

CMP

DELPHI 2 Developers' Solutions

DELPHI 2

开发人员指南

(美) Nathan Wallace 著
Steve Tendon

冯博琴 冯 岚 徐景民 等译

计算机软件开发
与程序设计
系列丛书



机械工业出版社

西蒙与舒斯特国际出版公司

计算机软件开发与程序设计系列丛书

DELPHI 2 开发人员指南

(美) Nathan Wallace 著
Steve Tendon

冯博琴 冯岚 徐景民 等译
李波 审校

机械工业出版社
西蒙与舒斯特国际出版公司

TP31

本书是为具有一些 Delphi 2 程序设计经验而又亟待提高的读者编写的。本书包含大量实例程序，内容丰富、新颖，书中介绍的八十余个“如何…”，可直接指导实际工作。

Nathan Wallace, Steve Tendon ; Delphi 2 Developers' Solutions

Authorized translation from the English language edition published by Waite Group Press.

Copyright 1997 by The Waite Group, Inc.

All rights reserved. For sale in Mainland China only.

本书中文简体字版由机械工业出版社和美国西蒙与舒斯特国际出版公司合作出版，未经出版者书面许可，本书的任何部分不得以任何方式复制或抄袭。

本书封面贴有 Prentice Hall 防伪标签，无标签者不得销售。

版权所有，翻印必究。

本书版权登记号：图字：01-97-0937

图书在版编目 (CIP) 数据

Delphi 2 开发人员指南/ (美) 华莱士 (Wallace, N.) . 坦东 (Tendon, S.) 著；冯博琴等译. --北京：机械工业出版社，1997.9

(计算机软件开发与程序设计系列丛书)

ISBN 7-111-05882-8

I . D... II . ①华... ②坦... ③冯... III . 软件工具, Delphi IV . TP311.56

中国版本图书馆 CIP 数据核字 (97) 第 20688 号

出 版 人：马九荣 (北京市百万庄南街 1 号 邮政编码 100037)

责任编辑：温莉芳

昌平环球印刷厂印刷 · 新华书店北京发行所发行

1997 年 9 月第 1 版第 1 次印刷

787mm×1092mm 1/16 · 56 印张 · 1371 千字

印数：0 001—5 500 册

定价：95.00 元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

译 者 序

回顾程序开发历史，我们发现近几年出现了许多革命性的技术，如面向对象技术、程序开发可视化技术、基于应用程序框架的开发技术等。融合了这些新技术的开发软件包使程序员的开发效率提高了好几倍。从此一种新的软件开发方法也应运而生了，这就是 RAD (Rapid Application Development) ——快速应用开发。一个成熟的 RAD 开发包应具备：

- 可视化及事件驱动开发技术
- 面向对象及基于应用程序框架技术
- 支持组件技术
- 支持网络编程及 Internet 编程
- 对数据库访问的支持

其主要目的是：

- 简化在图形界面下程序的开发方法
- 提高开发效率
- 提高软件的可重用性
- 加强软件的质量
- 支持更多更强的应用

纵观现有的 RAD 工具——Microsoft 的 Visual Basic、Sybase 的 PowerBuilder、IBM 的 Visual Age 及 Borland 的 Delphi，我们发现 Borland 公司的 Delphi 是其中的佼佼者，它是唯一符合上述要求的开发工具。

在 Delphi 之前，人们须在没有真正 OOP（面向对象编程）性能的可视化编程和没有可视化编程性能的 OOP 之间做出选择；而现在，Delphi 允许在一个具有真正 OOP 扩展的可视编程环境中使用它的 Object Pascal 语言。这种组合是革命性的，它使可视编程与一个功能极其强大的且面向对象的程序开发框架紧密结合起来。

《Delphi 2 开发人员指南》主要从编程原理及实例应用出发，它重点讨论了 Windows 程序设计的 11 个高级话题，其中包括了如何利用 Delphi 开发 Internet 最新的应用程序。这 11 个高级话题对每一个 Windows 程序员无疑都是一个挑战：掌握了这些，就有可能真正成为一个用 Delphi 开发 Windows 程序的高手。原来认为软件公司出品的高不可攀的软件，在你用 Delphi 完成之后，一定会大大提高自己对于软件开发的信心。另一方面，本书 80 个左右的“如何…”对初学者也颇有益处，同时对学习其他编程方法也很有帮助。可以肯定地说，只要

