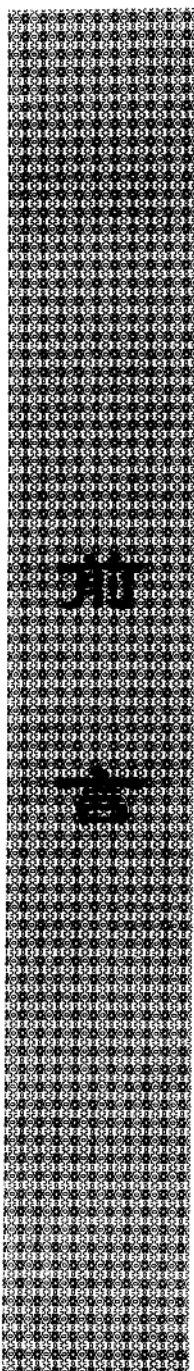


Borland C++ Builder 使用指南

张锋铭 陈 静 陈宇鸿 编



西安电子科技大学出版社



Borland 公司终于为 C++ 语言撩开了神秘的面纱, 于 1998 年推出了该公司最新的 32 位 Windows 开发工具 C++Builder。Borland C++Builder 不但继承了 C++ 结构化语言的特点, 而且采用了与微软公司的 Visual Basic 非常类似的用户界面。可以说, Borland C++Builder 是 Visual Basic 外观和 C++ 内涵相结合的一个优秀产物。

Borland 公司之所以推出这样的编程语言, 完全是市场竞争的结果。微软公司于 1991 年推出的 Visual Basic 1.0 给 Borland 公司的 C 语言市场造成了很大的冲击。随着 Visual Basic 高级版本的相继推出, 其功能越来越完善, 基本上能完成 C++ 语言 90% 以上的功能。但是由于 Visual Basic 应用程序的执行速度不够理想, 因此, 微软公司又推出了 Visual C++ 语言。虽然 Visual C++ 比传统的程序设计环境以及 Turbo C++、Borland C++ 具有更友好的界面, 但是, 单就应用程序界面设计来说, 它比 Visual Basic 逊色得多。因此, 一般的软件设计者都将这两个开发工具结合起来使用。Borland 公司开始似乎并不在意 Visual Basic 和 Visual C++ 所带来的市场冲击, 它把重点放在了对旧的 Turbo C 市场的维护上, 并相继推出了 Borland C++ 系列。尽管每一个 Borland C++ 高级版本都较前一个版本有较大的改进, 但用户越来越感觉到用 Borland C++ 编写 Windows 应用程序是一项费力的工作, 不如 Visual Basic 或 Visual C++, 尤其不如 Visual Basic 那样轻松自如。于是, 即便是一些老的 Borland C++ 用户也开始“投靠”VB 或 VC。

Borland 公司终于认识到自己在编程语言市场的被动地位, 经过几年艰辛的探索和非常大度的借鉴, 推出了相当于集 VB 和 VC 于一身的 Borland C++Builder。可以说 Borland C++Builder 不但适合对 VB 和 VC 都有所了解的老用户, 也适合对 C++ 有一定了解的初级用户。

作为一种 Windows 下的 GUI 式的 OOP 语言, Borland C++Builder 同 VB 一样, 强调可视性、反馈性和调试功能。

可视性: 用户在进行设计程序时不再像以往那样进行封闭式的代码编辑, 整个程序编辑过程方便直观。

反馈性: 用户和 Borland C++Builder 之间, 无论是在程序设计阶段, 还是在程序运行期间, 两者之间始终都有良好的交互和反馈。

调试功能: Borland C++Builder 提供了集成调试器, 大大方便

07/5/03

了用户对程序的调试过程。

Borland C++Builder 可以实现 Windows 的绝大多数高级功能,如多重窗口、对象的链接与嵌入(OLE)、动态数据交换(DDE)、动态链接库(DLL)函数的调用、字体、颜色选择以及文件列表对话框等。

由于 Borland C++Builder 刚刚推出不久,因此本书主要介绍编写常用应用程序的基本方法。其实当用户基本掌握了 Borland C++Builder 后,编写复杂的应用程序也不成问题。依据这个目的,我们将各章节的内容安排如下:

第一章:主要介绍 C 语言的发展状况、C++Builder 的概述、如何安装 C++Builder,另外我们还将 C++Builder 和 Borland C++以及 Visual Basic 进行了比较。

