

OSBORNE

Delphi nuts

程序员

Experienced

Programmers

必读

GARY CORNELL | TROY STRAIN



科学出版社
龍門書局



TP312

K22

387130

Delphi 程序员必读

康奈尔 斯特兰
- Gary Cornell Troy Strain 著

周关边 译

万博 校



科学出版社

龙门书局

1995

(京)新登字 092 号

JS195/14

内 容 简 介

Delphi for Windows 是 Borland 公司最新投放市场的先进产品。较同类产品 Visual Basic 4.0 和 PowerBuild 4.0 而言, Delphi for Windows 有许多优越之处, 是美国五家 PC 最著名杂志 95 年评比的最佳可视 Windows 应用程序开发系统。本书介绍 Delphi 环境的使用法, 说明利用 Delphi 设计 Windows 和 Windows 95 应用程序的技术。本书内容翔实, 实例典型, 讲述由浅入深, 适合大专院校师生、计算机爱好者使用。

需要本书的读者, 可直接与北京海淀 8721 信箱书刊部联系, 电话 2562329, 邮编 100080。

Gary Cornell & Troy Strain

DELPHI NUTS & BOLTS: FOR EXPERIENCED PROGRAMMERS

McGraw-Hill, 1995

版 权 声 明

本书英文版名为“Delphi Nuts & Bolts: For Experienced Programmers”, 由 McGraw-Hill 公司出版, 版权归 McGraw-Hill 公司所有。本书中文版由 McGraw-Hill 公司授权出版。未经出版者书面许可, 本书的任何部分都不得以任何形式或任何手段复制或传播。

Delphi 程序员必读

Gary Cornell Troy Strain 著

周关边 译

万 博 校

责任编辑 汪亚文

科学出版社

龙门书局 出版

北京东黄城根北街 16 号

邮政编码: 100717

双青印刷厂 印刷

新华书店北京发行所发行 各地新华书店经售

*

1995 年 8 月北京第一版

开本: 787×1092 1/16

1995 年 8 月第一次印刷

印张: 11 3/4

印数: 1—5000 册

字数: 259 000

ISBN 7-03-005022-3/TP·513

定价: 18.00 元

鸣 谢

编写一本书的最佳时刻之一是作者要对那些帮助过他的人表示感谢的时候,因为几乎没有一本书完全是由作者自己来完成的(当然,本书也不例外)。首先,要感谢的是 Osborne/McGraw-Hill 的工作人员,他们的耐心、富于献身精神的工作和所给予的帮助已经超出一种职业责任的程度了。向 Jeff Pepper, Ann Wilson, Janet Walden, Vicki Van Ausdall, Madhu Prasher, Kelly Vogel, Daniela Dell'Orco 和 Marcela Hancik 致意,谢谢你们! 作为一名技术审查人员, Steve Nameroff 所做的一切已经远远超出了作者所预期的(或者说所期望的),在此我们深表谢意!

我们还要感谢那些在 Borland 公司里创立 Delphi 的人们(遗憾的是,我们不知道他们的姓名)。Delphi 确实是一件很棒的产品。Nan Borreson 负责与 Borland 公司的作者联系,她对一些古怪的要求所赋予的耐心和她一贯的周到的帮助也超出了她的职责所应该做的。如果没有她所提供的资料,这本书也是无法完成的。

Gary Cornell 要感谢他的所有朋友们,他们对他的一些古怪的想法和偶尔的脾气都给予了宽容和理解。特别要感谢 Bruce, Caroline 和 Kurt, 没有他们的友谊和帮助, Cornell 负责的本书的那部分内容也是不可能写成的。

Troy Strain 要感谢他的妻子 Tracy, 要列举感谢的原因实在太多了,但他首先要感谢妻子对他所熬过的许多不眠之夜的理解。他还要感谢他 18 个月的儿子 Nathan, 小家伙最近在他父亲的计算机上发现了 Reset 键。还要特别感谢 Drew 和 Devri Bayless 的帮助。最后 Troy 要感谢他的弟弟 Robbie Ricker。