认真学习此书就必然会受益匪浅。

本书是用 Delphi 进行 Windows 程序开发的高级教程,使用者应对 Delphi 及 Object Pascal 十分熟悉。

本书由冯博琴主译。参加翻译的有冯岚(第1、2、3章)、邓辉(第4章)、苏迪(第5章)、吴振海(第6章)、徐景民(第7章)、朱丹军(第8章)、李钢(第9章)、叶盛(第10、11章)。全书由李波审校,冯博琴、冯岚统稿。

限于译者水平,译文中不准确之处在所难免,恳望同行和读者不吝赐教。

对本书英文原书配套光盘感兴趣的读者请与机械工业出版社华章公司联系。

电话:(010) 68451253

译 者

1997 年

前言

本书内容

在编程类书籍中，本书具有一定的突破性。不同于绝大多数的程序设计书籍（它们只提供代码片段和非常少量的“思想的证明”应用程序），本书向读者提供一组精选的、全新的且可立即使用的应用程序，本书的 80 个左右的“如何…”可以满足现实世界的需要，也可解决实际开发中的问题。本书要求读者应对 Delphi 1.0 和 Delphi 2.0 环境以及 Object Pascal 语言均非常熟悉。如果你对于现实问题需要一个定制的快捷的解决方法，就读读它吧！本书正是你梦寐以求的！

Delphi 揭密

Delphi 无疑是当前运行 Microsoft Windows 的 IBM PC 机适用的最先进的编程环境。但它的简便性使开发者将其仅仅看作一种原型工具，在 C 或 C++ 中使用时尤为如此。本书创建了与用 C 甚至是汇编语言编写的商用软件包完全同等的 7 个完善的应用程序，因此与陈旧的观念截然无关。在这里，Delphi 第 1 次释放出了它的所有功力，读者可以执行无限的递归搜索、无闪烁的位图拖拽、彩色光标操作以及让程序出类拔萃的所有其他诀窍。行动起来吧，Delphi 就在这里！

填充应用程序框架

本书介绍了代表当前计算机程序设计信息的新方法，这一方法被称为应用程序框架。它的思想就是开发一个骨架应用程序，然后可以加入一些短小而有用的功能模块。其每项功能都展示了一个实际程序设计环境的主流方法。因此，整个应用程序不仅可作为一个技术性的演示，而且还可解决实际任务。在这本书中，首先开发了 3 个小的实用程序，接着开发了一个复杂而功能强大的文件管理器，它甚至还包括了除 Windows 95 Explorer 之外的功能，最后则展示了带有 3 组 Internet TCP/IP 协议的客户机——FTP、UseNet NNTP News 和 POP3/SMTP Electronic Mail。

强大的功能

因为 Delphi 给出了完整的 Windows 95 和 Windows 3.x API，因此它可以做到 C 或 C++ 所做的任何事。这些事包括为 Internet 客户机使用 Winsock DLL、调用 Bit Block Transfer GDI 例程以允许快速的且无闪烁的位图移动通过整个桌面，还可以调换 Windows 桌面壁纸。另外，由于 Object Pascal 支持其他编译语言的所有功能，因此一些诸如递归、基于指针的串分析、函数数组以及运行时间的类型指派等复杂技术均可被开发出来。通过这本书，读者可以发现一些完善的软件工程技术，并在自己的程序中实际地运用它们。

078 151 / 03

问答格式

本书采用了问答系统。每一节均有一个确定的问题，接着再提出解决方案，然后讲解它是如何工作的及其原因。这一风格可以仅仅使用一个引用就决了两种特定的需求；而且通过从头到尾地阅读本书，读者可接受丰富的 Delphi 高级程序设计知识。

组件、共享软件、演示、图片和产品

与本书配套的 CD 中除包括了书中的所有源代码项目及帮助器文件外，还有其他 Delphi 书中所缺少的东西：即时可用的 VCL 模板组件。CIUPKC software 从书中提取代码并将其放入到 15 个以上即时可用的独立组件之中，且它还可用于 Delphi 1.0 中。CD 盘中的这些组件的源代码与书中的极为相似。在 CD 盘中还有适用于 Delphi 的其他演示，它们是从 CIUPKC software、Internet 和国际互联网共享软件及独立软件程序与组件中得到的。另外，Delphi Cheat Sheet 也在 CD 盘上，含有用 HTML 格式表示的一组数十个错误修复、诀窍、技巧、技术以及操作，它们来源于世界各地。

最后，书中插图所用的所有的屏幕视图均可在每一章下压缩的 PCX 文件中找到，它们颜色丰富，太可爱了。将它们解压缩，并用 Windows Paint Program 来观赏它们，如果愿意，就把它们打印出来。好好享受吧！