第二章:主要介绍 C++Builder 的集成开发环境,在此环境下如何使用表格编辑器、代码编辑器、组件模板、对象浏览器、对象资源,最后还给出了一个例子,用来说明编制程序的基本步骤。

第三章:主要介绍了 C++语言的基本数据类型、基本语句、对象、类、指针等,这些都是使用 C++Builder 编制程序的基本元素。

第四章:主要介绍如何建立美观的菜单、建立嵌套菜单和弹出菜单以及利用菜单模板创建菜单等。

第五章:主要介绍单行文本框、多行文本框、标签、列表框、组合框创建及控制以及滚动条、状态条、轨迹条、进度条的创建和使用。

第六章:主要介绍图形图像的控制、图片按钮以及如何利用画布进行绘图,最后给出了绘画实例。为了使用户更好地掌握图形图像的控制,还讲解了棒打昆虫和俄罗斯方块游戏的编制过程。

第七章:主要介绍基本鼠标事件及其响应,鼠标的拖曳与放下,最后还介绍了两个按钮设计小技巧。

第八章:主要介绍 windows 环境的应用,如多重文件界面 MDI、文件列表框、对话框以及 Win95 组件条。

第九章:主要介绍动态数据交换 DDE、DDE 模式、对话对象、主题和内容、DDE 客户、DDE 服务、连接事件以及连接方法等。

第十章:主要介绍有关集成调试器、错误类型、调试过程、控制程序执行的方法、暂停程序、检查程序数据值的状态并查看输出结果、评估和更改表达式的值、查看程序的低级代码以及通过更改程序的数据值来测试错误。

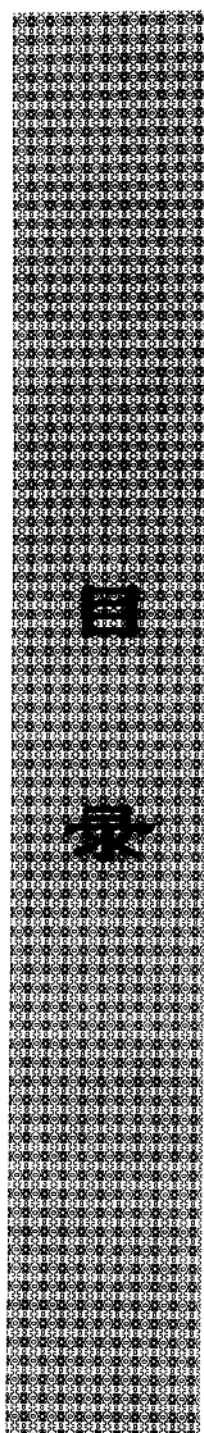
在附录 A 中,主要介绍了 C++的一些系统函数。在实际工作中,一个 C++用户对于一些常用系统函数已非常了解,而且这方面的资料也很多。但有时用户也会遇到一些陌生的函数,往往就是这些函数耽误和影响程序设计者的进程和思路,因此附录 A 侧重讲述非常用系统函数的使用。在附录 C 中,给出了棒打昆虫和俄罗斯方块游戏的 Borland C++源程序,用户可以将其与第六章中这两个游戏的 C++Builder 源程序进行比较学习,加深

对 C++Builder 语言的理解和运用。

本书是作者在熟练掌握 Borland C++Builder 的基础上编写而成的，也可以说本书是作者编程经验的一种总结。但由于作者本身的水平有限，书中肯定存在粗疏和错误，敬请读者指正。

编 者

1998 年 3 月



第一章 面向对象的程序设计语言和 C++Builder	1
1.1 C 语言的发展状况	2
1.2 C++Builder 概述	2
1.3 安装 C++Builder	5
1.4 C++Builder 和 BorlandC++ 的比较	7
1.5 C++Builder 和 Visual Basic 的比较	9
第二章 C++Builder 开发环境和程序设计概述	12
2.1 C++Builder 的开发环境	12
2.2 使用表格编辑器	17
2.3 使用代码编辑器	18
2.4 使用组件模板	19
2.5 使用对象浏览器	23
2.6 使用对象资源	24
2.7 对象资源使用选择	24
2.8 编制程序的基本步骤	25
2.9 创建和管理工程文件	28
第三章 C++Builder 程序设计语言元素	32
3.1 程序的注释	32
3.2 变量的类型、定义和作用域	33
3.3 基本语句	36
3.4 数组	41
3.5 指针	41
3.6 类和对象	44
第四章 菜单的制作与控制	48
4.1 建立美观的菜单	48
4.2 嵌套菜单和弹出菜单	52
4.3 使用菜单模板创建菜单	55
4.4 确认框和无线钮	57
第五章 输入和输出	60
5.1 输入和显示文本框的内容	60
5.2 标签的创建与控制	66
5.3 列表框的建立与控制	69
5.4 组合框的建立与控制	74
5.5 滚动条的建立及控制	76

