

C++ Builder 实用技术与典型案例

徐 科
杨朝霖 李滨涛 等编著

清华大学出版社

(京) 新登字 158 号

内 容 简 介

本书介绍了 C++ Builder 6.0 开发应用程序的一些实用技术, 并通过几个典型案例解决读者的一些具体问题。全书内容分“实用技术”和“典型案例”两大部分。“实用技术”部分的内容包括 C++ Builder 6.0 基础、绘图与打印、多线程、定制组件、异常处理、C++ Builder 特殊使用技巧等; “典型案例”部分的内容包括用 C++ Builder 开发图像处理应用程序、网络应用程序、CORBA 应用程序及基于 SOAP 的 Web Service 应用等。

全书力求实用性与典型性, 可供初学者作为入门的教材, 同时中高级使用者也能在书中找到感兴趣、受启发的内容。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

书 名: C++ Builder 实用技术与典型案例
作 者: 徐 科 杨朝霖 李滨涛 等编著
出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)
<http://www.tup.tsinghua.edu.cn>

责任编辑: 朱英彪

印 刷 者: 北京鑫丰华彩印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 23.5 字数: 534 千字

版 次: 2002 年 11 月第 1 版 2002 年 11 月第 1 次印刷

书 号: ISBN 7-900643-75-3

印 数: 0001~5000

定 价: 35.00 元(附光盘)

前 言

C++ Builder 的出现是程序开发人员，特别是 C++ 程序员的一大幸事！

如果你还在为 Visual C++ 开发的应用程序单调的、近乎千篇一律的用户界面而痛苦，为 Visual Basic 程序的低执行效率而烦恼，为 Delphi 不能使用 C++ 语言而叹息，那么就请你不要再犹豫，一起来学习 C++ Builder 吧！C++ Builder 可以为你解决以上的问题，它是绝对值得信赖的。

本书的目的就是为已经使用或即将使用 C++ Builder 的人员提供一些实用技术，并通过一些典型案例来介绍如何通过 C++ Builder 构筑起功能强大的应用程序。由于 C++ Builder 的内容非常丰富，涉及到的应用领域十分广泛，使用方面的技巧更是不能穷尽，为了能帮助读者尽快熟悉 C++ Builder，并用它来开发所面临的一些应用程序，本书将保持以下两个方面的特色：

1. 力求实用性

本书是作者在多年使用 C++ Builder 编程的基础上写成的，每一项实用技术和每一个典型案例都是作者的经验总结。虽然本书的篇幅不大，但是读者在学完每一部分内容后都会在某些方面得到启发和提高。书中的一些实例更是初学者难得的“敲门砖”，很有针对性，虽不是为每个应用程序“量身订做”，但只要稍加改变，便可成为解决手中编程任务的“法宝”。

2. 突出典型性

本书介绍的 4 个典型案例——图像处理、网络编程、CORBA 应用程序和基于 SOAP 的 Web Service 应用都是读者比较感兴趣，但目前的 C++ Builder 方面的书籍又很少深入讨论的内容。本书将对这些案例进行深入分析，突出 C++ Builder 在解决这些案例中的优势。通过这些案例，不仅可以使读者掌握 C++ Builder 的使用，而且可以为读者在这些领域的编程打下很好的基础。

全书共分两大部分，第一部分介绍实用技术，共有 6 章；第二部分介绍典型案例，共有 4 章。各章的安排如下：

第 1 章介绍 C++ Builder 的一些基础知识，包括 IDE、工程、窗体设计、多文档应用程序设计，本章的前面部分是针对 C++ Builder 5.0 编写的，在后面部分中添加了 C++ Builder 6.0 的 IDE 以及一些新的特色。第 2 章介绍绘图与打印方面的一些使用，并通过一个综合实例来说明如何用 C++ Builder 进行图形设计。第 3 章介绍如何开发多线程应用程序。第 4 章介绍如何在 C++ Builder 下创建用户自定义组件。第 5 章介绍 C++ Builder 的异常处理机制，以及如何通过异常处理保证程序的可靠性。第 6 章介绍 C++ Builder 中的一些特殊使用技巧，如在 C++ Builder 中如何使用 Windows 消息处理机制及用户自定义消息，注册表的

