

Borland C++ Builder

使用与开发指南

雷霖 李康 陈曦 陈义进 编
史惠康 审校



人民邮电出版社

PEOPLE'S POSTS &
TELECOMMUNICATIONS
PUBLISHING HOUSE



Borland C++ Builder 使用与开发指南

雷 霖 李 康 编
陈 曦 陈义进

史惠康 审校

人民邮电出版社

内 容 提 要

Borland C++ Builder 是 Borland 公司推出的全新的可视化编程工具, 完全面向对象, 能方便、快捷地开发 Windows 应用程序。本书从浅到深讲述了 Borland C++ Builder 的使用环境和编程技巧, 适合于自学 Borland C++ Builder 编程的初、中级读者。

Borland C++ Builder 使用与开发指南

◆ 编 雷 霖 李 康 陈 曦 陈义进
审 校 史惠康

◆ 人民邮电出版社出版发行 北京崇文区夕照寺街 14 号
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 18.5

字数: 454 千字

1998 年 5 月第 1 版

印数: 1—5 000 册

1998 年 5 月北京第 1 次印刷

ISBN 7-115-07097-0/TP·664

定价: 24.00 元

前言

自从美国微软 (Microsoft) 公司于 1995 年 8 月发布 Windows95 以后, 各种个人计算机 (PC) 机上的应用程序都开始向 Windows95 平台转移。因此无论是对于计算机爱好者还是专业程序员来说, 学习如何为 Windows95 编写应用程序都是一条必经之路。

遗憾的是, 传统的程序开发工具在编写 Windows 应用程序时需要高度的技巧和很大的编程量, 即使是在易用性上作了很大改进的开发环境如 Visual C++、Borland C++ 等, 其复杂的操作界面和巨大的源文件也是令初学者望而生畏的。因此快速应用程序开发工具 (RAD) 应运而生并取得了飞速发展, 微软的 Visual Basic、PowerSoft 公司的 PowerBuilder、Borland 公司的 Delphi 等开发环境使用接近口语化的语言、拖放式控件编程、大量的模板以及可重用技术和精心设计的数据库接口, 使得编程即使是对一个新手来说也是可能的了。RAD 工具的广泛使用, 极大促进了软件行业的发展和信息技术对各个社会领域的影响。

美国的 Borland 公司是在 1993 年推出其 RAD 工具 Delphi 1.0 的, Delphi 在秉承前身 Turbo Pascal for Windows 优点的基础上, 围绕 RAD 和数据库两个方面进行了大量改进。RAD 方面最值得称道的是 Delphi 的可视化组件库 VCL (Visual Component Library), 这是一个精心设计的对 Windows 应用程序函数接口 (API) 的包装, 它将许多基本模块封装为易于使用的组件形式; 数据库方面, Borland 公司开发的数据库引擎 BDE (Borland Database Engine), 大大简化了对数据库的操作, 程序员即使对数据库原理及 SQL (结构化查询语言) 不太了解, 也能够通过 BDE 很快作出漂亮的报表来。Delphi 以其技术优势取得了巨大成功并在 1995 年推出了 2.0 版, 在 1997 年推出了 3.0 版。

可是事物总是在发展的, 随着 C++ 成为 ANSI 标准, 这种新的面向对象的程序设计语言迅速成为了程序员最广泛使用的工具。比起 Delphi 所用的 Object Pascal 语言来, 要有多得多的书籍文章是讨论和介绍 C++ 的, 而且许多新技术的软件开发包, 如 ActiveX SDK、DirectX SDK 等都是 C++ 形式的。于是, Borland 公司继 Delphi 之后又在 1997 年推出了 C++ Builder, 另一个优秀的可视化编程环境。C++ Builder 1.0 在某种程度上可以说是 Delphi 的兄弟, 它也基于 VCL, 也使用 BDE, 甚至具有与 Delphi2.01 (比起 2.0 版来扩充了 Internet 编程功能) 几乎完全一样的易于使用的界面和组件库。但是更能吸引我们的是, 它是基于 C++ 的, 可以充分发挥 C++ 语言强大灵活的特点, 而且对于 Windows API、DirectX 等的使用也更加直接了。可以说, C++ Builder 在强大性和易用性之间搭起了一座桥梁, 程序员可以根据任务的复杂程度自己确定在两者之间的平衡点。

