

Borland C++ 3.1~4.0

Windows 应用程序界面设计

方 旭 张克强 曲文路等 编

北京航空航天大学出版社

Borland C++ 3.1~4.0

Windows 应用程序界面设计

方 旭 张克强 曲文路等 编

北京航空航天大学出版社

(京)新登字 166 号

图书在版编目(CIP)数据

Borland C++ 3.1~4.0 Windows 应用程序界面设计/方旭等编. —北京:北京航空航天大学出版社, 1995.5
ISBN 7-81012-541-9

I. B... II. 方... III. ①窗口(软件), Windows-应用程序-界面-设计②窗口(软件)-C语言-程序设计 IV. TP312C

中国版本图书馆 CIP 数据核字 (95) 第 08056 号

内 容 简 介

Borland C++ 4.0 具有运行于 Windows 的 IDE, 集成了著名的程序员编辑器 Brief 的功能, 可进行多目标工程管理; IDE 中的 ObjectBrowser 可以显示类的关系, AppExpert 可以快速生成 ObjectWindows 2.0 的 Windows 应用程序, ClassExpert 帮助用户修改和组织 AppExpert 生成的应用程序; Borland C++ 4.0 可以编译生成 DOS, Win16 和 Win32s 和 Windows NT 的应用程序, 拥有 GUI 界面的调试程序。本书是 Borland C++ 4.0 的最新参考书, 介绍了利用 Borland C++ 4.0 Resource Workshop 生成各种资源文件, 用 C 设计 Windows 应用程序的界面, 包括菜单、对话框、加速键、位图、图标、光标、字体和帮助系统。本书适用于使用各种 C 和 C++ 编写 Windows 应用程序界面的读者使用。

●书 名: Borland C++ 3.1~4.0 Windows 应用程序界面设计

Borland C++ 3.1~4.0 Windows YINGYONG CHENGXU JIEMIAN SHEJI

●编 著 者: 方 旭 张克强 曲文路 等 编

●责任编辑: 许传安

●出 版 者: 北京航空航天大学出版社

●地 址: 北京市海淀区学院路 37 号(100083)

●印 刷 者: 北京朝阳科普印刷厂

●发 行: 新华书店总店科技发行所

●经 售: 全国各地新华书店

●开 本: 787×1092 1/16

●印 张: 28.25

●字 数: 723 千字

●印 数: 5000 册

●版 次: 1995 年 4 月第 1 版

●印 次: 1995 年 4 月第 1 次印刷

●书 号: ISBN 7-81012-541-9/TP·145

●定 价: 35.00 元

0000100

编者前言

Borland C++的前身为 Turbo C。从 2.0 版开始, Turbo C 演化为 Borland C++ 和 Turbo C++ 两个系列。Borland C++ 后来又推出了 2.0, 3.0, 3.1 和 4.0; Turbo C++ 推出了 2.0 和 3.0。Borland C++ 非常庞大, 主要被专业程序员使用; 而 Turbo C++ 相对较小, 主要用于教学, 为初学者所喜爱。

从 Turbo C++ for Windows 1.0 开始, 集成环境运行于 Windows, 是一个真正的 Windows 应用程序。而 Borland C++ 从 4.0 开始, 是一个很好用的 Windows 应用程序开发环境, 要特别指出的是, 新的 IDE(集成开发环境)可在 Windows 中直接调试程序, 工程管理比以前更方便。

Borland C++ 和 Turbo C++ for Windows 配备了 Turbo Debugger for Windows 用来调试设计的 Windows 应用程序, 增强了包括图标、光标、位图、对话框、菜单、热键和字体等的资源设计。特别是在 Borland C++ 4.0 中, 资源设计已得到显著增强, 用来设计 Windows 应用程序的界面。Borland C++ 在 OWL(ObjectWindows Class Library)中封装了 Windows API(Application Program Interface)包括函数、数据结构和消息等, 可以加快读者用 C++ 开发 Windows 应用程序的速度, 减轻读者用 C++ 开发 Windows 应用程序的难度。

本书介绍如何利用 Resource Workshop 进行 Windows 界面程序设计, 内容包括菜单、对话框、加速键、位图、字体和帮助的设计。本书适用于 Borland C++ 和 Turbo C++ for Windows 以及各种演化版本。