引 论

Delphi 是一种在很短时间就打入市场的最为激动人心的程序语言产品。公众的舆论很少会为一件产品引起如此的轰动。其诱人之处究竟是什么?确切地说,什么是 Delphi,它可以做哪些工作?事实上,这是一个使用简便,但对开发 Windows 应用软件特别有力的工具,而且它使用了现代的编译器技术产生极快的可执行程序。Delphi 甚至具备真正的面向对象编程和客户-服务器数据库开发这样的功能。

如果这么说似乎还不足以证明各种有关宣传是真实的,那么请记住,有了 Delphi 之后, C 和 C++ 方面的专门知识即便是在开发最为复杂的应用软件时也是所需甚少的了(如果还需要的话)。使用者勿需再是一个 C 语言的编程专家,并且也勿需再用价值大约 20 磅的文件来开发一个速度极快的 Windows 应用软件了。

1. 本书简介

本书是针对高级程序员的 Delphi 简明指南。我们认为读者已经有了若干编程的经验(无论是用哪种语言编程),并且熟悉循环和判断结构的一些概念。总之,本书的宗旨就是向读者展示 Delphi 的优点,并告诉读者如何使 Delphi 成为自己编程工具的一部分。

更确切地说,我们要编写这部书时的愿望是要将 Delphi 的精华提取出来并清晰、简明地传授给读者。作者们尽自己的努力去强调那些掌握 Delphi 编程所必需的思想方法,所以即使是那些对传统的编程语言非常熟悉的人阅读本书也会有所收获。之所以采用这一方法是因为要将 Delphi 纳入旧的编程语言的框架中最终是徒劳的。在旧的范例中的思考方式是无法利用 Delphi 功能的优点的。

假定读者在知道是什么使 Delphi 保持活力之后,可以在需要时参考联机帮助以及随 Delphi 一起提供的手册。我们并没有太多的例子提供给读者,作者假设读者在此前已经编写过很多程序了。我们认为一旦读者熟悉 Delphi 的思维方式,就可以看懂 Delphi 所提供的范例程序。这样在需要时就可以参考这些程序。

2. 本书的编排

根据读者的基础和需要,本书也可以有很多种用途。对于熟悉 Visual Basic 的人们,可能就想跳过前面的章节而把注意力集中在讨论 Delphi 基础语言的后面章节中。对于熟悉 Pascal 和 C 但不熟悉像 Visual Basic 这种事件驱动语言的读者,可能就需要在前面章节中多花些工夫了。但后面的章节则可以简要地看一看(不管怎样,我们建议读者通读所有章节,因为无论读者具有何种语言背景,都会发现针对 Delphi 的内容)。

下面是 13 个章节的简要说明:

- (1) 第一章简要介绍了 Delphi,并给出了两个简单的例子。
- (2) 第二章描述了 Delphi 的环境。

(3) 第三章开始介绍定制窗口的概念,这是每一种 Windows 应用软件的核心(当然也是 Delphi 的核心)。读者将了解如何向定制窗口(称为窗体)中加入和操作基本的 Delphi 的对象,诸如命令按钮、输入文本所在区域、标签、时钟显示以及网格等等。

(4) 第四章和第五章则概括了在 Delphi 中用到的 Pascal 版本。

(5) 第六章简要介绍了面向对象编程。

(6) 第七章讨论了“异常”(exceptions),这是 Delphi 处理运行时错误的非常尖端的方法。

(7) 第八章讨论了调试技术。

(8) 第九章讨论了处理文件的技术。

(9) 第十章向读者介绍动态数据交换(DDE)和对象链接与嵌入(OLE)。

(10) 第十一章介绍了图形。因为 Microsoft Windows 是一个图形环境,Delphi 在这方面的功能是非常强大的。

(11) 第十二章讨论更为先进的用户接口功能,诸如操作一个鼠标的技术。

(12) 第十三章简要介绍 Delphi 的基本数据库特征。

说明 本书不讨论如何使用 Delphi 的开发客户服务器应用软件的问题。

3. 本书的有关说明

键名在文本中用的是大写字母。例如,CTRL 和 ALT 键就像现在这个样子。箭头及其他方向键则拼写出来,仍然用大写字母。如需要按下右箭头键时,就会说“按右箭头键(RIGHT ARROW)”。

