

Borland C++ Builder

编程技巧与实例

许冬岗 周涤非 聂文强 编著



西安电子科技大学出版社

Borland C++ Builder

编程技巧与实例

许冬岗 周涤非 聂文强 编著

西安电子科技大学出版社

1998

内 容 简 介

本书系统地介绍了 C++ Builder 开发 Win32 应用程序的基本思想及其编程技巧。C++ Builder 是 C++ 语言发展的一个划时代的里程碑,它结合了 Delphi 和 Borland C++ 的优点,将可视化开发环境、可视双向开发工具、可视组件库加入到 C++ 语言中,使开发人员不再需要在高效的底层控制与轻松的程序设计间做出艰难的选择,把开发者从繁杂的代码编制中解放出来,从而更多地关注于应用程序的功能。本书是为迫切想开发 Windows 应用程序的 C++ 程序员所写,它将通过大量实例展示 C++ Builder 作为一种独立的可视化开发工具所具有的优势。书中给出了非常丰富而实用的例子,它们由浅入深,覆盖了 Windows 编程的各个方面,并且易于修改和扩充。本书以书写器为例叙述了文件流的操作,以一个类似 Windows 中的画笔来说明图形的显示和打印输出,随后提供了三维真实感图形动画的例子。书中最后介绍 Windows 95/NT 特色编程,以多媒体播放机和定制组件作为本书的结束。

本书思想新颖、风趣易懂,可作为 Win 32 编程技术人员的自学教材,也可作为广大科技工作者、学生、程序员的参考资料。

Borland C++ Builder 编程技巧与实例

许冬岗 周涤非 聂文强 编著

责任编辑 戚文艳

西安电子科技大学出版社出版发行

地址:西安市太白南路 2 号 邮编:710071

空军电讯工程学院印刷厂印刷

各地新华书店经销

开本 787 × 1092 1/16 印张:16.75 字数:393 千字

1998 年 4 月第 1 版 1998 年 4 月第 1 次印刷 印数:1~6 000 册

ISBN 7 - 5606 - 0597 - 4/TP · 0300

定价:22.00 元

序 言

一、步入 32 位 Windows 的世界

Windows 漂亮的图形用户界面(GUI)、多方的多媒体服务、大量应用软件的支持、几乎无限制的内存资源、多任务的并发执行,促使用户、程序员迫不及待地跳上 Windows 这一 32 位的列车。是向 DOS 举行告别仪式的时候了,请追随 Win 95, Win NT 4, Win NT 5 的前进步伐吧!在发出这样的感叹后,程序员们发现:与 MS-DOS 相比,Windows 程序更加复杂,规模也更庞大,从而使得开发应用程序的难度增大了。

二、传统的编程方法面临困难

C 语言由于本身所具有的灵活、自由、简洁、高效等特点,而成为一种主导语言。C 语言可以直接调用 Windows API(应用程序接口),给我们带来不少方便。但是,由于 Windows 是基于消息和面向对象的机制,因此用 C 语言开发 Windows 应用程序难免凌乱、复杂,难于控制一大堆的 switch...case 语句(号称来自地狱的开关语句)。因而,采用面向对象的方法是必要的。从 C++, Delphi 的 Object Pascal, Visual Basic 可以看到这一趋势。

Borland C++, Visual C++ 等引入了可视化编程的概念。高度集成化的可视化工作平台极大地提高了程序开发效率;随着 OWL, MFC 类库不断的丰富,用它们来实现 Windows 新特性更加方便,编制出来的应用程序易于维护、移植、扩充。但 OWL, MFC 从本质上讲都是以控制方式为主的架构,它们始终摆脱不了繁重的代码编制工作,同时要求程序员要有较高的专业水平。从而使得许多程序员在潜心学习数月后,还是难于入手。于是 Visual Basic 大行其道, Delphi 更是在引入快速应用程序开发(RAD)后横扫天下。

三、为什么要用 C++ Builder?

选择开发工具的原则是,哪一个方便就用哪一个。对于 Borland C++ 和 Visual C++, 笔者并没有固定的好恶,时而用 Borland C++, 时而用 Visual C++, 因为它们实在是各有所长。身为一个 C++ 程序员,从 DOS 时代一直到 Windows 世界,笔者已经难于摆脱对 C++ 执着,光灵活简洁这一点就使 Fortran、Pascal 大失颜色,更不用说,它那几乎万能的指针,使得程序员能指哪打哪,像个神枪手。事实上,去掉了指针就不可能实现真正的面向对象,充其量只能是一种模拟而已。Windows 本身就是用 C 语言写的, Windows API 只可能支持 C 语言的直接调用。在这些方面, C++ 程序员可以说是占了先天和后天的优势。即使如此,刚着手 Windows 编程时,仍是困难重重,封装 API 虽花了不少力气,但效果却不是很好。有时,笔者会怀念 DOS(尽管这种念头只是一闪而过),但最终还是开始去尝试 OWL、MFC。它们确实做得不错,入手快,但烦恼接着也就来了。庞大的类库,控制方式为主的架构,熟悉它们实在不是轻而易举的事(这是一位 Delphi 程序员对 C++ 程序员进行攻击的原话),编程仍然是在大量的帮助和代码里漫游。