使用技巧，Windows API 函数的使用，如何开发多国语言程序，以及如何进行应用程序的分发与包装等，这些内容是读者在开发应用程序过程中经常会遇到的。第 7 章介绍如何用 C++ Builder 进行图像处理应用程序的开发，虽然目前在图像处理领域用 Visual C++ 居多，但 C++ Builder 也有自己的特色，其功能一点不亚于 Visual C++。第 8 章介绍如何用 C++ Builder 进行网络编程，主要包括大容量数据的网络传输、Web 浏览器的开发、网上聊天程序的实现及邮件收发系统的开发等。第 9 章介绍如何用 C++ Builder 开发 CORBA 应用程序，CORBA 技术在分布式应用中占有重要的地位，用 C++ Builder 开发 CORBA 应用程序将会减轻程序员的很多负担。第 10 章介绍如何用 C++ Builder 开发基于 SOAP 的 Web Service 应用，这是 C++ Builder 6.0 的特色。

本书的第 1 章由徐科、阳建宏编写，第 2 章由徐科、宋强编写，第 3、4、5 章由李滨涛编写，第 6、8 章由杨朝霖编写，第 7 章由徐科、宋强编写，第 9 章由屈蓉编写，第 10 章由阳建宏编写。

本书所配光盘中给出了书中所有实例的代码和可执行文件。光盘中的代码都在 C++ Builder 6.0 下调试通过，为了让使用 C++ Builder 5.0 的读者也能使用实例，光盘中还给出了所有实例的 C++ Builder 5.0 版本。

由于时间紧迫，加上作者水平有限，书中不足之处在所难免，希望广大读者和同行批评指正。

作者

2002 年 7 月

目 录

第 1 篇 实用技术

第 1 章 C++ Builder 6.0 基础	3
1.1 C++ Builder 的 IDE	3
1.1.1 工具栏	3
1.1.2 组件及对象观察器	5
1.1.3 程序的设计状态 (Design Time) 和运行状态 (Run Time)	6
1.2 工程 (Project)	6
1.2.1 C++ Builder 工程中使用的文件	6
1.2.2 工程管理器 (Project Manager)	8
1.2.3 使用窗体模板	9
1.2.4 使用工程模板和应用程序向导 (Application Wizard)	11
1.3 窗体的设计	12
1.3.1 TForm 组件的属性	12
1.3.2 TForm 组件的方法	15
1.3.3 TForm 组件的事件	16
1.3.4 动态创建窗体对象	17
1.3.5 创建应用程序启动画面	21
1.4 多文档界面应用程序的设计	23
1.4.1 多文档界面的自动创建	23
1.4.2 MDI 应用程序的窗体类型	23
1.4.3 主菜单	24
1.4.4 工具栏	28
1.4.5 状态栏	30
1.4.6 TOpenDialog 组件	30
1.4.7 TImageList 组件	34
1.4.8 TActionList 组件	34
1.5 C++ Builder 6.0 IDE 的新特性	37
1.5.1 对象树形浏览器 (Object TreeView)	37
1.5.2 可装载的视图 (Loadable View)	40
1.5.3 对 IDE 环境的一些改善	41

1.5.4	窗体设计器 (Form Designer)	47
1.5.5	对象观察器 (Object Inspector)	50
第 2 章	绘图与打印	53
2.1	Windows GDI 简介	53
2.2	TColor 类	54
2.3	TCanvas 类	57
2.3.1	画刷 (Brush)	58
2.3.2	画笔 (Pen)	59
2.3.3	字体 (Font)	61
2.3.4	TCanvas 的其他属性	62
2.3.5	TCanvas 的方法	63
2.4	打印	64
2.4.1	TPrinter	64
2.4.2	TPrinterDialog 和 TPrinterSetupDialog	65
2.5	综合例程	67
2.5.1	实例功能	67
2.5.2	实现步骤	67
2.5.3	代码编辑	71
2.5.4	技术要点	85
2.5.5	程序演示	86
第 3 章	编写多线程程序	89
3.1	多线程程序的结构和优越性	89
3.2	定义线程对象	90
3.3	同步线程	96
3.4	执行线程对象	98
3.5	实例编写	99
第 4 章	定制组件	106
4.1	组件概述	106
4.2	创建用户自定义组件	107
4.3	实例分析	108
第 5 章	异常处理	121
5.1	异常处理的作用	121
5.2	异常的基本语法	122
5.3	VCL 异常处理	123