在写这篇前言时作者了解到, Borland 公司已经在 1998 年 2 月正式发布了 C++ Builder3.0 版, 这是 Borland 公司在 98 年度发布的最重要的产品之一 (Borland 公司语), 其中增加了许多新技术和新功能, 这显示了 C++ Builder 的生命力和 Borland 公司推广它的

5/5/99/05

决心。把它介绍给广大编程爱好者是我们的心愿，衷心希望这本书能有助于读者对 C++ Builder 的学习。

本书的具体阅读方法，详见随后的导读部分。

本书由史惠康策划。一至七章由雷霖、李康、陈曦、陈义进编写，第八章由钟建平编写，第九章由郭立明编写，李雨编写了前言和全书导读并帮助编制了部分程序。另外参加编写的还有：张峰、李力、韩小云、范海波等。史惠康负责全书的统稿，并进行了审阅。

由于作者经验有限，纰漏在所难免，敬请读者批评指正。

编者

1998 年 3 月于北京

全书导读

本书的主要目的是帮助读者快速掌握 Borland C++ Builder 这一功能强大的可视化编程工具。读者只需具有其它任何一种编程语言或编程工具的使用经验即可。通过阅读本书，读者可以很快的熟悉 Borland C++ Builder 的界面并能熟练使用 Borland C++ Builder 所提供的各种编程和调试工具。

本书内容简介

本书共分为九章，每一章的前面有对本章内容的简要介绍，读者可以在阅读前对各章的内容作一大概的了解，有一定 Delphi 或 C++ 编程基础的读者可以跳过其中的某些章节。每章的末尾有对本章内容的小结，所列内容均为本章必须重点掌握或需要提高的，读者在学习和阅读时要注意参照。

以下是各章内容的简要介绍，读者可先对本书内容作一大概了解。

第一章 Borland C++ Builder 概述

在本书的第一章里，概要介绍了 BCB 的特征；BCB 中使用和创建的文件类型；可视化编程的基础及其基本要素：组件、窗体、特性和事件；还有如何管理一个工程项目等内容。对 Delphi 比较熟悉的读者只需略读本章即可。

第二章 进入 C++ 的世界

介绍了 C++ 的一些基本语法，以使读者对类、构造函数与析构函数、函数重载和继承等面向对象编程的概念有一个基本的了解，从而能在 BCB 集成开发环境中使用 C++ 语言编写代码。对 C++ 语言已经有一定了解的读者可以跳过本章。

第三章 Borland C++ Builder 的安装

介绍了 BCB 的安装和第一次运行的方法，读者可按照本章的步骤在自己的机器上正确的安装上 BCB，为正式学习和使用 BCB 作好准备。

第四章 响应用户的输出

全面介绍了与键盘和鼠标有关的编程方法。包括：对 OnKeyDown、OnKeyPress 和 OnKeyUp 基本事件的理解；如何处理键盘输入的焦点；鼠标编程的基本概念；以及与 Object Pascal 中的集合概念相应的 BCB 中的集合。

第五章 菜单设计

介绍了如何利用 BCB 的 MainMenu 组件和 PopupMenu 组件方便而快捷地创建主菜单和弹出式菜单及其菜单项，并用简单的编程实现菜单项的功能；并且介绍了多层菜单、动态菜单和菜单融合等能使应用程序更加充满活力的技术。

第六章 文本编辑器的设计

首先介绍了多页面界面和多文本界面技术,然后介绍了 Edit 组件、Memo 组件和 RichEdit 组件等文本编辑组件的用法,最后以一个实例说明了文本编辑器的制作,同时说明了文本编辑器中常用的对话框组件的用法。

第七章 文件管理

首先讲解了 BCB 中的文件类型以及文件的打开、创建、移动和删除等文件操作的标准过程,然后介绍了 BCB 所提供的 FileListBox、FileFilterComloList、DirectoryListBox 和 DriveComboList 等文件管理组件的用法,最后举了一个综合的例子,用以说明如何在应用程序中编写实现文件管理功能的代码。

