

万水计算机实用编程技术系列

Borland C++ Builder 实用编程技术

史惠康 主编

赵俊才 编写

中国水利水电出版社

内 容 提 要

本书从 C++ Builder 的基本概念起,由浅入深地讲述了如何使用 C++ Builder 进行应用程序开发。由最基本的控件使用,到开发复杂的数据库应用程序,循序渐进。全书使用通俗的语言,结合与章节内容相关的丰富的实例,辅以详细的注释,较为全面细致地介绍了使用 C++ Builder 开发 Windows 应用程序的方法。

无论是初学者还是具有一定开发经验的程序员都可以通过本书的学习,顺利地进入到一个以 C++ 语言为基础的,采用可视化技术的 Windows 应用程序开发天地中。

图书在版编目(CIP)数据

Borland C++ Builder 实用编程技术/史惠康主编. —北京:中国水利水电出版社, 1998.8

(万水计算机实用编程技术系列)

ISBN 7-80124-777-9

I. B… II. 史… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(98)第 20848 号

书 名	Borland C++ Builder 实用编程技术
作 者	史惠康 主编
出版、发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: www. waterpub. com. cn E-mail: sale@waterpub. com. cn 电话: (010)63202266(总机)、68331835(发行部)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京天竺颖华印刷厂印刷
规 格	787×1092 毫米 16 开本 34 印张 782 千字
版 次	1999 年 1 月北京第一版 1999 年 1 月北京第一次印刷
印 数	0001—5000 册
定 价	52.00 元

凡购买我社图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·侵权必究

前 言

提起 Turbo C、BorlandC++3.1，许多人会感到无比亲切。这两款由 Borland 公司推出的基于 C 语言的集成开发工具曾伴随了一代程序员的成长。虽然在 Borland C++3.1 中已经可以支持 Windows 程序设计，但是留给人们深刻印象的还是这两个集成开发环境在开发 DOS 应用程序中所表现出来的简洁高效。Turbo C、Borland C++3.1 以其在集成调试、代码优化以及执行效率等方面的出色表现，当之无愧地成为 PC 机中开发 DOS 应用程序的首选编译器。

当基于图形用户界面的 Windows 操作系统推出后，应用程序开发也逐渐向这方面转移，Borland 公司相继推出了 Borland C++3.1 的升级版 Borland C++4.0，Borland C++4.5 等，虽然每一版都是精心制作，但是并没有赢得用户的普遍认可。Borland 公司推出的以“对象窗口库(OWL)”为基础的程序设计方法因为不能适应人们开发 Windows 应用程序的要求而失去越来越多的用户，多数程序员面对如此繁琐晦涩的概念望而止步。

用户希望的是容易掌握、便于使用的平易近人的开发工具。

1995 年，Borland 公司推出了一款全新概念的开发工具——Delphi1.0，Delphi1.0 以“可视化组件库(VCL)”为基础，提供给用户一个可视化的开发环境。Borland 公司提出了基于组件的属性、方法和事件等概念上的程序设计方法，用户只需要将可视化组件加入到窗体中，就可以设计出在 Windows 环境下的简单的应用程序，在组件的事件需要函数中加入必要的控制代码，就可以让应用程序工作起来。

程序员根本无须去理会 Windows 窗口的创建、注册以及令人眼花缭乱的消息控制，就可以编写 Windows 应用程序，深为广大程序员喜爱。

Delphi 直接生成可执行代码，只要复制一个应用程序，就可以随时执行，而不象 VB 那样需要庞杂的支持库——稍有不慎，漏拷一个文件，就可以导致程序无法运行。

由于 Delphi 直接编译生成可执行代码，执行效率是非常高的，这一点是以解释执行的 VB 所无法相比的。

稍后推出的 Delphi2.0 是运行在 Windows95 和 WindowsNT 上的 32 位应用程序开发工具，Borland 公司又加入了许多更新的组件。

Delphi1.0、2.0 和新近推出的 Delphi3.0 都是以 Object Pascal 为开发语言，这是一种面向对象的强数据类型开发语言，编写出来的程序结构严谨。