书中的例子针对国内程序员的需要, 全部采用中文的菜单、对话框和提示界面, 每个程序都有很好的注释和解说, 便于理解和学习。书中的程序即可运行在中文 Microsoft Windows, 也可以运行在具有中文之星的西文的 Microsoft Windows 上。如果想在西文的 Microsoft Windows 上运行, 可以将菜单、对话框和提示改变西文重新编译后即可。本书完整的程序都通过调试。

目 录

第 0 章 Windows 概述	1
0.1 Windows 程序设计语言	1
0.1.1 Windows 环境快速浏览	1
0.1.2 使用 Windows 的优越性	1
0.1.3 新的 Windows 可执行文件的格式	6
0.2 Windows 编程:概念和词汇	7
0.2.1 Windows 的窗口	7
0.2.2 Windows 布局	7
0.2.3 C++ 中的 Windows 类	9
0.2.4 OOP 及 Windows	9
0.2.5 发送和接收 Windows 消息	11
0.2.6 访问 Windows 函数	14
0.2.7 Windows 头文件:Windows.h	14
0.2.8 Windows 应用程序的组成成份	15
0.3 Borland C++ Windows 工具	15
0.3.1 使用 make 文件	15
0.3.2 创建资源	16
0.4 使用资源工具	17
0.4.1 BRCC.EXE:资源编译器	17
0.4.2 RLINK:资源连结器	18
0.4.3 BRC.EXE 资源外壳	19
0.5 启 动	20
0.5.1 安装 Resource Workshop	20
0.5.2 启动 Resource Workshop	20
0.5.3 退出 Resource Workshop	22
0.5.4 获得帮助	22
第一章 Windows 程序设计技术	23
1.1 应用程序框架	23
1.1.1 有效地使用句柄	23
1.1.2 考察 Windows 应用程序中的成份	23
1.2 一个简单的应用程序和示例	33
1.2.1 画椭圆	36
1.2.2 画弦	37
1.2.3 画饼图瓣	38
1.2.4 画矩形	39

1.3	使用 SWA 开发一个正弦波应用程序	40
1.4	创建 Windows 饼图应用程序	43
1.4.1	1PIE 和 1PIE.DEF 文件	44
1.4.2	1pie.h 头文件	44
1.4.3	1PIE.RC 资源文件	52
1.4.4	1PIE.C 程序	53
第二章	Resource Workshop 基础	57
2.1	了解 Windows 资源	57
2.2	资源的类型	57
2.2.1	对话框(Dialog boxes)	58
2.2.2	菜单(Menu)	58
2.2.3	加速键(Accelerator)	58
2.2.4	字符串表(String table)	59
2.2.5	位图(Bitmap)	59
2.2.6	图标(Icon)	59
2.2.7	光标(Cursor)	59
2.2.8	字体(Font)	60
2.2.9	用户定义和 rcdata 资源	60
2.3	两种编辑器	61
2.4	资源文件的类型	62
2.4.1	资源编译器文件	62
2.4.2	资源文件	62
2.4.3	可执行动态连接库文件	62
2.4.4	对话文件	62
2.4.5	位图文件	62
2.4.6	图标文件	63
2.4.7	光标文件	63
2.4.8	字体文件	63
2.4.9	标识符文件	63
2.4.10	.DRV 文件	65
2.4.11	这些文件如何一起工作——一个工程的样例	65
2.5	管理所有的工程文件——工程窗口	67
2.6	忠告新用户	68
第三章	使用工程和资源	69
3.1	创建一个新工程	70
3.2	打开一已有的工程	70
3.3	使用 Project 窗口	73
3.3.1	在 Project 窗口显示信息	74
3.3.2	选择一个资源	75