C++程序员在期待着，希望有像 Delphi 一样的高效、舒适、方便的开发环境。国际刊物 PC WEEK 曾说：“C++程序员羡慕 Delphi，但还是不得不用 Windows API 的母语——C 来开发程序，这痛苦深深地折磨着 C++程序员。C++ Builder 正是治愈这痛苦的良方。它提供了和 Delphi 一样的源程序与可视化工具的双向交互，比微软 Developer Studio 更易学习和控制的开发环境。” Borland 公司积十数年开发 Borland C++ 和 Delphi 的经验推出的 C++ Builder 是 C++ 语言开发的一个划时代里程碑。它结合了 Borland C++ 和 Delphi 的优点，将可视化开发环境、可视双向开发工具、可视组件库加入到 C++ 语言中，使开发人员不再需要在高效的底层控制与轻松的程序设计间做出艰难的选择，把开发者从繁杂的代码编制中解放出来，更多的关注于应用程序的功能。C++ Builder 是 C++ 可视化编程革命的一大进步，首次实现了高效开发 Windows 应用程序而不用考虑 Windows 编程的机制。C++ Builder 不是可视化工具套件，而是独立的开发环境，可直接生成高效的可执行代码，它支持 ANSI C++，可编译 Borland C++，还可编译 Delphi 的代码。

四、本书适合您吗？

如果您是 Borland C++ 或 Visual C++ 的用户，使用 C++ Builder 会使您的开发效率倍增。如果您有过使用 Visual Basic 或 Delphi 的经验，那么，转到 C++ Builder 再容易不过了。C++ Builder 也可以说是 C++ 和 Delphi 的结合。唯一的要求是读者必须具有 C++ 编程的经验，否则本书对您就不太合适！读者不一定要具有 Windows 编程技术，但为了理解本书的最后一部分，读者最好能在看书的过程中逐渐熟悉这一技术。

五、书中的内容

我们在很短的时间里就和 C++ Builder 结下了深厚的感情。在不久前，我们对诸如 Delphi，C++ Builder 一类的 RAD 开发工具也是一无所知。先是耳闻，然后目睹，一经试用便为其倾倒。在利用 C++ Builder 开发《集成电路可制造性仿真与优化设计系统》，《集成电路多目标优化设计系统》，《集成电路真实缺陷的分形特征仿真系统》等应用程序时，我们发现其开发效率比以往提高了几乎一倍，并且非常容易地将我们以前利用 Borland C++ 及 Visual C++ 开发的基于 DOS 和 Windows 的大量代码，毫无障碍、非常平滑地移植到了 C++ Builder 下。

本书是为迫切想开发 Windows 应用程序的 C++ 程序员所写，它将通过大量实例展示 C++ Builder 作为一种独立的可视化开发工具所具有的优势。书中的例子非常实用和丰富，由浅入深地覆盖 Windows 编程的各个方面。每一个例子都易于修改和扩充。通过对例子的学习，读者会发现用 C++ Builder 编程就像搭积木一样容易。书中的每一个例子都是 Windows 编程在该领域的典型应用，颇具代表性，同时给出了用 C++ Builder 开发此类程序的各项技巧和方法。为了使每一个例子尽量简单、易于理解，我们去掉了所有不必要的修饰和控件，尽管对 C++ Builder 来说，要实现这些花俏的修饰是再容易不过的事了。

第 1 章“C++ Builder 简介”，着重介绍了 C++ Builder 的集成开发环境，以及常用工具的设置与使用。

第 2 章“常用控件和窗体的使用”，介绍 C++ Builder 编程的基本方法，详细说明了控件，窗体，工程文件的组成，窗体与控件，多个窗体之间的关系。例子包含：最简单的起步

程序、内存申请并测试、自绘式按钮、动态软件封面。

第 3 章“文件流的操作”，介绍了关于文件流的各种操作和 Win 32 下的文件系统。通过例子“文件锁”介绍了与文件操作有关的各种控件，并说明了文件锁定的方法。“分页式记事本”实现了一个程序员记事用的分页式编辑器，涉及了 Windows 里的各种公共对话框，并讨论了文本的打印输出。最后一个例子是“多文本格式的 MDI 应用程序”，在涉及文件流操作里，这可能是所有程序员所能遇到的最复杂的应用程序。在大多数介绍编程的书籍里，要么避而不谈，要么把它非常浅显的一笔带过。但本书非常清晰地把它介绍给读者，只要翻一翻书，读者便会发现利用 C++ Builder 来开发此类应用是多么的容易和简洁。