读者应达到的水平

虽然本书主要是针对 Delphi 中级和高级使用者，但即使是初学者也会从代码和项目的认真学习中受益非浅，所有的读者都应基本了解如何操作 Delphi IDE 以及如何在 Object Pascal 中进行程序设计。无论你的 Delphi 水平怎样，你都能从本书发现一些值得阅读的东西。

使用本书所需的环境

本书假设你至少有 Delphi 1.0 的桌面版本，这意味着一台带有 VGA 显示器和鼠标的 Windows PC 机。当然还需要 CD-ROM 驱动器，但是书中的代码可以不用它而用键盘敲进去（但并不赞成这样做！）。代码以 Delphi 1.0 和 2.0 两个格式给出，且对于兼容性问题作出了注释。同时也为这两个版本分别给出了 VCL 组件。

本书的结构

本书包括 11 章

第 1 章 实用程序

本章是做准备工作的，提供了 3 个漂亮的启发性质的实用程序。第 1 个是一个 Windows 屏幕保护程序，它可将用户选定的位图随机地在屏幕上移动，即演示了无闪烁的位图移动。第 2 个实用程序则讲述如何变换 Windows 壁纸，以及展示了过程中的一些神秘的 API 调用。最后一个实用程序则是著名的 NeXT 风格的任务栏的另一种实现方法（因其实际为 Berkeley 风格，作者考虑称其为 YANSTB），而且还解释了许多今后将要遇到的强劲的文件操作技术。

第 2 章 文件与目录

本章开始论述应用程序框架。在此章中，你将创建 CC File Manager 的框架，这是一个

适用于工作站的基于先进的 UNIX/X - Windows File Manager 之上的应用程序 (Windows 95 Explorer 是这个令人难以置信的应用程序的简单拷贝……)。在本章的学习中, 通过递归目录操作与文件搜索、文件属性设置和显示、立体边框色彩与基于异常的文件操作等, 读者将会真正了解 Windows 中的文件管理 (这并非一个琐碎的小问题, 因为 Delphi 的标准文件例程在其失败时并不触发异常)。在本章结尾, 读者就会拥有了一个强劲的小应用程序。

第 3 章 键盘与鼠标

这一章继续第 2 章的体系且集中讨论 Windows 中的键盘和鼠标。首先在讨论 SHIFT、ALT 和 CONTROL 键以及组合键 CONTROL - BREAK 和功能键 (甚至 F11 和 F12) 的过程中增加了一些例程, 接着重点围绕鼠标进行讲解: 提到了右击、创建及操作彩色图标光标甚至还有位图光标的技术 (当然在 Windows 95 中, 这仅仅是标准的 API 功能, 但在 Windows 3. x 中, 这确实是一个奇迹)。

第 4 章 打印与打印机

接着将打印机支持添加进 File Manager。在这里开发了一个特殊的对话框, 它封装了实际上适合于 Windows 3. x 的打印机的所有信息, 而且允许用户与之交互且控制其大多数信息。图形打印机选择、打印功能显示、错误捕捉打印以及访问打印队列都是这一潜在的小应用程序 (它包括大量的对 Windows 95 打印机 API 额外的支持, 而这些 API 功能远远强于 Windows 3. x 的 API) 的组成部分。本章还特别注意了屏幕打印问题, 讲解了一个简单的但卓有成效的方法来捕捉屏幕内容, 并将其打印出来或将其保存为一个位图。

第 5 章 帮助和帮助文件

本章进入了 Windows 程序设计中非常重要的但又经常被忽视的一部分, 即 Help。这一章将以其本机 RTF 格式展示一个完整的 Help 文件, 而且还包括 HPJ 控制文件。本章还详细解释如何将 Rich Text Format (富文本格式) 的格式化代码翻译为系统使用的 WinHelp 超文本文件。然后展示了一些技术将 Help 添加到在 Delphi 下创建的应用程序中, 这包括具有上下文敏感性自动弹出的 Help。最后完整地阐述了两个和 Delphi 一起发行的实用程序——KWGen 和 HelpInst, 并讲解了使用它将帮助添加到运行的 IDE 的打印对话框中, 而且还包括具有上下文敏感性的 Help!

第 6 章 图形和图像

本章通过 5 个“如何…”来最终完善 File Manager 框架。这 5 个“如何…”分别是: 如何创建和使用缩微位图而损失最小量的信息、在 Windows 3. x 下如何显示大于 64K 的位图、如何使随机位图如图标一样透明、如何读一个 256 色的 PCX 文件。将这些图像技术添加到 File Manager 之后, 给定文件中的任何图形信息都可显示在它的图标上。最后且很重要的一点就是提供了这一组件的 VCL 版本, 它同任何其他对话框一样, 可由 Execute 命令执行!