3.4	使用资源	75
3.4.1	编辑资源	75
3.4.2	添加一个资源	76
3.4.3	重新为资源命名	78
3.4.4	指定资源内存可选项	79
3.4.5	删除一个资源	81
3.5	使用标识符	81
3.5.1	添加一个标识符文件	81
3.5.2	在没有标识符文件下工作	82
3.5.3	使用资源编辑器添加标识符	82
3.5.4	添加、编辑、删除及列出标识符	83
3.6	存贮资源、文件和工程	84
3.6.1	File Save Project 命令	84
3.6.2	File Save File As 命令	84
3.6.3	Resource Save Resource As 命令	85
3.7	配置选项(Configuration preferences)	85
3.7.1	Undo Level	86
3.7.2	文本编辑器	86
3.7.3	包含路径(Include path)	86
3.7.4	Multi-Save	86
3.8	工程之间拷贝资源	86
3.9	使用二进制文件	87
第四章	创建对话框	89
4.1	启动 Dialog 编辑器	89
4.1.1	生成一个新的对话框	90
4.1.2	编辑一个已有的对话框	90
4.2	定制一个对话框	91
4.2.1	定义一个对话框	91
4.2.2	使用控制(control)	94
4.2.3	编辑成组的控制	102
4.2.4	取消修改(Undo Changes)	107
4.2.5	按钮控制	107
4.2.6	滚动条控制(Scroll bar control)	108
4.2.7	列表框控制(List box control)	108
4.2.8	编辑文本控制	110
4.2.9	静态控制	112
4.2.10	组合框控制 Combo box control)	113
4.2.11	自定义控制(Custom control)	115
4.3	测试一个对话框	116

4.4	存贮一个对话框	116
4.4.1	存贮工程	116
4.4.2	将一个对话框存贮到文件中	117
4.5	观察两个对话框	118
4.6	设置 Dialog 编辑器	118
4.7	一个工程的样例	120
4.7.1	创建一个新的对话框	121
4.8	相关函数	124
	CheckRadioButton	124
	DialogBox	125
	DialogBoxIndirect	125
	DialogBoxIndirectParam	127
	DialogBoxParam	128
	EndDialog	128
	GetDlgCtrlID	129
	GetDlgItem	129
	GetDlgItemInt	129
	GetDlgItemText	130
	MakeProcInstance	130
	MapDialogRect	130
4.9	示例程序	131
4.9.1	创建一个简单的“关于...”对话框	131
4.9.2	用对话框修改图形形状	137
4.9.3	用对话框输入文本	145
4.9.4	用对话框输入整数	152
4.9.5	用对话框读取实数	158
4.9.6	编写消息框程序	164
4.9.7	公用对话框	169
第五章	生成菜单	178
5.1	Menu 编辑器屏幕	178
5.1.1	Outline 区	179
5.1.2	Dialog Box 区	179
5.1.3	Test Menu 区	180
5.2	启动 Menu 编辑器	182
5.2.1	创建一个新菜单	182
5.2.2	编辑一个已存在的菜单	182
5.3	设置一个菜单	182
5.3.1	添加一个新语句	183
5.3.2	移动和拷贝	183

5.3.3	消除错误	183
5.3.4	设置一个菜单项	183
5.3.5	设置一个弹出命令	186
5.3.6	定义一个菜单分隔符	187
5.3.7	删除一个菜单语句	187
5.4	测试一个菜单	188
5.5	存贮修改	188
5.5.1	存贮工程	188
5.5.2	将菜单资源以文件形式存贮	189
5.6	编辑一个菜单资源正文	189
5.7	一个菜单样例	190
5.7.1	用文本编辑器创建一个菜单	191
5.7.2	用 Menu 编辑器来创建一个菜单	191
5.8	相关函数和数据结构	193
	AppendMenu	193
	CreateMenu	194
	CreatePopupMenu	194
	CreateWindow	194
	CreateWindowEx	201
	DeleteMenu	202
	DestroyMenu	202
	GetSubMenu	203
	GetSystemMenu	203
	InsertMenu	203
	LoadBitmap	205
	LoadMenu	205
	LoadMenuIndirect	205
	RegisterClass	206
	WNDCLASS	207
	SetMenu	208
	SetMenuItemBitMaps	208
	TrackPopupMenu	209
5.9	编程示例	210
5.9.1	创建一个菜单用来改变形状和大小	210
5.9.2	建立位图菜单	214
5.9.3	复杂菜单程序	221
5.9.4	非弹出式菜单程序	227
5.9.5	弹出式菜单示例	231
5.9.6	快速菜单示例	237