5.6	状态条的建立和控制	78
5.7	轨迹条的建立与控制	81
5.8	进度条的创建	84
第六章	C++Builder 的图像、图形功能	86
6.1	图形图像的控制	87
6.2	图片按钮	89
6.3	画布	91
6.4	绘画实例	101
6.5	游戏制作	105
第七章	鼠标事件的响应	133
7.1	基本鼠标事件	134
7.2	基本鼠标事件的应用	135
7.3	单击鼠标与双击鼠标	136
7.4	鼠标的拖曳与放下	138
7.5	按钮设计小技巧	141
第八章	环境应用	144
8.1	多重文件界面 MDI	144
8.2	文件界面	147
8.3	对话框	152
8.4	Win95 组件	158
8.5	实例	159
第九章	应用程序间的通信	168
9.1	动态数据交换	169
9.2	DDE 的模式	169
9.3	Service、Topic 及 Item	170
9.4	DDE 客户组件	170
9.5	DDE 服务组件	173
9.6	连接事件	174
9.7	连接方法	175
第十章	程序的运行及调试	178
10.1	关于集成调试器	178
10.2	错误类型	179
10.3	调试过程	180

10.4	控制程序执行.....	183
10.5	暂停程序.....	185
10.6	检查程序数据值的状态并查看输出结果.....	185
10.7	评估和更改表达式的值.....	187
10.8	查看程序的低级代码.....	188
10.9	通过更改程序的数据值来测试错误.....	190
附录 A	Borland C++ Bullder 函数库	192
附录 B	可视组件库	234
附录 C	两个游戏的 Borland C++ 源程序	235



面向对象的程序设计 语言和 C++Builder

本章内容:



★ C 语言的发展状况

★ C++Builder 概述

★ 安装 C++Builder

★ C++Builder 和 Borland C++ 的比较

★ C++Builder 和 Visual Basic 的比较

面向对象程序设计(Object Oriented Programming, 简称 OOP)是一种全新的程序设计方法,它第一次从根本上改变了半个世纪以来的编程模式。其基本思想是:通过建立和具体实际相对应的软件对象,并通过对这些对象的组合来创建具体的应用。所谓对象是指在传统的数据结构中加入处理这些数据的过程和函数所构成的实体,因此,对象是数据和程序的结合体。Windows 下的程序设计语言如 Borland C++、Microsoft C++、Visual C++ 和 Visual Basic 等都是面向对象的程序设计语言,虽然风格各异,但都具有共同的思维和编程模式。

面向对象的程序设计语言必须具备以下特征:

(1) 对象:对象是一种将数据和过程或函数合为一体的数据结构。虽然大多数编程系统都提供了数据和过程之间的关系,但只有面向对象系统才将数据及其处理过程和函数融合在同一对象中。

(2) 可编程性:面向对象的语言必须是可编程的,以实现对象的控制和应用。

(3) 可访问性:对象具有能对它的内部某些元素实施保护,同时对象属性和方法能被

外部其它程序访问。

(4) 可预见性: 对象的建立和删除对用户是可预见的。在 C++Builder 中, 用户既可以在程序设计时静态创建和删除对象, 也可以在程序运行时动态地建立和删除对象。

(5) 继承性: 通过对现有对象进行增加或修改来产生新的对象, 这是面向对象的程序设计语言所必须具有的功能。一般地, 最初的类被称为基类, 从基类扩展出来的类称为派生类。显然, 修改现有类进行派生是为了产生针对性更强、实用性更好的新类。

面向对象技术使用户可以忽视对象的内部机制和具体实现过程, 将精力集中在数据结构的组织和对象的相互联系上, 从而使用户能够方便、快捷、可靠地设计出各类软件产品。

1.1 C 语言的发展状况

1970 年美国贝尔实验室的 Ken Thompson 和 Dennis Ritchie 完成了 UNIX 的初版。与此同时, 他们还改写了由 Martin Richards 开发的 BCPC 语言, 形成了一种称为 B 的语言, 此后 B 语言又被进一步的改进和完善, 形成了我们现在称之为 C 的语言。