当需要用组合键来激活一个菜单项时,键名用一分开,并且各键名均用大写字母。例如,“按下 CTRL-A-B”指示读者应当按住 CTRL 键再按下 A 和 B 键。另一方面,ALT-F,P 是指示按下 ALT 和 F 键,然后按下 P 键——当然这时无须再按住 ALT 键了。

DOS 命令、文件名以及文件扩展名都以大写字母出现:COMMAND.COM,.TXT,等等。Microsoft Windows 在大多数情况下都被简称为 Windows。本书的作者们试图按照 Delphi 自身的约定来写。菜单选择用一个竖条来分开它们,例如,“选择 Run|Run”。内建函数与过程则用每个单词头一个字母大写的方式,诸如 Font,TextOut 等等。

最后,本书中有大量的提示和说明,前面均有不同图标指示。例如,一般的提示如下:

提示 因为本书是为那些资深的编程者提供的,我们没有提供很多的代码。Delphi 提供数以兆字节的程序是一种很好的补充。

还有一些提示是针对 Visual Basic 编程者的,一般像下面这样:

Visual Basic 提示: Delphi 在它的 DOC 目录中含有一个称为 VB2DELPHI.WRI 的文件,读者一旦对 Delphi 更熟悉之后就应当仔细地研究一下该文件。

最后,对那些使用过 Borland 早期 Pascal 语言的编程者们也有提示,像下面这样:

Pascal 提示: 读者不要试图将一个用 Borland 的 ObjectWindows 编写的程序与 Delphi 的 Visual Component Library(可视成员库)组合在一起。

目 录

鸣谢

引论

第一章 开始	1
1.1 为什么要开发 Windows 和 Delphi	1
1.1.1 如何开发一个 Delphi 的应用软件	2
1.2 Delphi 的操作	3
1.2.1 一个“Hello World”类型的程序	5
1.3 一个功能更强大的工程设计	9
第二章 Delphi 的编程环境	12
2.1 主屏幕的综览.....	12
2.1.1 标题栏.....	13
2.1.2 菜单条.....	13
2.1.3 加速条 SpeedBar	14
2.1.4 成员模板.....	15
2.1.5 初始的窗体窗口.....	15
2.1.6 对象观察窗口.....	16
2.1.7 代码编辑器(单元)窗口.....	16
2.1.8 工程设计管理器窗口.....	16
2.1.9 对齐工具板窗口.....	17
2.1.10 帮助系统	17
2.2 File(文件)菜单详述	18
2.3 编辑.....	19
2.3.1 编辑菜单.....	20
2.4 搜索(Search)菜单	21
2.5 察看(View)菜单	21
2.6 选项(Options)菜单	22
2.7 工具(Tools)菜单	23
2.8 加载与运行程序.....	23
第三章 设计一个窗体:成员、菜单以及事件	25
3.1 Object Inspector 的操作.....	25
3.1.1 简单的编辑器.....	26
3.1.2 下拉清单	26
3.1.3 对话框编辑器.....	26
3.1.4 嵌套的属性编辑器.....	27
3.2 窗体.....	27

3.2.1	窗体属性	27
3.2.2	窗体事件	30
3.3	成员	31
3.3.1	成员的综览	31
3.3.2	向窗体中加入成员	35
3.3.3	已存在成员的操作	36
3.3.4	成员的对齐	38
3.4	一些个别的成员	38
3.4.1	焦点属性与事件	39
3.4.2	标签	39
3.4.3	编辑框	40
3.4.4	Memo 成员	40
3.4.5	按钮与位按钮	40
3.4.6	ListBox 与 ComboBox 成员	41
3.4.7	CheckBox 与 RadioButton 成员	42
3.4.8	Timer(定时器)	42
3.4.9	String Grid 和 DrawGrid 成员	43
3.5	设计菜单	44
3.5.1	菜单设计工具的操作	46
3.6	窗体的 ASCII 表达	48
第四章	Delphi 编程基础	50
4.1	Delphi 中的语句	51
4.1.1	注释语句	51
4.2	运行与结束程序	52
4.2.1	Halt 语句	52
4.3	属性设置	52
4.3.1	布尔属性	53
4.3.2	With 语句	53
4.4	变量	54
4.4.1	变量类型	55
4.4.2	变量赋值	56
4.5	常量	57
4.5.1	变量与常量的范围	57
4.5.2	过程间的数值共享	58
4.5.3	保存数值	59
4.6	串	60
4.7	数字	61
4.7.1	数字的运算	61
4.7.2	Delphi 中数字的另一些说明	63