第六章 创建热键	245
6.1 使用 Accelerator 编辑器	245
6.1.1 Outline 区	246
6.1.2 对话框	246
6.2 启动 Menu 编辑器	247
6.3 启动 Accelerator 编辑器	248
6.3.1 创建一个新的热键表	248
6.3.2 编辑一个已有的热键表	248
6.4 编辑一个热键表	248
6.4.1 选择一个热键	249
6.4.2 使用对话框	249
6.4.3 移动和拷贝一个热键	251
6.4.4 删除一个热键	251
6.4.5 取消和重做修改	251
6.5 检测重复的键组合	251
6.6 为热键表创建资源正文	251
6.7 创建一个热键表样例	252
6.8 相关函数	254
AccessResource	254
FreeResource	254
LoadAccelerators	254
LoadResource	254
6.9 示例程序	255
第七章 创建一个字符串表	262
7.1 启动 String 编辑器	262
7.2 编辑字符串表	262
7.2.1 Windows 和字符串	263
7.2.2 输入新字符串	264
7.2.3 编辑已有的字符串	265
7.2.4 编辑字符串表的资源正文	265
7.3 存贮字符串表	266
7.4 测试字符串表	266
7.5 创建字符串表样例	266
7.6 相关函数	268
DrawText	268
ExtTextOut	269
TabbedTextOut	270
TextOut	270
7.7 程序示例	271

7.7.1 动态链接库程序	271
7.7.2 字符串应用程序	274
第八章 Paint 编辑器的使用	282
8.1 启动 Paint 编辑器	282
8.1.1 装入光标、字体和位图资源	282
8.1.2 装入图标	283
8.2 资源的可用特征及 Paint 编辑器功能	283
8.3 前景色和背景色	284
8.4 使用 Tools 板	284
8.4.1 Pick Rectangle 工具	284
8.4.2 剪刀(scissors)	285
8.4.3 Zoom 工具	286
8.4.4 橡皮擦(eraser)	286
8.4.5 笔	286
8.4.6 刷子	287
8.4.7 喷雾器	287
8.4.8 颜料罐	287
8.4.9 Line 工具	288
8.4.10 Text 工具	288
8.4.11 绘制空框架	288
8.4.12 绘制填充的框架	289
8.4.13 Hand 工具	289
8.4.14 模式选择	290
8.5 使用两个窗口区	290
8.6 变比放大	291
8.6.1 使用变比热键	292
8.6.2 观察图象的变比放大部分	292
8.6.3 移动一个变比放大的图象	292
8.6.4 在变比放大的窗口上显示网格	293
8.7 读取状态行	293
8.7.1 当前绘制工具信息	293
8.7.2 菜单命令说明	294
8.8 使用颜色	294
8.8.1 为源选择颜色的数目	294
8.8.2 指定前景色	295
8.8.3 指定背景色	295
8.8.4 在光标或图标中定义透明和反相的区域	295
8.8.5 显示和隐藏 Colors 板	296
8.9 设置颜色	296

8.9.1	编辑 Colors 板的颜色	296
8.9.2	改变透明和反相区域的颜色	298
8.10	向资源中加入文本	298
8.10.1	调整文本	299
8.10.2	选择字体、尺寸和文本式样	299
8.11	选择刷子或喷雾器的形状	299
8.12	选择绘画模式	300
8.13	选择线型	301
8.14	调整选中的区域	302
8.15	改变选中区域的尺寸	303
8.16	设置全局 Paint 编辑器选择项	303
8.16.1	Draw on both images 选项	304
8.16.2	Grid on zoomed images 选项	304
8.16.3	Save with default device colors 选项	304
第九章	创建图标	305
9.1	启动 Paint 编辑器	306
9.1.1	创建新图标	306
9.1.2	编辑已有的图标	307
9.2	设置图标	308
9.2.1	设计问题	309
9.2.2	缩放图标	309
9.2.3	使用透明和反相区域	309
9.2.4	使图标产生三维效果	310
9.3	测试图标	311
9.4	存贮图标	311
9.4.1	存贮工程	312
9.4.2	把图标资源作为文件来存贮	312
9.5	向图标资源中加入图象	313
9.6	删除图标和图象	313
9.6.1	删除图标资源	314
9.6.2	删除一个图标图象	314
9.7	编辑图标资源正文	314
9.8	创建样例图标	315
9.8.1	创建新图标	315
9.8.2	改变透明颜色	315
9.8.3	画计算器	315
9.8.4	加上三维效果	316
9.8.5	画分类帐页	317
9.8.6	把图象拷贝到一个不同的颜色格式	317