第八章 图形图像编程

首先介绍了图形图像编程中最常用的画布对象 (TCanvas Object) 的属性和方法,然后介绍了一些常用的图形组件如 Shape 组件、PaintBox 组件和 Image 组件的用法。最后讨论了图形重绘、屏幕外位图和拖放操作等图形图像编程的技术和技巧。

第九章 异常处理与程序调试

介绍了增强程序的健壮性并提高开发效率的办法,包括如何处理异常和调试程序两部分。首先介绍的是如何在程序中增加错误处理模块,包括利用 VCL 内置的异常处理模块和增加自己的异常事件处理;另外还介绍了如何在 BCB 的集成开发环境 (IDE) 中使用其功能强大的内置调试器 (Integrated Debugger) 来调试应用程序。对于一个优秀的程序员来说,这一章的内容是必须掌握的。

另外,本书提供了很多实例,有些还提供了源程序,读者最好边阅读边照本书的做法在计算机上试着做一做。

目 录

第一章	Borland C++ Builder 概述	1
1.1	几个基本问题	1
1.1.1	BCB 的形式	1
1.1.2	关于 VCL	2
1.1.3	RAD 与 BCB	3
1.2	BCB 快速入门	3
1.2.1	进入 BCB 环境	4
1.2.2	BCB 集成开发环境 (IDE) 介绍	4
1.2.3	设计简单的用户界面	5
1.2.4	改变对象的属性	8
1.2.5	编写事件处理过程	11
1.2.6	使用联机帮助 Help	13
1.3	BCB 工程项目的设计与管理	13
1.3.1	创建多窗体应用程序	14
1.3.2	管理多窗体应用程序	16
1.3.3	使用模板与向导创建窗体	18
1.3.4	使用工程模板和应用程序向导	19
1.4	VCL 库中常用的组件介绍	20
1.4.1	常用的文本相关组件	20
1.4.2	按钮和检查框组件	22
1.4.3	分组、分界组件	23
1.4.4	图形、图象组件	24
1.4.5	关系图、文件列表组件	24
1.4.6	滚动组件	25
1.4.7	网格、表格组件	25
1.4.8	多媒体 (MultiMedia) 和 OLE 组件	26
1.4.9	菜单组件	26
1.4.10	定时器组件	26
1.4.11	使用公用对话框组件	27
1.5	本章小结	29
第二章	进入 C++ 的世界	31
2.1	编写一个简单的 C++ 程序	31
2.1.1	编写 Hello.cpp 程序代码	32
2.1.2	编译并链接 Hello.cpp	33
2.1.3	执行 Hello.exe	33
2.2	对象	34

2.3	栈对象的例子	36
2.4	构造函数与析构函数	40
2.4.1	构造函数	40
2.4.2	析构函数	43
2.4.3	构造函数和析构函数的调用问题	44
2.5	this 指针的使用	44
2.6	Static 类型的成员	46
2.7	派生类和层次及虚函数	47
2.7.1	派生的类	48
2.7.2	类的层次	50
2.7.3	虚函数与虚基类	51
2.8	本章小结	65
第三章	Borland C++ Builder 的安装	67
3.1	Borland C++ Builder 1.0 的安装	67
3.1.1	系统环境准备	67
3.1.2	开始安装	67
3.1.3	安装类型选择	69
3.1.4	安装路径选择	71
3.1.5	正式安装 Borland C++ Builder	72
3.2	修补 Borland C++ Builder 1.0 的错误	73
3.2.1	第一个修补程序	73
3.2.2	第二个修补程序	74
3.2.3	更新 Microsoft SDK Help 文件	75
3.2.4	更新 BCB Help 文件	75
3.2.5	更改错误 HELP 文件	76
3.2.6	NetMange 的国际互联网组件的更新	76
3.3	安装后第一次运行 Borland C++ Builder 1.0	76
3.3.1	创建 BCB 快捷方式	76
3.3.2	运行 Borland C++ Builder 1.0	78
3.4	本章小结	79
第四章	响应用户的输入	81
4.1	处理键盘输入	81
4.1.1	理解三个基本键盘响应事件	81
4.1.2	键盘输入的焦点	91
4.2	处理鼠标输入	93
4.2.1	理解鼠标编程的基本概念	93
4.3	BCB 中的集合	100
4.3.1	集合的基本概念	100
4.3.2	TshiftState 类	100

