



Borland C++[®] 4.5 for Windows[™]

编程指南


PROGRAMMING WINDOWS WITH BORLAND C++[®] 4.5

〔美〕William Roetzheim 著

陈晓明 孙晓安 王 理 陈亚琼 郝蔚萍 译

宋建云 蒋喆奇 刘湘凌 校

如果你是一个 Windows 程序员，或者是第一次使用 Borland C++ 4.5，那么只要你拥有本书并充分利用其语言精炼而又涉猎广泛的信息资源，你就完全能够编写、编译并调试出十分健全耐用的应用程序。本书作者 William Roetzheim 先生既是一位技艺精湛的程序员，又是一位经验丰富的语言教师。为了让读者更快地学会并增强编程能力，他将把重点放在 Borland 公司提供的形象直观的有关工具上。



本书附有一张软盘，它包含本书中所出现的所有源程序代码，并提供了一些有价值的实用程序。



ZIFF-DAVIS
ZD
PRESS

电子工业出版社

72.87424

C394

73.87429

C394

图 8 清

Borland C++ 4.5 for Windows 编程指南

PROGRAMMING WINDOWS WITH BORLAND C++ 4.5

[美] William Roetzheim 著

陈晓明 孙晓安 王 理 陈亚琼 郑莉萍 译

宋建云 蒋啸奇 刘湘凌 校

电子工业出版社

(京)新登字 055 号

内 容 简 介

本书以最新最强大的 Borland C++ 4.5 为开发环境,它不要求读者具有面向对象和 Windows 程序设计的专门知识。本书将证明学习 C++ 并用其开发 Windows 程序既不困难也不复杂。当你随本书进入类、对象和窗口世界时,你不仅会领略到 OOP 和 Windows API 的强大,更能在编写基于 C++ 的 Windows 应用程序时体会到程序设计的乐趣!

本书分十二章和六个附录。涵盖了从系统安装、配置到用 C++ 编写、编译和调试一个现代 Windows 应用程序的全部知识。另外还有面向对象设计、类重用、COCOMO/W 模型等方面大量极具参考价值的示例和忠告,本书所附软盘含有各种示例文件,可以极大地节省读者编程和调试时间。



Copyright© 1994 by Ziff-Davis Press All rights reserved.

Ziff-Davis Press and ZD Press are trademarks of Ziff Communications Company.

本书英文版由美国 Ziff-Davis Press 出版, Ziff-Davis Press 已将中文版独家版权授予北京富国电子信息有限公司。未经许可,不得以任何形式和手段复制或抄袭本书内容。

Borland C++ 4.5 for Windows 编程指南
PROGRAMMING WINDOWS WITH BORLAND C++ 4.5

[美] William Roetzheim 著

陈晓明 孙晓安 王 理 陈亚琼 郑莉萍 译

宋建云 蒋啸奇 刘湘凌 校

责任编辑 张 丽

电子工业出版社出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行 各地新华书店经销

北京顺义天竺颖华印刷厂印刷

北京富国电子信息有限公司排版

开本:787×1092 毫米 1/16 印张:23.25 字数:570 千字

1995 年 6 月第一版 1995 年 6 月第一次印刷

印数:0001—5000 册 定价:46.00 元

ISBN7-5053-3104-3/TP·1091

著作权合同登记章

图字:01-95-367号

出版说明

计算机科学技术日新月异，为了引进国外最新计算机技术，提高我国计算机应用与开发的水平，中国电子工业出版社与美国**Richina Media Holdings Limited**合资兴办的北京富国电子信息有限公司取得了美国**Ziff-Davis Press**的独家版权代理。**Ziff-Davis Press**授权本公司通过电子工业出版社等出版机构全权负责的中国大陆出版该公司的中文版和英文版图书。

美国**Ziff-Davis Press**是全美最大的计算机出版商之一，在全世界96个国家中都有它的书刊，它出版的书籍、杂志和光盘，主办的展览和会议，提供的咨询和网络服务，形成了整个行业潮流的主导。我们优选翻译出版的第一批图书是**Ziff-Davis Press**的最新计算机图书，并采用了该公司提供的电子文件，由我公司采用当今世界一流的图文系统排版制作。提高了图书质量并大大缩短了图书的出版时间，从根本上改变了以往翻译版图书要落后原版书较长的“时差”现象，这在电子技术日新月异的时代具有深远的意义。今后我们还将陆续推出**Ziff-Davis Press**的最新计算机图书和软件，为广大读者提供更好的服务，传递更多的信息。