第 4 章是“二维、三维图形显示技术”，事实上，这一章比较全面地介绍了图形、图像的各项内容。它将向您阐述：Windows 图形设备接口、图像显示方法、C++ Builder 中的各种绘图操作、图形的打印输出、C++ Builder 与 OpenGL 的接口、三维真实感动画技术。这一章的内容相当丰富，提供的例子也多。它们是图像浏览器、个人画笔及三个 OpenGL 的例子，从简单的插值图形显示到三维的真实感动画。

第 5 章是“Windows 32 特色编程”，它是 Microsoft 的 Windows 家族，从 16 位的 Windows 3.0 一直发展到 Win 95、Win 98、Win NT 5.0，逐渐实现操作系统内核的融合。地道的基于新型操作系统的应用程序应当追踪操作系统的特性变化，从而与操作系统天衣无缝的集成在一起。这一章的主要内容包括：实现通用控制的控件、界面一致性、进程的创建与检测、多线程、任务条通知区应用。对于进程，多线程，任务条通知区分别举了一个例子来说明。

第 6 章是“C++ Builder 的多媒体创意”，它是本书最愉快和轻松的部分，主要介绍了多媒体的基本概念，多媒体应用程序开发的原则和方法，C++ Builder 多媒体控件的属性、方法和事件以及利用 MediaPlayer 控件开发音频、视频播放程序的技巧与实例。这些实用化例子包括：波表音频播放、CD 唱片播放、VCD 视频播放以及 AVI 视频播放等。

第 7 章是“定制组件”，它是 C++ Builder 的最高层应用，也是本书最复杂的部分，要求读者熟知原有 VCL 类库结构、精通面向对象程序设计。我们以简单明了的方式叙述了 C++ Builder 中最复杂的应用。

本书另配磁盘一张，内含书中所有实例的源程序。每一个实例所在的目录对应一个 Readme.txt 文件。Readme.txt 文件详细地介绍了每个实例的可视化开发步骤，结合磁盘上所提供的源代码，读者可以很快地掌握本书的精髓，并据此自行开发优秀的 Windows 应用程序。磁盘已制成 Windows 下通用的安装盘格式，欲购者请与西安电子科技大学出版社或作者联系。

本书由许冬岗编写第 2、3、4、5、7 章，聂文强编写第 1 章，周涤非编写第 6 章。由许冬岗统编全书。由于作者水平有限，时间仓促，书中错误在所难免，恳请读者批评指正。

不要再犹豫，迅速加入 C++ Builder 的队伍！

作者

1998 年 1 月于西安

目 录

第 1 章 C++ Builder 简介	1
1.1 面向对象的可视化开发	2
1.2 集成开发环境(IDE)介绍	3
1.3 对象观察器的使用	5
1.4 模板对象的使用	6
1.5 C++ Builder 的文件管理	10
1.6 菜单编辑器	12
1.7 设定用户的工具栏	14
1.8 设定用户的组件板	15
1.9 程序调试器设置	17
 第 2 章 常用控件和窗体的使用	 19
2.1 静态文本控件 TLabel	19
2.2 标准按钮控件 TButton	20
2.3 简单的起步——Hello.C++ Builder	21
2.4 工程中文件的组成	22
2.5 单行编辑框控件 TEdit	25
2.6 状态条控件 TStatusBar	26
2.7 丰富的内存资源	26
2.8 位图按钮控件 TBitBtn	32
2.9 图像控件 TImage	33
2.10 菜单 TMainMenu	33
2.11 操纵控件的方法	34
2.12 多个窗体的结合使用	37
2.13 动态软件封面	39
 第 3 章 文件流的操作.....	 42
3.1 驱动器组合框控件 TDriveComboBox	42
3.2 目录列表框控件 TDirectoryListBox	43
3.3 文件列表框控件 TFileListBox	43
3.4 列表框控件 TListBox	44
3.5 文件打开与锁定	45
3.6 快捷按钮控件 TSpeedButton	51
3.7 标签控件 TTabSet	51
3.8 公共对话框控件	52
3.8.1 文件打开对话框控件 TOpenDialog	52

