# 技术内幕

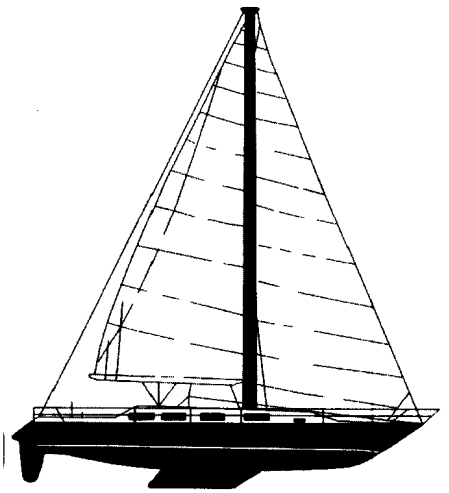
# C++ Builder 5

乔林 费广正 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

# C++ Builder 5

# 技术内幕



乔林　费广正等　编著

中国铁道出版社
2000年·北京

（京）新登字 063 号

## 内 容 简 介

本书通过大量深入的实例，主要讲解了面向对象程序的基础知识、异常处理的基本方法、类的继承与重载、多态性与动态联编等，因为它们是创建 C++ Builder5 部件的核心技术。同时全书的宗旨意在告诉读者应该遵循什么样的步骤，应该采取什么样的思考方法，以及如何将自己的思考转化为正确的程序代码。强调只有在“干中学”才能有最大的收获。

**图书在版编目（CIP）数据**

C++ Builder 5 技术内幕/乔林等编著. —北京：中国铁道出版社，2000.7

ISBN 7-113-03797-6

Ⅰ.C… Ⅱ.乔… Ⅲ.C 语言—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2000)第 32849 号

**书　　名**：C++ Builder 5 技术内幕

**作　　者**：乔林　费广正等

**出版发行**：中国铁道出版社（100054，北京市宣武区右安门西街 8 号）

**策划编辑**：严晓舟

**特邀编辑**：孙晓博

**封面设计**：冯龙彬

**印　　刷**：北京市兴顺印刷厂

**开　　本**：787×1092　1/16　印张：28.75　字数：695 千

**版　　本**：2000 年 7 月第 1 版　2000 年 7 月第 1 次印刷

**印　　数**：1～4000 册

**书　　号**：ISBN 7-113-03797-6/TP·459

**定　　价**：45.00 元

# 前　言

C++ Builder 5 是一个全新的可视化的 C++ 编程环境，它为我们提供了一种方便、快捷的 Windows 应用程序开发工具。它使用了 Microsoft Windows 图形用户界面的许多先进特性和设计思想，采用了弹性的、可重用的和完整的面向对象程序语言（Object-Oriented Programming Language）。对于广大的程序开发人员来讲，使用 C++ Builder 5 开发应用软件和数据库应用程序，无疑会大大地提高编程效率，而且随着应用的深入，我们将会发现编程不再是枯燥无味的工作——C++ Builder 5 的每一个设计细节，都将带给我们一份难以表达的惊奇。

## C++ Builder 5 的新特征

笔者曾使用过 Pascal、C、Borland C++、Visual C++、Visual Basic 和 Delphi 编写程序，开发过应用系统，但从没有一个系统像 C++ Builder 5 这样给我们留下如此深刻的印象。虽然 C++ Builder 5 中的每个部件都是指针，会使有一定基础的读者感到畏惧（正如笔者最初也都曾对此表示怀疑一样），但这也正是让人备觉兴奋的地方（读者最终也会理解这种方式，并对它情有独钟）。通过本书的学习，读者将会了解到，对象的委托与关联只有用指针才能最好的描述。

C++ Builder 5 实际上是 C++ 语言的一种版本，但它与传统的 C++ 语言有着很大的差别。一个 C++ Builder 5 程序首先是应用程序框架，而这一框架正是应用程序的“骨架”。在骨架上即使没有附着任何东西，也仍然可以严格地按照设计运行。程序员的工作只是在“骨架”中加入自己的程序。缺省的应用程序是一个空白的窗体（Form），我们可以运行它，结果得到一个空白的窗口。这个窗口具有 Windows 窗口的全部性质：可以放大缩小、移动、最大化和最小化等，但我们却没有编写一行程序。因此，可以说应用程序框架通过提供所有应用程序共有的东西，为用户应用程序的开发打下了良好的基础。C++ Builder 5 已经为读者做好了一切基础工作——程序框架就是一个已经完成的可运行应用程序，只是不处理任何事情。我们所需要做的，只是在程序中加入完成所需功能的代码而已。