4.4 本章小结	105
第五章 菜单设计	107
5.1 菜单界面的设计	107
5.2 菜单代码的设计	112
5.2.1 Check 标记的使用	113
5.2.2 弹出式菜单	116
5.2.3 动态菜单	117
5.2.4 快捷键的使用	120
5.2.5 命令激活和命令失效	122
5.2.6 菜单的提示(Hint)	122
5.2.7 融合 MainMenu 对象	123
5.2.8 修改系统菜单	127
5.3 本章小结	128
第六章 文本编辑器的设计	129
6.1 多页面界面 (MPI)	129
6.1.1 概述	129
6.1.2 Notebook 组件与 TabSet 组件	129
6.1.3 TabbedNotebook 组件	133
6.1.4 TabControl 组件	134
6.1.5 PageControl 组件	135
6.1.6 动态创建页面	136
6.2 多文本界面 (MDI)	137
6.2.1 概述	137
6.2.2 创建 MDI 父窗体	138
6.2.3 创建 MDI 子窗体	138
6.2.4 创建 MDI 主菜单	139
6.3 常用文本编辑组件介绍	142
6.3.1 Edit 组件	142
6.3.2 Memo 组件	144
6.3.3 RichEdit 组件	145
6.4 创建文本编辑器	147
6.4.1 完善子窗体	147
6.4.2 设置字体	151
6.4.3 设置段落格式	159
6.4.4 剪切、拷贝、粘贴和取消功能	162
6.4.5 查找与替换	164
6.4.6 文件的打开与保存	167
6.4.7 文件打印	172
6.5 本章小结	176

第七章 文件管理	177
7.1 文件类型和标准过程	177
7.1.1 文本文件	177
7.1.2 二进制文件	178
7.1.3 INI 文件	178
7.1.4 C++ Builder 的文件管理标准过程	179
7.2 文件组件的应用	181
7.2.1 文件组件及其相互关系	181
7.2.2 程序的设计	182
7.2.3 程序功能和实现	183
7.2.4 文件管理器的实现	187
7.3 文件的应用	202
7.3.1 概述	202
7.3.2 设计步骤	202
7.3.3 二进制文件的打开和创建	204
7.3.4 二进制文件的读入	207
7.3.5 增加记录	207
7.3.6 修改记录	208
7.3.7 文件和系统的关闭	210
7.4 本章小结	211
第八章 图形图象编程	213
8.1 画布对象 (TCanvas Object)	213
8.1.1 什么是画布对象	213
8.1.2 Canvas 的属性	213
8.1.3 TColor 类型	214
8.1.4 TPen 对象	215
8.1.5 TBrush 对象	217
8.1.6 Canvas 的方法	218
8.1.7 使用 Canvas 的属性和方法绘图	221
8.2 图形程序设计技术	230
8.2.1 图形类关系	230
8.2.2 使用 PaintBox 组件	231
8.2.3 使用组件画图	233
8.2.4 拖放编程	240
8.2.5 使用图象编辑器(Image Editor)	243
8.2.6 图形重绘	245
8.2.7 动画效果	246
8.3 本章小结	250
第九章 异常处理与程序调试	251

9.1 异常基本理论	251
9.1.1 错误的来源	251
9.1.2 关键词与术语	252
9.2 BCB 中的异常类	252
9.2.1 一个新的异常处理体系	252
9.2.2 BCB 中的异常类	253
9.2.3 创建自己的异常类	258
9.3 实际使用异常	258
9.3.1 VCL 中的缺省异常处理	259
9.3.2 try 块和 catch 块	259
9.3.3 唤起与重引发异常	262
9.3.4 创建适合自己程序的异常类	263
9.3.5 分析异常信息	263
9.3.6 建立自己的异常处理入口	270
9.4 程序调试简介	271
9.4.1 调试的准备和开始	271
9.4.2 程序运行的控制	272
9.4.3 断点的使用	273
9.4.4 监视数据的值	274
9.5 本章小结	282