对于大多数习惯了使用 C 语言开发应用程序的程序员来说，放弃得心应手的 C 语言实在难以接受。

1997 年 Borland 公司推出了以 C 语言为开发语言的可视化开发工具——C++ Builder，在完全继承了 Delphi 的优点的基础上，提供了对 C 语言的支持。用户可以使用 C++ Builder 提供的组件库组织窗口界面，用自己所熟悉的 C 语言编写控制代码，使用起来简洁方便。

可视化开发可以大大缩短选项周期，即使是一个从来没有开发 Windows 应用程序经验的人，也可以使用 C++ Builder 很快地编写出 Windows 程序，因为 C++ Builder 向屏蔽了繁杂的 Windows 底层控制操作，向用户提供直观友好的开发界面。

由于应用程序中结构化的控制语句全部是由 C++ Builder 自动生成的，用户只要使用鼠标拖点操作进行界面设计，在 C++ Builder 生成的函数中写入控制代码，比较起传统的 C++ 程序开发，使用 C++ Builder 可以节省大量的开发时间，简洁的代码量也为程序的维护带来很大的方便。

C++ Builder 的易用性决不是牺牲性能为代价的，使用 C++ Builder 同样可以开发功能复杂的应用程序，这是以 C++ Builder 所提供的如下优点为保证的：

- 高速的编译、链接技术：C++ Builder 的编译和链接速度比其他 C/C++ 开发工具快许多。
- 高速的运行效率：C++ Builder 的代码经过严格优化，充分保证程序运行效率。
- 双编译引擎：可以同时编译 C/C++ 代码和 Delphi 的 Object Pascal 程序代码。
- 对 C++ 的支持：支持 Ansi C++，可以编译 Visual C++ 的代码。
- 强大的数据库支持能力。
- 对中文的支持：采用双字节内核。

本书面向广大渴望进入 Windows 程序设计世界的程序员，以浅显的语言，详细的说明，并结合实例介绍如何使用 C++ Builder 进行程序设计。

本书第一章介绍了 C++ Builder 的特点和一些基本概念，第二章详细介绍了 C++ Builder 集成开发环境，第三章介绍了使用 C++ 进行程序设计的基本知识，从第四章起由浅入深介绍如何使用 C++ Builder 进行程序设计，每一章都有详细的说明和与组件密切相关的示例，帮助程序员尽快掌握组件的特性和使用方法。在第十九章介绍了如何使用 C++ Builder 集成调试环境进行应用程序调试。

所有章节遵照由浅入深的原则逐渐展开，结合示例重点介绍了常用组件的使用方法。本书的目的在于引导读者逐步熟悉 C++ Builder，掌握使用 C++ Builder 进行程序设计的方法。

C++ Builder 本身就具有很高的易用性，相信每一个有信心的程序员都能够通过本书顺利进入 C++ Builder 的程序设计世界，去感受 Windows 应用程序的无尽魅力。

本书由史惠康策划编写，参加本书编写的有赵俊才、王士中、郭美山、石利文、伊俊文、魏山豹、李虹、张江华、程少杰、蔡未华、上官杰、赵福祥、文仲玉、朱元开、宋子荣、郭菁、张治东、王晓红、张江冰、暂东海、赵建荣、钱吉化、弓占星、刘浩、司马士、周幸存、万利静、师磊、艾花、左清伟、高创旷、魏鑫，赵俊才对全书做了统稿工作，赵俊萍负责部分章节的录入工作。

由于经验有限，时间仓促，书中错漏难免，恳请读者批评指正。

目 录

前言

第一章 C++ Builder 概述	1
1.1 C++ Builder 特点	2
1.2 C++ Builder 基本概念	4
1.3 C++ Builder 文件类型	8
1.4 小结	11
第二章 集成环境使用说明	12
2.1 主窗体	12
2.2 Object Inspector 窗	16
2.3 窗体设计窗	21
2.4 文本编辑窗	23
2.5 使用联机帮助	32
2.6 C++ Builder 组件说明	35
2.7 工具栏设置	44
2.8 小结	46
第三章 C++ Builder 程序设计基础	47
3.1 基本语法规则	47
3.2 数据类型	48
3.3 表达式	53
3.4 基本控制流程	55
3.5 函数	58
3.6 类与对象	59
3.7 程序设计风格	63
3.8 小结	64
第四章 创建应用程序	66
4.1 新建应用程序	66
4.2 新建窗体 (Form)	70
4.3 向窗体中加入控件	79
4.4 编写事件响应函数	85
4.5 保存文件	88
4.6 编译并运行程序	89
4.7 使用模板	90
4.8 窗体控制	95