4.8	类型转换	63
4.9	重复操作(循环)	63
4.9.1	确定的(For)循环	64
4.9.2	非确定性循环	65
4.10	作出判断	66
4.10.1	循环语句与 If—Then 语句一起使用	67
4.10.2	Case 语句	68
4.11	关于运行时对象操作的进一步说明	68
4.11.1	方法	69
4.11.2	应用软件对象	69
4.11.3	屏幕对象	70
4.11.4	Canvas 对象	71
4.11.5	Printer 对象	61
4.12	信息框与输入框	72
4.12.1	InputBox 函数	74
4.13	多窗体的应用软件	75
4.13.1	窗体的保存	76
4.13.2	去除工程中的一个窗体(及其相关的单元)	77
4.13.3	指定工程的主窗体	77
4.13.4	运行时的窗体动作	77
第五章	高级编程技术	80
5.1	类型重述	80
5.1.1	顺序数据类型	80
5.1.2	集合	81
5.2	数组	82
5.2.1	一维数组	83
5.2.2	串列表	84
5.2.3	多维数组	85
5.2.4	记录	86
5.3	指针	88
5.3.1	指针与内存管理	89
5.3.2	零结尾串和 PChar	90
5.4	函数与过程	91
5.4.1	内建的子程序	92
5.4.2	Windows API 的使用	93
5.5	用户定义的函数与过程	94
5.6	递归	97
5.7	工程与单元的管理	98
5.7.1	单元的进一步说明	99

5.8 编译器	100
5.9 goto	102
第六章 对象.....	104
6.1 开始	104
6.1.1 OOP 的词汇表	105
6.2 在 Delphi 中建立一个对象	107
6.3 可视成员库对象	109
6.3.1 窗体声明的私有与公用区域	110
6.4 在运行时建立 Delphi 对象	113
6.4.1 在运行时建立新的可视对象	113
6.5 ObjectBrowser	114
6.5.1 编译器选项及它们对于对象浏览器的影响	115
6.5.2 深入检查	116
第七章 出错与异常的处理.....	117
7.1 被保护的代码块	117
7.2 运行库的异常	118
7.2.1 RTL 异常	118
7.2.2 处理异常的类别	120
7.2.3 再提出一个异常	120
7.3 保护源的分配	121
7.4 用户定义的异常	122
7.4.1 声明一个异常(Exception)对象	122
7.4.2 静异常	122
第八章 测试与调试工具及技巧.....	123
8.1 调试工具及其功能	123
8.2 编写易于调试的程序	124
8.2.1 逻辑故障	124
8.3 加入调试信息	125
8.4 计算/修改对话框.....	125
8.4.1 再谈利用“计算/修改”对话框的调试.....	126
8.5 跟踪进入	126
8.6 暂时中止程序的运行	126
8.6.1 监视	127
8.7 关于调试的最后几点说明	128
8.7.1 事件驱动故障	128
8.8 编程风格	128
第九章 文件操作	130
9.1 文件命令	130
9.1.1 其他文件和输入输出函数	131

9.2 文件系统成员	133
9.2.1 文件列表框构件	133
9.2.2 目录列表框成员	134
9.2.3 驱动器组合框成员	134
9.2.4 筛选组合框成员	134
9.2.5 将各文件成员组合起来	134
9.3 文本文件	135
9.3.1 从文本文件中读出信息	136
9.3.2 从文本文件中读取数字	137
9.4 格式化文件	138
9.4.1 格式化文件中的记录	139
9.5 非格式化文件	139
9.6 命令行信息	140
第十章 与其他 Windows 应用程序通讯	141
10.1 剪贴板	141
10.1.1 剪贴板格式和图形传送	142
10.2 动态数据交换(DDE)	143
10.2.1 在设计阶段建立 DDE 链接	143
10.2.2 用代码建立 DDE 链接	144
10.2.3 DDE 构件的数据成员,成员函数及事件	145
10.3 OLE	147
10.3.1 使用 OLE	148
10.4 在设计阶段技术使用 OLE2	149
10.4.1 特殊粘贴(Paste Special)	150
10.4.2 OLE 数据成员	150
10.5 通过代码创建 OLE	150
第十一章 图形	153
11.1 图形对象	153
11.2 图形学基础	154
11.2.1 图像对象的进一步讨论	155
11.3 Canvas 图形基础	155
11.3.1 颜色	155
11.3.2 黑点	155
11.3.3 画图形	156
11.3.4 笔	156
11.3.5 刷子	157
第十二章 高级用户接口	158
12.1 向工程中增加工具棒和状态条	158
12.1.1 创建工具棒或状态条	158