C 语言大约形成于 1972 年, 1973 年 Dennis Ritchie 把 UNIX 系统的 90% 又用 C 语言进行了改写, 并在 PDP-11 小型机上完成了用 C 语言及汇编语言编写的 UNIX 操作系统的调式, 并将其投入运行, 因而随着 UNIX 的移植和推广, C 语言也得到了推广和发展。美国 Borland 公司于 1989 年推出了 TurboC2.0, 它是一种很好的程序设计语言。但是, 随着软件处理对象从简单的数字和字符串发展到目前的图、文、声、像, 信息量越来越大, 越来越复杂, 因而对程序的设计方法提出了更高的要求, 随之在 80 年代出现了一种崭新的程序设计方法——面向对象的程序设计方法。结构程序设计的基本单位是模块, 而面向对象的程序基本单位是对象, 因此 C 语言又进一步发展和充实, 出现了支持面向对象的程序设计语言 C++。

Borland 公司继推出面向程序设计的语言 TurboC++ 之后, 又相继推出了 BorlandC++3、BorlandC++4 和 BorlandC++5, 使 C 语言得到了很大发展和完善。但是, 与 Microsoft 公司的产品 Visual Basic 语言相比, BorlandC++ 语言在对象的可视性和可编程性上逊色许多。为此, Borland 公司于 1997 年 6 月推出 C++Builder1.0 测试版, 征集用户意见与建议, 并将于 1998 年正式发布 C++Builder 这一全新的程序设计语言, 与 Microsoft 公司的 Visual Basic 及 Visual C++ 语言相抗衡。

1.2 C++Builder 概述

1.2.1 什么是 C++Builder

Borland C++Builder 是一个面向对象的、可视的编程环境, 是 Microsoft Windows 95 and Windows NT 支持下的快速应用程序开发软件(Rapid Application Development), 它支持客户程序与服务程序之间的动态数据交换。利用 C++Builder, 用户能够使用很少的编程代码高效地创建 Windows 应用程序。

C++Builder 提供了一个可重复使用的组件的完整库和一套 RAD 设计工具, 包括应

用程序和表格模板,以及编程 wizards。当用户启动 C++Builder,便立即被置于一个可视的编程环境。在这个环境下,C++Builder 为用户提供了所需的设计、开发、检测和调试程序的手段。

C++Builder 集中处理两种类型的对象:一种是称之为表格的窗口,另一种是在表格上绘制的各种控件。C++Builder 为这两种对象提供了标准的事件处理函数或过程,程序设计者可在此基础上编写实际的事件处理程序以响应用户的动作。C++Builder 以事件驱动方式执行应用程序,一个 C++Builder 应用程序一般都由一个消息等待循环、若干对象以及和对象相应的事件函数组成。消息等待循环用以接受由 Windows 发来的各种用户请求事件,如窗口内容改变、鼠标移动或按键等事件,Windows 将此事件或消息传递给相应的事件函数,并调用事件函数处理之。

1.2.2 C++Builder 的功能特点

C++Builder 具有强大的图形用户界面和程序开发接口,使程序设计者可以充分利用其图形环境和 Windows 系统资源。其主要功能如下:

(1) 创建 Windows 应用程序所需的各种对象,包括命令按钮、无线按钮、检查框、列表框、组合框、文本框、滚动条、图像框等对象;

(2) 实时响应鼠标和键盘事件,还可响应鼠标的拖曳与放下等复合事件;

(3) 可随时根据需要显示或隐藏各种控件、表格或窗口等;

(4) 可直接访问系统剪贴板,也可直接控制打印机水平;

(5) 可直接调用 Windows 的系统函数;

(6) 通过调用动态数据交换 DDE(Dynamic Date Exchange)和动态链接库 DLL(Dynamic Link Library)方法,可实现和其它 Windows 应用程序之间的相互通信。例如,利用 DLL,C++Builder 可以方便地调用 Windows 的系统函数。

此外,C++Builder 向程序设计者提供了一套功能强大的调试工具,使得程序设计更为容易。

1.2.3 C++Builder 是一种 GUI 式的 OOP 语言

Windows 以其直观形象的图形用户界面而倍受用户青睐,图形用户界面已成为软件界面的一种趋势和标准。但是,在 GUI(Graphic User Interface 图形用户界面)环境下的程序设计者必须考虑面临的一些新问题,即视窗式设计问题,事件函数代码编程问题以及怎样实现应用程序和用户的友好交互性问题,而这实际上是面向对象程序设计语言编程要实现的主要功能目标。C++Builder 很完善地实现了这三个功能。