在空白窗口的背后，应用程序的框架正在等待用户的输入。由于我们并未告诉它接收到用户输入后作何反应，窗口除了响应 Windows 的基本操作（移动、缩放等）外，它只是接受用户的输入，然后再忽略。C++ Builder 5 将 Windows 编程的回调、句柄处理等繁复过程都放在一个不可见的覆盖物下面，这样读者可以不为它们所困扰，轻松从容地对可视化部件进行编程。

坦率地说，一名优秀的编程人员需要一点小聪明。在 Windows 操作系统出现以前，编程实际上是一种激动人心的智力体验。而随着 Windows 操作系统的不断升级和普及，编程正在变成越来越枯燥无味的事情。相信读者对使用 Visual C++ 开发应用程序的繁琐和细节之多已经不厌其繁，而 C++ Builder 5 完全绕开了这些细节，将激动人心的编程体验重新带回到 C++

编程中来。

如果要快速开发应用程序，读者将会发现 C++ Builder 5 是非常容易使用的编程工具。C++ Builder 5 支持全部的高级 C++ 特性，例如模板（template）、名字空间（name space）和操作符重载（operator overloading），此外，C++ Builder 5 也完全支持全部的 Windows API 函数，例如割边（cutting-edge）函数 DirectX、OLE 自动化（automation）和 ActiveX。

在大多数情况下，使用 C++ Builder 5 进行编程都是非常容易的，这并不意味着 C++ Builder 5 是非常低级，功能不全的开发工具。事实上，C++ Builder 5 完全可以完成任何复杂的编程任务，我们既可以编写 C++ Builder 5 代码，也可以直接编译 Delphi 源代码，我们甚至可以使用 MFC 类库！

## 本书的读者对象

就笔者各人的观点，今天最优秀的程序员不是要对操作系统知之甚详，而是要对真实世界的应用知之甚详。本书不是为初级或中级程序员编写的读物，因此，笔者假设读者有较广泛的编程经验；而且相对于少而精，笔者更倾向于希望读者的知识面广而杂。会用汇编写 Windows 95/98/NT 应用程序是值得称道的，但不了解面向对象的技术和应用程序建模方法学却是不可接受的。

我们这里要指出三点：

首先，本书是基于图形用户界面的，特别是基于 Windows 95/98/NT 界面的，这样就很容易掌握一些最重要和最基本的概念。

其次，本书中的内容没有一项是特别的技术。关于 C++ Builder 5 语言的精确定义可以从各种各样的资料上得到，其中最重要的一个就是 C++ Builder 5 的联机帮助系统。当然，只有理解了事情怎样做以及为什么这样做时，这些介绍才能有所帮助，否则语言描述本身就太抽象了。

最后，本书的宗旨是通过大量深入的实例讲解 C++ Builder 5 的核心内容。与简单地完成某项编程任务相比，我们更关心 C++ Builder 5 如何完成这项编程任务。因此，本书包含了部分 Delphi 代码，这部分代码是深入了解 VCL 的基础。我们希望引入的 Delphi 代码没有给读者带来任何不便。

正是基于上述三点考虑，贯穿整本书的是读者应该遵循什么样的步骤，应该采取什么样的思考方法，以及如何将自己的思考转化为正确的程序代码。因为只有在“干中学”才能有最大的收获，所以本书包含了大量实例，这些例子大部分都较难，而且在很多情况下还有一定的关联。我们希望读者在阅读本书的同时也上机实践。

## 本书的组织

本书是这样组织的。本书的第 1 章简要介绍了 C++ Builder 5 的可视化编程环境。第 2 章着重讨论 C++ Builder 5 的语言扩展。第 3 章和第 4 章则讨论 C++ Builder 5 与 VCL 的关系以及 C++ Builder 5 的核心技术——事件与委托模型。第 5 章为一个实际的文本编辑器例子，通过该实例，读者将掌握创建 C++ Builder 5 应用程序所需要考虑的基本问题。从第 6 章到第