4.9 一个实例.....	100
4.10 应用程序属性设置.....	105
4.11 小结.....	106
第五章 文本输出与编辑.....	107
5.1 Lable 对象.....	107
5.2 Edit 对象.....	113
5.3 MaskEdit 对象.....	119
5.4 Memo 对象.....	123
5.5 RichEdit 对象.....	129
5.6 小结.....	133
第六章 使用按键.....	134
6.1 常规按键: Button.....	134
6.2 位图按键: BitBtn.....	142
6.3 快捷键: SpeedButton.....	145
6.4 按键使用示例.....	148
6.5 小结.....	153
第七章 设计菜单.....	155
7.1 MainMenu 菜单.....	155
7.2 PopupMenu 菜单.....	159
7.3 小结.....	162
第八章 选项设定: 使用复选框、单选键.....	163
8.1 复选框: CheckBox.....	163
8.2 单选键: RadioButton.....	165
8.3 小结.....	166
第九章 成组对象.....	167
9.1 成组框: GroupBox.....	167
9.2 单选键组: RadioGroup.....	168
9.3 面板: Panel.....	177
9.4 对象位置固定化.....	180
9.5 小结.....	188
第十章 使用列表.....	189
10.1 列表框: ListBox.....	189
10.2 组合框: ComboBox.....	198
10.3 小结.....	201
第十一章 多页组件.....	202
11.1 Notebook 对象.....	202
11.2 TabSet 对象.....	208
11.3 TabbedNotebook 对象.....	215

11.4	TabControl 对象	217
11.5	PageControl 对象	219
11.6	小结	224
第十二章	滚动对象	225
12.1	滚动条对象: ScrollBar	225
12.2	上下键对象: UpDown	230
12.3	TrackBar 对象	233
12.4	滚动框对象: ScrollBox	237
12.5	进度条对象: ProgressBar	243
12.6	小结	246
第十三章	鼠标控制	247
13.1	改变鼠标形状	247
13.2	响应鼠标按键事件	255
13.3	响应鼠标移动事件	258
13.4	实例: 制作对象分隔控件	266
13.5	小结	270
第十四章	键盘控制	271
14.1	响应字符按键事件	275
14.2	响应任意按键事件	280
14.3	响应按键松开事件	287
14.4	控制焦点转移	290
14.5	小结	296
第十五章	图形与图像处理	298
15.1	Canvas 对象	298
15.2	Shape 控件	318
15.3	Image 对象	321
15.4	PaintBox 对象	328
15.5	小结	329
第十六章	视图对象	330
16.1	使用 OutLine 控件	331
16.2	使用 TreeView 控件	336
16.3	ListView 控件	341
16.4	ImageList 对象	347
16.5	一个实例	348
16.6	小结	356
第十七章	拖放控制	357
17.1	设置拖动方式	357
17.2	进入拖动状态	359

17.3	拖动对象.....	361
17.4	结束拖动状态.....	367
17.5	小结.....	378
第十八章	应用程序对象.....	380
18.1	应用程序属性读取与设置.....	381
18.2	使用联机帮助.....	383
18.3	输出信息.....	386
18.4	显示提示信息.....	398
18.5	处理消息.....	406
18.6	最小化应用程序.....	414
18.7	运行和终止应用程序.....	415
18.8	小结.....	431
第十九章	联机调试.....	432
19.1	编译源代码，加入调试信息.....	432
19.2	区别错误，分类调试.....	436
19.3	设定程序运行方式.....	437
19.4	使用断点.....	439
19.5	监视内存变量.....	442
19.6	使用 Call Stack 窗口观察函数调用情况.....	444
19.7	生成可发布代码.....	445
19.8	小结.....	446
第二十章	使用对话框.....	447
20.1	打开文件对话框： OpenFileDialog.....	447
20.2	保存文件对话框： SaveDialog.....	451
20.3	字体选择对话框： FontDialog.....	452
20.4	颜色设置对话框： ColorDialog.....	454
20.5	打印对话框： PrintDialog.....	456
20.6	打印机设置对话框： PrinterSetupDialog.....	458
20.7	查找对话框： FindDialog.....	459
20.8	替换对话框： ReplaceDialog.....	461
20.9	小结.....	463
第二十一章	系统对象.....	464
21.1	定时器： Timer.....	464
21.2	文件列表框： FileListBox.....	467
21.3	目录列表框： DirectoryListBox.....	469
21.4	文件过滤组合框： FilterComboBox.....	471
21.5	驱动器组合框： DriveCombBox.....	473
21.6	媒体播放器： MediaPlayer.....	475