第 7 章 本地数据库程序设计

数据库程序设计是一个重要的问题, 每一个 Delphi 程序员迟早都会遇上它。这一章探讨了各种各样的数据库程序设计技术, 例如用数据表表示继承关系、使用 Borland Database Engine 的接口技术、创建内存表、执行 SOUNDEX 搜索、确保注册安全、创建表驱动用户接口等等。虽然本章提到的技术并非围绕一个单一体系, 但其方法仍使读者能够便利地再次使用绝大多数代码, 而且更重要的是使用该解决方法所涉及到的设计和思想。

第 8 章 客户机/服务器数据库编程

在前一章的基础上，这一章通过与 Delphi 一起发行的 Local Interbase Server 的便利条件，进一步扩展了客户机/服务器数据库编程技术。此处所探讨的最为重要的问题是在运行时创建表、询问数据库的特殊功能、在 Interbase 中生成唯一的键（通过触发、生成器和保存的过程），最后还讨论了事务，特别是在事务处理中对于任何客户机/服务器工程都尤为重要的称作“多并发事务”的那一部分。

第 9 章 Internet I: Winsock 与 FTP

本章创建 CC Internet Command Center 框架应用程序并由强劲的 FTP (File Transfer Protocol, 文件传输协议) 客户机功能启动它。在这时还开发了一个完整的 OOP Winsock 工具及一个 FTP 对象，它甚至还允许一个带有远程和本地文件列表功能的全球范围的 FTP 浏览器、ASCII 及二进制的下载和上载，以及用数量少得令人吃惊的程序代码来创建一个图形过程指示器！

第 10 章 Internet II: USENET 新闻组与 UUCode

本章通过 NNTP 协议向 ICC 添加了 UseNet 新闻的无限多的可能性。工作在相同的显示环境下，新闻客户机添加了该功能以管理新闻组列表、支持任意数量的讨论组、获取文章组数、头和文本，并将该信息保存为文本文件信息或将其保存在系统中，也可创建和下载文章。最后，解释了强劲的 UUCode（在这里，UU 代表 Unix to Unix 之意，通常称为 UUEncode 和 UUDecode）功能，并详细地讲述了在 UUCodeObject 完全工作时，可从网络中获取多重邮递、无缝隙地将其置为一体以及输出一个大型的二进制文件，或相反地从一个单一的二进制文件中创建所需的邮递文件。在本章中，CC 新闻就是好新闻！

第 11 章 Internet III: POP3/SMTP E-Mail 和 MIME

最后，ICC 应用程序添加了电子邮件以及 POP3 收件和 SMTP 发件电子邮件协议。它还创建并保持了邮箱邮件队列和询问答复。最后解释了 UUCode 和 MIME 的电子邮件版本，且创建并包括了一个完整的 MIME 编码器和译码器对象。关于这一点，CC Internet Command Center 已准备好将 Internet 放在你的指尖上！

错误消除

本书第 9 章给出的程序代码中有些错误，它会导致文件下载时出现混乱的二进制代码。本书作者已对此进行更正，读者可到下述两个地址 <http://WWW.waite.com> 或 <http://WWW.tde.com/ciupke> 进行查询。

Delphi 2.0 打印

由于有了新的强劲的 Windows 95 打印 API，所以在与本书配套的 CD 中放置了一个特殊的“如何…”的内容存储在 WIN95PRN/Chapter4 目录之下的一个 DOC 文本文件，它可用 Word for Windows 或 Wordpad 查看，该文件附带了所有的源代码。