5.4 实例分析.....	124
第 6 章 C++ Builder 特殊使用技巧	126
6.1 消息使用技巧.....	126
6.1.1 Windows 系统消息	126
6.1.2 对系统消息的截获.....	127
6.1.3 实例编写.....	129
6.1.4 用户自定义消息.....	132
6.2 注册表的使用技巧.....	137
6.2.1 什么是注册表.....	137
6.2.2 如何操作注册表.....	138
6.2.3 TRegistry 使用实例.....	139
6.3 Windows API 函数的使用	146
6.3.1 如何使用 Windows API	146
6.3.2 实例程序的实现原理.....	146
6.3.3 捕捉程序的实现.....	148
6.3.4 工具程序的实现.....	161
6.4 多国语言程序的开发.....	167
6.5 应用程序的分发与包装.....	168
6.5.1 应用程序的打包.....	168
6.5.2 安装程序的制作.....	170
6.5.3 开发应用程序的帮助系统.....	175

第 2 篇 典型 案 例

第 7 章 C++ Builder 开发图像处理应用程序	187
7.1 Windows 位图基础	187
7.1.1 数字图像基本概念.....	187
7.1.2 TBitmap 类	188
7.1.3 TJPEGImage 类	189
7.1.4 调色板	190
7.1.5 TBitmap 深入分析.....	197
7.2 通过 TBitmap 实现图像的特殊显示.....	202
7.2.1 图像的旋转.....	202
7.2.2 动态显示.....	207
7.2.3 淡入淡出.....	215
7.3 图像处理.....	218
7.3.1 图像数据的存取方式.....	218

7.3.2 图像处理的基本方法.....	222
7.4 图像处理例程.....	224
第 8 章 C++ Builder 开发网络应用程序	257
8.1 一个实时的图像传输程序.....	257
8.1.1 常用网络组件介绍.....	257
8.1.2 程序实现思路.....	258
8.1.3 例程实现.....	261
8.2 建立一个 Web 浏览器	276
8.2.1 WWW 简介	276
8.2.2 程序设计思路.....	277
8.2.3 实例实现.....	277
8.3 开发一个网上的多用户聊天程序	283
8.3.1 Socket 介绍.....	283
8.3.2 TClientSocket 和 TServerSocket 介绍	284
8.3.3 实例编写.....	288
8.4 开发一个简单的邮件发送和接收系统	303
8.4.1 组件使用介绍.....	303
8.4.2 实例编写.....	308
第 9 章 C++ Builder 开发 CORBA 应用程序	321
9.1 CORBA 简介.....	321
9.2 实现 CORBA 的基本步骤	323
9.3 接口定义语言 (IDL)	324
9.4 CORBA 高级编程.....	332
9.4.1 内存管理.....	332
9.4.2 对象引用.....	336
9.4.3 命名服务.....	337
9.4.4 可移值对象适配器.....	338
9.5 例程实现.....	340
第 10 章 C++ Builder 6.0 开发基于 SOAP 的 Web Service 应用	346
10.1 XML 简介.....	346
10.2 XML 与 HTML、SGML.....	346
10.3 XML——第二代 Web 语言，下一代网络应用的基石	348
10.4 SOAP 与 Web Services 简介.....	350
10.4.1 为什么需要 SOAP 和 Web Services.....	350
10.4.2 什么是 SOAP	351

10.4.3 SOAP 与 CORBA、COM/DCOM 的区别	351
10.4.4 HTTP + XML = SOAP	353
10.4.5 未来的发展	354
10.5 用 C++ Builder 6.0 开发一个 Web Service 应用实例	354