21.7 打印机: Printer.....	480
21.8 小结.....	487
第二十二章 数据库应用程序设计	488
22.1 BDE 基础.....	488
22.2 使用 BDE Configuration	489
22.3 使用 Database Desktop.....	495
22.4 使用 Data Access 控件	501
22.5 使用 Data Control 控件	522
22.6 小结.....	533
附录: 常用对象的属性、方法及事件.....	535

第一章 C++ Builder 概述

Borland 公司是一个著名的软件开发公司，该公司推出的多款编译器，是程序员深为喜爱的开发工具。

早先发布的 C 语言编译器 TurboC、Borland C++3.1 以其高速、高效、优秀的代码优化技术和便利的集成调试功能深深地赢得了程序员的喜爱，很长的一段时期内，Borland 公司的这两款编译器都是在 PC 机上使用 C 语言开发 DOS 应用程序的首选工具。

随着 Windows 系统的普及，Borland 以 Borland C++3.1 为起点开始支持 Windows 应用程序开发。在支持标准的 C++ 开发的基础上，Borland 推出了带有自己公司特点的基于“对象窗口库(OWL)”的面向对象的 Windows 应用程序开发技术，尽管 Borland 对 OWL 技术倾注了无数心血，Borland C++ 一再升级，但始终未收到良好的市场回报。

不可否认，OWL 的确存在着技术上的优秀性，但是，与 DOS 编程相比，开发 Windows 应用程序本来就要繁琐得多，再去使用 OWL 这种建立在要求对基本的 Windows 程序开发比较熟悉的基础上的开发技术，使很多程序员望而却步。

在很长的一段时间内，始终没有一款能为广大程序员普遍接受的 Windows 应用程序开发工具。对 Windows 应用程序开发工具的基本要求是：易掌握，易使用。直到 Microsoft 推出 Visual Basic(即 VB)才使这种情形发生改观。

在 VB 中，使用了可视化开发技术，将晦涩的 Windows 基本概念隐藏在可视化的组件(Component)之后，程序员只需要将系统提供的组件摆放在窗体(Form)上，然后使用鼠标单击组建，写出组件所对应的功能的实现代码，VB 会自动生成 Windows 应用程序，直观、方便、易用。

VB 发布后不久，在 1995 年，Borland 公司推出了自己的第一款基于可视化技术的 Windows 应用程序开发工具——Delphi1.0，在 Delphi 中，Borland 充分发挥了可视化技术的优点，提供了更加丰富的组件和更强大的数据库技术支持。

Delphi 仍旧保留了 Borland 在编译技术上的高速、高效的优点，而且直接生成可执行代码，无论编译速度还是执行速度都远远超过 VB，尤其是 Delphi 直接生成可执行代码，比起 VB 的解释执行方式，效率更要高出许多。

使用 Delphi 生成的应用程序可以独立运行，无需象 VB 那样需要众多的支持软件，所以深为程序员喜爱。

随后推出的基于 32 位应用程序开发的 Delphi2.0 在功能和组件的种类上都有了很大的丰富和完善，与其同期推出的 VB4 相比，依旧在效率和速度上占有无可异议的优势。包括 1997 年推出的 Delphi3.0 在内，Delphi 始终保持对 VB 的领先性。所以有人称 Delphi 为“VB Killer”也是不无根据的。

然而十分遗憾的是，无论 VB 还是 Delphi 都不支持 C 语言开发。尤其对于 Delphi 这样出色的开发工具来讲，不能不算一大缺憾。

1997 年, Borland 公司推出了支持 C++ 语言的可视化开发根据 C++ Builder, 在继承 Delphi 的优点的基础上, 提供了全面的对 C 语言的支持。