9.9 相关的函数	318
ArrangeIconicWidnows	318
CopyIcon	319
CreateIcon	319
DestroyIcon	319
DrawIcon	320
ExtractIcon	320
IsIconic	320
LoadIcon	321
OpenIcon	321
9.10 图标程序示例	321
第十章 创建光标	325
10.1 启动 Paint 编辑器	325
10.1.1 创建新光标	326
10.1.2 编辑已有的光标	327
10.2 设置光标	328
10.2.1 设计问题	328
10.2.2 缩放光标	328
10.2.3 使用透明和反相区域	328
10.2.4 设置光标的热点	329
10.3 测试光标	330
10.4 存贮修改	330
10.4.1 存贮工程	330
10.4.2 将光标资源以文件形式存贮	331
10.5 编辑光标资源正文	331
10.6 相关的函数	332
ClipCursor	332
CopyCursor	333
CreateCursor	333
DestroyCursor	333
GetClipCursor 3.1	334
GetCursor	334
GetCursorPos	334
LoadCursor	335
SetCursor	335
SetCursorPos	336
ShowCursor	336
10.7 光标示例程序	336

第十一章 创建位图	340
11.1 启动 Paint 编辑器	340
11.1.1 创建新位图	340
11.1.2 编辑已有的位图	342
11.2 设置位图	342
11.3 存贮位图	343
11.3.1 存贮工程	343
11.3.2 以文件形式存贮位图	343
11.4 测试位图	344
11.5 相关的函数	344
CreateBitmap	344
CreateBitmapIndirect	344
CreateCompatibleBitmap	345
CreateDIBitmap	345
CreateDiscardableBitmap	347
GetBitmapBits	347
GetBitmapDimension	347
GetBitmapDimensionEx	348
LoadBitmap	348
SetBitmapBits	348
SetBitmapDimension	349
SetBitmapDimensionEx	349
SetMenuItemBitMaps	349
11.6 位图示例程序	350
11.6.1 位图动态链接库	350
11.6.2 位图引用程序	351
第十二章 创建字体	356
12.1 启动 Paint 编辑器	357
12.1.1 创建新的字体资源	357
12.1.2 编辑已有的字体资源	357
12.2 设置字体资源	358
12.2.1 修改字体图象	358
12.2.2 为字体定义字符集	358
12.2.3 创建不同宽度的字体	360
12.2.4 定义字体资源的头	361
12.2.5 改变尺寸和属性	363
12.2.6 删除一个字体图象	363
12.3 存贮字体资源	364
12.3.1 存贮工程	364

12.3.2	以文件形式存贮字体资源.....	364
12.4	把字体资源加入到用户的应用程序中.....	364
12.4.1	用 Turbo Pascal 创建.FON 文件	364
12.4.2	用 C++ 创建字体文件	365
12.5	测试字体.....	365
12.6	一个字体资源样例.....	365
12.7	相关函数.....	367
	AddFontResource	367
	ChooseFont	367
	CreateFont	369
	CreateFontIndirect	371
	CreateScalableFontResource	373
	EnumFontFamilies	373
	EnumFonts	374
	GetFontData 3.1	374
	RemoveFontResource	375
12.8	字体示例程序.....	376
12.8.1	使用 TrueType 字体输出水平和垂直字符串	376
12.8.2	输出不同字体大小的字符串.....	379
12.8.3	使用 Script 字体输出水平和垂直字符串.....	382
12.8.4	使用 TrueType 字体输出水平和垂直字符串	385
第十三章	创建用户定义的资源.....	388
13.1	创建资源类型.....	388
13.2	加入用户定义资源.....	389
13.3	编辑用户定义资源.....	389
13.3.1	在资源正文中输入数据.....	390
13.3.2	管理存放在独立文件中的数据.....	391
13.4	测试用户定义资源.....	391
13.5	使用 RCDATA 资源类型	391
13.6	删除用户定义资源.....	391
第十四章	HC;WindowsHelp 编译程序	393
14.1	建立一个开发系统:开发周期	393
14.1.1	用户如何看待 Help 系统	393
14.1.2	Help 作者如何看待 Help 系统	394
14.1.3	程序员如何看待 Help 系统	395
14.2	规划 Help 系统	395
14.2.1	拟定规划.....	395
14.2.2	决定标题文件结构.....	397
14.2.3	设计 Help 标题	398