目 录

译者序

前言

第 1 章 实用程序 1

1.1 如何创建一个 Windows 屏幕保护程序 ... 1

1.2 如何创建一个 Windows 壁纸转换器 ... 25

1.3 如何创建一个 NeXT 风格的任务栏 ... 42

第 2 章 文件与目录 69

2.1 如何在 Delphi 中创建一个定制的图形 File Manager 70

2.2 如何对定制的 File Manager 添加完整的错误控制 112

2.3 如何给定制的 File Manager 添加标准列表框功能 121

2.4 如何用定制的 File Manager 完成整个目录及驱动器导航 128

2.5 如何用定制的 File Manager 找到在驱动器任何位置上的文件 133

2.6 如何在定制的 File Manager 中执行完整的嵌套目录操作 140

2.7 如何将个人程序图标加到定制的 File Listbox 中 147

2.8 如何在定制的 File Manager 中显示并修改文件的属性 151

2.9 如何为定制的 File Manager 添加拖放功能 164

第 3 章 键盘和鼠标 172

3.1 如何对 Delphi 中的 Shift、Alt 和 Ctrl 与鼠标点击组合作出响应 173

3.2 如何对 Delphi 中的鼠标右键点击作出响应 191

3.3 如何在 Delphi 中,添加对功能键(尤其是 F11 和 F12)的支持 193

3.4 如何在 Delphi 中,捕获类似于 Ctrl-Break 的特殊按键 199

3.5 如何触发 Delphi 中的 Numlock、Scroll

Lock 和 Capslock 键 202

3.6 如何在 Delphi 中,对 Home 键和 End 键作出应答 210

3.7 如何在 Delphi 中,用箭头键移动鼠标 ... 213

3.8 如何在 Delphi 中,创建彩色的鼠标光标 222

3.9 如何在 Delphi 16 中,建立 Windows 95 风格的动态彩色鼠标 231

3.10 如何同 Solitaire 一样用位图作为鼠标光标 238

第 4 章 打印与打印机 248

4.1 如何在 Delphi 中确定可用的打印机 249

4.2 如何在 Delphi 中确定能够打印的内容 259

4.3 如何在 Delphi 中控制运行时刻的打印机 262

4.4 如何在 Delphi 中检测打印错误 272

4.5 如何在 Delphi 中检查和改变打印队列 275

4.6 如何在 Delphi 中进行图形的屏幕打印 276

4.7 如何在 Delphi 中把结果输出到文件而不是打印机中 280

4.8 如何在 Delphi 中预览打印结果 285

第 5 章 帮助和帮助文件 290

5.1 如何创建 WinHelp 可阅读的 Windows 3.X 帮助文件 290

5.2 如何使 Delphi 中的应用程序获得 Win Help 帮助 303

5.3 如何使 F1 引发一个 WinHelp 主题 ... 305

5.4 如何为 Delphi 中定制的 VCL 组件增加设计阶段的 WinHelp 帮助 307

5.5 如何为 Delphi 中定制的 VCL 组件增加对象检查器 WinHelp 帮助 309

第 6 章 图形和图像	311	9.5 如何用 Delphi 显示远程 FTP 服务器的目录	611
6.1 如何在 Delphi 中创建缩微图像	311	9.6 如何使用 Delphi 浏览远程 FTP 服务器上的文本文件	616
6.2 如何在 Delphi 中将缩微图像保存为文件以备后用	323	9.7 如何使用 Delphi 将远程 FTP 服务器上的文本文件下载到本地的 PC 机硬盘中	622
6.3 如何在 Delphi 中装入大型位图文件	328	9.8 如何使用 Delphi 将远程 FTP 服务器上的二进制文件下载到本地的 PC 机硬盘中	627
6.4 如何在 Delphi 中将任意位图转换成带透明色彩的图标	340	9.9 如何使用 Delphi 将本地 PC 机硬盘中的文本文件装载到远程 FTP 服务器中	632
6.5 如何在 Delphi 中从 Paintbrush 里还原 16 及 256 色 PCX 文件	344	9.10 如何使用 Delphi 将本地 PC 机硬盘中的二进制文件装载到远程 FTP 服务器中	638
第 7 章 本地数据库程序设计	366	第 10 章 INTERNET II:USENET 新闻组与 UUCODE	649
7.1 如何在一个窗体内显示多个记录对象	367	10.1 如何在 Delphi 中与 UseNet 新闻组 NNTP 服务器联系	650
7.2 如何用数据库表来表示和显示继承关系	374	10.2 如何用 Delphi 从 NNTP 服务器获得可利用的 UseNet 新闻群组列表	685
7.3 如何同 Borland 数据库引擎接口	381	10.3 如何用 Delphi 获得 UseNet 新闻群组中所有可利用的文章	697
7.4 如何创建 TableScanner 对象	388	10.4 如何用 Delphi 维护 NEWSRC 文件	702
7.5 如何创建内存表	394	10.5 如何在 Delphi 中获得 UseNet 新闻群组文章的头标	706
7.6 如何用 SOUNDEX 算法执行语音搜索	400	10.6 如何在 Delphi 中获取 UseNet 新闻群组中文章的文本	713
7.7 如何确保本地数据库的注册安全	406	10.7 如何在 Delphi 中执行 UU 解码以将文章文本转换成二进制文件	719
7.8 如何创建表驱动的选项卡式的用户界面	419	10.8 如何用 Delphi 发布 UseNet 新闻群组文章	740
第 8 章 客户机/服务器数据库编程	428	10.9 如何在 Delphi 中使用 UU Encode 放置二进制文件,并将其作为 UseNet 新闻群组的文章	746
8.1 如何创建 Interbase 数据库	429	第 11 章 INTERNET III:POP3/SMTP E-MAIL 和 MIME	755
8.2 如何在运行状态下创建数据库表	433	11.1 如何在 Delphi 中与 E-mail POP3/SMTP 服务器联系	756
8.3 如何揭示数据库服务器的性能	457		
8.4 如何在 Interbase 表中创建唯一关键字	466		
8.5 如何进行多事务处理	470		
第 9 章 INTERNET I:WINSOCK 与 FTP	478		
9.1 如何使用 Delphi 在 Internet 上编写具有 Winsock 规范的应用程序	479		
9.2 如何使用 Delphi 获得 Internet 提供的各种服务	553		
9.3 如何使用 Delphi 登录到文件传输协议(FTP)服务器	561		
9.4 如何使用 Delphi 获得远程 FTP 服务器的目录	596		

