Borland C++ Builder 程序设计

刘　华　等 编著

清 华 大 学 出 版 社

（京）新登字 158 号

内 容 简 介

本书以 C++ Builder 5.0 为蓝本用 14 章的篇幅详细介绍利用 C++ Builder 进行程序设计的基本技术，它们包括 C++ 面向对象的基本概念、VCL 提供的基本 Windows 组件、常用窗体的设计技术、Windows 常用应用程序框架的设计技术，以及利用调试技术和异常处理机制处理应用程序中可能出现的错误的技术。另外，本书还在上述基本技术基础之上简要介绍访问文件、数据库应用程序和多媒体应用程序设计技术，为读者进一步提高应用程序设计能力提供帮助。

本书使用简单明了的语言详细介绍基本的应用程序设计技术，而在对于一些常见的高级技术，则主要通过示例程序加以简要介绍。本书通过大量示例程序说明常见应用程序的设计方法，它们的所有代码都可以在本书的附带光盘中找到。

本书可以作为想快速学习 Borland C++ Builder 程序设计人员的入门教材，其中部分内容还可作为初、中级程序员的参考资料。

书　名：Broland C++ Builder 程序设计
作　者：刘　华　等
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
印刷者：北京清华园胶印厂
发行者：新华书店总店北京发行所
开　本：787×960 1/16　**印张**: 32.5　**字数**: 724 千字
版　次：2001 年 6 月第 1 版　2001 年 6 月第 1 次印刷
书　号：ISBN 7-900635-10-6
印　数：0001~5000
定　价：55.00 元

前　言

1．关于 C++ Builder 5.0

Borland C++ Builder 的发展很具戏剧性，其最初版本竟被 Borland 公司作为赠品，这当然与 Borland 公司的最初发展重点有关。然而，随着 Delphi 提供的 RAD 程序设计技术的不断成熟，以及 Borland 公司在 RAD 工具的不断领先和它逐渐将重点向 C++转移，都使热心于 C++ Builder 的程序员信心百倍。

从 1.0 到 5.0，C++ Builder 给程序员带来的欢呼难以数尽，一次次的技术突破给人的影响是无法忘怀的。实际上，只要仔细浏览 C++ Builder 各个版本提供的帮助文档，并仔细浏览其不同版本的 What's New，所有这些不言自明。发展至今，C++ Builder 和 VCL 的成熟速度几乎令人咋舌。勿庸置疑，Borland 公司在应用程序开发技术的发展上占有绝对的优势。

从目前来看，要将 C++ Builder 推下历史舞台的只能是 Microsoft 的 Visual C++。当然，这里我们不能肯定它们各自的发展前景和命运如何。但是，来自网上的一个比较结果却使我们大受鼓舞，不论从运行速度、资源利用效率，还是从代码多少来看，C++ Builder 的集成编译器生成的应用程序都占有绝对优势，甚至由 Visual C++ 进行速度优化的应用程序在速度方面仍然大大落后于 C++ Builder 没有使用优化选项生成的应用程序。容易理解，Borland 公司在 C/C++ 编译器领域的明显技术领先地位使我们看到 C++ Builder 的光明未来。

诚然，利用 Visual C++ 开发与操作系统密切相关的应用程序是比较明智的选择，然而几乎没有程序员（包括 Visual C++ 程序员）认为 Visual C++ 在加速应用程序开发方面占有任何优势，而 Visual C++ 的 MFC 更令初学者望而生畏。

发展到 5.0 版本，C++ Builder 的技术已经非常成熟。集成开发环境、VCL、集成编译器、集成链接器、集成调试器、对 COM/COM+ 技术的支持和辅助开发工具都得到进一步发展。在 C++ Builder 集成开发环境中开发应用程序的过程能够让程序员享受 RAD 技术带来的实惠，读者不妨试一试，并与其他开发工具做一个简单比较。

2．关于本书

作者编写本书的目的非常明确，帮助 C++ Builder 初学者和具有一定基础的用户提高使用 C++ Builder 开发应用程序的技术。而本来的意图是要为中高级读者编写一本参考用书，但是作者本人进入 C++ Builder 的艰难历程和苦苦煎熬的记忆促使我改变写作方向，而将重点放在帮助 C++ Builder 的初学者和具有一定基础的初级用户，因为从他们身上能够看到 C++ Builder 的信心。