第一章 Borland C++ Builder 概述

Borland C++ Builder (以下简称 BCB) 是 Borland 公司继 Delphi 之后推出的又一个优秀的可视化编程环境, 它在 32 位 Windows 环境下为我们提供了一种极具吸引力的快速 Windows 应用程序开发系统 (RAD)。它基于最流行的面向对象程序设计语言 C++, 采用领先的数据库技术, 并结合使用了图形用户界面 (GUI) 的许多先进特性和设计思想。对于正在或将要使用 BCB 开发应用软件广大的程序开发人员来讲, 了解 BCB 的数据库功能、可视形式继承以及 ActiveX 支持, 了解它对 Windows API 的方方面面的支持, 充分领略它的超级面向对象结构和支持本地组件的优势, 无疑会大大地提高编程效率。

作为本书的第一章, 以下将概要介绍 BCB 的特征; BCB 中使用和创建的文件类型; 可视化编程的基础及其基本要素: 组件、窗体、特性和事件; 还有如何管理一个工程项目。对 Delphi 比较熟悉的读者只需略读本章即可。

1.1 几个基本问题

1.1.1 BCB 的形式

C++ 是一种优秀的计算机程序设计语言, 它的语法简洁精炼, 程序代码效率很高, 而且, 较之 C 有更加丰富的数据结构及更加强大的功能, 所以很多开发者都把它作为首选的开发语言。而 Borland 公司的 Delphi 是基于 Object Pascal 的应用程序开发系统, 这限制了它的使用范围, 因此 Borland 公司又于 97 年推出了它的 C++ 姊妹产品 BCB。

BCB 是一个基于 C++ 语言的应用程序开发系统, 但它远不止于此。一个 BCB 程序首先是应用程序框架, 而这一框架正是应用程序的“骨架”。在骨架上即使没有附着任何东西, 仍可以严格地按照设计运行 (图 1-1 就是启动 BCB 后自动生成的程序框架)。缺省的应用程序是一个空白的窗体 (Form), 可以运行它, 得到如图 1-2 所示的一个空白的窗口, 这个窗口具有 Windows 窗口的全部性质: 可以被放大缩小、移动、最大最小化等, 但开发者却没有编写一行程序。因此, 可以说应用程序框架通过提供所有应用程序共有的东西, 为用户应用程序的开发打下了良好的基础。BCB 已经为开发者做好了一切基础工作, 在框架中提供了所有应用程序所共有的东西——程序框架就是一个已经完成的可运行应用程序, 只是不处理任何事情。

在空白窗口的背后, 应用程序的框架正在等待用户的输入: 如击键、鼠标单击和鼠标的移动等。窗口除了响应 Windows 的基本操作如: 改变尺寸、移动、缩放等以外, 只是接受用户的输入, 然后再忽略, 因为开发者并未告诉它接收到用户输入后作何反应。BCB 把 Windows 编程的回调、句柄处理等繁复过程都隐藏在后台自己完成了, 这样开发者可以不为它们所困扰, 轻松从容地对可视组件进行编程, 他们所需要做的, 只是在“骨架”中加入程序, 往程序中加入完成所需功能的代码而已。

理解这种程序运行机制的关键在于弄清激励与响应的概念。用户在 BCB 窗体中产生

输入即是发出激励，而编写好的作为响应的程序代码便会执行。如果开发者没有编写响应程序，就不会有任何响应。激励被称为事件而响应被称为过程。在 BCB 程序开发中整个“编程”部分包括开发直接或间接的对应于事件的程序代码。