北京富国电子信息有限公司

1995年6月

鸣 谢

我非常愿意感谢以下各位朋友对本书的编辑和出版所做出的贡献: Heidi Steele、Cort Day、M. D. Barrera、Tony Jonick、Howard Blechman、Carol Burbo 和 Margo R. Hill。我还要特别感谢 Rob Hummel 为我所花费的大量时间以及他的极为宝贵的教学经验。谢谢各位朋友,没有你们,我就不可能完成这本书。

引 言

《Borland C++ 4.5 for Windows 编程指南》主要着眼于利用 Borland C++ 编译器和应用程序框架来开发 Microsoft Windows 的应用程序, Borland C++ 的应用程序框架带有用于 Windows 开发的 ObjectWindows 类库。虽然重点强调软件开发周期的编码阶段,但是也提供了有助于面向对象设计和软件测试及优化的忠告和实例。

本书的主要目的不是使你成为 C++ 程序设计专家或者 Windows 程序设计专家,相对而言,我的目标要小得多。我希望能够确保在你读完本书之后,你就能够轻松自如地用 C++ 完整地编写、编译并调试一个中等规模的 Windows 应用程序。

学习 C++ 并用它来开发 Microsoft Windows 程序既不是十分困难也不是十分复杂的。随着我们对类、对象和窗口世界探索的不断深入,你将会发现面向对象程序设计和 Windows 应用程序接口(或称之为 API)的优美和强大,也许更为重要的是,你将会发现你不仅能够编写出基于 C++ 的 Windows 程序,而且也不会丧失程序设计的趣味性。

0.1 本书的读者对象

本书是一本中等技术水平的书籍,它需要在 C 语言方面有某些背景知识,并且需要对 Microsoft Windows 有一个总体的理解。但是,本书并没有假定读者具有任何 C++ 或者 Microsoft Windows 应用程序开发方面的深入知识。本书的重点在于利用程序设计实例来说明 Windows 和 C++ 所必需的基本概念。

另外,我假定你对于 MS-DOS 和 PC 专有的 C 编译器(不一定是 Borland 的编译器)很熟悉。一般来说,你应该能够轻松自如地在一台 PC 上完整地编写、编译并调试一个 C 程序。我并不指望你对于面向对象程序设计或 Windows 程序设计有任何了解。

0.2 为什么选用 Borland C++

在本书的第一版本中,C++ 编译器的选择是显而易见的。Borland 的编译器明显地比其最接近的竞争对手(Microsoft C++)领先不少,但是差距已经缩小。随着 Microsoft 的 Visual C++ 和 Microsoft 基础类(MFC)的宣布,Microsoft 在 C++ 编译器领域已经变得非常具有挑战性,另外,诸如 Visual Basic 和 Gupta SQL 之类的 4GL 也已经变得相当流行。尽管竞争加剧,我却继续迷恋 Borland C++。虽然没有单一的驱动因素可以说明我迷恋的理由,但是 Borland C++ 编译器在广泛的各个细小的方面提供了更好的特性,以至在总体性能上它要比简单组合这些特性强得多。

下面是 Borland 编译器超过其竞争对手的优点的几个例子:

- 生成 16 位和 32 位 Windows 代码的能力以及生成带有或不带有内部覆盖支持的 DOS 代码的能力
- 支持 Pentium
- 全面支持 ANSI C++ 的异常处理
- ObjectWindows Library (OWL), 这是一个将通过与 Novell 协同约定来支持广泛的 GUI 平台(包括 Unix 和 Macintosh)的类库
- 一个利用图形用户接口操作的非常复杂的集成调试器
- 一个多目标的项目管理器,它允许你基于一个项目中的文件之间和诸如组成项目的动态链接库之类的部件之间的依赖关系来控制编译。