3.8.2 打印对话框控件 TPrintDialog	53
3.9 编辑框控件 TMemo	54
3.10 分页式记事本	55
3.11 文本打印输出	63
3.12 多文本格式编辑控件 TRichEdit	66
3.13 MDI 多文档界面应用程序	66
3.13.1 多文档接口	66
3.13.2 支持多文本格式的 MDI 应用程序	67
第4章 二维、三维图形显示技术	85
4.1 Windows 图形设备接口	85
4.2 检查框控件 TCheckBox	86
4.3 两个窗体结合显示图像	87
4.3.1 位图、元文件和图标	87
4.3.2 TPicture 类	88
4.3.3 图像显示	90
4.4 滚动框 TScrollBar	96
4.5 具有打印功能的画笔	97
4.5.1 TCanvas 类	97
4.5.2 画笔	97
4.5.3 图形打印输出	114
4.6 使用 OpenGL	115
4.6.1 OpenGL 简介	115
4.6.2 Win 32 与 OpenGL 的接口	116
4.6.3 Win 32 下 OpenGL 的程序设计技巧	116
4.6.4 在 C++ Builder 中使用 OpenGL	118
4.7 时钟 TTimer	129
4.8 帧缓存和动画	129
4.9 三维动画实例	130
4.10 特定消息的映射	141
4.11 光照效果的三维真实感动画	143
第5章 Windows 32 特色编程	165
5.1 一致的界面	165
5.2 创建进程并检测	168
5.2.1 Win 32 的进程概念	168
5.2.2 创建进程的方法	169
5.2.3 C++ Builder 中创建进程	171
5.3 画图框控件 TPaintBox	175
5.4 如何从 TThreads 派生类	176
5.5 多线程的使用	177
5.6 同步对象	191
5.7 任务条通知区应用程序	192

第 6 章 C++ Builder 的多媒体创意	201
6.1 多媒体的概念	202
6.2 多媒体应用程序的开发	203
6.2.1 多媒体应用程序的分类	203
6.2.2 多媒体编程的一般原则	203
6.3 媒体播放机控件 TMediaPlayer	204
6.3.1 TMediaPlayer 控件的主要属性	205
6.3.2 TMediaPlayer 控件的方法(Method)	207
6.3.3 媒体控件 TMediaPlayer 的一些有用事件	207
6.4 一个简单的媒体播放器——动听的 WAV	208
6.5 最简单的视频播放器——优美的 AVI	212
6.6 五脏俱全的 CD 播放器	213
6.7 完美的多媒体播放器	222
 第 7 章 定制组件	 236
7.1 C++ 程序员的注意事项	237
7.2 定制组件	237
7.2.1 定制组件的起点	238
7.2.2 定制组件的步骤	238
7.2.3 创建属性	240
7.2.4 创建事件	244
7.2.5 创建方法	248
7.2.6 注册组件	248
7.2.7 制作帮助	248
7.3 组件定制实例	249
7.3.1 饼图控件的定制过程	249
7.3.2 使用饼图控件	255

第 1 章

C++Builder 简介

你 可能有过开发 Windows 应用程序的经历, 虽然能利用 Windows 提供的工具设计出漂亮的用户界面, 但也为此付出额外的精力。Visual Basic 的出现, 使得开发 Windows 应用程序变得容易, 但随之就会发现: 由于 Basic 语言自身的缺陷, 使得用 Visual Basic 编程时会觉得有些东西用 Visual Basic 很难实现。Delphi 在各个方面都有很好的表现, 可对 C/C++ 编程人员来说, 要在 Delphi 上面进行开发, 首先得花精力先学一些 Pascal 语言方面的知识。

Borland 公司最新发布的 C++ Builder 是建立在 C/C++ 语言之上, 不管是可视化开发, 还是实现复杂的功能等, 都与 Delphi 同样地杰出甚至更好。

C++ Builder 的集成开发环境 (IDE) 提供了可视化的窗体设计、对象观察器、组件板、工程文件管理器、完全集成的代码编辑器和程序调试器, 使用户能实现真正的高速应用程序开发和对用户代码、资源的彻底管理; 对象组件的拖放设计, 使开发应用程序只需简单地从可视化的组件板里选择出所需的组件, 把它放在窗体里面, 并通过对象观察器设定其属性, 而对应于组件所需的事件句柄已由系统自动生成, 用户所要做的只是在对应地方插入自己的代码。

C++ Builder 提供的双向交互程序设计工具 (Two - Way - Tools), 通过可视的窗体设计和代码编辑器, 使用户能对应用程序代码直接进行控制。在可视的窗体里设计和在代码编辑器里进行开发是完全一样的。如果用户在可视窗体设计环境进行开发, 则系统在代码编辑器里就帮你生成对应的代码; 反之也是一样的。用户可以灵活地在二者间选择。

C++ Builder 采用完全的面向对象程序设计风格, 任何组件都是一个对象。系统提供的标准对象模板使用户能通过对象观察器改变其属性来满足自己的需要, 通过事件句柄来产生和响应 Windows 的各种消息。所以应用程序满足健壮性要求, 利于维护和扩充, 完全支持 Windows95 和 Windows NT。

C++ Builder 还提供了所有可视组件对象的完整代码，用户可以通过它作为例子，迅速地掌握开发自己组件的方法，也可以在此基础上实现代码的重用。

C++ Builder 拥有 C++ 的编译技术，因此，用 C++ Builder 编译生成的执行文件 EXE 或是 DLL，具有 C/C++ 语言编译生成的执行代码的效率，执行代码具有高效性和多平台的兼容性。C++ Builder 还支持多种语言编写的代码，如 Borland C++、Visual C++、Delphi、Power Builder、Visual Basic、Unix C++ 等。对以往的此类资源，只需用 C++ Builder 重新编译，仍然可以利用，并还可使它具有新语言的特征。