本书的内容定位与它的编写目的是紧密相连的。在介绍 C++ Builder 设计应用程序的基本技术时，作者尽可能使用简单明了、通俗易懂的语言，而为了能够为初学者提供进一步提高程序设计技术的基础，并为具有一定基础的初级用户提供丰富的学习内容，本书在介绍基本程序设计技术时不时插入一些作者在设计大量应用程序过程中总结出来的点滴经验，而且本书的内容安排上也将尽力体现这个目的。在本书开始的几章中，作者不啻使用非常详细和繁琐的介绍方法使初学者尽快进入 C++ Builder 的世界，然后在后面章节中迅速向纵深方向突进，希望籍此最终将初学者和 C++ Builder 的初级用户带入具有一定深度的程序设计技术中。

本书的内容编排上还有一个非常突出的特点，那就是使用大量提示语句。凡是使用灰色背景的文字前面都带有“注意”、“说明”、“提示”、“问题”和“小技巧”等字样，这些都是作者在写作过程中插入的与上下文紧密相关的一些程序设计经验。尽管这样的经验对读者设计应用程序的过程没有很大影响，但是在设计应用程序过程如果遇到提示文字中叙述的情形，它们的作用就会变得非常明显，所以读者不妨也仔细浏览这些内容。

学习应用程序设计技术与简单理论的学习不同，学习者需要的是简单的启发和大量的示例程序，所以本书在介绍使用 C++ Builder 设计应用程序的基本技术时，尽量使用大量示例程序说明它们的具体使用方法。丰富的示例是本书的一个非常重要的特点。另外需要说明的是，在光盘上能找到本书的所有示例程序的源代码。

为了方便读者学习和查阅本书的内容，本书还编排了全书索引，读者可以通过附录上的索引迅速定位具体内容。

3．本书导读

本书分成 14 章按照由浅入深的方式扼要、全面地介绍 C++ Builder 常用的应用程序设计技术，下面分别介绍各章的主要内容，并说明它们使用的对象。

第 1 章　开发环境概述

本章分成“C++ Builder 的历史和未来”、“BCB 与 Delphi”和“集成开发环境”等 3 个小节简单介绍 C++ Builder 5.0 的来龙去脉，展望其未来，并在简要分析其与姊妹产品 Delphi 之间的关系后扼要说明 C++ Builder 5.0 集成开发环境的主要组成元素，为下面的介绍奠定技术基础。

第 2 章　C++ 语言和面向对象

本章分成‘C++ 语言的基本点’和‘在 BCB 中学习 C++’两个小节，分别介绍 C++ Builder 使用的 C++语言面向对象的主要特点和在 C++ Builder 中如何编写体现面向对象特征的简单应用程序。

第 3 章　应用程序工程的管理

本章分成“创建我的‘Hello World！’”、“删除整个桌面”和“修改程序标志”等 3 个小节介绍在 C++ Builder 中管理工程资源的方法，并介绍利用工程选项对话框修改工程特性参

数的方法，从而读者可以定制自己的工程而生成独具特色的应用程序。

第 4 章　应用程序的用户界面元素

本章分成“设计菜单”、“几种常用组件”、“设计标准 Win32 用户界面”和“使用系统功能”4 个小节分类介绍 C++ Builder 提供的应用程序的基本组成元素——组件。容易理解，它们是构成应用程序的基石，也是初学者学习程序设计技术的基础。如果读者已经对 C++ Builder 的常用组件有所了解，则可以跳过本章的学习，但是需要注意的是，本章在介绍 C++ Builder 的基本组件同时还列举了大量示例程序，它们仍然值得学习。

第 5 章　复杂窗口和对话框

本章分成“窗体及其调用”、“选项对话框”和“模仿‘资源浏览器’”3 个小节介绍应用程序使用多个窗体的方法，并介绍如何设计选项风格对话框，模拟 Windows 的资源管理器设计类似对话框。本章是作者对多类常用窗体进行总结后得到的，读者可以参考作者的分类方法学习窗体的设计方法。

第 6 章　设计应用程序框架

本章分成“对话框应用程序框架”、“SDI 应用程序框架”、“MDI 应用程序框架”和“控制台应用程序框架”4 个小节介绍几类常见应用程序框架的设计方法。本章的内容与前面两个章节的内容构成一个有机的整体，它们是程序员设计应用程序必须具备的 3 个基本素质。如果读者对应用程序框架比较清楚也可以跳过本章的学习。但是，这里仍然需要强调的是，本章在介绍应用程序的框架时也使用了大量示例，它们仍然是值得学习的。