第 1 篇 实用技术

本篇内容

- ◆ 第 1 章 C++ Builder 6.0 基础
- ◆ 第 2 章 绘图与打印
- ◆ 第 3 章 编写多线程程序
- ◆ 第 4 章 定制组件
- ◆ 第 5 章 异常处理
- ◆ 第 6 章 C++ Builder 特殊使用技巧

第 1 章 C++ Builder 6.0 基础

C++ Builder 已经成为当今世界上最流行的 Windows 编程工具之一，它的流行得益于它综合了 RAD (Rapid Application Development, 即快速应用程序开发) 的简单易用和 C++ 语言的强大功能。无论对于编程爱好者、专业编程人员、中小公司的软件开发小组，还是专业软件公司的大型开发队伍，C++ Builder 都是一种开发应用程序的合适工具。

本章将介绍 C++ Builder 的一些基础知识，包括 C++ Builder 的 IDE (集成开发环境)、VCL (可视化组件库)、工程 (Project)、窗体 (Form) 的设计等，并且还将介绍如何创建多文档界面应用程序，以及程序界面设计时经常用到的菜单、工具栏、状态栏的设计。当前 C++ Builder 的最新版本为 6.0，为了使读者熟悉和掌握 C++ Builder 6.0，本章从第 5 节开始介绍 C++ Builder 6.0 的 IDE 环境以及一些使用技巧。通过这些内容的介绍，读者将对 C++ Builder 有一个基本的认识，并且可以自己动手创建一些简单的应用程序。

1.1 C++ Builder 的 IDE

IDE 是集成开发环境 (Integrated Development Environment) 的英文缩写，它将程序开发时所需要用到的一些操作界面集成到一起。C++ Builder 5.0 的 IDE 如图 1.1 所示，包含了工具栏、主菜单、组件面板以及一些常用窗口，如对象观察器、窗体、代码编辑器、工程管理器、CPU 观察器等。