C++ Builder 提供的 SQL 工具，使用户能轻松自如地开发客户/服务器应用程序，数据库程序等。

C++ Builder 所需的系统配置：

- Intel 486 微机或以上；
- 操作系统为 Windows95 或 Windows NT3.51, 4.0；
- 16 MB 的内存（推荐用 24 MB）；
- VGA 显示器或更高；
- 鼠标；
- 所需硬盘：

标准版：	75 MB
专业版：	100 MB
客户/服务器版：	130 MB

1.1 面向对象的可视化开发

面向对象的概念贯穿 C++ Builder 编程的始终。C++ Builder 把一切视为对象，编写面向对象的程序意味着能够重复使用你的工作成果并使程序维护起来更为容易，程序扩充起来也更方便，原因在于各对象间是相互独立的，而对象间的联系仅通过特定的消息来取得。

继承性意味着可以在已有对象类型的基础上创建新的对象，这个新对象将继承其父辈对象的所有属性和方法，并且在衍生对象添加新的行为和特性时非常地灵活。

利用 C++ 编程的程序员可能早就体会到面向对象编程的优点，但实际做起来还是有很多困难，程序员往往在诸如对象的认定、类的设计、类的封装等方面感到头痛。然而在 C++ Builder 中，这些问题变得非常容易。类的设计与封装系统自身已帮你完成，你只需在适当的地方，写上你想要做什么的代码就行了。C++ Builder 在面向对象的基础上实现了真正的可视化编程，而且做起来非常容易。

1.2 集成开发环境(IDE)介绍

C++ Builder 是一个运行在 Windows 下的“可视化”开发环境,可以用来创建支持客户/服务器和 Windows95、Windows NT 的应用程序。C++ Builder 提供了非常丰富的可视化组件的标准对象,能非常容易地创建程序中的可视部分。如果没有这些工具,程序中的可视部分只能全部以代码的形式编写出来,如果你编写过 Windows 应用程序,就会知道一个 Windows 程序的可视部分从设计到它的最后成功有多么地不容易。较之于传统的编程方式,在 C++ Builder 中做这些事的过程更类似于桌面的设计。

进入 C++ Builder 集成开发环境后,在常规的启动状态下,C++ Builder 的界面有五个组成部分,如图 1-1 所示。

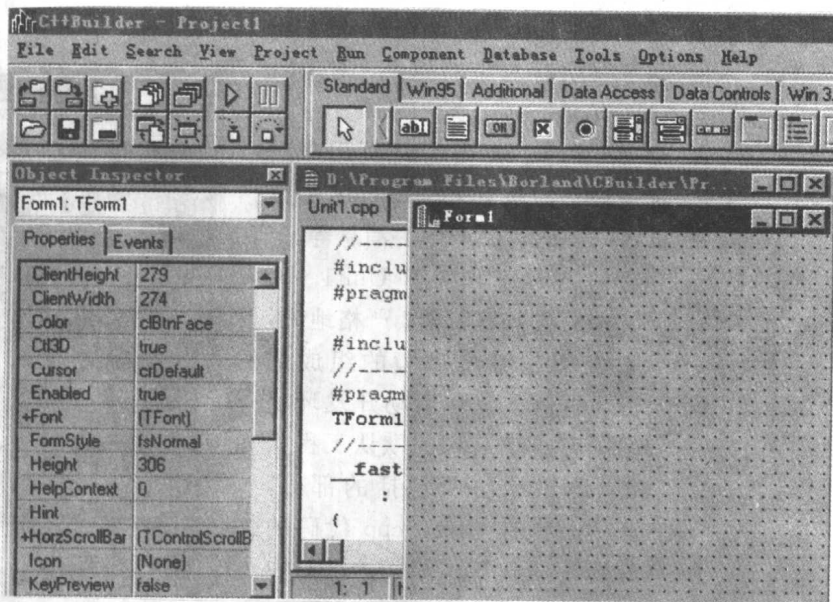


图 1-1 C++ Builder 集成开发环境

- Main Menu: 主菜单部分(见图 1-2),位于窗口的最上面,它提供了 C++ Builder 的全部操作功能。

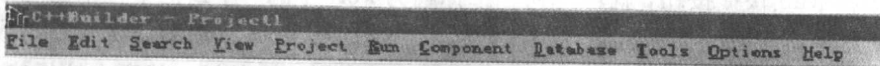


图 1-2 主菜单

- Toolbar: 工具栏部分(见图 1-3),位于窗口的左上侧,其上的按钮代表的是某些菜单命令的快捷形式。显然,使用快捷形式比使用菜单更方便。

- Component palette: 组件板部分(见图 1-4), 它是 C++ Builder 的可视组件库 (VCL), 是进行程序设计的主要工作区, 总共有 11 页标记页, 分别是 Standard、Win95、Additional、Data Access、Data Controls、Win3.1、Internet、Dialogs、System、QReport、Activex。