11.2 如何用 Delphi 获取 POP3/SMTP 服务器上的邮件.....	770	11.5 如何在 Delphi 中发送 SMTP E-mail 消息或响应.....	841
11.3 如何用 Delphi 维护邮箱文件.....	812	11.6 如何在 Delphi 中将二进制文件作为 MIME SMTP 消息发送.....	864
11.4 如何在 Delphi 中执行 MIME 解码将消息文本转换成二进制文件.....	824		

第 1 章 实用程序

Windows 程序员最经常接受的任务之一就是开发一项可完成单一用途的简单应用程序，即实用程序。完成这项“简单”工作的工作量可能会很大。本章将着重于 3 个看起来容易的实用程序的介绍，其内容将涉及到有关 Windows 编程环境的深奥知识。及技术，包括如何产生用户界面友好的应用程序，如何平滑地移动位图（无闪烁），如何在用户指定的目录下创建所有文件的列表，以及如何为可执行文件（或与文件扩展名关联的可执行文件）获取缺省图标。

1.1 如何创建一个 Windows 屏幕保护程序

屏幕保护程序是 Windows 最高级的实用程序之一。该程序经常使用公司标识，它们既美观又实用。本节将创建一个接受用户输入的完整的屏幕保护程序，并用它来选择它的运动位图。同时，还将演示一种用于在屏幕上移动位图而无恼人的重画闪烁的强有力的机制。

1.2 如何创建一个 Windows 壁纸转换器

Windows 所拥有的特性之一就是在 PC 机上引入了壁纸的概念，每一台计算机都可依据个人的口味设计背景位图，从而使在整天盯着屏幕的同时获得适当的放松。这一节将帮助读者开发一个壁纸转换器，它可以和 Windows 桌面直接交互，因此也就确保了与所有其他应用程序之间的协调性。为了允许选择随机显示，用户需开发包含有一个目录和若干文件的列表框，且这些文件需带有在目录中已给出的扩展名的所有文件名（这是一项标准列表框不支持的功能）。

1.3 如何创建一个 NeXT 风格的任务栏

NeXT 是苹果 Steve Jobs 的独创，是带有整洁用户接口的很棒的计算机。其特性之一就是有一个任务栏，在这个任务栏上的按钮带有各个程序的图标，当将某一按钮按下时，就会执行其应用程序。Windows 95 中也有类似的系统。在这一节中，读者可使用 Delphi 创建自己的 Windows 版本，并将学会如何在程序的可执行文件中直接得到程序的图标，还将掌握哪一个文件与一个给定的文件扩展名有关联。我们还将展示一个复杂的链接列表系统，该系统允许存储每一个应用程序所需的信息到堆中，而不是在有限的数据段中。

1.1 如何创建一个 Windows 屏幕保护程序？

* 复杂程度：初级

问题的提出

公司要求开发一个用公司标识作为运动位图的屏幕保护程序。我能否不必花费 3 个月的时间阅读手册就完成这项任务吗？

实现技术

Windows 屏幕保护实用程序只有一个标准的可执行文件（以 SCR 为扩展名），而且还附加了一些苛刻的条件。屏幕保护程序应能接受 Windows Shell 输入的命令行、维护配置文件、

在屏幕上随机放置位图，并在鼠标点击或键盘输入后迅速、干净地自行关闭。神奇的 Delphi TCanvas 组件应运而生，并带有灵巧的 TIniFile 对象。太令人吃惊了！

实现步骤

运行 CCSAVER.EXE 后将会看到类似图 1-1 所示的画面。如想保留“SMCIUPKC.BMP”为所选位图，则按 Text Saver 按钮，就会出现图 1-2 的显示。反之则按 Browse for Filename 按钮选择新位图，也可在 Edit 控制框中直接输入文件名及路径。接着再按 Test Saver 按钮，这样就使用了所选定的位图而非图中所示的缺省位图。按照下示步骤创建自己的 CCSAVER 版本。