12.2 通用对话框.....	159
12.2.1 通用对话框的应用.....	160
12.3 MDI 表	162
12.3.1 Windows 菜单和 Arrange 函数.....	162
12.4 再谈 Delphi 的显示操作——Z 顺序	163
12.5 监视鼠标.....	163
12.5.1 鼠标事件处理句柄.....	164
12.5.2 拖动与定位操作.....	166
12.6 帮助系统.....	168
第十三章 数据库综述.....	170
13.1 现代数据库通用词汇.....	170
13.2 Delphi 数据库结构	171
13.2.1 数据访问对象构件.....	172
13.2.2 数据控件页上的对象构件.....	173
13.3 数据库窗体专家.....	173
13.4 最后说明.....	175

第一章 开 始

这一章将给读者有关 Delphi 及 Delphi 编程后面的一些思想的综览。读者还将看到建立一个简单的 Delphi 程序是何等的容易。本章后面所介绍的两个程序对使 Delphi 变得功能强大的事件驱动和面向对象模型作了说明。本章(包括本书)没有提及如何安装、如何启动 Delphi, 以及如何在 Windows 环境中操作窗口等等——我们认为读者应该知道这些内容。但是, 我们并不认为 Delphi 或 Windows 编程方面的任何专业技术是为读者已经掌握的——只需具备在任何环境下一般水平的编程经验即可。

注意 在 Help 菜单中的 Interactive Tutors 是对本章材料的很好的补充。

1.1 为什么要开发 Windows 和 Delphi

图形用户界面, 或称为 GUIs(发音为“gooies”)已经使微型计算机工业产生了变革。它们证明了这样一句俗语“一幅图像胜过千言万语”仍然是真理。不再是 DOS 用户们过去看到的(而且有些人已经久已畏之了)C: >提示符, 现在的用户所看到的是一幅桌面系统的景象, 里面充满了图标以及使用鼠标与菜单项的程序。可能从长远来看, Windows 应用软件的“样子”与应用软件给操作者的“感受”相比其重要性就不如后者了。Windows 应用软件一般都有一个前后一致的用户界面, 这样用户就可以在应用软件的掌握上多花些工夫, 而不必过多去考虑在菜单上与对话框里哪个键是做哪样工作的了。

在程序员们对 GUIs 感到既冗长又混乱的时候, 初学者们却似乎对它们情有独钟, 于是人们希望 Windows 的程序是基于 GUIs 模型的(因而具有良好的外观和操作体验)。因此, 如果要开发用于 Windows 的程序, 开发者们就需要有一个工具用于高效率地开发基于 GUI 的应用软件。

有很长一段时间并没有这种工具出现。因为 Visual Basic 是在 1991 年才问世的, 因此开发 Windows 的应用软件就比开发 DOS 应用软件要困难的多。编程者要考虑很多的问题, 诸如鼠标在做什么操作, 用户将菜单置于哪里以及他或她是单击还是双击一个指定处等等。开发 Windows 应用软件要求有熟练的 C 编程知识, 而完成一项简单的任务也要数页的代码, 即便是专家也会犯难(Microsoft Windows Software Development Kit 是 C 编译器所需要附加的, 它有 9 磅半重)。

Visual Basic 在它的头三个版本中就开始改变这一过程了; Delphi 则将产生一场革命。复杂的 Windows 应用软件现在开发起来只需原来所用时间的一个零头。编程错误也不像过去那样经常发生了, 即便发生了错误, 那也只是一些容易发现和更正的错误。简言之, 有了 Delphi, Windows 的编程就不仅仅是高效率的而且是一项很有意思的工作了(绝大多数情况下就是如此)。