14.2.4	图形图像.....	400
14.3	建立 Help 标题文件	401
14.3.1	选择编写工具.....	401
14.3.2	构造 Help 标题文件	401
14.3.3	对 Help 标题文件进行编码	401
14.3.4	插入图形图像.....	407
14.3.5	标题文件管理.....	408
14.4	建立 Help 文件	410
14.4.1	创建 Help 工程文件	411
14.4.2	指定标题文件.....	411
14.4.3	指定建立标志.....	412
14.4.4	指定选择项.....	412
14.4.5	指明替换的上下文串.....	416
14.4.6	上下文相关标题的映射.....	417
14.4.7	靠引用来包含位映像.....	418
14.4.8	Help 文件的编译	418
14.4.9	设计用户程序访问 Help 系统	419
14.5	Help 系统例子	425
14.6	相关函数.....	427
	WinHelp	427
14.7	帮助示例.....	428

第0章 Windows 概述

Borland 主要的 Windows 开发语言是 C 和 C++。汇编语言在对时间效益要求较高的部分起了重要的作用,而所有重要的应用程序,例如 Windows 本身,都是用 C 或 C++ 编写的。这一版本的 Borland 编译器提供了所有在 C++ 环境下开发 Windows 程序所需的工具。

本书后面章节开发的 Windows 应用程序都是用此编译器提供的工具设计的。在安装 Borland C++ 编译器时,须指明要求安装 Windows 和 DOS 的开发工具。

这一章分为三节。第一节涉及有关 Windows 的语言定义和术语,同时也讨论图形环境。第二节讨论应用程序开发者经常使用的 Windows 项,将讨论 Windows 的成份,如边框、图标、位图等。第三节描述 Windows 资源及建立这些资源的 C++ 工具。Windows 资源包括图标、光标、位图、菜单、热键、对话框及字体。Borland C++ 提供的资源工具是 Resource Workshop。

0.1 Windows 程序设计语言

Borland Windows 是工作于 DOS 上的图形操作环境。这一操作环境将弹出式菜单、运行 Windows 编写的应用程序,及 DOS 下的标准应用程序等结合在一起。图形用户界面是未来的用户界面,而 Windows 现在便提供了这种功能。

0.1.1 Windows 环境快速浏览

如前所述,Windows 是基于图形的多任务操作环境,这一环境工作于 DOS 之上。为这一环境开发的程序(即专门为 Windows 编写的程序)有一致的外观及命令结构。这样,为学习使用各种 Windows 应用程序提供了方便。

为了辅助程序员开发 Windows 应用程序,Windows 提供了大量的内部函数,使用这些函数可实现弹出式菜单、滚动条、对话框、图标,及其他友好的用户界面的特性。程序员可利用 Windows 提供的图形编程语言的优越性,很容易地格式化并输出文本,在这一过程中可选用各种字体和间距。

Windows 允许应用程序以不依赖于硬件的方式处理显示模式、键盘、鼠标、打印机、串行口及系统时钟。不依赖于设备或硬件意味着,同一个应用程序可在硬件配置不同的各种计算机上同样地运行。

0.1.2 使用 Windows 的优越性

同传统的 DOS 下的基于文本方式的环境相比,Windows 对使用者和程序员有着大量的优越之处。Windows 提供了多种的编程能力,其中包括标准化图形界面、多任务能力、OOP 程序设计方法、内存控制、不依赖于硬件及使用动态连接库(DLL)。

0.1.2.1 图形用户界面

最引人注目的 Windows 特性是标准化的图形用户界面,这对用户来说是很方便的。一致的界面使用图形,或称图标,代表磁盘驱动器、文件、子目录及多种操作系统命令及操作。图0.1