0.3 硬件和软件的需求

你的开发环境最低限度需要配置一台 80486 机器、8MB 内存、一个 120MB 的硬盘、一个鼠标器、一套 VGA 兼容的视频卡和显示器。一个值得推荐的配置应该由一台 80486 计算机、16MB 内存、一个 300MB 的硬盘、一个鼠标器、一套 VGA 兼容的视频卡和显示器组成。尽管更快的机器将带给你更加优异的性能,但是我相信你将会发现,这套推荐配置的性能可以满足你在开发和测试应用程序两方面的要求。

本书假定你已经购置了 Borland C++ 4.0 版(或更高版本)的编译器。这个编译器包含了一个用于创建和编辑肖像、菜单之类程序资源的工具(Resource Workshop)和一个 Windows 专有的类库(ObjectWindows Library ,或者称为 OWL)。

除了 Borland C++ 编译器版本之外,你还必须拥有一套 Microsoft Windows 3.0 版或更高版本的副本(本书中的例子都是利用 Windows 3.1 版完成测试的)。

如果你的硬盘驱动器上还没有安装这两个程序,那么第一章提供了安装和配置这两个程序的一些指导。

0.4 本书的内容

在这套书/盘中封装的材料包括十二章正文和六个附录,外加一张软盘,这张软盘包含了正文中出现的项目的源代码,可以节省你在录入和调试方面的大量时间和精力。以下几节将给出本书和磁盘内容的一个概貌,你可以先睹为快。

0.4.1 各章节导读

第一章提供了 Microsoft Windows 和 Borland C++ 4.5 的安装和配置指南。如果你已经安装了这些程序,并且已经用它们成功地编译了一个简单的 Windows 应用程序,那么你可以跳过这一章。

第二章和第三章的目的在于帮助你完成从 C 到 C++ 的过渡。如果你已经是一个有经验的 C++ 程序员,那么你也可以跳过这两章。

第四章到第十一章讨论了利用 C++ 开发 Windows 应用程序的一般性论题以及利用 Borland 的 OWL 开发 Windows 应用程序的特殊技能。如果你能毫无问题地跳过这些章

节,那么你多半不应该去买这本书!

第十二章讨论调试 Windows 应用程序和剖析 Windows 应用程序。

附录 A 是 Borland 的 OWL 的一个概要,它应该成为一份有用的参考资料。附录 B 和附录 C 提供了有关面向对象设计、类重用、程序设计准则等方面的指导,也提供了其它涉及大型程序设计的论题。附录 D 包含了一个针对 Windows 的成本估算模型及其源代码。在这一附录中的该模型的源代码也可以作为真实 Windows 应用程序的一个有价值的例子。附录 E 包含了 Tracker Support Classes 程序的源代码,该程序已在正文中讨论过。附录 F 描述了随附磁盘的内容并给出了安装磁盘文件的指导。

在学完本书之后,你也许想就某些问题与我讨论。我很希望你能这么做,请通过电子邮件 71542,1717@compuserve.com 与我联系。

0.4.2 磁盘的内容

本书随附的磁盘包含了你在通读本书过程中将要使用的各个例子文件。这些文件包括 Tracker 程序、COCOMO/W 成本估算程序以及其它程序。这张磁盘将大大节省你在录入、调试等方面花费的时间和精力,并且为你提供了可以用在你自己的应用程序中的各种程序。

概 览

引 言	(1)
第一章 Borland C++ 4.5 综述	(1)
第二章 转向对象	(18)
第三章 在 C++ 里支持对象	(28)
第四章 Windows 程序的基本元素	(49)
第五章 菜单	(59)
第六章 对话框	(130)
第七章 图形设备接口	(159)
第八章 打印机和绘图仪	(177)
第九章 生成联机帮助	(192)
第十章 位图和中介文件的使用	(199)
第十一章 鼠标器、键盘和定时器	(230)
第十二章 调试和剖析	(243)
附录 A Borland ObjectWindows Library 概要	(251)
附录 B 建立一个类重用库	(253)
附录 C C++ 的程序设计标准	(256)
附录 D 使用 COCOMO/W 估算 Windows 程序成本	(267)
附录 E Tracker 支撑类源代码清单	(279)
附录 F 盘上所包含的内容	(352)