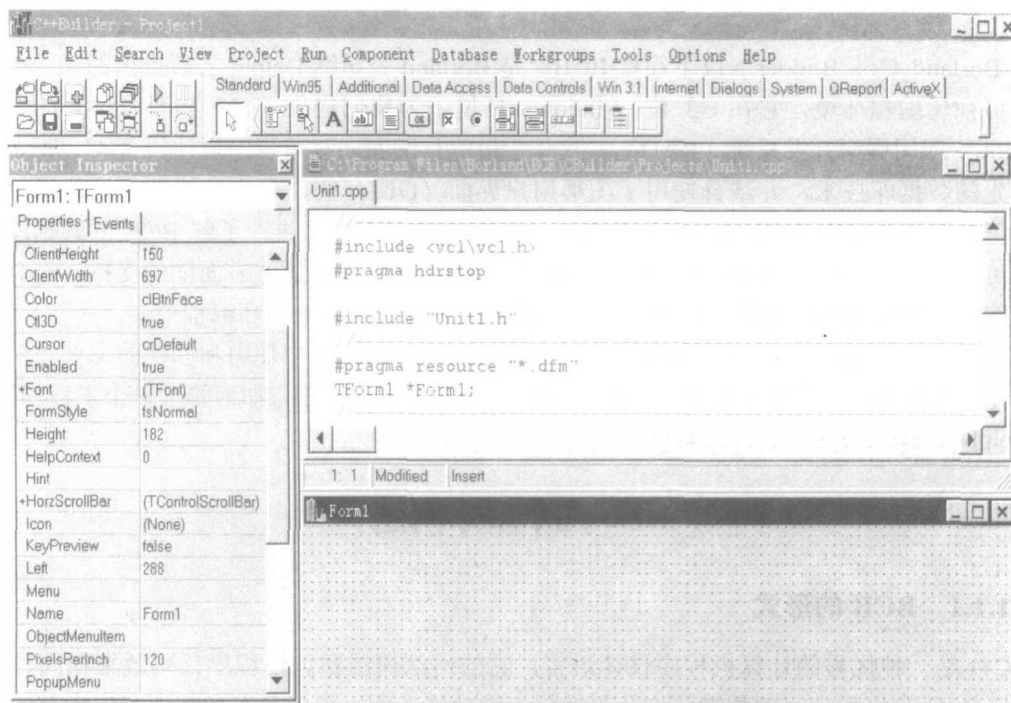


图 1-1 运行中的 Borland C++ Builder

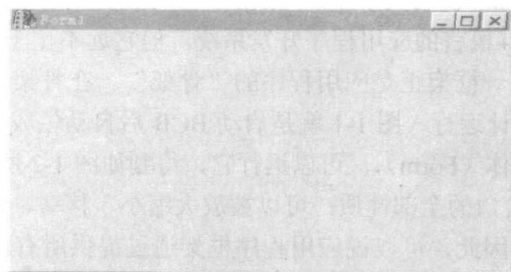


图 1-2 程序框架运行结果

1.1.2 关于 VCL

Borland 公司在 BCB 之前，还推出过另外一个著名的基于 C++ 的开发工具 Borland C++ (BC)，其最新版本是 97 年发行的 5.02 版。那为什么还要推出 BCB 呢？因为在与 Microsoft 的基于“微软基础类库”(MFC)的 C++ 开发工具 Visual C++ 竞争市场中，Borland 公司基于“对象视窗库”(OWL)的 BC 可以说是节节败退；另一方面，Borland 公司基于“可视组件库”(VCL)的可视化开发工具 Delphi 则一经推出就获得了很大成功，凭借其强大功能和易用性胜过了 Microsoft 的 Visual Basic，被誉为“VB Killer”。所以 Borland

公司决定推出 C++ 和 VCL 强强联合的新产品——Borland C++ Builder。

VCL 是可视化组件库的简称，它是一个面向对象的函数库，完全支持所有面向对象编程的标准概念如继承、多态和封装等。VCL 是 Delphi 与 BCB 的基础，这就如同 OWL 是 BC5 的基础、MFC 是 MSVC 的基础一样；而 VCL 与后两者的主要区别在于：它是建立在组件（Components）、特性（Properties）和事件（Event）等关键概念之上的；而 OWL 与 MFC 则没有这些特征。

