

Delphi 4.0

Delphi 4.0

Delphi 4.0

# Delphi 4.0

## 使用指南

卓越文化艺术公司

主编 / 王益进 杜吉祥

### 编程人员必备

2.35	0.001	1
2.36	0.243	
3.7	0.387	
1.11	0.265	
1.29	0.3	5
2.37	0.37	6
4.47		
5.90		
3.47	0.667	
2.38	0.77	
4.48		
2.39		
3.48	0.67	
3.49	0.466	14
4.5	0.765	15
3.46	0.66	16
2.37	0.667	17
2.39	0.373	18
2.37	0.367	19



航空工业出版社

# Delphi 4.0 使用指南

主 编 王益进 杜吉祥

编 委 姜晓铭 肖 阳

见 英 张 磊

航空工业出版社

1999

## 内 容 提 要

本书向读者系统地介绍了最新可视化软件开发工具——Borland Delphi 4.0（以下简称 Delphi 4.0）。

全书总共分为 12 章。第 1 章到第 3 章是 Delphi 的基础入门，带领读者浏览 Delphi 的可视化开发环境、介绍 Delphi 的菜单功能、完成一个小例子，使读者熟悉 Delphi 的开发环境。第 4 章和第 5 章向读者简单介绍 Delphi 所使用的编码语言——Object PASCAL，帮助读者更好地掌握 Object PASCAL 语言，使读者能够更好地理解本书后面的内容，也为读者以后的发展做好准备。第 6 章和第 7 章向读者介绍 Delphi 对工程文件的管理以及 Delphi 所提供的编辑和调试工具，进一步帮助读者打下良好的程序设计基础。第 8 章向读者介绍了数据库基础知识，使读者对 Delphi 所应用的关系数据库的概念和机制有一个较深入的了解。第 9~11 章通过大量的例子向读者介绍 Delphi 4.0 中与数据库有关的控件，包括 QuickReport 和 Decision Cube 页上的报表控件，读者通过学习这 3 章的内容，将可以开发出美观实用的 Delphi 数据库应用程序。第 12 章向读者介绍了结构化查询语言——SQL 的基础知识，使读者能够更好地使用关系数据库的查询功能。

本书结合生动的例子详细地介绍了 Delphi 4.0 的数据库开发机制。全书图文并茂，既适合初学者入门学习，也适合中级读者作为 Delphi 数据库开发的进阶读物。是一本具有较好指导性的计算机图书。

### 图书在版编目（CIP）数据

Delphi 4.0 使用指南 / 王益进等编. — 北京：航空工业出版社，1999.8

ISBN 7-80134-478-2

I. D… II. 王… III. Delphi 语言 IV. TP312

中国版本图书馆 CIP 数据核字（1999）第 24361 号

航空工业出版社出版发行

（北京市安定门外小关东里 14 号 100029）

北京云浩印刷厂印刷

全国各地新华书店经售

1999 年 8 月第 1 版

1999 年 8 月第 1 次印刷

开本：787×1092 1/16

印张：19.75

字数：480 千字

印数：1-12000

定价：29.80 元

本社图书如有缺页、倒页、脱页、残页等情况，请与本社发行部联系调换。联系电话：010-65859701 或 64941995

# 前 言

Microsoft Windows 是基于图形界面的多任务、多窗口操作系统。自 1983 年 Windows 问世以来，日趋完善，Windows 标准已经不断被广大用户接受和认可，从而成为了计算机软件设计的标准。

早期的 Windows，没有很好的应用程序开发工具，那时的程序员形容开发一个 Windows 程序，就像经过一场可怕的梦魇。经过多年的实践和探索，很多程序员发现，随着面向对象技术的进步，开发一般的 Windows 应用程序后很多的工作是重复的，于是在很多大软件公司里形成了自己的开发系统。随着这些开发系统的不断进步和完善，在 20 世纪 90 年代中期，一些公司开始将这些称为“代码生成器”的工具，推向市场。当用户第一次看到这些可视化开发工具时，心中都充满了一种期待已久的惊喜。软件开发已经成为了大多数计算机工作者力所能及的事情。