第 7 章　文件和目录处理功能

本章是为了提高初学者应用程序设计技术而引入的，它分成“目录和逻辑驱动器”、“使用文件的版本信息”、“使用文件指针”、“使用文件句柄”和“文件的流操作”5 个小节介绍常用的文件访问技术。很显然，学会并使用这些文件访问技术可以大大丰富应用程序的系统资源访问功能，这是本书所有读者必须学习的重点章节。

第 8 章　应用程序的调试

本章分成“应用程序的错误概述”、“BCB 的调试器及其设置”、“BCB 的调试技术和辅助调试工具”和“调试应用程序的例子”4 个小节介绍 C++ Builder 设计的应用程序中经常出现的错误，并说明这些错误的排除方法，进而引入 C++ Builder 的应用程序调试技术，介绍使用集成调试器调试应用程序的基本方法和经常使用的辅助调试工具。调试技术是程序员必须掌握的基本技术之一，所以本章也是所有读者必须学习的重点章节之一。

第 9 章　应用程序的异常处理

本章分成“C 和 C++ 的异常处理技术”、“BCB 的异常处理技术”、“使用 VCL 的异常处理功能”和“使用自定义异常类”4 个小节介绍 C++ Builder 支持的另一种错误处理机制——异常处理，并通过示例程序说明这种机制的使用方法。本章也是本书的重点章节之一。

第 10 章　多媒体编程

多媒体编程不是本书的重点，这是为了提高读者学习的兴趣，并为进一步提高多媒体程

序设计技术而引入的。它分成“简单的媒体播放器”、“控制媒体播放器”和“一个 CD 播放器的例子”3 个小节介绍使用多媒体组件设计多媒体应用程序的基本方法。如果读者需要详细学习多媒体应用程序的设计方法可以参考清华大学出版社的同类专题图书。

第 11 章　基于 BDE 的数据库连接技术

本章简要介绍 Borland 公司最成熟的一种数据库连接技术——基于 BDE 的数据库连接技术，它分成“BDE 概述”、“连接数据库”和“控制数据库中的数据”3 个小节介绍 BDE 技术的核心，并说明使用基于 BDE 技术的组件连接数据库的基本方法。本章可为读者学习数据库应用程序设计技术奠定基础。

第 12 章　基于 ADO 的数据库连接技术

本章分成“BCB 对 ADO 技术的支持”、“使用 ADO 组件连接数据库”和“ADO 数据集组件”3 个小节，通过简单示例说明 C++ Builder 基于 ADO 数据库连接技术的组件的使用方法，并利用这些组件实现上一章实现的数据库连接功能。

第 13 章　数据报表

本章分成“QuickReport 组件概述”、“设计数据报表”和“使用数据报表”3 个小节介绍 C++ Builder 提供的报表功能。很显然，报表组件能够大大丰富数据库应用程序的功能，本章也是读者必须学习的重点章节之一。

第 14 章　FTP 客户端应用程序

本书并没有展开介绍 C++ Builder 提供的网络应用程序设计技术，而在最后利用一章的篇幅，分成“Internet 类组件”、“TNMFTP 组件”和“一个 FTP 客户端的例子”3 个小节简要介绍 C++ Builder 提供的网络应用程序设计技术，并通过一个简单的示例程序——FTP 客户端应用程序，说明网络组件的使用方法。

本书在最后还提供 4 个附录，以方便读者学习本书的内容。

4．本书约定

为了方便读者学习本书，在编写本书时，作者使用了一些统一的约定，从而保持风格一致，这些约定主要包括：

- 本书所有中文菜单项都以【】括起，例如【文件】，而多级菜单以“|”隔开，例如【文件】|【打开】,【工具】|【字体】|【颜色】。
- 本书对所有控件对象的命名采取统一的方式：控件名称的简称+控件对象的名称，例如 btnOk；而所有控件本书都采用统一的名称约定，例如 TButton 简称为 btn。
- 本书中插入一些用“说明”、“注意”、“提示”、“问题”和“小技巧”标记的内容，用来辅助读者的学习。
- 在提示性文字前面使用“注意”、“说明”、“提示”、“问题”和“小技巧”等字样，以引起读者的注意。

5．致谢

本书成书过程中，受到众多同仁的支持和帮助，在此表示感谢。其中，Borland **公司驻中国的技术支持对于本书的成书有重要作用，作者表示诚挚感谢。另外，本书还受到薛阳、刘剑锋、罗宏宇、李言良、韩宝龙、宋岩磊等人的大力帮助，在此一并表示谢意。**