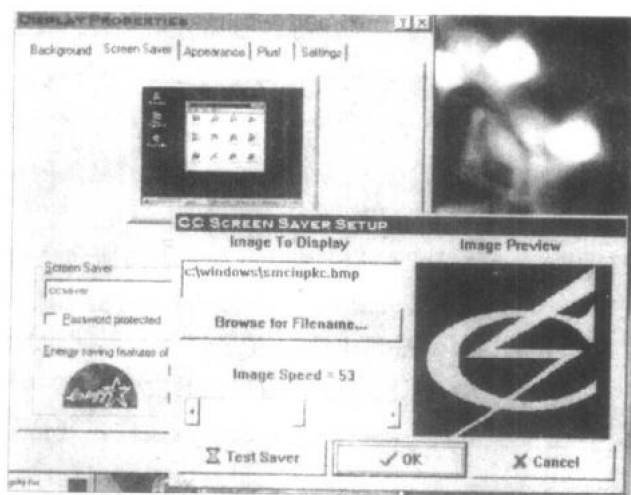


图 1-1 CCSAVER 运行时显示的配置对话框

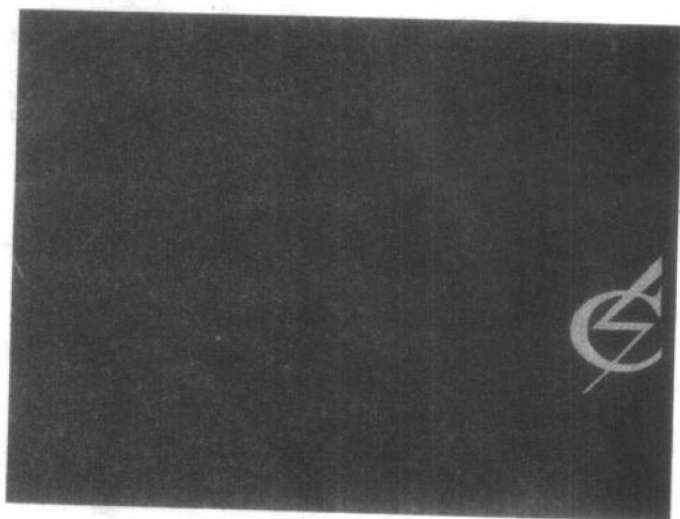


图 1-2 CCSAVER 运行时，在屏幕保护程序模式中显示其运动位图

1. 创建一个新的空白窗体项目并命名为 CCSAVER.DPR，将其单元命名为 CCSCREEN.PAS。为该项目添加另一个窗体——一个标准的底部按钮对话框窗体，再将其单元命名为 CCSAVERSU.PAS。

2. 将 CCSCREEN 的 Form1 重命名为 CCScreenSaverForm, 并添加如表 1-1 中所示的属性、组件以及设置。

表 1-1 CCSAVER 项目中的 CCSCREEN 单元的属性及设置

组 件	属 性	设 置
CCScreenSaverForm	Type	TForm
	Left	66
	Top	119
	Width	266
	Height	147
	Caption	CC Screen Saver
	BorderIcons	
	+ bsSystemMenu	false
	+ bsMinimize	false
	+ bsMaximize	false
	BorderStyle	bsNone
	FormStyle	fsStayOnTop
	Color	clBlack
	Position	poDesigned
CCSSTimer	WindowState	wsMaximized
	Type	TTimer
	Enabled	false
	Interval	1000

3. 激活 CCScreenSaverForm 中的 Object Inspector 并移至 Events 页在 OnShow 域内双击, 并在屏幕保护程序出现时添加以下代码初始化它 (注意你只需输入那些用黑体显示的语句, 其余均为 Delphi 自动产生)。

```

| This header is Created by Delphi; put in the calls to set up the starting
| mouse position, the speed of the bitmap motion, and the starting bitmap
| coordinates chosen randomly. Also set up the message hook and hide the
| cursor for neatness sake. Finally, get the necessary canvases to show
| the moving bitmap smoothly. Last, load the bitmap to use as an image!
procedure TCCScreenSaverForm.FormShow (Sender: TObject);
begin
    | Seed the random number generator |
    Randomize;
    | Use an API call to get the position of the mouse cursor at startup |
    GetCursorPos ( StartingMousePosition );
    | Set the speed to move the bitmap between 910 and 100 ms |