在众多的可视化开发工具中，当年的 Borland 公司推出的基于 Object Pascal 的 Delphi 不愧是一朵奇葩。笔者从 Delphi 1.0 推出时就开始使用 Delphi 进行软件开发，对 Delphi 的功能机制和特点有较深入的了解。当年的 Delphi 以它的编译速度以及高效的生成代码，被称为当时的 VB 杀手。基于结构化的原代码以及大量的原代码公开，使得 Delphi 不仅为用户提供提供了快捷的开发环境，而且也给了使用者发挥自己创造力的空间。这也是 Delphi 多年来一直受到广大软件开发者欢迎的原因。

Delphi 除了拥有其他可视化开发工具的优点外，还提供了很好的数据库支持。Delphi 配有 Database Engine，可通过 SQL Links、ODBC 访问多种数据库，而且还提供了强大的开发 Client/Server 模式数据库应用程序的能力。在数据库前端应用程序的开发中，Delphi 使用了 Multi Session 和 Thread Safe 的数据库引擎、数据库过滤器 (Filter)、Visual Query Build、查询引擎和最新的数据更新模式等，最大限度地为 Delphi 的用户提供了方便。

本书通过实例介绍 Delphi 4.0 的控件使用和数据库开发机制，将作者多年来应用 Delphi 开发数据库应用程序的经验融入其中，希望能够为读者在使用 Delphi 4.0 的过程中提供一点帮助。读者的收获就是我们的成功。

由于水平有限，书中难免存在错误与不足，欢迎广大读者批评指教。

编 者  
1999 年 6 月



# 目 录

<b>第 1 章 新一代编程工具——代码生成器 .....</b>	<b>1</b>
1.1 应用程序开发工具 .....	1
1.2 强大的数据库功能 .....	4
<b>第 2 章 揭开面纱：一个简单的例子 .....</b>	<b>5</b>
2.1 窗体——你的舞台 .....	5
2.2 控件的放置 .....	5
2.3 设置控件的属性 .....	6
2.4 响应事件 .....	7
2.5 运行一个应用程序 .....	8
<b>第 3 章 浏览 Delphi 的 IDE .....</b>	<b>9</b>
3.1 IDE 概貌 .....	9
3.2 基本组成 .....	9
3.2.1 主菜单 .....	10
3.2.2 工具条 (Toolbar) .....	10
3.2.3 控件模板 .....	10
3.2.4 窗体设计器 (Form Designer) .....	11
3.2.5 代码编辑器 (Code Editor) .....	11
3.2.6 对象观察器 (Object Inspector) .....	12
3.3 菜单结构 .....	14
3.3.1 File 菜单 .....	14
3.3.2 Edit 菜单 .....	15
3.3.3 Search 菜单 .....	16
3.3.4 View 菜单 .....	16
3.3.5 Project 菜单 .....	17
3.3.6 Run 菜单 .....	18
3.3.7 Component 菜单 .....	19
3.3.8 Database 菜单 .....	19
3.3.9 Tools 菜单 .....	20
3.3.10 Help 菜单 .....	20
3.4 配置自己的 IDE .....	22

3.4.1	自定义工具条.....	22
3.4.2	自定义控件模板.....	23
3.4.3	自定义代码编辑器.....	24
3.4.4	设计界面的布局及保留.....	25
3.5	小结.....	26
<b>第 4 章 Object Pascal 简介 .....</b>		<b>27</b>
4.1	数据及运算.....	27
4.1.1	标识符 (Identifier) .....	27
4.1.2	保留字和标准指令.....	28
4.1.3	常量 (Constants) .....	29
4.1.4	变量 (Variant) .....	30
4.1.5	简单数据类型.....	30
4.1.6	结构类型.....	34
4.1.7	数据的运算及操作.....	36
4.2	Object Pascal 程序结构.....	40
4.2.1	顺序结构.....	40
4.2.2	分支结构.....	42
4.2.3	循环结构.....	45
4.2.4	子程序.....	49
4.2.5	指针.....	54
4.3	小结.....	56
<b>第 5 章 面向对象的程序设计 .....</b>		<b>57</b>
5.1	面向对象的程序设计 .....	57
5.1.1	概念.....	57
5.1.2	特点.....	58
5.2	Delphi 的面向对象实现——Object Pascal 语言 .....	59
5.2.1	对象与类.....	59
5.2.2	类的封装性.....	61
5.2.3	类的继承性.....	62
5.2.4	类的多态性.....	64
5.2.5	类的高级特征.....	65
5.3	小结.....	67