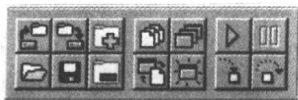


图 1-3 工具栏

其中, 不同类别的组件对象在不同的标记页上。

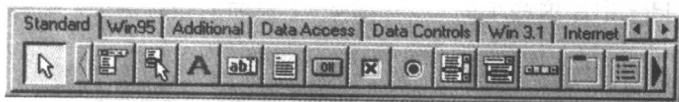


图 1-4 组件板

- Object Inspector: 对象观察器(见图 1-5), 用它可考察所有对象的属性。对象观察器有 2 页标记页, Properties 和 Events。Properties 标记页用来设定组件对象的属性, Events 标记页用来发送对象产生的消息等。

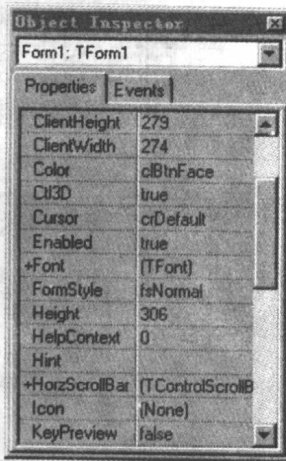


图 1-5 对象观察器

- Unit1.cpp、Form1: 窗体 Form1 和装有 Unit1.cpp 文件的代码编辑器; 严格地说, 它们不是编辑环境的组成部分, 而是系统在启动开发环境的同时自动装入的默认工程文件 (Project1) 所包括的部分。其中, 装有 Unit1.cpp 代码的编辑框叫代码编辑器, 是用户用来设计代码的地方。Form1

为窗体, 是 C++ Builder 的可视编辑环境部分, 也是用户用来设计应用程序的桌面部分。

在集成开发环境里, 系统已经装入了默认工程文件, 这个工程文件就是生成如图 1-6 所示的窗体, 标题为 Form1。在主菜单的 Run 下拉菜单里, 选择 Run 或是使用热键 F9, 或在工具栏里选择 Run 快捷按钮, 就可得到一个标题为 Form1 的、什么也不包括的 Windows 窗口, 这个窗口具有 Windows 窗口的所有属性, 如能改变大小, 移动等。我们一句代码也没有写, 就得到了一个。如果用传统的代码编写来得到同样窗口的话, 就不会这样容易了。实际上, 要用 C++ Builder 做一个很复杂的界面, 也不用费多少力。

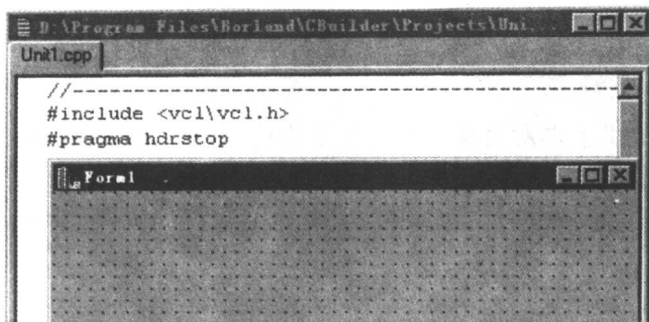


图 1-6 代码编辑器

1.3 对象观察器的使用

对象观察器 Object Inspector 可以在程序设计阶段对任何组件对象的属性进行设定和观察,同时,对象观察器还在 Events 标记页里产生选定对象响应的消息和发送消息的句柄。对象在设定时的界面是什么样,最后应用程序得到的也是什么样。

对象观察器由两个标记页组成,属性页(Properties)和事件页(Events),属性页用来设定选中对象的属性,如字体、颜色、显示标题等。事件页需要用户来设定,在发生由事件页上标注的事件时,该对象便根据相应的事件句柄调用用户的代码。

启动 C++ Builder,用鼠标点中对象观察器的属性页(Properties),对象观察器里包含的就是 Form1 窗体的各项属性。对象观察器的标题栏下面一行显示为 Form1:TForm1,它表示下面的内容为窗体 Form1 的属性。如果改变属性页里的内容,例如,把 Caption 栏的 Form1 改变为 Hello,则会发现窗体的标题 Form1 已变成 Hello;同样,也可以改变其它的属性。用户不用执行代码,就能知道运行结果是什么。

在 Form1 中加入其它的组件对象也很简单。比如加入按钮对象,可在组件板中选择 Standard 页中的 Button 对象(用鼠标点中图 1-7 中有 OK 字样的按钮,并确定它已凹陷),把鼠标移到窗体 Form1 中,按下鼠标左键,在 Form1 中就出现了一个标准按钮,这时在对象观察器的标题栏下面显示的是 Button1:TButton,其所列的内容就为刚加入的对象按钮的属性。至于如何改变其属性和怎样发送、响应消息,后面的例子将会讨论。读者也可以自己使用帮助试着去体会这一过程。