本书之所以能够出版，离不开清华大学出版社的大力支持和帮助，在此代表本书的所有作者对清华出版社表示诚挚的感谢。

作　者

2001.2.17

目　　录

第 1 章　开发环境概述

从本章开始，就可以开始 C++ Builder 的学习。不管是第一次接触 C++ Builder 的新朋友，还是已经基本熟悉它的老朋友，只要还没有完全掌握用它开发应用程序的必要技术，那么都可以从本章开始获取你所需的东西。

为了让更多的人熟悉 C++ Builder，本章将首先概述它的产生和发展过程中出现的一些轶事，以及它产生的背景和必然性，提高读者对其未来的信心。然后，本章还将介绍 C++ Builder 的核心——VCL，以及与其姊妹产品 Delphi 的关系。最后，简单介绍集成开发环境中的一些基本工具，以使读者能够对这些辅助开发工具有一个初步了解，便于下面的学习。

1.1　C++ Builder 的历史和未来

1.1.1　起源及其背景

在 Borland（Inprise）公司的软件家族中，C++ Builder 属于新成员，甚至在推出编译器的初期，C/C++ 编译器根本不是 Borland 的看家本领。事实上，Borland 公司是从 Pascal 编译器起家的，只要明确它的第一个产品是由 15 年前 Anders Hejlsber 编写的第一个 Turbo Pascal 编译器，就很容易理解了。Borland 公司成功的关键就是这个 Pascal 编译器只有稳定、优雅的特性与极其快速的编译能力，而且，因为当时 Pascal 是使用最为普遍的教学语言，因此，Borland 就自然成为一方霸主。

当然，Borland 公司不会将其发展的着眼点只放在 Pascal 上，因此，在 Turbo Pascal 编译器取得成功后，它迅速迈出一个非常关键的步骤——推出了 Turbo C/C++ 编译器，这使 Borland 公司再一次达到其发展过程中的一个“极点”。随着 C/C++ 编译器的不断发展，Turbo C/C++ 的版本也不断提高，其性能也不断成熟。

但是，Windows 操作系统的产生给 Borland 公司迎头一击，Turbo Pascal 和 Turbo C/C++ 编译器被冷落自然成为一种必然。为了应付形势的变化，沿着 Turbo C/C++ 的发展轨迹，Borland C++ 3.x/4.x/5.x 的出现也是一种必然。然而，直至 Borland C++ 5.0，Borland 公司在与 Micorsoft Visual C++的竞争中并没有占有明显的优势，甚至产生了令人非常不愉快的结果。然而，就在 Borland C++与 Microsoft Visual C++厮杀的过程，Borland 公司找到了另一条捷径——Delphi。

尽管 Turbo Pascal 已经渐渐被历史遗忘，但是，它的开发小组并不是一堆不会复燃的死

灰，Anders Hejlsber 又一次成为 Borland 公司的救星。他再度领导开发小组，推出具有划时代意义的新一代软件开发工具——Delphi。而 Delphi 一经推出后，迅速成为“快速应用程序开发”（即 RAD，Rapid Application Development）这个新概念名副其实的领导者。可以说，Delphi 的推出使 Borland 公司在与 Microsoft 的竞争中转危为安，而 Delphi 则迅速成为众多程序员青睐的快速开发工具。

此时，Borland C++ Builder 的产生才具有所有必备的条件，它在 Delphi 的哺育下不断发展壮大。当然，Borland C++ Builder 发展到 5.0 版本时仍然不能抛开 Delphi，因为它的核心 VCL 仍然是 Delphi 的；从某种意义上说，Borland C++ Builder 就是 Delphi for C++。尽管目前 Borland C++ Builder 还不能离开 Delphi，但在不久的将来，用 C++ 重写 VCL 必将会使 Borland C++ Builder 完全摆脱 Delphi 的束缚。而将来 Borland C++ Builder 是否能够替代 Delphi，则不得而知。不过，在短期内，这是不会发生的，毕竟 Borland C++ Builder 是喝 Delphi 的“奶”长大的。

1.1.2 世纪末软件革命的候选领袖

随着软件开发技术的不断发展，在众多应用程序开发工具的竞争硝烟中，众多程序员能够嗅出一股软件革命的气味，而这样的革命是在世纪之交发生的，因此，称为世纪末的软件革命。正如工业的不断发展必然导致所谓的工业革命一样，软件业的发展也会导致软件革命，而这场革命的领导者是谁，还不得而知。