<b>第 6 章 应用程序、文件和工程管理器</b> .....	<b>68</b>
6.1 组织一个工程.....	68
6.2 应用程序的组成.....	69
6.2.1 工程文件.....	69
6.2.2 窗体文件.....	70
6.2.3 单元文件.....	71
6.2.4 其他文件类型.....	76
6.3 大管家——工程管理器 .....	76
6.3.1 工程管理器窗口.....	77
6.3.2 在工程中增加和取出文件.....	77
6.3.3 保存工程及文件.....	78
6.3.4 工程文件的编译和链接.....	79
6.4 详细的工程环境设置信息 .....	80
6.4.1 Forms 选项卡 .....	80
6.4.2 Application 选项卡.....	81
6.4.3 Compiler 选项卡 .....	81
6.4.4 Linker 选项卡 .....	83
6.4.5 Directories/Conditionals 选项卡 .....	85
6.4.6 Version Info 选项卡.....	86
6.4.7 Packages 选项卡.....	88
6.5 对象仓库.....	89
6.6 小结.....	90
<b>第 7 章 编辑与调试</b> .....	<b>91</b>
7.1 使用程序编辑器.....	91
7.2 设置自己喜欢的程序编辑器选项 .....	92
7.2.1 Editor.....	92
7.2.2 Display .....	95
7.2.3 Color.....	96
7.3 无微不至的帮助——Code Insight 功能.....	99
7.3.1 配置 Code Insight .....	99
7.3.2 Code Completion .....	100
7.3.3 Code Parameters .....	103
7.3.4 Tooltip Expression Evaluation.....	103
7.3.5 Tooltip Symbol Insight .....	104
7.3.6 Code Templates.....	105
7.4 调试器.....	106

7.4.1	设置调试选项.....	107	
7.4.2	跟踪运行与设置断点.....	109	
7.4.3	利用表达式计算器监看变量值 .....	112	
7.4.4	用 Watch 窗口监看变量值 .....	112	
7.5	小结.....	114	
<b>第 8 章 数据库系统基础概论 .....</b>			<b>115</b>
8.1	数据库管理系统.....	115	
8.2	数据库应用程序.....	117	
8.3	Delphi 的数据库系统.....	118	
8.3.1	Delphi 的数据库特性.....	118	
8.3.2	Delphi 可以访问的数据源.....	120	
<b>第 9 章 应用 Data Access 和 Data Controls .....</b>			<b>121</b>
9.1	Delphi 数据库编程的基础知识 .....	121	
9.1.1	Delphi 对各种数据库的支持.....	121	
9.1.2	数据表格.....	123	
9.2	快速入门: 一个小例子 .....	124	
9.3	Delphi 数据库应用程序的基本结构 .....	128	
9.4	DataSource 控件.....	129	
9.4.1	DataSource 的属性.....	129	
9.4.2	DataSource 的事件.....	131	
9.5	Table 控件.....	131	
9.5.1	TTable 的属性: .....	132	
9.5.2	TTable 的方法: .....	139	
9.6	Query 控件.....	144	
9.6.1	TQuery 的属性.....	144	
9.7	使用 DBNavigator 控件 .....	146	
9.7.1	DBNavigator 控件上的按钮.....	146	
9.7.2	使用 ShowHint 属性显示帮助 .....	147	
9.7.3	Confirm Delete 属性.....	148	
9.8	使用 DBGrid 控件.....	149	
9.8.1	DBGrid 简介.....	149	
9.8.2	使用 Columns Editor.....	151	
9.8.3	设定 Options 属性.....	154	
9.9	使用 SQL 语言 .....	156	
9.9.1	使用 SQL Builder 建立查询.....	161	