(1) 视窗式 GUI 界面设计功能: C++Builder 可以建立两种方式的对象——表格和控制图。用户设计程序界面,实际上就是在表格窗上设计各种控制图,如命令按钮、文本框、图像框、列表框及各种滚动条等。而这一切工作如同我们在白纸上设计图案一样简单快捷,用户可以移动框的位置、改变框的大小和增删控制图等。C++Builder 是一个真正的具有 GUI 视窗界面设计功能的语言。

(2) 事件驱动及其编程: 对任一对象,C++Builder 都定义了一系列相关的事件,用户可对这些事件进行编程。但并不是对每一事件都必须进行编程,而只需编写那些有待作出

响应的事件函数或过程即可。为了方便编程，C++ Builder 为每个事件都定义了其首尾，格式为：

```
void __fastcall 表格名称::事件名称(参数)
{
}

```

我们来分析一下 C++ Builder 中对对象事件的响应过程。例如，在表格窗上有一个对象名为 Button1 的命令按钮，该对象具有 OnClick、OnKeyPress、OnMouseMove、OnStartDrag、…… 事件，其事件响应过程可用图 1.1 表示。

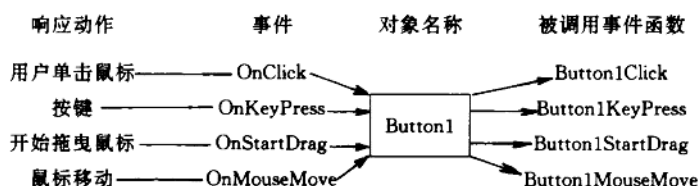


图 1.1 对象响应事件的过程

对象的事件函数可以实现各种各样的功能，如计算、输入输出、操纵界面等。然而对象本身的改变是通过修改其属性值实现的。

对象属性：属性是对象特征的描述。对每一种类型的对象，C++ Builder 均为其定义了一个相应的属性集。每一属性有属性名和属性值两个参数，用户可通过属性名引用或设置属性值。我们以状态条对象为例加以说明。设表格窗上有一个名为“StatusBar1”的状态条，如图 1.2 所示。其某些属性参数示意为：

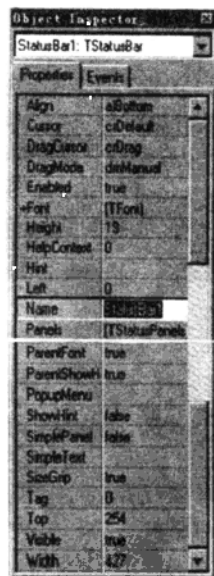


图 1.2 对象浏览器中的状态条属性

对象浏览器中左栏为属性栏，右列为属性值。其中，Height、Left、Top 和 Width 属性确定状态条在表格窗上的位置和大小，其值为像素点数。Name 属性确定状态条的名称。Font 属性确定状态条中显示字符的字体、样式、大小等。Align 属性确定状态条在表格窗中的排列。通过对这些属性的设置，用户可以使状态条以自己满意的面貌出现。

C++ Builder 中的对象除了反映其自身特征的属性和响应事件的函数之外，还定义了若干操作对象的方法。例如调色板对话框提供了 TColorDialog 方法、Execute 方法及 TColorDialog 方法，用于对调色板对话框进行控制。

C++ Builder 正是从对象的事件函数、对象属性及对象方法这三个方面完整地描述了一个对象，同时也利用这三种途径实现对象之间的联系和对对象的各种操作，从而实现

GUI 式的面向对象程序设计。

1.3 安装 C++Builder

1.3.1 系统配置

1. 最小系统配置

为了安装和运行 Borland C++Builder, 用户必须具有下述最小系统配置:

- (1) 16 MB RAM(推荐 24 MB);
- (2) 80486 processor 或以上;
- (3) Windows 95 or Windows NT (3.51 or 4.0);
- (4) 80~175 MB 可用硬盘空间, 取决于你所选择的安装方式。

2. 压缩安装所需配置

通过压缩安装可以对 C++Builder 进行最小安装, 压缩安装包括 C++Builder 以及 Borland Database Engine。这种安装不包括 Database 桌面、本地 InterBase 或 SQL 连接。压缩安装大约需要 80 MB 硬盘空间。在安装选择页面上选择“Compact”可以进行压缩安装。

3. 完全安装所需配置

C++Builder 的完全安装包括所有可用工具、Windows 集成 IDE, 以及允许 32 位 Windows 开发。在安装选择页面上选择“Full”可以进行完全安装, 大约需要 125 MB 硬盘空间。

