0客户机数据集支持过滤表达式变量和维护总量，以及允许面向对象的域类型。

12．MIDAS3 支持

Delphi 5.0提供了许多对多层应用的控制，包括refresh／resync 支持和对数据包中包含的数据及更新如何应用等方面的控制。一个新类TDataSetProvider允许用户使用nested表从一个支持Master／detail的数据集中取得数据，或发送数据到它们之中。另外，用户数据集的增强功能使参数传递到应用服务器，或保存用户信息到数据包中更为方便。

利用Delphi 5.0中的MIDAS 3 (Multi－tier Distributed Application Services Suite)，用户可以创建高效灵活的Delphi程序，以满足日益增长的事务量和用户数的需要。MIDAS 3支持所有的分布式计算标准，如CORBA、COM和MTS等，可以实现现有系统和电子商务程序之间的无缝集成。它的特点表现为：没有界限的数据请求，在移动或窄带宽的环境中可实现更多控制；WebConnection组件能够在保证安全性的前提下，帮助用户把程序移到防火墙外；服务对象库增强了扩展性。

13．增强数据库功能

Delphi 5.0的数据连接组件允许开发人员调整数据模型，以便建立数据模块或窗体。修改的数据连接组件和Borland Database Engine(BDE)允许用户使用新的数据服务器类型，包括Access 97和Oracle 8到SQL的新扩展(包括ADTs——Abstract Data Types，抽象数据类型)、数据、索引和nested表。全新的InterBase Express数据库组件使得InterBase数据库连接和服务器的事务控制更方便快捷。可视化Query Builder已被SQL Builder替代，这是一个新的智能查询创建器。

14．强大的 COM 技术支持

Delphi 5.0的DAX技术有力地将COM技术溶入任何的网络开发中，使其更方便地支持MTS、ActiveX、MIDAS和自动化对象与服务器开发。

15．支持 XML

Delphi 5.0提供了对XML的支持，从而简化了数据分布，优化了数据交换。用户可以方便地创建高效率的Internet程序，来把数据迅速发布到Internet上。新版Delphi对XML数据的支持使得开发人员能够快速建立起具有可移植性和扩展性的系统，以迎接Internet正面临着的又一次冲击波——电子商务。

16．支持 HTML4

Delphi 5.0支持HTML4，使用户能在Web应用中创建各种类型的动态瘦客户端程序：把HTML4和XML结合起来使用，从而创建动态的极瘦客户端程序，用来满足Internet上各种任务的需求。

17．支持 ADO

ADO(ActiveX Data Objects)是微软提供的一项技术。通过ADO，我们可以方便地访问各种类型的数据库，特别是OleDB数据库。ADO已成为访问数据库的新的标准接口。Delphi 5.0增添了对ADO的支持，是为了让用户能迅速实现对终端用户用来做商业决策的数据的一致性访问，结合Delphi本身的开放式数据元件结构，用户可以很快地建立应用程序，用来把自己的商业数据通过Internet发送给客户、最终用户以及整个销售环节。通过ADO，Delphi 5.0能让用户快速访问关系型或非关系型数据库，以及E__Mail和文件系统。

18．InterBase Express

利用InterBase Express，系统集成商和独立软件商可以快速开发高性能的应用系统。把Delphi 5.0和InterBase Express结合起来使用，便可以在只需少量维护工作且只占用很少系统资源的数据库上进行高效率、高性能的开发。有了InterBase Express后，用户不需要使用BDE便可以创建一个优秀的嵌入式系统。

19．TeamSource

Delphi 5.0集成了一个与微软的SourceSafe功能相似的源代码管理工具——TeamSource。这样，工作组共同进行的开发工作会合作得更好更快，工作更有效率。TeamSource建立在现有的源代码管理工具的版本引擎的基础上，并把它们扩展成一种工作流程的模式，从而简化了工作组开发的源代码管理工作。

20．Borland Translation Suite

这是Delphi 5.0包含的一个国际化和本地化工具，它能帮助用户同时开发多语言版本，管理和维护各种资源的翻译代码，从而加速产品的国际化开发步伐。现在利用Delphi 5.0开发的系统可以在世界范围内应用，帮助用户开拓全新的世界范围内的市场。