关于 VCL 还要说明的一点是它是由 Object Pascal 写成的，也就是说它用的是和 Delphi 一样的代码库，而非同一个库又由 Pascal 改写为 C++ 的。不过这对使用者来说并非特别重要：Object Pascal 比起 C++ 来有其弱处，但在速度与代码效率上相差并不大，即使用 C++ 为 BCB 完全重新改写过 VCL，BCB 编出来的程序也不见得就能更快一些或是更小一些。

1.1.3 RAD 与 BCB

RAD 设计的方法学包括创建包含可以重复使用的组件的窗体以及为这些组件创建特定的事件代码，还包括生成一个可执行的和调试产生的最后的应用程序。在所有的 RAD 工具中，最早也是最著名的要数 Microsoft 的 Visual Basic (VB) 了，此外还有 PowerBuilder、Delphi 等，现在 BCB 作为一个新的角色也加入到竞争行列中来了。要选择合适的 RAD 工具，开发者应该考虑几个因素，最主要的是看哪个工具能更大程度地提高生产能力，具体说来有以下几个方面：

1. 快速的应用程序开发。包括环境本身的使用是否简单易用，开发者可以重复应用的组件是否丰富，辅助工具是否完备等。
2. 性能。对有大量市场的应用程序来说，高性能是保持竞争优势的关键。
3. 开发数据库应用程序。RAD 工具的一个重要（可能是最大市场）的应用方向是数据库，因而是否适于开发数据库应用程序也应作为一个重要的考虑因素。

现有的各种 RAD 工具各有其优点，而且随着版本的提高日益优秀，很难说哪一个能明显优于其余产品。而 BCB 应该是相当不错的选择：它的集成开发环境界面继承了 Delphi 2.0 的优秀传统，易学好用；32 位版本的 SQL Link 使得连接到 Oracle、MS SQL Server 6.x、DB2 和 SyBase System 更为方便；查询引擎优化后的 BDE 不但功能增强了，速度也比 Delphi 1.0 中的版本快了很多。BCB 还提供了 OLE Automation 功能，可以很方便地创建 OLE Automation Object。此外它还提供了新的存储各种资源的工具即所谓的“Object Repository”（对象仓储），用 Object Repository 可以存储所开发的表格及其它资源，体现了软件重用的设计思想。

而 BCB 还不止是一种 RAD 工具，开发者还可以用它编写 API 代码的程序，可以说只要是 BC 和 MSVC 能做的事情它都能完成，如编写控制台程序、开发组件和 ActiveX 组件、编写 DLL 程序等等。从这个角度讲，BCB 是一个强大的系统。

1.2 BCB 快速入门

在这一节中，我们来开发一个浏览图形文件的小程序。随着开发的过程，逐步介绍

BCB 的主要组件及其操作方法, 以使开发者对 BCB 的可视化编程有一个直观、快捷的了解。

1.2.1 进入 BCB 环境

启动 BCB, 先启动 Windows 95 并打开 Borland C++Builder 程序组, 双击“C++Builder”程序项以启动 BCB 应用程序。(可以参见图 1-1)

为避免隐藏在 BCB 后的其它程序扰乱版面, 分散开发者的注意力, 不妨在启动 BCB 前关掉或最小化其它应用程序, 这样屏幕上就只留下 BCB 窗口可见了。

加载 BCB 后, 屏幕上会出现四个窗口:

1. 标题为“C++Builder - Project1”的 BCB 主窗口。
2. Object Inspector 窗口。任何时候可以按下 F11 键使其可见。
3. 标题为“Form1”的窗体 (Form) 窗口。
4. 标题为“Unit1.cpp”的代码编辑窗口。刚启动时这一窗口的大部分被“Form1”窗体所掩盖。将“Form1”窗体移开, 或单击 Form1 窗体下方的状态行, 或按下 F12 键在窗体窗口和代码编辑窗口之间切换, 可以使其全部可见。在“Form1”窗体的任意可见位置单击鼠标, 可以使主窗体可见。

以下我们将对这四个窗口分别进行介绍。

1.2.2 BCB 集成开发环境 (IDE) 介绍

主窗口 (Main Form)