目 录

引 言	(1)
第一章 Borland C++ 4.5 综述	(1)
1.1 安装和配置 Microsoft Windows 3.1	(1)
1.2 安装 Borland C++ 4.5	(3)
1.2.1 安装 Borland C++ 的准备工作	(3)
1.2.2 开始安装过程	(4)
1.2.3 将编译器传送到另外一台计算机上	(6)
1.3 使用集成开发环境	(7)
1.3.1 建立一个新的项目	(7)
1.3.2 迅速查看合适的菜单选择项	(9)
1.3.3 其它 IDE 的特性和能力	(10)
1.4 使用 Borland AppExpert	(10)
1.4.1 应用程序	(11)
1.4.2 主窗口 (MAIN WINDOW)	(13)
1.4.3 多文档界面子窗口/视图 (MDI CHILD/VIEW)	(15)
1.4.4 试验 APPEXPRT	(15)
1.5 使用 Borland ClassExpert	(16)
第二章 转向对象	(18)
2.1 什么是对象	(18)
2.1.1 类和对象	(18)
2.2 对象的特性	(19)
2.2.1 封装	(19)
2.2.2 抽象	(19)
2.2.3 继承	(19)
2.2.4 多态性	(19)
2.2.5 解释面向对象的概念	(20)
2.3 应用封装	(21)
2.4 应用抽象	(22)
2.5 应用继承	(23)
2.6 应用多态性	(25)
2.7 继承与包容	(26)
2.8 转变到面向对象的观点上	(27)
第三章 在 C++ 里支持对象	(28)
3.1 C++ 的背景	(28)
3.2 C 和 C++ 之间的相似之处	(28)

3.3	C 和 C++ 之间的区别	(29)
3.3.1	对 C 类型系统 (typing system) 的扩充	(30)
3.3.2	只是使 C++ 成为更加完善的 C 的扩充	(30)
3.3.3	为 C++ 提供面向对象支持的扩充	(30)
3.4	C++ 如何实现封装	(31)
3.4.1	类	(31)
3.4.2	访问控制	(33)
3.4.3	友元 (Friends)	(35)
3.5	C++ 的抽象	(38)
3.5.1	运算符重载	(38)
3.5.2	换型 (CAST) 运算符	(39)
3.5.3	重载其他的运算符	(40)
3.5.4	在重用库中重载运算符	(42)
3.6	C++ 的继承	(42)
3.6.1	从一个父类上继承	(42)
3.6.2	从多个父类中继承	(44)
3.7	C++ 的多态性	(44)
3.7.1	静态联编	(44)
3.7.2	虚函数	(46)
3.8	构造函数	(46)
3.9	析构函数	(48)
第四章	Windows 程序的基本元素	(49)
4.1	Windows 适用于什么地方	(49)
4.1.1	理解 TRACKER	(50)
4.1.2	TRACKER 主窗口	(51)
4.2	菜单	(52)
4.3	对话框	(53)
4.4	图象表示	(54)
4.5	输出设备	(55)
4.6	联机帮助	(55)
4.7	输入设备	(55)
4.8	创建一个应用程序	(58)
第五章	菜单	(59)
5.1	使用 AppExpert 创建 Tracker 1.1	(59)
5.1.1	改变 AppExpert	(60)
5.2	使用 ClassExpert 修改菜单	(62)
5.2.1	创建并且修改菜单	(64)
5.2.2	使用 ClassExpert 和 Resource Workshop 的一个简便方法	(66)
5.3	菜单约定	(73)
5.4	高级的菜单特性	(74)
5.5	修改加速条	(75)
5.5.1	创建新的位图	(75)