21．改善的 IDE

Delphi 5.0中的集成开发环境(IDE)有了很大改进，能极大地提高开发效率，它主要依靠简化读写和浏览代码的操作来提高开发效率。代码编辑器让浏览本单元或相关单元的内容变得容易，工程管理器中的拖放支持使得从已打开的工程中或资源管理器的对话框中选择文件，并增加到工程的文件中变得更简单，使用工程管理器中多个工程管理的功能让用户能同时编辑多个工程的文件，应用Delphi 5.0的可视化窗口设计让用户能够轻松地从模板中选择创建Internet程序、分布式计算及Windows程序等。与Delphi 4.0相比，Delphi 5.0的IDE

新特点主要包括：桌面的用户化设置增强了用户对开发环境的控制；编辑键映射功能可以让编辑器按照用户的习惯来工作；工程浏览可以帮助用户更好地理解代码、操作VCL；带有树视图和数据图表视图(Data Diagram View)的DataModule设计器可以帮助用户充分理解程序中的数据；资源文件与工程管理器和编译器集成在一起；动作列表保存了开发计划；控制面板向导可以帮助用户定制应用程序的属性。

功能的扩展必定相伴着一定的兼容问题，以下再简要的说明一下Delphi5.0兼容性：

- Delphi 5.0包的扩展名被改动，由原来的.DPL替换为.BPL。
- 如果已经创建了自己的设计包，Delphi 5.0要求在安装到Delphi 5.0之前必须重建该包。为了能够识别其他包的制造者(如C++Builder 3、Delphi 5.0等)，内部包格式也做了修改。
- C++Builder 4的.BPL文件不能在Delphi 5.0上使用。如果用户购买了第三方的Delphi组件，那么，必须使用它们提供的源代码。
- Delphi 5.0不能打开Delphi 4.0的DPK文件，而是自动转换成Delphi 5.0包新的源格式。
- 改变所有SYSDATE实例到GetDate。不能使用Year、Month、Day、Hour、Minute或Second提取时间、日期等。
- OnDragOver和OnDragDrop事件源参数被修改。在Delphi 4.0中，OnDragOver和OnDragDrop事件可以使用TDragControlObject的源参数执行，不需要知道其内部拖动控制。在Delphi 5.0中，事件源参数实际上控制拖动操作的开始。
- DWORD、UINT、HResult、OLE_HANDLE和API处理类型被改变。DWORD、UINT、HResult、OLE_HANDLE和所有的API处理类型被定义为无符号的2位LongWord，它们不再与Integer类型兼容。
- 新的Int64／LongWord类型。为了进行算术运算，整型结果导致转到Int64，避免溢出。因此，某些不适合于Integer类型的变量将处理成Int64(64位Integer)/LongWord(无符号32位Integer)类型。详细资料请看帮助信息。
- Delphi 5.0提供了NetManage的Internet组件。
- 与Oracle 8兼容。在SELECT陈述中，关键字OR UPDATE会锁住结果集，用户需要将QLPASSTHRU MODE设为SHARED NOAUTOCOMMIT。

1.3 本 章 小 结

1．问：Delphi 和 VC++、VB 相比有什么优越性？

答：和VC相比，Delphi更简单、更易于掌握，而在功能上却丝毫不逊色；和VB相比，Delphi则功能更强大、更实用。可以说Delphi同时兼备了VC功能强大和VB简单易学的特点。它一直是程序员至爱的编程工具。

2．问：Delphi 5.0 和以前的版本相比有什么新的特点？

答：Delphi 5.0在Object Pascal语言扩展、工程管理功能、类实现代码探测器、模块导向和代码浏览器，以及可连接的工具窗等21个方面均有了新的改进，尤其是在数据库处理方面更加强大。