9.10 其他一些 Data-Aware 控件 .....	164
9.10.1 DBText 控件 .....	164
9.10.2 DBEdit 控件 .....	165
9.10.3 DBListBox 控件 .....	166
9.10.4 DBComboBox 控件 .....	168
9.10.5 DBMemo 控件 .....	170
9.10.6 DBImage 控件 .....	171
9.10.7 DBCtrlGrid 控件 .....	173
9.10.8 DBLookupComboBox 控件 .....	174
9.11 对数据表格进行编辑 .....	176
9.11.1 移动文件指针 .....	176
9.11.2 修改数据表格 .....	181
9.11.3 对数据表格进行遍历 .....	183
9.12 TField 类 .....	185
9.13 进行快速查找 .....	189
9.14 向应用程序添加计算功能 .....	192
9.14.1 添加计算字段 .....	192
9.14.2 求数据表格的总和 .....	193
9.15 数据表格的连接、建立主从关系 .....	194
9.16 运行时间创建数据表格 .....	198
<b>第 10 章 用 QReport 控件制作快速报表 .....</b>	<b>201</b>
10.1 几个基本控件 .....	201
10.1.1 QuickRep 控件 .....	201
10.1.2 QRSubDetail 控件 .....	208
10.1.3 QRBand 控件 .....	210
10.1.4 QRDBText 控件 和 QRDBLabel 控件 .....	213
10.2 最快速生成快速报表的方法 .....	214
10.3 快速报表的边条 .....	218
10.3.1 QRChildBand 控件 .....	218
10.3.2 QRGroup 控件 .....	219
10.3.3 边条的组织 .....	220
10.4 在快速报表中加入图形 .....	221
10.4.1 有关图形的控件 .....	221
10.4.2 一个例子 .....	223
10.5 生成快速报表的应用编程 .....	225
10.5.1 QReportLabel 模板 .....	225
10.5.2 Master/Detail 模板 .....	231

10.5.3 应用快速报表编程.....	239
10.6 快速报表的计算功能.....	246
10.7 快速报表的图表功能.....	253
10.8 小结.....	260
<b>第 11 章 Decision Cube 页上的控件.....</b>	<b>261</b>
11.1 Decision Cube 的机制.....	261
11.2 一个简单的实例.....	264
<b>第 12 章 SQL 基础知识.....</b>	<b>266</b>
12.1 SQL 历史.....	266
12.2 SQL 概貌及其特点.....	266
12.3 SQL 数据查询功能.....	268
12.4 SQL 数据操纵功能.....	281
12.5 本章小结.....	286
<b>附录 A 常用函数.....</b>	<b>287</b>
A.1 算术函数.....	287
A.2 时间函数.....	287
A.3 内存管理函数.....	288
A.4 文件函数.....	288
A.5 浮点函数.....	290
A.6 流程控制函数.....	290
A.7 I/O 函数.....	291
A.8 堆函数.....	291
A.9 其他函数.....	291
A.10 次序函数.....	292
A.11 指针函数.....	292
A.12 格式化函数.....	292
A.13 Pascal 风格字符串函数.....	293
A.14 NULL 结尾字符串函数.....	294
A.15 文本文件函数.....	295
A.16 转换函数.....	296
A.17 无类型文件函数.....	296

---

<b>附录 B 控件简介 .....</b>	<b>297</b>
B.1 Standard 页 .....	297
B.2 Additional 页 .....	298
B.3 System 页 .....	298
B.4 Win32 页 .....	299
B.5 Dialogs 页 .....	300
B.6 Internet 页 .....	300
B.7 QReport 页 .....	301
B.8 Decision Cube 页 .....	302