Delphi 的一些优于 Visual Basic 前三版的地方是:

- (1) 用 Delphi 开发的应用软件比 C 或 C++ 开发的应用软件运行速度快。

(2) 有了 Delphi, 编程者可以建立真正的可执行软件, 包括 DLLs (DLL 系指动态链接库, 它是 Windows 编程的基础)。

(3) 编程者可以按照面向对象编程的范例来建立可重复的对象 (见第六章)。

注意 尽管 Delphi 的客户/服务器版本在本书未提及, 但是 Delphi 的 Client/Sever 同样使得这种软件的开发更为容易了。

1.1.1 如何开发一个 Delphi 的应用软件

要开发一个 Delphi 应用软件, 第一步是先计划一下用户将看到的内容, 换句话说, 就是设计屏幕。例如, 需要什么样的菜单呢? 应用软件所用的窗口该有多大? 要有多少个窗口? 用户要单击以激活应用软件的命令按钮应当放在哪里? 应用软件是否要有空间以便输入文本 (即编辑框和记忆字段)?

在 Delphi 中, 编程者置于他或她所设计的窗口中的对象就称为成员。读者以 70 个以上的成员开始自己的布置, 并且可以通过购买或自己建立成员来增加成员。关于 Delphi 提供成员的概述请参见第三章。

注意 如果读者对面向对象编程有些熟悉, 那么就要小心, 因为 Delphi 成员实际上是面向对象编程 (OOP) 意义上的对象 (有关面向对象的编程以及 Delphi 的成员是如何符合 OOP 原则的更多说明请参见第六章)。

Delphi 与 Visual Basic 都具备让编程者设计屏幕的功能。编程绘制用户界面, 几乎就像是在用一个画笔程序一样。此外, 当编程者绘制完界面之后, 放置于空白窗口中的命令按钮及其他成员会自动识别用户的操作, 诸如鼠标移动和按钮单击等。Delphi 还带有一种菜单设计功能, 该功能使得普通菜单与弹出式菜单的建立就是一个捕捉的过程, 与 Visual Basic3 不同的是, Delphi 使重复使用编程者先前建立的界面特征变得容易了。

只是在设计了界面之后, 一切工作才又变得像传统的编程一样了。Delphi 中的成员将识别诸如鼠标单击这样的事件; 对象对事件的响应取决于所编写的代码。编程者几乎总要编写代码以使成员去响应事件。这就使得 Delphi 编程从根本上与 Turbo Pascal 早期版本中的面向过程编程不同了。

常规编程语言中的程序是从上往下执行的。对于旧的编程语言来说, 运行从第一行开始, 然后按照需要随着程序流方向移动。而一个 Delphi 的程序通常不是这样的。Delphi 程序的核心是一组独立的代码块, 它们只能由要求它们识别的事件所激活, 也只能响应这些事件。这是基本的变化。现在就不是编程者设计一个程序来做编程者认为要发生的事, 而是由用户来支配的了。

Delphi 中告诉编程者如何响应鼠标单击这类事件的编程代码是从 Delphi 称为“事件过程”的程序段中开始的。一个事件过程就是只为响应某个外部事件而执行的代码段。在几乎所有的情况下, Delphi 程序执行要么是一个事件过程, 要么就是为某个事件过程所用来帮助过程完成其工作的。实际上, 为了强调 Delphi 与普通编程语言的根本不同, 主要在于文件中使用了术语“工程”而不是“程序”来指代构成 Delphi 应用软件的程序代码与用户界面的组合。

下面总结一下设计 Delphi 应用软件的步骤:

(1) 定制用户看到的窗口。

(2) 确定窗口中成员识别什么事件。

(3) 对那些事件编写事件过程(以及使那些事件过程工作的辅助过程)。

下面是当应用软件运行时所发生的情况:

(1) Delphi 为每个成员可识别的所有事件监控窗口和每个窗口中的成员(这些事件如鼠标移动、单击、击键等等)。

(2) 当 Delphi 测到一个事件之后,如果没有对事件的内建的响应,Delphi 即检测应用软件以检查是否已经为该事件写了一个事件过程。

(3) 如果已经编写了事件过程,Delphi 就执行组成事件过程的代码,并返回步骤 1。

(4) 如果没有编写该事件过程,Delphi 就等待下一事件再返回步骤 1。

这些步骤将持续不断地循环,直到应用软件结束为止。通常一个事件必须在 Delphi 要做任何操作之前发生。这样事件驱动程序就是反应式的而不是主动式的了——如此使得它们对用户更为友善。

由于 Delphi 的编程语言是基于 Object Pascal 的,通过使用现代模态和面向对象技术就很容易建立超大型的程序。Delphi 还提供一种高级的出错处理技术,以防止最终毁坏应用软件。Delphi 编译器是非常快的——这意味着任何对纠正例行编程和排印错误所需的变化都成了一次捕捉,这些错误在制作一个应用软件时是很普遍的。在编程者开发一个应用软件时就有个广泛的联机帮助系统提供快速的参考。

运行 Delphi 所需的环境

Delphi 是一具复杂的程序,要与 Delphi 一道工作,需要下面这些条件:

(1) 至少是 386CPU,6 兆 RAM(实际上是一个快速的 486 或奔腾处理器,并且至少有 8 兆的 RAM)。客户/服务器编辑需要 8 兆,不过实际上至少得要 16 兆。

(2) 至少有 30 兆的硬盘自由空间(客户/服务器编辑的安装则需要 90 兆!)

(3) 鼠标或与鼠标兼容的指示设备。

(4) Microsoft Windows 3.1 或以增强模式运行。

注意 尽管可以在 Windows NT 或 Windows 95 下运行 Delphi 1.0,它也只是作为 16 位的应用软件来实现的。

1.2 Delphi 的操作

在这一节中读者将看到使用 Delphi 来制作两个程序的每一个过程。这两个程序一个是修改传统的“Hello World”程序,另一个则是位图观察器。观察器特别能说明 Delphi 在开发 Windows 应用软件的效率——它只用了两行代码!在 C 或 C++中相同数量的功能则需要百来行的代码。

在启动 Delphi 时,初始屏幕的样子就像图 1.1 所示的那样。在下一章里读者将看到更多的 Delphi 环境,但是现在还是先不要过多去考虑环境问题只是按照书中步骤走或许会更容易一些。

请注意屏幕中央空白的窗口,上面有网格点。这是读者将要定制的窗体。用网格来调整成员如命令按钮和列表框等在屏幕上的位置(读者将在第三章里学习这方面的内容)。当运

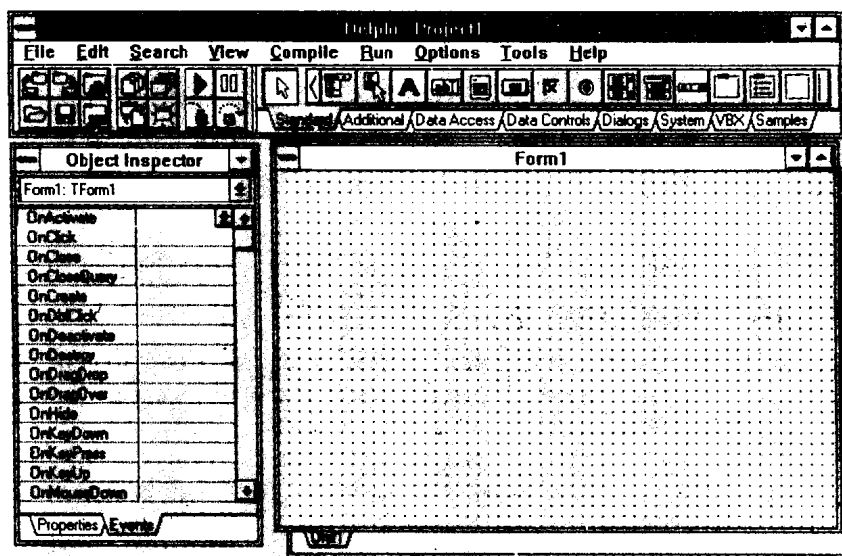


图 1.1 最初的 Delphi 屏幕

行该工程时(或是编译该工程,这样就能够脱离 Delphi 环境,单独运行),窗体就成为用户所看到的窗口。

在空白窗体的顶部是带有标题的标题栏(“标题”(Caption)是 Delphi 专指在窗体的标题栏中出现的内容)。现在这个窗体就称为 Form1,在开始一个新的工程设计时,Delphi 就将其作为默认的标题。