3．Delphi 5.0 和 Delphi 4.0 以及以前的版本如何兼容？

答：Delphi 5.0的兼容性需要考虑以下方面：

- Delphi 5.0包的扩展名被改动，由原来的.DPL替换为.BPL。
- 如果已经创建了自己的设计包，Delphi 5.0要求在安装到Delphi 5.0之前必须重建该包。为了能够识别其他包的制造者(如C++Builder 3、Delphi 5.0等)，内部包格式也做了修改。
- C++Builder 4的.BPL文件不能在Delphi 5.0上使用。如果用户购买了第三方的Delphi组件，那么，必须使用它们提供的源代码。
- Delphi 5.0不能打开Delphi 4.0的DPK文件，而是自动转换成Delphi 5.0包新的源格式。
- 改变所有SYSDATE实例到GetDate。不能使用Year、Month、Day、Hour、Minute或Second提取时间、日期等。
- OnDragOver和0nDragDrop事件源参数被修改。在Delphi 4.0中，OnDragOver和OnDragDrop事件可以使用TDragControlObject的源参数执行，不需要知道其内部拖动控制。在Delphi 5.0中，事件源参数实际上控制拖动操作的开始。
- DWORD、UINT、HResult、OLE_HANDLE和API处理类型被改变。DWORD、UINT、HResult、OLE_HANDLE和所有的API处理类型被定义为无符号的2位LongWord，它们不再与Integer类型兼容。
- 新的Int64／LongWord类型。为了进行算术运算，整型结果导致转到Int64，避免溢出。因此，某些不适合于Integer类型的变量将处理成Int64(64位Integer)/LongWord(无符号32位Integer)类型。详细资料请看帮助信息。
- Delphi 5.0提供了NetManage的Internet组件。
- 与Oracle 8兼容。在SELECT陈述中，关键字OR UPDATE会锁住结果集，用户需要将QLPASSTHRU MODE设为SHARED NOAUTOCOMMIT。

第 2 章　Pascal 的语法简介

2.1　标记、常量和变量

标识符是Delphi应用程序中一些量的名称，这些量包括变量（var）、常量（const）、类型（type）、过程（procedure）、方法（Method）及其他，Object Pascal 在应用标识符时，必须首先说明它们。Object Pascal是强类型语言，它的编译器可以检查确保赋给变量或属性的值是正确的类型，以便于您改正错误。因为Object Pascal是编译语言，所以Delphi的执行速度要比使用解释语言快得多。在使用标识符前说明它们，可以减少程序的错误并增加代码的效率。

常量在说明时就被赋予了一个值，在程序执行过程中是不可改变的，在Pascal 中用const关键字来声明，和C语言里constant 关键字一样，但是也有区别，主要表现在Object Pascal 语言中不需要在对常量赋值时声明常量的类型，Pascal 语言的编译器能根据常量的值自动判断常量的类型并分配内存；但对整型常量，编译器在运行时跟踪它的值但不对它分配内存，当然，在声明变量时，也可以指定变量的类型，这样就能完全控制编译器对常量的处理。

下面的例子告诉你怎样声明常量：

```
Const WM_message=5000（未制定变量的类型）；
Const I: Integer=10（指定变量的类型）；
```

变量是程序代码中代表一个内存地址的标识符，而此地址的内存内容在程序代码执行时可以被改变。在使用变量前必须对它进行说明，即对它进行命名，并说明它的类型。在所有变量说明之前加上保留字var。变量说明左边是变量的名称，右边则是该变量的类型，中间用“ : ”隔开。如：

```
var
    Value ，Sum : Integer;
    Line : String;
```

变量又分局部变量和全局变量。

通过图2.1分析局部变量和全局变量，图2.1表示一个含有两个库单元的工程，每个库单元中又各有三个过程或事件处理过程。

此图中每一个矩形代表一个程序模块。如果您在程序D中说明一个变量，则只有过程D可以访问它，它是D的局部变量。如果您在程序B中说明一个不属于任何过程的变量，那么过程D、E、F都能使用它，因为以上的过程都属于库单元B，因此这个变量对以上过程是全局的，但对于程序库单元B它是局部的。一般来讲，应尽量把标识符说明成局部的，这样会增加程序对数据的保护，使它不会被不小心修改。只有当几个程序模块分享数据时，才需要考虑使用全局的说明。

需要全本请访问 www.ertongbook.com