目前，Windows 下的 RAD 开发过程已经成为一种事实上被公认的应用程序高效开发方法，而在 RAD 领域可能成为领袖的两位候选人只能是 Delphi（或者 Borland C++ Builder）和 Microsoft Visual Basic（或者 Microsoft Visual C++）。这里之所以将 Delphi 和 C++ Builder 罗列在一起，是因为在这场竞争中它们代表了同样的技术，惟一的差别是它们支持的程序设计语言不同，这对 Pascal Object 和 C++ 来说仅仅意味着语法的不同。而将 Microsoft Visual Basic 和 Microsoft Visual C++ 放在一起的原因是，从某种意义上说，Microsoft Visual C++ 不能算得上是真正的 RAD，仅仅可以归入可视化开发工具一类，而这里之所以仍将它列出来，是因为它可能成为 RAD。在这场 RAD 竞争中，Microsoft Visual Basic 成为领袖的可能性不会很大，这在很大程度上是受限于 Basic 语言的。

目前，Borland C++ Builder 有点始终令众多程序员心有余悸：VCL 库是由 Delphi 提供的，因此，它的发展必然受制于此，从而使 Borland C++ Builder 无法充分发展其优势。另外，在与 Microsoft 的 RAD 竞争时，Borland 公司也没有与系统实现完全无缝结合的优势。当然，上述 Borland C++ Builder 面临的问题将随着其进一步发展而得到解决。

首先，Borland 将其重点放在 Borland C++ Builder 上已成为事实，用 C++ 重写 VCL 将成为一种必然。实际上，Delphi 只是 Borland 公司在竞争中的前锋，以发挥其在 Pascal 上的技术优势，而随着 Borland C++ Builder 的不断发展壮大，Delphi 隐居后台也似乎将是合情合理的了。

随着软件开发技术的不断标准化，Borland 公司在与其他 RAD 工具的竞争中其劣势将不断被削弱，而其优势将不断得到加强，最终使其成为这场世纪末软件革命的领袖——相信每一位 Borland C++ Builder 的程序员都应该具有这样的信心。

为了叙述上的方便，在下面章节中 Borland C++ Builder 将一律用 BCB 代替。

1.2　BCB 与 Delphi

众多程序员习惯于将 BCB 与 Delphi 称为 Borland 公司的姊妹产品，这是有道理的。BCB 和 Delphi 不仅在外观上非常相似，它们使用的底层库 VCL 也完全相同。因此，使用 BCB 与 Delphi 的方法基本相同。然而，将 BCB 称为 Delphi 的“孩子”也具有一定的道理，BCB 完全是喝着 Delphi 的“奶”长大的，相同版本的 BCB 总是在 Delphi 后推出，实际上，Delphi 就是 BCB 的前锋。

1.2.1　BCB 与 Delphi 有相同“长相”

只要打开 BCB 与 Delphi 的集成开发环境，对它们有相同的“长相”这一说法是不难理解的。一般情况下，BCB 总是随着 Delphi 的版本更新而不断更新其版本的，因此 BCB 的每一个版本总是落后于 Delphi。这里的比较当然是在相同版本的 BCB 和 Delphi 之间进行的。两者的相似点几乎不能罗列完，独立于具体语法的特点必然是两者共有的，这里可以罗列如下几个最具有代表性的。

除了[帮助]菜单的菜单项不完全相同，其他菜单完全相同，而菜单的功能也完全相同。

两者的帮助系统基本相似，除了与具体语言相关的部分，其他部分使用相同的帮助文件。

注意：

特别需要说明的是，在 BCB 中打开的 VCL 使用 Object Pascal 语法，而不是 C++语法，因此，许多关键字都不相同。另外，BCB 还引入 C++ 不支持而 Object Pascal 支持的 Set 类型。除此之外，BCB 还引入许多 C++ 中没有的关键字，例如 published、__property 等。

两者的工具栏包含的工具按钮完全相同（当然，程序员可以自定义工具栏）。

两者的组件栏包含的组件完全相同（当然，程序员可以自定义组件的排列方式，也可以安装其他组件）。

两者对工程的管理方式完全相同，提供的工程管理器也完全相同。

两者提供的代码编辑器完全相同。当然，它们支持的语法不相同，使用的创建新窗体的方式也不相同。最明显的差别是，C++ 中存在头文件的概念，而在 Object Pascal 中则不存在这个概念。因此，在创建新窗体时，BCB 将自动创建窗体对应的窗体文件（FRM 文件）、代码文件（CPP 文件）和头文件（H 文件），而 Delphi 则不需要创建头文件。