# 第1章 新一代编程工具——代码生成器

本章通过对软件开发工具的发展的简单介绍，提出了代码生成器的概念，使读者对 Delphi 的编程机制有一个简单的了解。

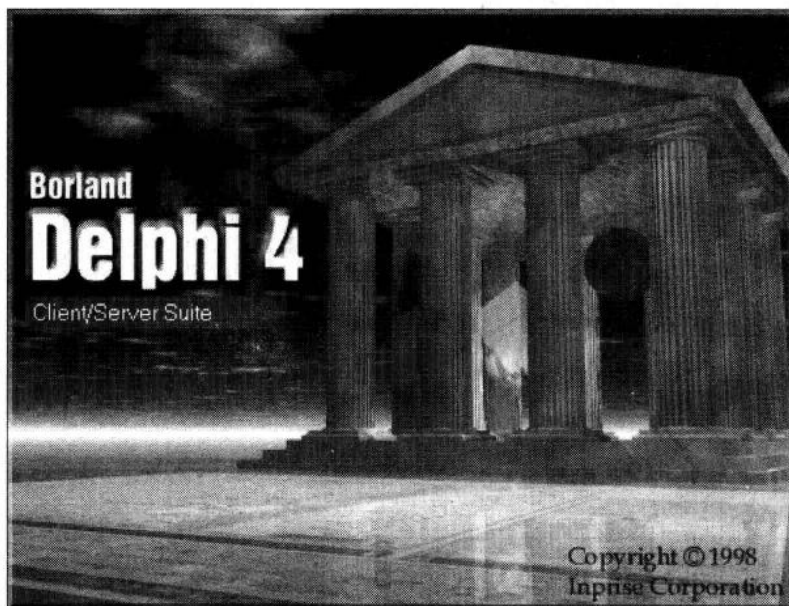


图 1.1 Delphi 的欢迎界面

## 1.1 应用程序开发工具

应用程序开发工具 (Application Developing Tool) 作为一种开发软件的特殊软件，其发展过程标志着整个计算机软件体系的发展和进步。

最初的计算机语言是仅以“0”、“1”为编码的、冗长的、从编码上没有任何直观意义的“机器语言” (Machine Language)，又称“二进制语言”。虽然这种语言的执行效率非常的高，但是由于人工编写机器语言程序非常繁琐、易出错，所以当时只有极少数的专业人员才能掌握。

后来人们提出了“汇编语言” (Assembler Language)，又称“符号语言”。它用一些简单的符号来代替机器语言中的二进制编码然后再由特定的系统软件将这种语言转换成目标程序 (机器语言) 交给计算机执行。汇编语言比机器语言看起来较直观，不易出错，但一个复杂的程序包括许多指令，写起来还比较繁琐，所以直到现在仍然只有少数专业人员才能使用。

随着计算机软件理论的发展，出现了根据算法编写的“算法语言”（Algorithmic Language）。一个算法语言程序是由许多语句组成的，一个语句可以对应多条机器语言指令。所以用算法语言编写程序更加简单、方便、直观、不易出错。另外，算法语言是“硬件无关”（不依赖于机器）的，所以又称为“高级语言”。

今天，为了使开发出来的软件能容易而迅速地被改变，同时也是为了让更多的非专业人员能够使用计算机语言，出现了一种高生产率的语言——“非过程化语言”。这种语言可以根据用户对自己需求的高级描述直接产生无差错的代码，使语言对用户“友好”，使非专业人员也能独自使用计算机解决自己的问题。