BCB 的主窗口位于屏幕的上端, 包括菜单 (Menu)、加速条 (Speed Bar) 和组件选项板 (Component Panel)。Menu 是下拉式主菜单。Speed Bar 位于主窗口的左下端, 由两排共 14 个加速按钮组成。这些按钮是菜单功能的快捷方式, 各种图标直观地表示了它能执行的动作。Component Panel 由一行、若干页对象按钮所组成, 利用它来选择需要的组件并将它放到窗体中去。

对象检查器 (Object Inspector)

Object Inspector 窗口含有两页: Properties 页显示窗体中当前被选择组件的属性信息, 并允许改变对象的属性; Events 页列出了当前组件可以响应的事件。按动 Object Inspector 下端的“Events”页标签, 使得 Events 页可见, 这一定事件后边的空白处, 可以定义对象接受到相应事件时执行的动作。首次启动时, Object Inspector 窗口显示的是当前窗体 Form1 的属性。Object Inspector 根据对象属性的多少, 决定是否有滚行显示。移动滚行条, 可以查看当前对象的全部属性。

此外, Object Inspector 上还有对象选择器 (Object Selector), 位于 Object Inspector 上方的下拉式菜单中。它显示了窗体上所有组件的名称和类型, 也包含窗体本身。开发者可以用 Object Selector 很容易地在窗体的各个组件之间切换, 也可以快速地回到窗体本身。当窗体中含有较多的对象时, 开发者会发现这是切换对象尤其是回到窗体的最快捷途径。

想使 Object Inspector 一直可见, 可将鼠标移到 Object Inspector 上, 按动右键, 以启动 Object Inspector 的弹出式菜单, 将其设置为 Stay On Top。这对初学者常是一个很重要的设置方式。

窗体窗口

Forms 窗口是开展大部分设计的工作区域。首次启动 BCB 时显示的是窗体 Form1。可以把组件放在窗体中, 通过移动位置、改变尺寸等操作随心所欲地安排它们, 以此来开发应用程序的用户界面。开发者可以把窗体想象成一个可以放置其它组件的容器。窗体上有栅格 (Grids), 供放置组件时对齐位置用, 要改变栅格的设置, 只要选中菜单中的 Options 下的 Environment..., 然后在 Form Designer 组中修改 “Display Grid” 和 “Grid size X”、“Grid size Y” 选项即可。在程序运行时 Grids 是不可见的。

一个真正的应用程序可能有不止一个窗口, 开发者可以选用不同的窗体进行设计。其它窗体可以是对话框 (Dialog Box)、数据录入框等。

代码窗口

代码窗口一开始处于窗体窗口之下。因为在 BCB 中, 设计用户界面直接在窗体中进行, 运行结果和设计模板完全一致。当组件被放到窗体上时, BCB 会自动生成大部分的用户界面代码。开发者所应做的只是在它为之开发者生成的框架中加入完成所需功能的程序段而已。点动 Form1 的状态行使代码窗口可见。图 1-3 显示了空窗体 Form1 的代码窗口。

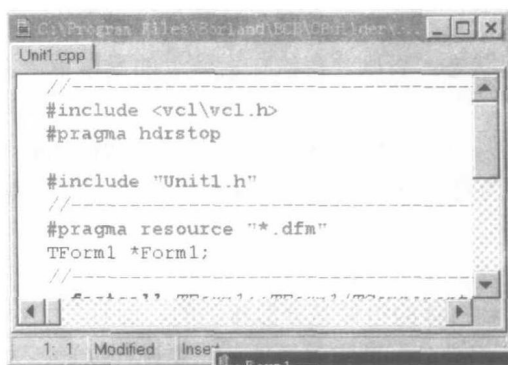


图 1-3 BCB 的代码编辑器

这个窗口中是代码编辑器, 可以在其中书写 BCB 应用程序的源代码。当程序中含有不止一个窗口时, 会有几个库单元的源程序出现在代码编辑器中。代码编辑器的标题条中显示了当前正在编辑的库单元文件名。要查看某一特定程序的源代码, 只需用鼠标点动写有该库单元文件名的页标签, 就可以对该库单元进行编辑了。

1.2.3 设计简单的用户界面

首先从空白窗体开始工作。在菜单上选用 File 中 New Application 菜单项来生成新的工