1.1.1 工具栏

工具栏提供了对 C++ Builder 中常用菜单项的快速访问。可以根据个人的爱好来定制工具栏，添加和删除工具栏中的一些功能。

定制工具栏的步骤如下：

第 1 步 在工具栏的任意位置右击。

第 2 步 在弹出的菜单中选择菜单项 “Customize” (定制)。

第 3 步 屏幕上将显示 “Customize” 对话框，如图 1.2 所示。

第 4 步 单击 Commands 标签。

现在就可以用拖放的方式把需要的命令添加到工具栏中了。如果要从工具栏中删除不想要的功能，只需把该按钮拖到工具栏外即可。

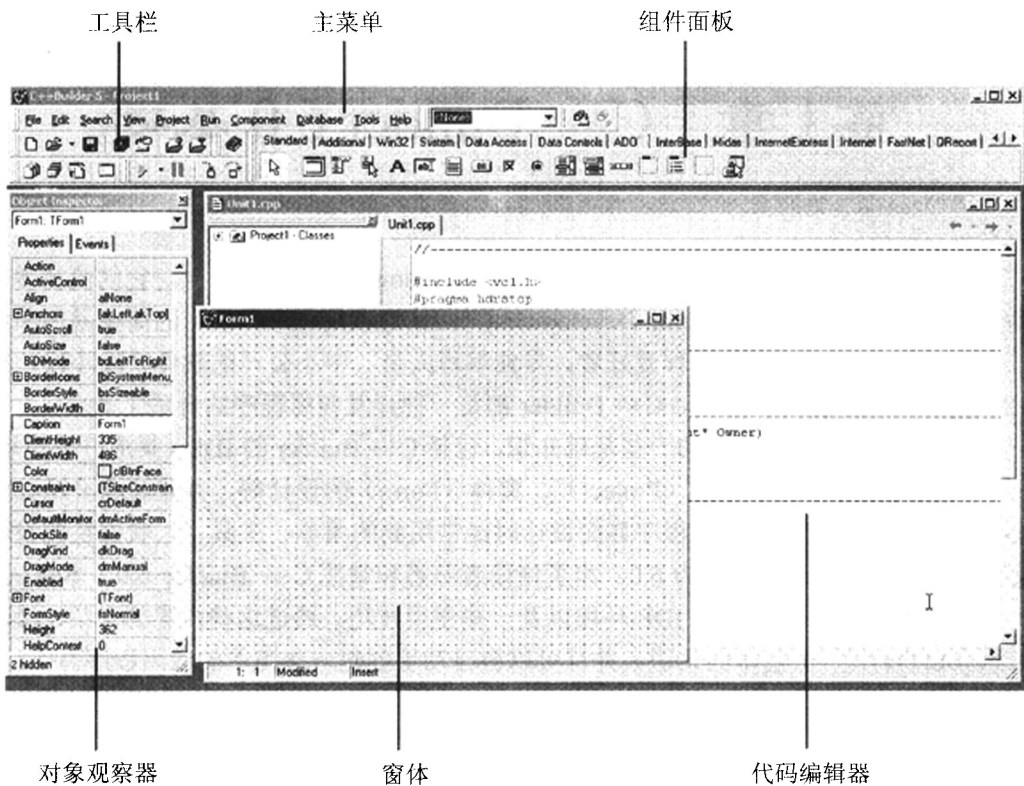


图 1.1 C++ Builder 5.0 的 IDE

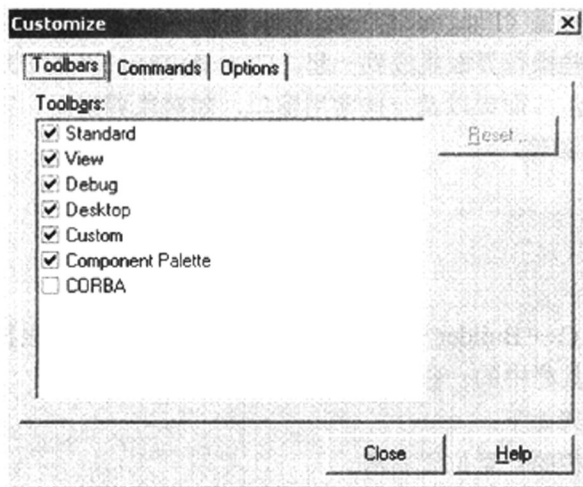


图 1.2 “Customize”对话框

1.1.2 组件及对象观察器

C++ Builder 的 RAD 特性之一是它提供了 VCL，并通过 VCL 来进行应用程序的开发。VCL 中包含了大量的组件，用户可以很轻松地从中选取任意一个需要的组件（如按钮），并把它拖放到程序设计过程中的工件区中。一个组件实质上就是一个对象，它具有属性、方法和事件三个特性。

组件的属性用来描述组件的特征，如组件的大小和位置、组件中所用的字体和颜色、组件是否可用（Enabled）等。组件的属性可以通过对象观察器来修改，也可以通过代码修改。按下快捷键 F11 或选择“View|Object Inspector”菜单命令都可以弹出对象观察器窗口。单击对象观察器窗口中的“Properties”标签，在弹出的选项卡中可以设置组件的属性。对象观察器中列出了组件的大部分（或者说常用）属性，但不是全部的属性。对象观察器中没有列出的属性可以在代码中修改。可以通过 C++ Builder 中的联机帮助来查看组件的所有属性。

组件的方法就是组件的成员函数。组件的方法能够访问所有公共的、保护的和私有的属性以及该组件的数据成员，也就是通常意义上的成员函数。

事件用来描述组件的动作，如鼠标的移动、字体的变化和某个通信端口有数据流输入等。不同的组件都有各自对应的事件，Windows 能够自动响应窗口中所发生的一切事件，但不一定所有的事件都有响应。如果程序需要对某个事件作出响应，那么需要在该事件的事件句柄中添加相应的处理代码。下面就以为组件 TButton 添加 OnClick 事件处理函数为例，来说明添加事件句柄的步骤：