5.5.2	输入现有的位图	(78)
5.5.3	修改加速条和应用程序图符	(79)
5.6	使用状态条	(82)
5.7	增加应用程序专有的代码	(83)
5.7.1	向 Tracker 增加应用程序专有的类	(83)
5.7.2	在你的项目中增加类	(86)
5.8	理解并修改生成的代码	(87)
5.8.1	TrackerApp 类	(88)
5.8.2	trackerMDIClient 类	(98)
5.8.3	trackerMDIChild	(108)
5.8.4	trackerWindow 类	(118)
5.9	利用 ObjectWindows 进行编程	(125)
5.9.1	窗口信息的封装	(125)
5.9.2	抽象	(126)
5.9.3	自动消息响应	(126)
5.10	Windows 函数和消息	(126)
5.10.1	Windows 函数的类型	(126)
5.10.2	Windows 消息源	(128)
5.10.3	消息的区域	(129)
5.11	始终让用户参与进来	(129)
第六章	对话框	(130)
6.1	Tracker 应用程序	(130)
6.1.1	利用 ClassExpert 增加对话框	(130)
6.1.2	修改 SetCoordDialog 代码	(132)
6.1.3	为调用对话框增加代码	(135)
6.2	定义对话框的需求	(136)
6.2.1	对话框子控制	(137)
6.2.2	对话框设计说明表	(138)
6.2.3	Tracker 对话框的设计说明表	(139)
6.3	使用 Resource Workshop 来创建对话框	(139)
6.3.1	启动对话编辑器	(139)
6.3.2	给对话框命名	(141)
6.3.3	增加控制	(142)
6.3.4	设置控制属性	(144)
6.3.5	将控制对齐	(146)
6.3.6	测试对话框	(147)
6.3.7	保存对话框	(147)
6.4	编程策略	(147)
6.5	与对话控制通信	(148)
6.5.1	编辑控制	(149)
6.5.2	列表框	(150)
6.5.3	组合框	(152)
6.5.4	分组框	(153)

6.5.5	按钮	(153)
6.5.6	复选框	(153)
6.5.7	单选按钮	(154)
6.6	作为对象的对话框控制	(154)
6.6.1	TControl 类	(155)
6.6.2	TScrollBar 类	(155)
6.6.3	TStatic 类	(156)
6.6.4	TEdit 类	(156)
6.6.5	TListBox 类	(156)
6.6.6	TComboBox 类	(156)
6.6.7	TButton 类	(157)
6.6.8	TCheckBox 类	(157)
6.6.9	TRadioButton 类	(157)
6.7	使用传送缓冲器	(157)
6.8	始终让用户参与进来	(158)
第七章	图形设备接口	(159)
7.1	设备上下文(device context)	(159)
7.2	输出文本	(163)
7.2.1	使用 TextOut()	(164)
7.2.2	使用 DrawText()	(164)
7.2.3	改变文本输出特性	(164)
7.2.4	在窗口中给文本定位	(165)
7.3	选择字体	(166)
7.3.1	创建一个字体	(166)
7.3.2	选择新字体	(167)
7.3.3	删除字体	(167)
7.4	图形原语	(167)
7.4.1	画线原语	(167)
7.4.2	区域填充原语	(170)
7.4.3	选择画笔和填充模式	(171)
7.5	窗口、视域和映象方式	(173)
7.6	Tracker Paint() 成员函数	(174)
第八章	打印机和绘图仪	(177)
8.1	OWL 打印支持	(177)
8.1.1	TPrinter 类	(177)
8.1.2	TPrintout 类	(177)
8.1.3	TPreviewPage 类	(178)
8.2	打印机转义码	(178)
8.3	有关打印机的提示	(180)
8.3.1	打印机联编处理	(181)
8.3.2	禁止 Print Manager (打印管理器)	(181)
8.3.3	同样的内容输出到屏幕和打印机	(181)

8.3.4	设置激光打印机上的剪贴区域	(182)
8.4	Tracker 中与打印有关的类	(182)
8.4.1	APXPrintOut 类	(183)
8.4.2	PreviewWindow 类	(185)
第九章	生成联机帮助	(192)
9.1	帮助文件开发概述	(192)
9.2	帮助主题概述	(193)
9.3	创建 RTF 文件	(194)
9.3.1	在页中增加上下文串	(194)
9.3.2	增加一个目录	(194)
9.3.3	增加转移链接	(194)
9.3.4	增加弹出链接	(195)
9.3.5	增加一个关键字索引	(195)
9.4	修改帮助文件	(195)
9.5	适用于 Word for Windows 的一个帮助创建支持工具	(196)
9.6	创建帮助项目文件	(196)
9.7	编译帮助文件	(197)
9.8	把帮助文件链接到应用程序中	(197)
第十章	位图和中介文件的使用	(199)
10.1	MetaTest 程序	(199)
10.1.1	运行 MetaTest	(199)
10.2	中介文件的使用	(200)
10.2.1	非艺术人员使用的中介文件	(201)
10.3	作为资源的中介文件	(203)
10.3.1	增加用户定义的资源	(203)
10.3.2	用户中介文件作为资源使用	(205)
10.4	清退位图	(206)
10.5	MetaTest 源代码	(209)
第十一章	鼠标器、键盘和定时器	(230)
11.1	鼠标器基础	(230)
11.2	鼠标光标形状	(234)
11.2.1	用 Resource Workshop 创建一个光标	(235)
11.3	Windows 键盘驱动程序	(236)
11.4	键盘消息	(237)
11.5	重映象键	(240)
11.6	Windows 定时器	(241)
第十二章	调试和剖析	(243)
12.1	调试 Windows 应用程序	(243)
12.1.1	编译器和链接器报警	(243)
12.1.2	用 MessageBeep() 和 MessageBox() 进行调试	(244)