```

```

CCSSTimer.Interval: = 305 - ( SetupDialog.TheImageSpeed * 3 );
| Start the timer |
CCSSTimer.Enabled: = true;
| Hook the message processing call into the Application object |
Application.OnMessage: = ShutDownTheScreenSaver;
| Use an API call to hide the cursor for neatness! |
ShowCursor ( false );
TheDisplayBitmap : = TBitmap.Create;
| Load the image file name and signal an error if unable to
| $ I + |
try
    TheDisplayBitmap.LoadFromFile ( SetupDialog.TheImageFilename );
except
    | abort on any io error at all! |
    on E: EInOutError do
        begin
            | Signal the error condition |
            ErrorDialog ( SetupDialog.TheImageFileName + ' ' +
                GetIOErrorMessage ( E.ErrorCode ) + '! Aborting Saver!' );
            | abort the saver |
            exit;
        end;
    end;
end;
| Get a Position where the image can move at least a little |
CurrentXLocation : = Random ( ( Screen.Width - ( TheDisplayBitmap.Width * 2 )) + 1;
CurrentYLocation : = Random ( ( Screen.Height - ( TheDisplayBitmap.Height * 2 )) + 1;
| Start out moving right and down |
CurrentDeltaX : = CCSaver.JumpIncrement;
CurrentDeltaY : = CCSaver.JumpIncrement;
| Create the two bitmaps to handle smooth animation |
TheSavedBackgroundBitmap : = TBitmap.Create;
TheWorkspaceBitmap : = TBitmap.Create;
| Set the bitmaps to appropriate height and width values |
TheWorkspaceBitmap.Width : = Screen.Width;
TheWorkspaceBitmap.Height : = Screen.Height;
TheSavedBackgroundBitmap.Width : = TheDisplayBitmap.Width;
TheSavedBackgroundBitmap.Height : = TheDisplayBitmap.Height;
| Blank out the workspace's canvas |
with TheWorkspaceBitmap.Canvas do
begin
    brush.color : = clblack;
    pen.color : = clblack;
    Brush.style : = bsSolid;
    Rectangle ( 0 , 0 , Screen.Width , Screen.Height );
end;
| Put the background behind the saved bitmap into the save buffer |

```

```

TheSavedBackgroundBitmap.Canvas.Copyrect ( Rect ( 0, 0,
TheDisplayBitmap.Width,
TheDisplayBitmap.Height), TheWorkspaceBitmap.Canvas, Rect (
    CurrentXLocation,
    CurrentYLocation, CurrentXLocation + TheDisplayBitmap.Width,
    CurrentYLocation +
    TheDisplayBitmap.Height ));
| Put the image in the workspace |
TheWorkspaceBitmap.Canvas.Copyrect ( Rect ( CurrentXLocation ,
    CurrentXLocation ,
    CurrentXLocation + TheDisplayBitmap.Width , CurrentYLocation +
    TheDisplayBitmap.Height ),
    TheDisplayBitmap.Canvas , Rect ( 0 , 0 , TheDisplayBitmap.Width ,
    TheDisplayBitmap.Height ));
| And bias it to the screen to start off |
CCScreenSaverForm.Canvas.CopyRect ( Rect ( CurrentXLocation ,
    CurrentYLocation,
    CurrentXLocation + TheDisplayBitmap.Width , CurrentYLocation + TheDisplayBitmap.Height),
    TheWorkspaceBitmap.Canvas , Rect ( CurrentXLocation , CurrentYLocation ,
    CurrentXLocation +
    TheDisplayBitmap.Width , CurrentYLocation + TheDisplayBitmap.Height ));
end;

```

4. 重复操作以向 OnHide 事件添加代码, 用以当屏幕保护程序被隐藏时关闭它。

```

| Delphi Created this; put in calls to unhook the message handler and turn off |
| the timer and show the cursor again. Then free all the bitmaps. |
procedure TCCScreenSaverForm.FormHide (Sender: TObject);
begin
    | Set the message handler hook to nil to abort message processing |
    Application.OnMessage := nil;
    | Turn off the timer until needed again |
    CCSSTimer.Enabled := false;
    | Make an API call to turn on the cursor again |
    ShowCursor ( true );
    | Free all the bitmaps till next time |
    TheDisplayBitmap.Free;
    TheSavedBackGroundBitmap.Free;
    TheWorkspaceBitmap.Free;
end;

```

5. 使用相同技术添加对 OnActivate 事件的处理程序, 它可以确保当屏幕激活时 (这正是屏幕保护程序所做的), 该窗体最大化以占据整个屏幕。

```

| Delphi Created this ; use it to maximize the window to the whole screen |

```