4. 安装冲突

为了避免启动安装系统文件时的内部冲突, 我们建议用户在安装 Borland C++Builder 前, 关闭所有其它正在运行的应用程序。为了获得正在运行应用程序列表, 用户可按下面的步骤执行:

如果用户在 Windows95 下, 请查看屏幕底部的任务状态条, 或者同时按下 Ctrl+Alt+Delete 键, 打开关闭任务列表(单击“取消”退出列表), 关闭所有正在运行程序, 如图 1.3 所示。

如果用户在 Windows NT 4.0 下, 请查看屏幕底部的任务状态条, 或者同时按下 Ctrl+Alt+Delete 键, 选择“任务管理器”打开正在运行任务列表(选择文件菜单下的退出项退出任务列表)。

如果用户在 Windows NT 3.51 下, 则在背景上双击鼠标, 打开任务列表(单击取

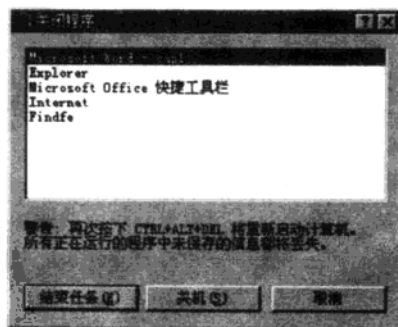


图 1.3 关闭任务列表

消退出列表)。

5. 安装 C++ Builder、Local Interbase 和 InstallShield Express

(1) 安装 Borland C++ Builder(所有软件包): 从光盘驱动器上安装, 运行\SETUP\CBUILDER\SETUP.EXE 文件。

(2) 安装 Local InterBase(客户/服务和专业包): 从光盘驱动器上安装, 运行\SETUP\LOCALIB\DISK1\SETUP.EXE 文件。

(3) 安装 InterBase Server(客户/服务软件包): 从光盘驱动器上安装, 运行\SETUP\SERVERIB\DISK1\SETUP.EXE 文件。

(4) 安装 InstallShield Express(客户/服务和专业软件包): 从光盘驱动器上安装, 运行\SETUP\ISX\SETUP.EXE 文件。

1.3.2 安装 C++ Builder

C++ Builder 的安装程序能够自动检测硬件设备, 并调节至最佳的设定状态。整个安装过程相当简单, 步骤如下:

(1) 将安装光盘放入光驱;

(2) 单击 Windows95 的“开始”按钮, 并单击弹出菜单中的“运行”, 显示运行对话框, 如图 1.4 所示;

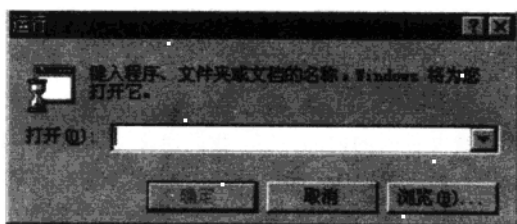


图 1.4 运行对话框

(3) 在“打开”文本框中输入“\SETUP\CBUILDER\SETUP.EXE”, 或者单击“浏览”按钮, 在浏览对话框中打开安装光盘上的 SETUP.EXE 文件, 此时运行对话框中的“确定”按钮可选;

(4) 单击“确定”按钮, Windows 开始安装 C++ Builder;

(5) 依照屏幕指示回答问题即可顺利完成安装。

值得一提的是, 在安装过程中, 安装程序会询问用户选择何种安装类型, 如图 1.5 所示。有三种类型可供选择:

(1) Full——完全安装, 推荐大多数用户使用;

(2) Compact——压缩安装, 对 C++ Builder 进行最小安装;

(3) Custom——用户安装, 用户自己选择安装选项, 推荐经验用户使用。

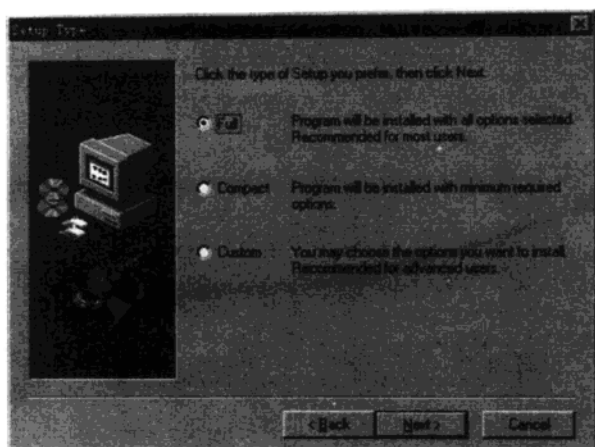


图 1.5 安装类型

1.4 C++ Builder 和 Borland C++ 的比较

与 Borland C++ 相比, C++ Builder 特点主要表现在以下几个方面:

(1) 友好的设计界面。C++ Builder 为用户提供了友好的用户界面。一个 C++ Builder 的初学者就可编制出生动的应用程序。而当一个 C++ 的初级用户在初次面对 Borland C++ 4.5 时往往会显得不知所措。图 1.6 是 Borland C++ 4.5 的设计界面, Borland C++ 将自己隐藏得太深了, 对用户来说其界面极不友好。尽管 Borland C++ 提供了 OWL

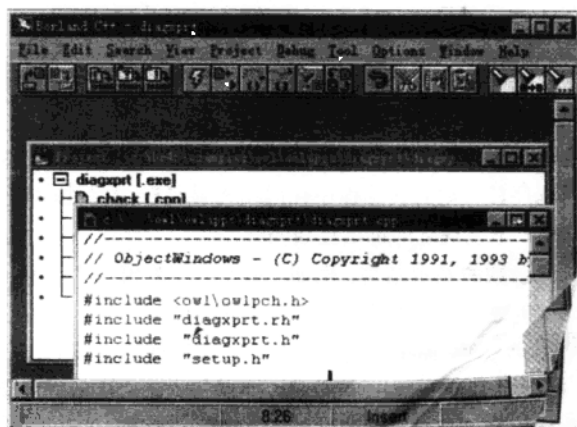


图 1.6 Borland C++ 4.5 的用户界面

技术,但 OWL 使用起来却十分不方便。例如,在 OWL 环境下的 Basic Option 选项中(如图 1.7),由于在开始阶段没有选择 Vertical scroll 等选项,如果在继续进行设计时发现必须使用垂直滚动条等功能项,那么用户就不得不在源程序中加入大量的代码,而使用 C++ Builder 在设计用户界面时就有很大灵活性,而且简单易行。

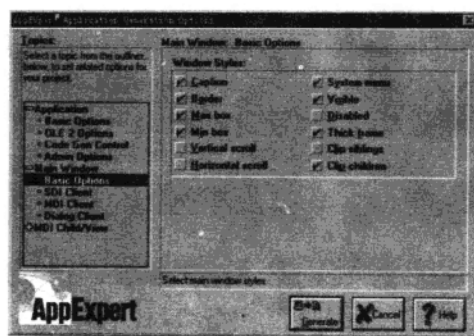


图 1.7 使用 OWL 进行编程

(2) 高效的可视化编程环境。C++ Builder 具有可视化的编程环境,为用户提供了可视的窗口设计(Visual Form Designer)、对象浏览器(Object Inspector)、组件模板(Component Palette)、工程文件管理器(Project Manager)以及相当完善的集成编辑器和调试器(Editor and Debugger)。

(3) 采用拖放式的程序设计方法。用户只需简单地从可视组件模板上选中一个组件,并将其放置在表格上,然后在对象浏览器中设置其属性即可使组件具有特定的功能。上述操作完成后 C++ Builder 会自动产生事件句柄,而且事件句柄可以被访问和编辑。

(4) 提供了属性、方法和事件。C++ Builder 为对象提供了属性(Properties)、方法(Methods)和事件(Events)这三个元素。属性可使用户轻松地设定标题、背景颜色等特性;方法,也称成员函数,它支持代码对用户的操作作出一定的反应;事件主要使应用程序响应一些特征状态的改变,如鼠标的移动,表格的缩放。

(5) 程序代码格式清晰。传统的 C 程序是过程式结构,灵活性较差;C++ 引入了面向对象设计技术,但当程序较大时,其源程序代码结构显得较为混乱;C++ Builder 的程序代码是模块式结构,每一个模块之间都用虚线隔开。

(6) 自动生成对象函数的基本结构。例如,用户在表格窗口上设置一个按钮,当用鼠标单击此按钮时,就会生成基本的响应事件函数结构。C++ Builder 的这种特性使用户不必太关心应用程序的东西,而把主要精力放在处理事件的内核上。

其实从 C++ 到 C++ Builder 是 C 语言的一个跨进,无论从用户界面的外部还是从整个编程环境的内核上讲它们都有很大的区别。上述的 C++ Builder 与 Borland C++ 的不同之处仅仅是从表象上,读者一个概念,要想真正地了解 C++ Builder,用户应细细体会本书的后续内容。