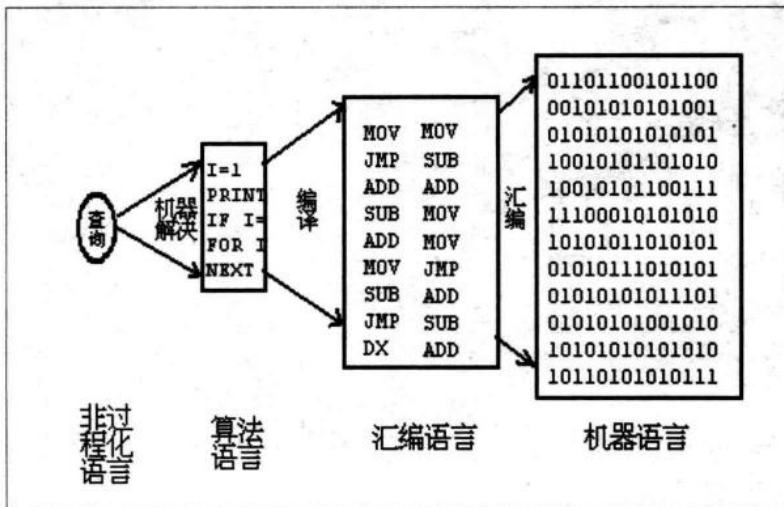


图 1.2 几代程序语言之间的关系

在计算机语言不断发展的同时，与计算机语言同样重要的程序设计思想也在不断发展。由戴克斯特拉（E. W. Dijkstra）和霍尔（C. A. R. Hoare）提出的结构化程序设计思想，是程序设计发展史上的一个重要的里程碑。结构化的程序设计主张所有的程序都由顺序结构、选择结构、循环结构、过程和函数四种基本结构组成，避免了 GOTO 语句所带来的混乱。在程序设计方法上主张采取自上向下的、逐步求精的方法，为多个人同时编程提供了方便。1971 年瑞士的沃斯（N. Wirth）教授提出了世界上第一个结构化的程序设计语言——Pascal。它具有功能强、应用广，编译和运行效率高，可移植、易推广等许多优点。

随着软件设计的日益发展，为了更好地适应多人协作编程，出现了更先进的面向对象编程（Object-oriented Programming）的思想。这种思想在结构化编程的基础上强调了“封装”（Package），要求每个对象做到自身完备、相对独立，通过特定的接口与其他对象联系。由于对象具有自我独立性，所以方便了软件的调试和维护。在面向对象的程序设计中最先提出了“继承”（Inheritance）的概念，使对象可以从自己的“祖先”（Forefather, Ancestor）那里继承原有的属性和方法，提高了资源（代码）的重复利用率，从而提高了软件的开发效率，降低了软件的成本。

但是非常可惜的是，对象并不与传统程序设计和编程方法兼容，如果一个语言只是部分面向对象反而会使情形更糟。除非整个开发环境都是面向对象的，否则对象产生的好处



还没有带来的麻烦多。一些早期的具有 OOP 性能的程序语言如 C++, Pascal, Smalltalk 等, 虽然具有面向对象的特征, 但不能轻松地画出可视化对象, 与用户交互能力较差, 为了设计良好的界面, 早期的 Windows 程序员是艰苦的, 他们要在看不见任何效果的情况下编写大量的代码, 并进行反复的调试才能成功。

在这种情况下, 早期的 Windows 程序员们是值得同情的。一个 Windows 软件的设计过程就像做一场噩梦, 让人筋疲力尽。他们期待着一个伟大的飞跃。而在这种时候 Delphi 的出现就是一个伟大的飞跃, 由于 Delphi 是完全面向对象的, 而且由于它的控件可以方便地画出自身, 所以很容易形成可视化编程, 这使得它成为一种触手可及的, 大大降低软件开发成本的开发工具, 从而具有强大的吸引力。

作为第四代开发工具的 Delphi, 以经过改进的 Object Pascal 为原码, 采用可视化编程技术, 是一个结合了原有的编译器和 IDE (集成开发环境) 的“代码生成器”。由于 Delphi 拥有空前完备的对象包装, 和许多方便快捷的向导 (Wizard) 与专家 (Expert), 以至于用户在使用 Delphi 编程的时候就像在使用“非过程化语言”一样方便。您不必自己建立对象, 只要在提供的程序框架中加入完成功能的代码, 其余的都交给 Delphi 去做。欲生成漂亮的界面和结构良好的程序丝毫不必绞尽脑汁, Delphi 将帮助您轻松地完成。它允许在一个具有真正 OOP 扩展的可视化编程环境中, 使用它的 Object Pascal 语言。这种革命性的组合, 使得可视化编程与面向对象的开发框架紧密地结合起来。对于绝大多数非专业的应用程序来说, 你甚至可以在设置完控件的属性后不写或只写几条非常简单的代码就可以完成你的设计。拥有了 Delphi, 谁都可以大声得说: “软件开发就是这样一件简单的事, 人人都能开发自己的软件!”。

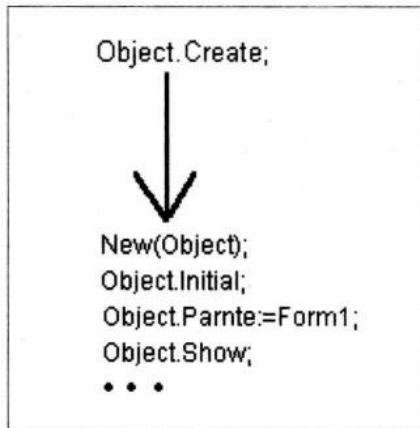


图 1.3 面向对象的程序结构

看到这里, 你也许会问: “用这么简单的编程方式产生的代码效率如何呢? ”。不用担心, 由于 Object Pascal 具有编译快和运行效率高等诸多优点, Delphi 根本不需要用牺牲所产生代码的效率来提高其易用性和编译速度。相反, Delphi 提供了大量由专业人员所编写的例程, 在大多数情况下使用 Delphi 提供的例程甚至会比你自己编写这些例程有更高的效率。