10 章我们将深入讨论 C++ Builder 5 语言规范。这些章节主要涉及面向对象程序的基础知识、异常处理的基本方法、类的继承与重载、多态性与动态联编。这部分内容事实上是本书的重中之重，我们之所以在这部分花费最多的笔墨，是因为它们是创建 C++ Builder 5 部件的核心技术。没有对面向对象编程的深入了解，我们绝不可能创建一个漂漂亮亮的可以推向市场的 C++ Builder 5 部件。第 11 章由此讨论创建部件的基本方法。本书的第 12 章则是一个例外，我们不希望读者在掩卷长思的时候觉得本书枯燥无味，因此，第 12 章着重讨论如何编写 C++ Builder 5 的游戏程序。

本书的内容较深，没有阅读前面的内容就直接学习后面的章节可能会给读者带来一定的麻烦，当然对相应内容有相当程度了解的读者又另当别论了。

本书的第 1 章由林杜执笔，第 2 章由费广正执笔，第 3 章和第 4 章由乔林执笔，第 5 章由王程铭执笔，第 6 章由乔木执笔，第 7 章和第 9 章由周岚执笔，第 8 章由金传恩执笔，第 10 章和第 11 章由汪巨涛执笔，第 12 章由乔森执笔。全书由乔林统一规划组织。书中如有错误，当是作者水平有限，恳请读者谅解。

## 对本书读者的忠告

在开始阅读本书之前，有必要先搞清楚一个基本事实。使用 C++ Builder 5 编写的程序在代码的大小和性能上与使用 OWL 和 MFC 编写的代码大致是相当的。那些认为 C++ Builder 5 难以完成一些特殊任务的观点是错误的。任何可以使用 Visual Basic、PowerBuilder、Borland C++ 或 Visual C++ 可以完成的任务都可以用 C++ Builder 5 来完成，甚至在某种程度上可以完成得更好，也更加艺术。快速开发工具可能会给人留下 C++ Builder 5 功能不全的错误印象，但是任何使用 C++ Builder 5 进行过深层次软件开发的编程人员都会反驳这种想当然的错误观念。事实上，除了使用起来更容易，设计起来更方便快捷以外，C++ Builder 5 完全具备 OWL 和 MFC 的全部功能。

与 Visual C++ 不同，C++ Builder 5 和 Visual Basic 都是快速开发工具。但是照笔者的看法，它们是不可比较的，Visual Basic 确实是一种非常漂亮的产品，但它并不是一个严肃的开发工具，而 C++ Builder 5 却是真正的 C++ 编译器。与 C++ Builder 5 和 Delphi 相比，Visual Basic 更像一个小玩具。

部件（component）对象的世界总让人想起又慢又难以理解，而且还充斥着错误的 ActiveX 控件。与 ActiveX 控件相比，C++ Builder 5 的部件又快又小又容易使用。对象链接与嵌入（OLE）确实是一种强有力的技术，但它缺少 C++ Builder 5 的 VCL 代码所具有的基本敏感性、速度与灵活性。

倾向于使用 C++ 语言的程序员总是欣赏它编程时的深入性和编译后程序代码的运行速度与可优化能力。因此，鉴于 C++ 语言的这种灵活性和高效率，如果要进行实用程序的开发，养成良好的编程风格是非常有必要的。无论是编写什么样的应用程序，如果对程序中任何一段代码的功能和要完成的任务不太明了，我们都推荐读者仔细思考并打上标记。事实上，笔者就是这么做的。

从某种程度上说，任何程序员都不是完美的。因此，本书所有的例子都基于这样一种假设，那就是笔者不仅会犯错误，而且很快就会犯错误。正是基于这样一种假设，本书才有了

足够的自由度为实际的应用程序尝试一定程度的创造性解决方案。读者一定要注意，一定程度的创造性是必要的，但太专业或充满技巧的代码则是 bug 的温床，然后是多个不眠之夜的前奏。

最优秀的程序员也会犯错误，而且是非常频繁地犯错误！笔者承认并接受这个事实，然后试图寻找一种安全的编程方案，以使得程序在笔者无法预料的情况下也能够安全地运行。

正确的代码总是应该能够以快速的方式完成指定的功能。笔者倾向于编写既不太抽象（太一般）也不太敏感（太特殊）的代码，这种特殊与一般的编程折衷方案正是笔者所欣赏的。至少，这样的代码既能完成指定的功能，也能在笔者没有对它厌倦之前产生创造的欲望，这正是编程的辩证法。