12.1.3	用 printf() 调试	(246)
12.1.4	一些涉及多方面问题的提示	(246)
12.2	剖析 Windows 应用程序	(249)
12.2.1	它是如何工作的	(249)
12.2.2	工作性能的改善	(249)
12.3	编写你自己的 C++ Windows 程序	(250)
附录 A	Borland ObjectWindows Library 概要	(251)
附录 B	建立一个类重用库	(253)
附录 C	C++ 的程序设计标准	(256)
C.1	面向对象设计的表示法	(256)
C.1.1	外部设计	(256)
C.1.2	内部设计	(257)
C.1.3	继承关系	(258)
C.1.4	构成关系	(258)
C.2	用于 C++ 开发的文件组织	(259)
C.3	C++ 头文件的结构	(260)
C.4	C++ 源代码文件的结构	(264)
C.5	为简单代码辩护	(266)
C.6	程序设计准则的样板	(266)
附录 D	使用 COCOMO/W 估算 Windows 程序成本	(267)
D.1	COCOMO 的基本模型	(267)
D.1.1	环境因子	(267)
D.1.2	估算页数	(268)
D.2	使用 COCOMO/W 的一般步骤	(271)
D.3	使用 COCOMO/W	(273)
D.4	把 COCOMO/W 应用于 C++	(277)
附录 E	Tracker 支撑类源代码清单	(279)
附录 F	盘上所包含的内容	(352)
F.1	安装及使用该盘	(352)
F.2	盘上文件清单	(352)

第一章 Borland C++ 4.5 综述

为了理解本书中的例子,你必须正确地安装和配置 Microsoft Windows 3.0 版本或后续版本,以及 Borland C++ 4.5 版本或后续版本。这里的应用程序也将能够在 Windows for Workgroups 中正常地工作。

本章详细地提供了每一步的具体指导,以便帮助你安装和配置所有必要的软件。在本章结束时,我将引导你去完成编译并运行你的第一个 Microsoft Windows C++ 程序的全部过程。

1.1 安装和配置 Microsoft Windows 3.1

本节将指导你去完成 Microsoft Windows 3.1 版本的整个安装过程。如果你缺少硬盘空间,我还向你提供一些用于删除 Windows 文件的提示。如果你已经安装了 Microsoft Windows 3.1 版本,那么就可以跳到本章的下一节。在本节结束时,你应当已经拥有安装好的 Microsoft Windows,并且在 CONFIG.SYS 文件中已加入了 HIMEM.SYS 和 SMARTDRV.SYS 初始化行。

Microsoft Windows 3.1 版本保存在大致 7 张 3.5 英寸的软盘中。Windows 大致需要 9 兆字节的硬盘空间,如果你安装不止一台打印机或绘图仪的话,那么所需空间就稍微多一些。如果你已经在硬盘上装有 Windows 3.0 版本,那么安装过程将自动地更新过去存在的文件。在安装 Windows 以前,你一定要了解可能连接到计算机上的打印机和绘图仪的品牌和型号,以及网络类型。我强调“可能连接”这句话,是因为你所拥有的这些设备目前多半尚未连接,但是它们很快就会连接到计算机上。

你的安装过程是通过插入第一张软盘并且运行软盘上的 SETUP 程序开始的。屏幕上大部分指令都是不释自明的。在安装过程的开始,会提示你用 Express setup 还是用 Custom Setup, Express setup 允许配置程序通过确定你当前已经安装的硬件而自己简单地回答许多问题。我建议你使用 Express setup,因为这种缺省的选择通常是合理的。