在 C++ Builder, Borland 不再使用 OWL 技术, 而是以被 Delphi 证明为性能出色的“可视化组件库”(VCL)为基础, 将 Windows 基本概念封装在组件类中, 使用 C++ Builder 提供的组件, 程序员可以使用自己得心应手的 C 语言开发应用程序。

开发的过程也是很简单的, 在窗体中加入组件, 为组件编写事件响应程序, 由 C++ Builder 生成 Windows 应用程序。C++ Builder 生成的也是可执行代码, 可以随时随地发布。

这里, 简单介绍 C++ Builder 的特点。

1.1 C++ Builder 特点

1.1.1 可视化的开发技术

可视化技术带来的方便在于:

1.1.1.1 大大缩短学习周期

对于一个有经验的 C/C++ 程序员, 从学习到掌握象 Borland C++ 和 Visual C++ 这样的工具, 通常要占用半年的时间, 才可以能比较熟练地使用。

而对 C++ Builder 的学习可以只需要几周的时间, 甚至可以在学习期间就开始进行开发工作。它无需要求苛刻的 Windows 基本知识, 而只要将设计思想通过 C++ Builder 提供的丰富的组件表达出来, 即可得到高效运行的应用程序。

1.1.1.2 缩短开发周期

C++ Builder 将 Windows 底层隐藏起来, 提供给用户的是简洁使用的界面, 用户只需构造自己的窗体, 编写与应用程序功能密切相关的控制代码, C++ Builder 会自动生成常规的 Windows 应用程序所必须的部分。大大减少了程序员的代码量。

对应用程序界面特别是窗口的设计, 可以在应用程序设计阶段同步进行, 而且可以随时修改, 比起使用系统工具生成界面再加入到应用程序中的做法要快捷得多。

1.1.1.3 提高程序的可维护性

代码量的大量减少, 尤其与应用程序并非密切相关的 Windows 底层知识的隐藏, 都使得使用 C++ Builder 开发的应用程序更加可读, 从而提高了可维护性。

另外, 由于 C++ Builder 以窗体为通常的分隔对象, 所以, 应用程序各组成部分的界面比较清晰, 便于差错检查或部分功能的改动处理。

1.1.2 高度的可重用性

传统的 C/C++ 开发工具是以应用程序的结构为中心, 而 C++ Builder 则是以组件为中心, 这样就使 C++ Builder 具有比传统开发工具更高的可重用性。

1.1.2.1 可重用的组件库

C++ Builder 提供的组件库是非常丰富的, 共有 100 多个组件。包含了构造 Windows95 应用程序界面的从标签(Label)到树形视图(TreeView)在内的全部组件; 还包含有两组用于

数据库开发的专用组件，可以通过这两组组件，轻松实现数据库程序设计，尤其是界面设计，使数据与界面尤其融为一体；为了满足更新技术的应用程序开发，C++ Builder 还提供了一组 Internet 组件和一组 ActiveX 组件，使程序员能够快速实现 Internet 应用程序和包含 ActiveX 技术的应用程序开发。

C++ Builder 提供了这些 VCL 组件的全部原始代码，程序员可以直接修改这些代码来重新构造符合自己需要的 VCL 组件。

C++ Builder 良好的界面使用户可以充分利用 C++ 技术轻松地继承 VCL 组件的特性生成新的组件。

1.1.2.2 可重用的对象存储库(Object Repository)

C++ Builder 用 Borland 数据引擎(BDE)的数据存储能力结合一个图形化的用户界面，向程序员提供一个可重用的重新设计框架。程序员可以将数据模块、设计规则、原始对象和窗体等存入对象存储库。

C++ Builder 通过对对象存储库的管理使窗体以至完整的应用程序都具备了可重用性。对应用程序新的开发可以通过继承、参考或者复制等方法直接应用到新的设计中。

1.1.3 双编译引擎

C++ Builder 内部带有两个编译引擎，使 C++ Builder 可以同时编译 C/C++ 语言和 Object Pascal 语言。

在 C++ Builder 内部使用最新的 Borland C++ 5.02 版的 C/C++ 语言编译器用来编译 C/C++ 语言代码，Borland C++ 5.02 编译器的良好性能使 C/C++ 代码得到很好的优化。

同时在 C++ Builder 还有一个可以编译 Object Pascal 代码的 Delphi 编译器，也就是说，C++ Builder 可以编译 Delphi 程序。