第 1 步 选中一个 TButton 组件 Button1。

第 2 步 按下 F11 键，弹出对象观察器（也可选择“View|Object Inspector”菜单命令）。

第 3 步 单击对象观察器中的“Events”标签。

第 4 步 在弹出的选项卡中的 OnClick 项上双击（OnClick 是单击 Button1 时产生的事件），在代码编辑器中将会自动生成以下代码：

```
//-----  
void __fastcall TForm1::Button1Click(TObject *Sender)  
{  
  
}  
//-----
```

这一函数就是单击 Button1 事件的事件句柄，在大括号中可以添加对单击 Button1 事件的响应代码，如：

```
//-----
```

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    Application->Terminate();
}
//-----
```

以上的代码表示当单击 Button1 时，程序将会关闭。

1.1.3 程序的设计状态（Design Time）和运行状态（Run Time）

由于 C++ Builder 采用可视化方式开发应用程序，因此在 C++ Builder 中应用程序有两种状态，一种是设计状态，一种是运行状态。设计状态是指以可视化界面的方式对程序进行设计，主要是设置组件的属性，设计状态时所要完成的工作一般是应用程序的一些初始设置；运行状态是指在程序运行过程中完成的工作，这部分工作往往通过代码实现。设计状态和运行状态的概念在下面的内容中会经常出现。

1.2 工程（Project）

C++ Builder 通过工程来创建应用程序。工程是一些 C++ 源代码、窗体文件和其他类型文件的集合，它们共同定义了应用程序。C++ Builder 使用特殊的工程文件来存储工程的结构，并且提供了与工程有关的各种选项。

1.2.1 C++ Builder 工程中使用的文件

C++ Builder 工程由很多不同类型的文件组成，有些文件是 C++ Builder 自动生成的，有些文件是程序员创建的，还有些文件是在编译过程中生成的。为了介绍 BCB 工程中使用的文件，首先按下面的步骤创建一个新的工程：

第 1 步 选择“File | New Application”菜单命令新建一个新的工程，然后选择“File | Save Project As”菜单命令保存该工程文件。

第 2 步 弹出“Save Unit1 As”对话框，对话框中要保存的是主窗体的文件名称（*.cpp 类型）。由于是主窗体，所以一般可用“main.cpp”文件名来保存（文件名的选取可凭个人兴趣，但要注意选取有意义的名称）。注意选择路径，如选择“d:\test”目录。设置完这些后，单击“保存”按钮。

第 3 步 弹出“Save Project2 As”对话框，对话框中要保存的是工程文件的名称（*.bpr），工程文件以 bpr 为后缀名，默认情况下是 Project2.bpr。由于工程文件的名称与最后生成的

应用程序名称一样，所以最好选用能表示应用程序特征的名称。在文件名编辑框中输入“test”，然后单击“保存”按钮。注意工程文件与窗体文件应该在同一目录下。

虽然我们只设置了两个对话框，但是查看一下“d:\test”目录，便会发现里面已经有6个文件了，其中“test”名的有3个，分别是 test.bpr、test.cpp、test.res。下面分别介绍这几个文件的作用。

- **test.bpr** 文件：前面已经介绍了它是工程文件。事实上，工程文件是一个文本文件，它包含了建立工程所需的选项和规则。可以用“记事本”之类的工具打开它，并对它进行编辑。当然，不鼓励这样做，这方面的修改应该在 IDE 下进行。
- **test.cpp** 文件：工程主源文件。test.cpp 文件中包含了应用程序的入口（起始）代码、各种宏（如 USEFORM、USEUNIT）等。在标准的 Windows 应用程序中，工程主源文件以 WinMain() 函数作为应用程序的入口。在控制台程序中，这个函数是 main()，而在 DLL 程序中，这个函数是 DllEntryPoint()。与其他自动生成的源文件不同的是，工程主源文件没有相对应的头文件（.h 文件）。一般不需要对工程主源文件进行修改，除非是在一些特殊的场合，如制作程序启动时的画面等。