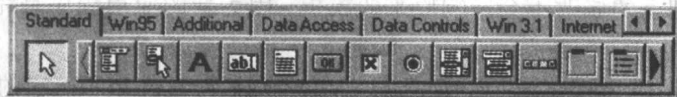


图 1-7 选择一个 OK 按钮

1.4 模板对象的使用

C++Builder 提供了丰富的模板对象供用户使用,而且还支持用户自己定义的模板。在 File 菜单里着重介绍 New 和 New Application 项。其中 New Form 是往应用程序添加一个新窗体, New Unit 是添加一个对应 Form 窗体的代码编辑框。其它菜单项都是我们常见的。

在主菜单的 File 菜单里选择 New 菜单项,就得到如图 1-8 所示的对话框,此对话框有个专有的名字,叫对象陈列框(Object Repository)。

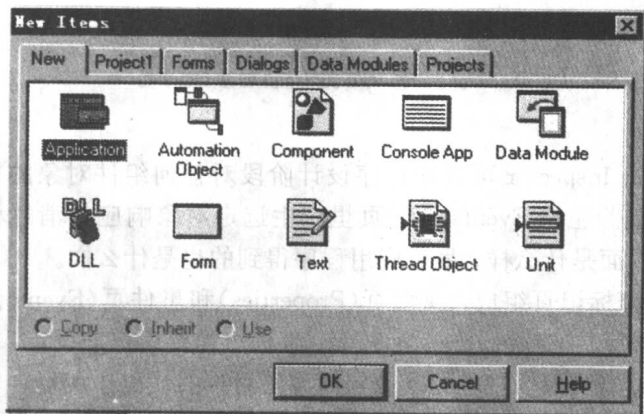


图 1-8 对象陈列框

对象陈列框也分为几个标记页。要说明的是,标记页的页数是可以改变的,用户可以增加或是删减标记页,也可以增加或是删减模板对象、增加自己的应用程序模板等。

用户可以在对象陈列框里选择适合自己的模板对象,在此基础上再进行应用程序开发。可以用继承的方法,也可以用拷贝的方法。当然,用户也可以不用对象模板,或仅仅是参考里面的一些对象。只是一切都从头开始,不方便而已。

例如,选择 Projects 标记页里面的 MDI Application 应用程序模板,确定靠近下部的 Copy 项被选中,然后选择 OK 键,这时,会得到如图 1-9 所示的对话框,该对话框让用户选择刚拷贝的模板应放在何处,选择好驱动器和路径名之后,再选择 OK 键,就开始了一个在模板基础上的应用程序。用户可以在模板的基础上增减内容,可以接着开发,这显然方便了用户,使用户不用每次都从头做起。

用户在模板基础上的修改不会影响原来的模板,因为新的应用程序是完全独立于对象陈列框模板的。

MDI Application 应用程序模板对象是一个完整的工程文件模板,它的界面部分如图 1-10 所示,该模板已把一个常用的应用程序的界面部分设计好了。用户可增加其它的功能,但是功能的实现部分还需用户来完成。用户在窗体里选中一个对象,如图中的 OpenDialog 按钮,这时 Object Inspector 自动选择在 OpenBtn: TSpeedButton 项,在 Properties 标记