可以编译 Delphi 程序代码对推出不久 C++ Builder 来说是非常有用的，由于 Delphi 发布时间较长，已经普遍得到用户认可，更有许多程序员为 Delphi 设计了新的组件，据说在 Internet 网上有 1000 多个 Delphi 组件，C++ Builder 可以完全使用使用这些组件，大大丰富了 C++ Builder 的组件库。

C++ Builder 的 VCL 库也是用 Object Pascal 编写的，之所以这样，主要是利用 Delphi 的可靠的技术。但是，并不是说使用 Object Pascal 编写的应用程序的效率比 C/C++ 的效率要低很多，Borland 的代码优化技术已经使二者的差距越来越小。即使使用 C++ 重写 VCL 库，生成的代码速度也不会明显高出很多。

1.1.4 高性能的编译、链接工具

在编译 DOS 应用程序时，普通的 C/C++ 工具的编译和链接速度上的影响不很明显，但是在代码量巨大的 Windows 应用程序前，对编译器的速度要求是很高。Microsoft 似乎更依赖于硬件的性能，这一点在使用 Visual C++ 编程时尤为明显，只要机器性能略差，带来的将是对一个程序编译一遍的漫长等待。

Borland 公司为解决这个问题，花费了大量精力用于改进行优化 C/C++ 编译和链接上，使得 C++ Builder 的编译、链接速度明显高于其他 C 编译器。

1.1.5 对 C++ 语言的全面支持

C++ Builder 对 C 语言的支持是目前所有基于 C/C++ 的开发工具中最齐全的。

C++ Builder 支持 Ansi C++，可直接编译 C++ Builder、Visual C++ 的代码。

1.1.6 强大的数据库处理能力

C/C++ 语言在数据库方面的处理能力一直是不太理想的，当然这与 C 语言本身就不是数据库程序设计语言有关。

C++ Builder 提供了对数据库的支持，使用 C 语言和 C++ Builder 的组件，就可以编写出数据库处理程序。

在 C++ Builder 中提供了两组组件专门用于数据库控制，这些控件的功能是很强大的，直接使用这些控件，无需复杂的代码，就可以生成一个功能比较完善的数据库应用程序，结合 Borland 数据引擎 BDE/IDAPI 可以实现更强的功能。

C++ Builder 所提供的 Borland 数据引擎是一种非常成熟的数据库连接技术，提供了三种数据库访问技术：

- 直接存取 DBase、FoxBase、FoxPro 和 Paradox 等文件型数据库。
- 提供一个标准的 ODBC 接口，通过整个接口可以访问任何一种支持 ODBC 的数据库。
- 提供一个高效的 SQL Link 数据库驱动程序，允许直接存取 Oracle、Informix、Sybase、MS SQL Server、DB2 以及 Borland 的 InterBase。

1.1.7 对中文的支持

C++ Builder 采用双字节内核设计，程序员无需再担心在界面中出现“半个汉字”的尴尬局面。C++ Builder 的 VCL 组件也和中文完全兼容。

1.2 C++ Builder 基本概念

下面介绍几个与 C++ Builder 应用程序密切相关的概念。

1.2.1 窗体(Form)