1.5 C++Builder 和 Visual Basic 的比较

1.5.1 程序设计基本界面的比较

C++Builder 基本界面包含：菜单条、工具条、组件模板、对象浏览器、表格窗、代码编辑窗口，如图 1.8 所示。而 Visual Basic 基本界面包含：菜单条、工具条、工具框窗口、表格窗、计划窗口，如图 1.9 所示。C++Builder 与 Visual Basic 基本界面的不同主要表现在以下方面：



图 1.8 C++Builder 程序设计的基本界面

(1) C++Builder 基本界面中菜单条上共有 11 个菜单项，而 Visual Basic 只有 8 个菜单项。C++Builder 增加了 Search、Project、Database、Tools 和 Component 菜单项。

(2) C++Builder 中的工具条位于菜单条左下方，是常用菜单功能的图标表示，其目的是为了简使用户操作。而 Visual Basic 界面中的工具条位于菜单条的下方。

(3) C++Builder 菜单条右下方的组件模板是 C++Builder 为用户提供的所有可用组件，它等同于 Visual Basic 界面中的工具框窗口，利用这些组件用户可以在表格窗口上设计各种控制图。与 Visual Basic 不同的是，C++Builder 将组件进行了归类，组件数目也增加很多。

(4) C++Builder 界面左下方为对象浏览器，用户可以直接对每个组件所代表的对象的属性进行设置；单击对象浏览器中的 Event，可以定义对象的驱动事件。而 Visual Basic 界面中没有对象属性窗口，用户必须按下 F4 键，属性设置窗口才会显示。

(5) C++Builder 的表格窗与编程窗与 Visual Basic 的表格窗与编程窗几乎相同，唯一

的区别是 Visual Basic 的事件函数在不同的编程窗口中, 而 C++ Builder 的事件函数都在一个编程窗口中。

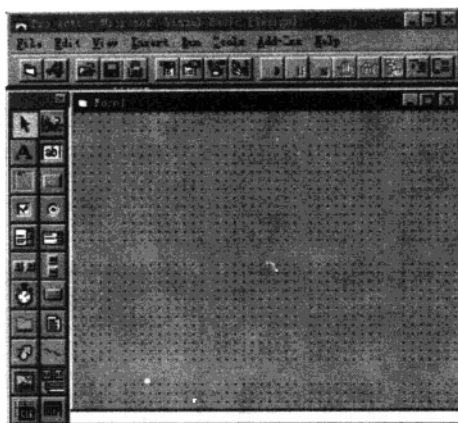


图 1.9 Visual Basic 程序设计界面

通过比较我们看出, C++ Builder 同 Visual Basic 相比, 其界面设计更加合理、直观, 具有更好的交互性。

1.5.2 表格窗上组件(控制图)的比较

1. C++ Builder 为用户提供了更多的组件

与 Visual Basic 相比, C++ Builder 提供了更多的组件, 并将这些组件进行了系统的分类, 这样做的目的是使用户能够高效、快捷地设计出更好的软件产品。例如 C++ Builder 的 Win95 组件模板, 从左到右分别为: Tab 控件(TabControl)、页控件(PageControl)、目录树(TreeView)、列表(ListView)、图像列表(ImageList)、头控件(HeaderControl)、文本(RichEdit)、状态条(StatusBar)、轨迹条(TrackBar)、进度条(ProgressBar)、增值减值(UpDown)、热键(HotKey)。而 Visual Basic 中没有这些组件(或控制图), 但用户可以自行设计 VB 控件, 扩展它的功能。

2. C++ Builder 改进了 VB 组件

C++ Builder 改进了 VB 组件, 例如它的 Standard 组件模板(如图 1.10 所示)中所列组件与 VB 工具框窗口中的控制图基本相同, 但文本框有一定程度的改进, 并且增加了无线钮组(RadioGroup)。VB 只有一个文本框控制图 TextBox, 而 C++ Builder 将文本框分为单行文本框(Edit)和多行文本框(Memo), 这样用户在创建文本框时就可以在二者之间进行选择, 而不必再去设置 Text 控制图的 MultiLines 属性。

C++ Builder 增加了无线钮组(RadioGroup), 使无线钮的分组变得非常简单。用户只需单击对象浏览器中的 Items 属性, 打开字符串编辑器, 输入所需按钮名称, 即可建立一