页里用户可以设定这个快捷按钮的属性。

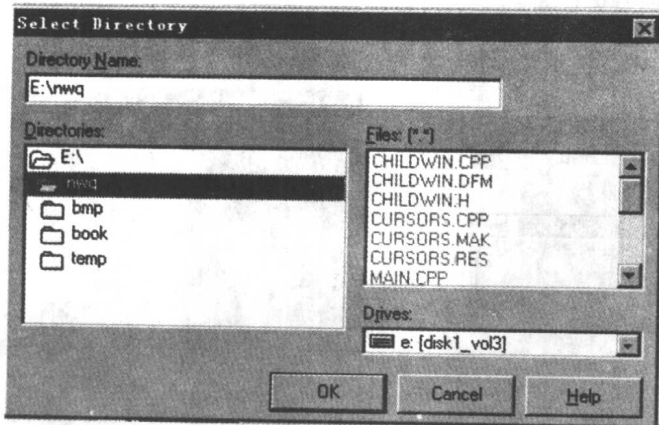


图 1-9 选择路径对话框

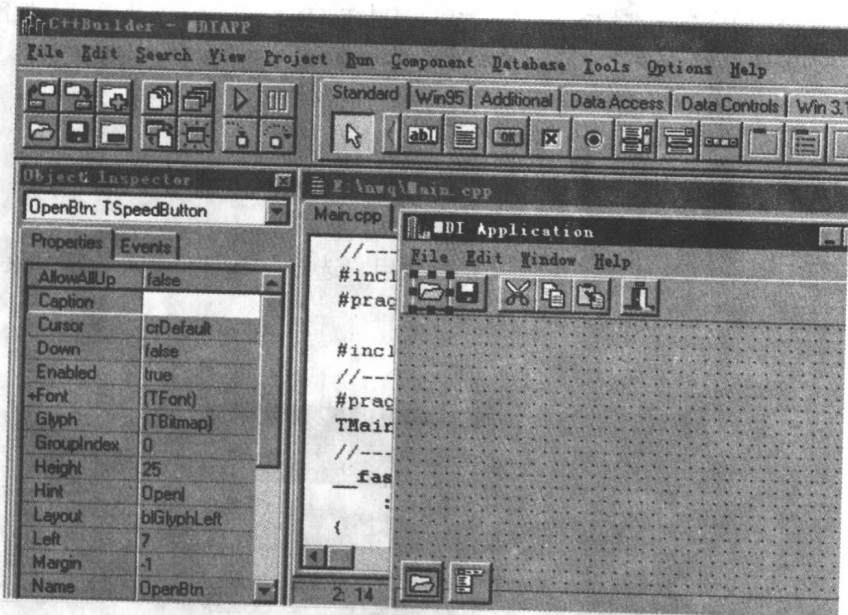


图 1-10 MDI 应用程序界面

在 Events 标记页里，可以设定该按钮的事件句柄。如图 1-11 中显示的是响应该按钮按下事件的事件句柄，事件句柄的代码要在代码编辑器里进行设计。用鼠标双击 OnClick 项里的 OpenItemClick，系统自动地转到代码编辑器的对应的地方，用户就可以插入自己的代码。

很多情况下，用户不得不开发一系列类型相同，差别很小的应用程序，如果每次都从头开始的话，就会增加程序员负担，这是没有必要的。但是，若不从头开始，对象陈列框里又没有用户需要的对象模板。为此，C++ Builder 提供了让用户增加自己对象模板的功

能, 使用户开发应用程序更加方便。

如果用户想加入自己定义的对象模板, 例如, 要想把具有如图 1-12 所示窗体的工程文件对象模板加入到对象陈列框里, 可按下面步骤进行(当然得首先设计出这个窗体):

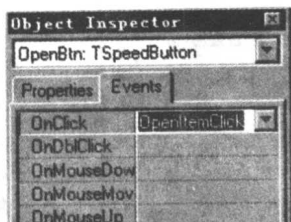


图 1-11 设定按钮的事件句柄

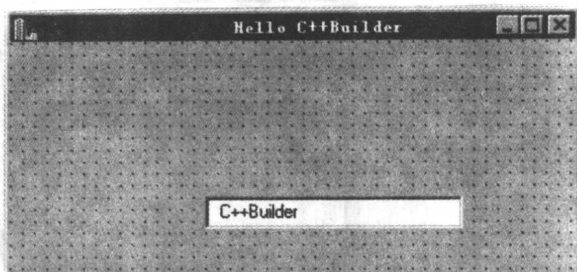


图 1-12 对象模板

(1) 在 File 菜单里选择 Save All 把这个工程文件保存, 重新取一个名字, 如 CForm (也可以用原来的默认名字, 但最好另取名字)。窗体名字任意取为 NewForm。这些操作也可以用 Toolbar 里的快捷按钮来完成。文件和窗体的扩展名系统会自动加上。

(2) 在主菜单的 Project 下拉菜单里, 选择 Add To Repository... 菜单项, 把刚保存的窗体对象加入到对象陈列框里。这时会得到如图 1-13 所示的对话框。其中:

- Title 指要装入的对象的名字, 如取为 CForm;
- Description 对该对象的描述, 用户可以自己定义;
- Page 指该对象应加入到对象陈列框的哪一个标记页里;
- Author 指创建该对象的作者。

Select an icon to represent this project 是要求用户选择一个图标来表示该对象, 用户点击 Browse... 按钮, 装入适当的图标即可。

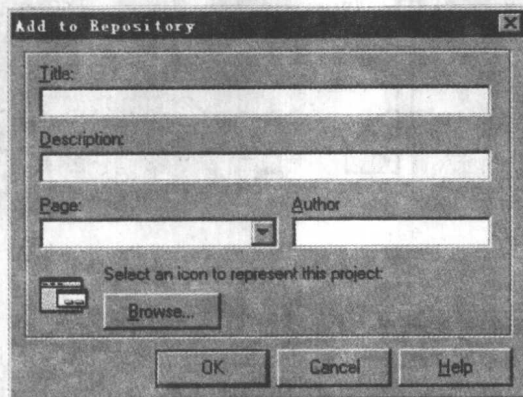


图 1-13 Add To Repository 对话框

(3) 上述整个操作完成后, 选取 OK 键, 就在对象陈列框里加入了用户新定义的对象模板。