窗体(Form)是用来承载组件的，当应用程序设计完毕，窗体在运行阶段就对应与 Windows 应用程序的一个窗口(window)。

在设计阶段，程序员可以为每一个窗口产生一个空白窗体，如图 1-1 所示，窗体的大小代表了应用程序中窗口的大小。

窗体中的网状点是用来方便程序员摆放组件而加入的，程序员可以以这些点为参考规划组件的位置。

这些点在生成的应用程序窗口中不会出现，如图 1-2 所示。

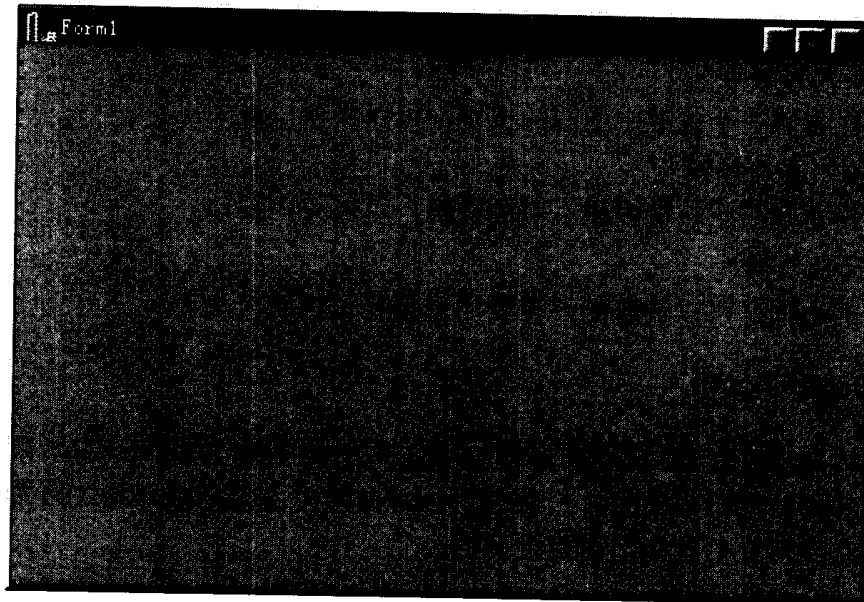


图 1-1 空白窗体

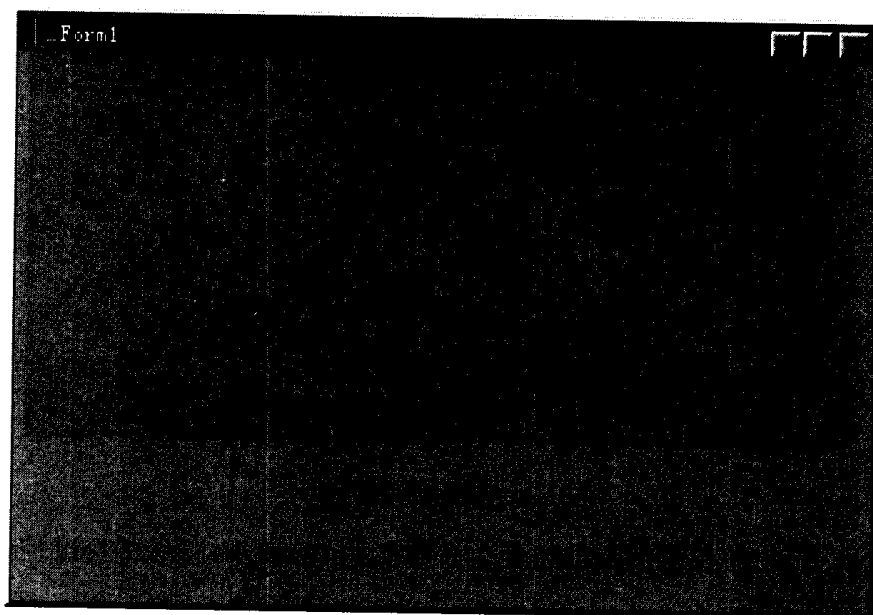


图 1-2 应用程序窗口

有几个与应用程序窗口相关的概念说明如下：

1.2.1.1 标题栏

标题栏位于窗口的正上方，用来显示窗口的标题，如果当前窗口是活动窗口，标题栏的背景色通常为蓝色；如果当前窗口是非活动窗口，标题栏的背景色通常为深灰色。当然，通过 Windows 桌面的颜色配置方案可以重新设置标题栏颜色。

1.2.1.2 窗口图标

窗口图标位于标题栏左侧，是符合 Windows 图标资源定义的标准图标。

1.2.1.3 最小化按键

窗口标题栏右侧三个按键中最左侧的按键，可以将窗口最小化。

1.2.1.4 最大化按键

窗口标题栏右侧三个按键中中间的按键，可以将窗口最大化。

1.2.1.5 关闭窗口按键

窗口标题栏右侧三个按键中最右侧的按键，可以关闭窗口。

1.2.1.6 窗口边框

窗口四周的边框是可以调整的，主要用来调整窗口的大小。

1.2.2 组件(Component)

组件是构成应用程序窗口界面的基本元素，C++ Builder 提供有 100 多种标准组件，有文本标签(Label)、编辑栏(Edit)、主菜单(MainMenu)、列表框(ListBox)等等，如图 1-3 所示，这些都是我们在后续章节重点学习的。