在 Form1 窗口的左边是对象观察器(Object Inspector)(如果读者没有看见它,就按 F1 或 ALT+V,O 使其出现)。Object Inspector 是用来定制放置在窗体当中的不同成员和窗体自身的。

现在,请把注意力集中在名为 Form1 的窗体中。读者应当能自如地改变该窗体的大小和位置。在很多 Delphi 应用软件中,窗体的尺寸与位置在完成设计时候(通常称为设计时)的样子就是用户在运行时所看到的样子。这并不是说 Delphi 就不让读者改变大小与位置来设计工程了(见第四章),实际上,Delphi 根本的特征就是它可以动态地变化以响应用户事件的能力。

一种对所有的 Microsoft Windows 应用软件都通用的调节窗体大小的方法是先在窗体内用鼠标点一下,以便将其激活(读者应该能够区分哪个窗口是激活的,因为被激活窗口的标题栏总是强调显示的)。然后移动鼠标到窗体的“热点”之一上(在一个窗体中,热点就是窗体的边或角)。鼠标箭头即变为双头箭头。在这一点上就可以拖动窗体以改变其大小和形状了。类似地,要移动窗体时,可以单击标题栏然后将窗体拖到一个新的位置。窗体的大小与位置就是 Delphi 称为窗体属性的例子。

要启动第一个示范软件时,要这样做:

(1) 通过在某些热点上操作窗体或拖动窗体在屏幕上移动来改变窗体的默认的大小、形状和位置。

U4P111A3J'RXZ

(2) 通过从 Run 菜单中选择 Run 来运行工程设计(在这样做的时候,Delphi 实际上是在操作者的计算机中建立一个可执行文件然后运行该文件)。

注意,读者所看到的是一个普通的 Windows 窗口,其大小、形状以及位置与设计时所留给窗体的是一样的。

下面,请注意在运行这个新的工程设计时,弹出的窗口有标准的 Windows 特征,如可调整大小的边框,一个控制框(在左上角)以及最大化和最小化按钮(在右上角)。

这显示了 Delphi 的一个最重要的特征:设计者窗体已经具备 Windows 下的特征,而无需设计者做什么工作。

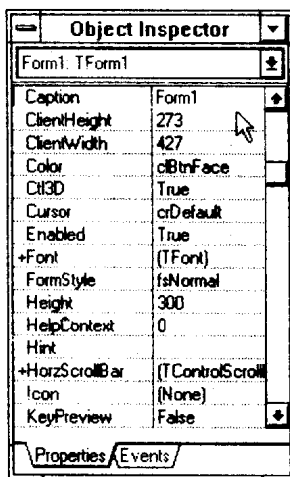
按下 ALT-F4 就返回开发环境中(注意,读者自己的应用软件自动地响应这种关闭 Windows 应用软件的标准方式。这说明很重要的一点,在很多情况下,不需要设计者做任何事情,Delphi 应用软件可以像 Windows 用户所期望的那样操作)。

1.2.1 一个“Hello World”类型的程序

现在我们将告诉读者如何编写一个工程设计,它将显示“Hello new user”在一个窗口的标题栏中以响应用户单击鼠标的操作。我们将使其稍稍复杂化一点,使这个窗口的标题栏开始时显示“Waiting for a click!”

下面是几个必要的步骤:

- (1) 移动鼠标至 Object Inspector 上并在标有 Caption(标题)的项上点一下。



- (2) 输入语句:Waiting for a click! 到 Object Inspector 的 Caption 项的右列中。

我们所做的工作就称为“设置一个窗体的属性”。如果按下 F9 或选择 Run/Run 表示运行这个工程设计(记住那是 Delphi 对处于开发阶段的应用软件的称呼),那么将会看到图 1.2 所示的那样。