- **test.res** 文件：资源文件。资源文件中包含了应用程序的图标、版本号等信息。并不是所有类型的应用程序都有资源文件。资源文件中存储的是与工程有关的图像、光标、图标等。开发人员可以利用 C++ Builder 附带的 Image Editor 工具来制作图标资源，利用 C++ Builder 附带的 brcc32.exe 程序来制作更为强大的资源文件。

“main”名称的文件也有3个，分别是 main.cpp、main.h、main.dfm。main.dfm 文件是窗体文件，它包含了窗体的可视化设计的信息，如窗体上的组件、组件的位置、字体等。在 C++ Builder 5.0 中，.dfm 文件在默认情况下以文本格式保存，但也可转换成二进制格式保存。如果需要，可以编辑.dfm 文件，但不鼓励这样做，而应该在 IDE 下对窗体进行修改。main.cpp 和 main.h 是典型的 C++ 语言文件名称，这里不再赘述。

如果需要在工程中使用多个窗体，那么可以选择“File | New Form”菜单命令，或者选择“File | New”菜单命令，在“New Items”对话框中单击“New”标签，在弹出的选项卡中选择“Form”项。然后保存该窗体，该窗体将会自动加入到工程中。

窗体是有可视化界面的，而 C++ Builder 中的有些源文件不需要可视化界面，如包含一些数学运算函数的源文件等。这类的源文件在 C++ Builder 中称为单元（Unit）文件。可以选择“File | New”菜单命令，从“New Items”对话框中单击“New”标签，在弹出的选项卡中选择“Unit”项来生成单元文件。单元文件由.cpp 和.h 文件组成。

除了上面所说的这些文件类型外，C++ Builder 中的文件还有：

1. 包文件

事实上，包就是一种能被 C++ Builder 应用程序共享的动态链接库。通过包可以在应用程序之间共享类、组件和数据，如 C++ Builder 中的可视化组件基本上包含在 VCL50.bpl 包中，C++ Builder 创建的大部分应用程序都可以从这个包中共享一些代码。与包有关的文件包括：

- **Package Project Options (.bpc)**文件。这一文件可以理解为包的工程文件，但它只适用于制作包。在选择“File | New Package”菜单命令时会生成这个文件。
- **Borland Package Library (.bpl)**文件。这是包的运行库文件，它的作用与.dll 文件类似，都是将可执行代码链接到应用程序中。不过.bpl 文件包含了 C++ Builder 特有的一些性质。
- **Borland Import Library (.bpi)**文件。这是包的导入库文件，类似动态链接库中的.lib 文件。它包含了动态链接.bpl 文件所需的导入记录，以及从相应的.bpl 文件中导出的数据。每次编译包的源文件时，都会生成一个.bpi 文件。
- **.lib** 文件。静态链接库文件，包含了包文件中单元目标文件 (.obj) 的静态库，用于将导出函数和数据静态链接到一个应用程序中。

2. 桌面选项文件

C++ Builder 的 IDE 设置可保存在桌面选项文件中，下一次打开工程时，就会自动将 IDE 还原为上一次退出工程时的设置。

桌面选项文件的后缀名为.dsk，它与工程文件保存在同一目录下，如果要自动保存这个文件，则可按下面的步骤设置：

第 1 步 选择“Tools | Environment Options”菜单命令。

第 2 步 在弹出的“Environment Options”对话框中单击“Preferences”标签。

第 3 步 在“Autosave options”组合框中选择“Project desktop”项。

这样，在退出工程时就会自动保存桌面选项。

3. 备份文件

在每次保存.bpr、.dfm、.cpp、.h 文件时，C++ Builder 会自动为这些文件创建一个备份文件。所有备份文件的扩展名都会加上前缀“~”，如.bpr 文件变为~bpr，.cpp 变为~cpp 等。

1.2.2 工程管理器 (Project Manager)

选择“View | Project Manager”菜单命令或按下“Ctrl+Alt+F11”键便可打开工程管理器，如图 1.3 所示。

工程管理器的主体部分列出了工程中包含的窗体及单元的名称和路径。工程管理器的主要功能有：

- 增加或删除工程
- 增加或删除窗体及单元
- 拖放功能
- 选择一个工程

这些功能大多可以通过右击“ProjectGroup1”或工程名称（如 test），在弹出的菜单中进行操作来实现。