如果安装没有成功,请检查你的硬盘,以保证有足够的可用空间,然后再试装一遍。如果仍然不成功,则你的 Windows 安装盘可能有问题,应该更换。在配置程序成功地安装 Windows 之后,你应该在 CONFIG.SYS 文件中看到下面几行(假如你使用 DOS 6.2 或后续版本的话):

```
device=C:\HIMEM.SYS
DEVICE=C:\DOS\SMARTDRV.SYS 2048 1024
DOS=HIGH
```

这里的关键是确保 HIMEM.SYS 驱动程序的安装,以及 SMARTDRV.SYS 作为一个磁盘超高速缓存的安装。HIMEM.SYS 设备驱动程序在 640K 到 1M 字节的 DOS 存储区。

SMARTDRV.SYS 是一个与 Windows 兼容的磁盘超高速缓存程序。上面的范例指明超高速缓存的最大空间为 2M 字节,最小空间为 1M 字节。

你还应该在 AUTOEXEC.BAT 文件中修改 PATH 变量,以便包含 Windows 目录,以及 Borland C++ 所使用的所有动态链接库的位置。假定你把 Windows 和 Borland C++ 都安装到 C 驱动器上,则 PATH 变量应该包括下面的路径:

```
PATH=C:\WINDOWS;C:\BC4X\BIN
```

Windows 将运行在两种模式之一中:标准的和增强的。实址模式是为了和早期的机器以及只具备有限的存储器(640K 或更少)的机器(基于 8086 的)兼容而设计的,并且同早期的 Windows 应用程序相兼容,但是 Windows 3.1 版本不再支持这种模式。脱离实址模式不会影响你的开发或运行你的应用程序。

标准模式是一种保护模式,它允许 Windows 使用大到 16M 字节的常规存储器。

增强模式需要一个 386 或 486 处理器,以及至少 2M 字节的存储器。它也是一种保护模式,实际上,它除了以下两点之外,其余和标准模式相同。

1. 能够利用 386/486 芯片的存储器分页性能,当需要时允许 Windows 通过交换 4K 页面把硬盘空间当作存储器使用。

2. 在 386/486 芯片的虚拟-86 方式下运行多个虚拟 DOS 应用程序。

通常,只要你的应用程序在开发和执行过程中不受影响,不论用户运行的 Windows 是处于标准模式下或是在增强模式下都没有关系。当你自己运行 Windows 的时候,可以简单地认为 Windows 自动地运行在对于硬件来说最为合适的模式上。

如果你的硬盘空间有限,那么你可以删除一些开发本书中的应用程序所不需要的 Windows 文件。这种情况下,我通常删除在表 1.1 中列举的文件。

表 1.1 通常可以放心地删除的 Windows 文件

文件	说明
* BMP	可以显示出奇特背景的用于程序管理器的位图。CHESS.BMP 是相当大的。
SOL.*	一个独立的游戏(很有趣但不是必须保留的)。
CALENDAR.*	一种显示日历的程序。只在用户打算使用时才有用。
TERMINAL.*	一种基于 Windows 的终端程序。我喜欢自己的终端仿真软件包,而不关心由 Windows 提供的某些终端程序。
CARDFILE.*	一种用于显示并修改屏幕上的重叠卡片的程序。只在打算使用时才有用。
WRITE.*	一种基于 Windows 的相当笨拙的字处理器。不要将这种字处理器误作为 NOTEPAD.EXE。NOTEPAD.EXE 是一种梗概编辑器(许多程序,包括 Borland C++,都要调用 NOTEPAD.EXE 进行简单的编辑)。
*.TXT	包含 Windows 信息的 README 类型的文件。在删除它们之前你应该事先读一下这个文件,或者你可以有选择地保留它们用作参考。

随着 Windows 的成功安装,以及你的计算机的重新引导,新配置开始生效。你现在应该暂停 Windows 的操作,开始证实一切是否可以正常地工作。如果你是刚开始使用 Windows 的新手的话,那么应该运行 PIFEDIT 程序,以便了解对话框是如何工作的,运行