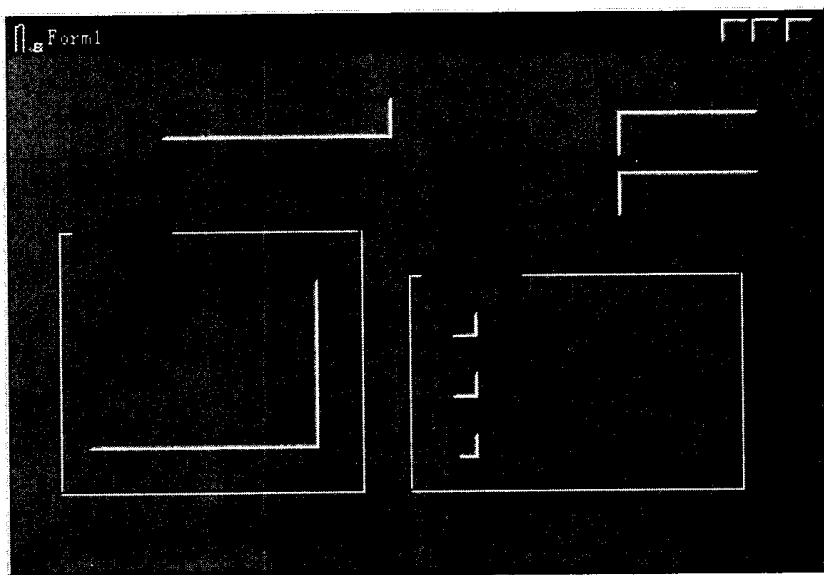


图 1-3 向窗体中加入组件

事实上 C++ Builder 提供的标准组件大多数都是 Windows 系统的标准控件(Control)，所以书中也以控件相称。

组件排列在组件板上，如图 1-4 所示，设计窗体界面时，可以使用鼠标直接点击相应组件，将其加入到窗体上。



图 1-4 C++ Builder 组件板

1.2.3 类(Class)

类是面向对象的应用程序设计的基本概念。类是对一组性质相同的实体的程序描述，它由概括了具有共同性质的数据和函数组成，在应用程序设计中，类被作为样板来建立对

象。

在 C++ Builder 中每一个组件都对应一个类，通常是由组件名称前加一个字母“T”构成。例如按钮 Button 对应的类是 TButton。

通过类来定义对象。

1.2.4 对象(Object)

同类一样，对象也是面向对象应用程序设计的基本概念。对象就是具有类所定义类型的变量。

当程序员把一个组件加入到窗体中，实际上就是在应用程序中定义了一个具有一个组件类所定义类型的对象，所以加入到窗体中的组件成为对象。

1.2.5 属性(Property)

属性是组件类的数据成员，主要用来记录和控制组件的特性与状态。属性可以分为三类：

- 常规属性：这一类属性主要控制组件的状态，读取属性值可以得到组件相关状态，改变属性值可以重新设置组件状态。
- 只读属性：这一类属性只是在应用程序运行阶段反映组件的相关状态，读取属性值可以得到组件状态，但是不允许改变属性值。
- 设计属性：这一类属性只是在应用程序设计阶段使用，用来设计界面。当程序运行后，就不允许再改变这些属性的设置。

1.2.6 方法(Method)

方法实际上是对象提供的公用成员函数，用来对对象本身进行相关控制或者改动对象内部私有数据。

在应用程序中可以调用组件提供的方法对组件进行较复杂的控制。

例如关闭窗口体可以调用窗体(Form)提供的方法 Close()：

```
Form1->Close();
```

1.2.7 事件(Event)

事件是指外界的输入引起的某种状态的变化，例如移动鼠标、最大化窗口等。

与传统应用程序的顺序执行不同的是，Windows 是以“事件驱动”的，当某事件发生时，转去执行相应的事件处理程序，关于事件何时发生都是不可预测的。

1.2.8 事件处理(Event Handler)

当某事件发生时，作为对此事件的响应，Windows 就转去执行相应的事件处理程序，通常是一个函数。

在 C++ Builder 中，事件的处理都是通过事件响应函数完成的。

应用程序的执行也正是在不断的处理事件中展开的。