## 1.2 强大的数据库功能

Delphi 在将原来部分对象化的 Object Pascal 扩展为完全对象化的同时，还为数据库提供了强大的支持。Delphi 3.0 除了自带了 Database Desktop, SQL Monitor, BDE Administrator, Data Migration Wizard 等数据库软件外，还提供了许多专门支持数据库的控件。其中除了分别用于数据访问和数据控制的数据 Access 页控件（包括支持 SQL 内嵌的 Query 控件）和数据 Control 页控件外，还提供了能够更简便而且美观的输出数据库数据的强大的 QReport 页和 Decision Cube 页控件。在 Delphi 3.0 中还提供了可以自动生成数据库应用程序框架的 Database Form Wizard，使您可以在一条代码都不用写的情况下实现对一个关系数据库的追加、访问、修改和删除等功能。

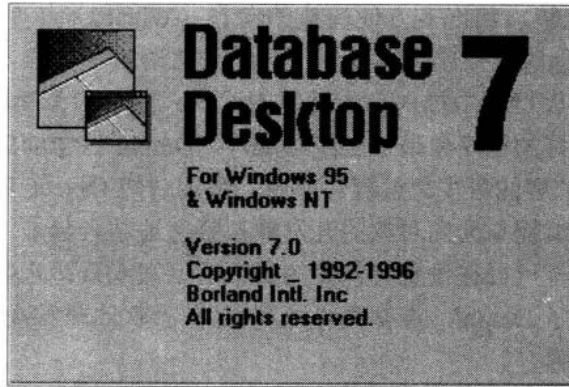


图 1.4 Database Desktop 的标志

可以说，Delphi 在提供了非常方便的，以 Pascal 为原码的可视化编程环境的同时，也提供了丝毫不亚于专业数据库管理软件的强大的数据库支持。这也正是今天很多专业人员选择 Delphi 来开发数据库软件的原因所在。

本书将会在大家面前展开一幅强大的 Delphi 数据库开发环境的图景，并以一些小巧的例子揭示数据库编程的原理，希望本书能将它的知音引入数据库编程的奇妙世界。

## 第2章 揭开面纱：一个简单的例子

本章将利用 Delphi 开发一个简单的例子，使读者对 Delphi 有一个初步的认识。这是一个很简单的程序，用 Edit 控件向 ListBox 控件里添加数据，以帮助读者了解 Delphi 的编程风格。在这里我要向读者介绍一下什么是控件，控件是一种集成化的工具，就好比一种有智能的机器人一样，他们每人都有一种特殊的技艺，你的任务就是根据你的需要，叫他们去干他们各自所会的任务，然后把这些任务按照计划组合起来。过去人们开发软件，不使用这种集成化的东西，什么细枝末节都要自己开发，开发速度极其缓慢，自从 Delphi 横空出世后，一切就都改变了。有时用鼠标点几下，一个可以和微软的画笔或记事本相媲美的程序就编成了。现在就让我们跨进这个奇妙的世界吧！

### 2.1 窗体——你的舞台

首先，从 File 菜单中选择 New Application，建立一个新的工程，这时屏幕上将出现以下窗体（图 2.1）。

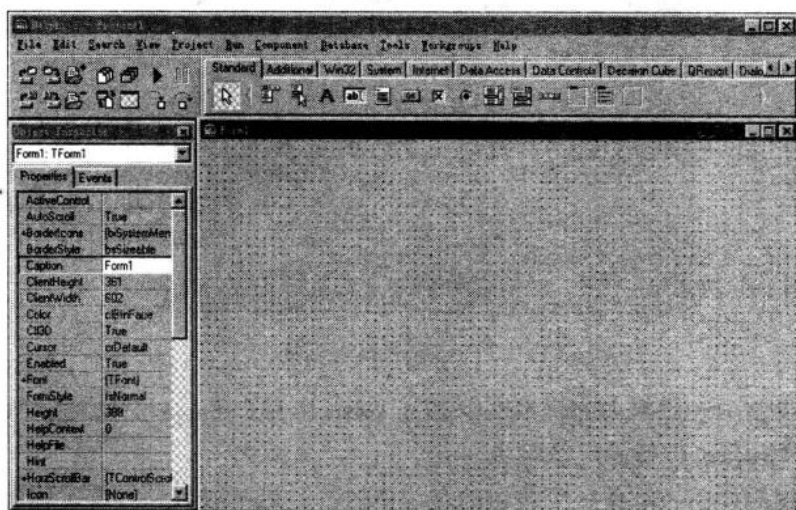


图 2.1 一个空的窗体。

如果把一个 Delphi 程序员比作一个画家的话，那么窗体（Form）就是一张空的画布，在这张画布上，您可以充分发挥自己的想象力，没有做不到，只有想不到！

### 2.2 控件的放置

下面，我们要在窗体上放置一个 Edit 控件，一个 ListBox 控件和三个 Button 控件。

首先,从控件面板(Component Palette)上选择 Standard 页。先从中选取 Edit 控件(Edit),然后把鼠标光标移动到窗体的左上角,并单击鼠标左键,一个 Edit 控件就放置到了窗体上。

然后,用同样的办法可以将一个 ListBox 控件放置在窗体上,使得 ListBox 控件位于 Edit 控件的下方。为了调整 ListBox 控件的大小,可以单击窗体上的 ListBox 控件,并把鼠标移动到 ListBox 控件右下的黑点上,按住左键并拖动鼠标,就可以改变 ListBox 控件的大小了。

现在,我们要在窗体上放置三个按钮,先按住 Shift 键,然后再从选中控件面板的 System 页上选定按钮控件(Button),这样就可以连续在窗体上放置同一类型的控件。分别把三个按钮放置在窗体的右侧,然后在控件面板上选择最左边的光标按钮就可以转换到结束位置。

现在,所有的控件都已经放置完毕,但看上去可能有一些参差不齐,现在我们的工作就是将这些控件对齐。首先我们要使 ListBox 控件和 Edit 控件的左边对齐,选中 Edit 控件后按住 Shift 再左单击 ListBox 控件将其同时选中或者干脆用鼠标将 Edit 控件和 ListBox 控件同时选中(直接圈入即可),接着从 Edit 菜单中选中 Align 命令,弹出了以下的对话框。

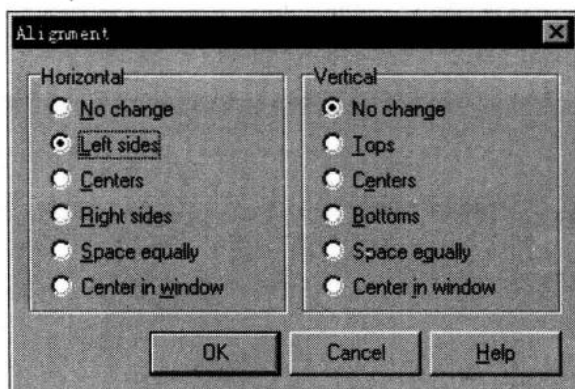


图 2.2 Alignment 对话框

分别在水平方向和垂直方向栏中选中 Left sides 和 No changes 然后选中 OK,就可以使 Edit 控件和 ListBox 控件的左边对齐。

然后我们用同样的方法使三个按钮的左边也对齐。这样放置控件的工作就结束了。

## 2.3 设置控件的属性

现在我们要对各个控件的属性进行设置,使得 Edit 控件初始时显示空白,并且设置三个按钮的标题。

要清除 Edit 控件的文本,先要在窗体窗口中选中 Edit 控件。Object Inspector 窗口就会显示 Edit 控件的各个属性。要改变 Edit 控件的文本,可以在 Object Inspector 窗口中单击 Text 属性项,接着就可以改变 Text 属性的值了。只要简单地将 Text 属性的值设置成空