优秀的程序员永远不要尝试成为一个完美主义者，苛求完美只会将你引入歧途。稍微降低一点你的目标，保证所编写的程序可以工作，然后再要求它“似乎是无 bug 运行的”就行了。程序越大越复杂，错误也就越多，Windows 95 的 5000 个 bug 就是最有力的例证（见：*IEEE Computer*，1998，No.7，pp107-109）。笔者从不期望自己编写的程序是完美无缺的，这也可以作为对读者的忠告。

# 目 录

# 第一章　C++ Builder 5 可视化开发概念

本章主要讨论 C++Builder 5 的开发环境与工作原理。相应的主题包括：如何使用 C++ Builder 5 的开发环境；如何创建一个简单的播放电影、WAV 和 MIDI 等多媒体数据的应用程序；如何关闭 C++Builder 5 的快速开发工具窗口直接编写 Windows API 代码；如何在运行时间动态地在堆（heap）中创建部件；如何在运行时间动态地设置事件句柄（handlers）或闭包（closures）；如何使用异常。

本章包含了所有涉及这些基本概念的样本程序。

## 1-1 C++ Builder 5 可视化开发环境

运行 C++ Builder 5 的过程非常简单，从“开始”菜单的“程序”中选择“Borland C++ Builder 5 | C++ Builder 5”选项即可。加载后的屏幕布局如图 1-1 所示。

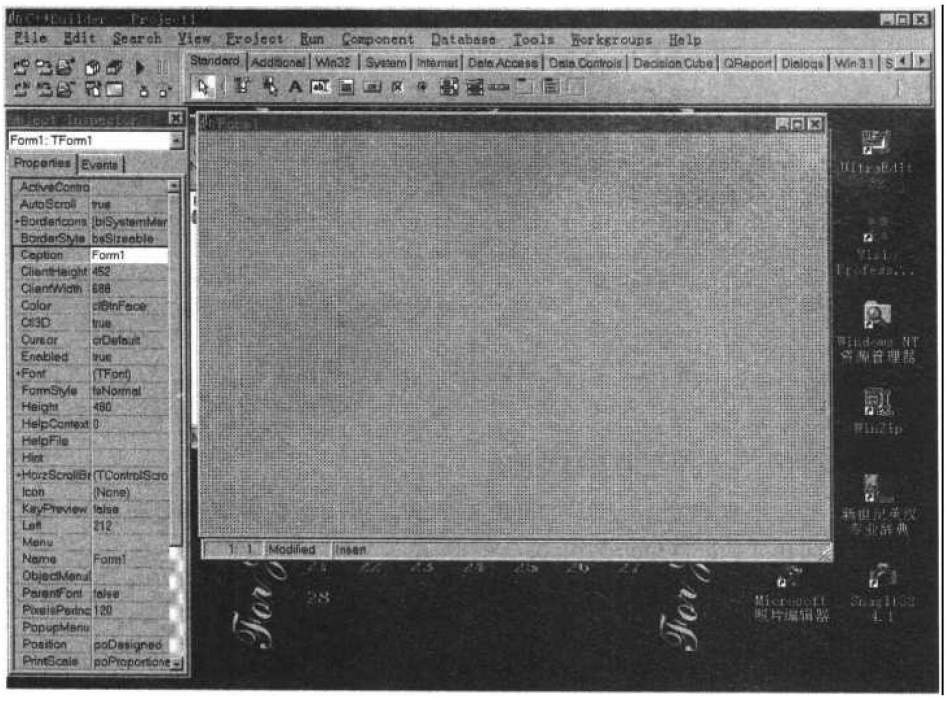



图 1-1　C++ Builder 5 环境的几个组成部分

首次加载 C++ Builder 5，屏幕上会出现四个窗口：

- 标题为“C++ Builder - Project1”带有菜单、工具栏（SpeedBar）和部件面板（Component Palette）的 C++ Builder 5 主窗口；

- 对象观察器（Object Inspector）；
- 标题为“Form1”的窗体设计器（Form Designer）；
- 标题为“Unit1.cpp”的代码编辑器（Code Editor）。刚启动时这一窗口的大部分被“Form1”窗体所掩盖。将“Form1”窗体移开，或单击 Form1 窗体下方代码编辑器窗口的状态行，可以看见其全部。在“Form1”窗体的任意可见位置单击鼠标左键，可以使窗体设计器恢复为可见状态。

对初学者而言，这个环境也许有些混乱。为减小工具表面上的不连贯造成的混乱感觉，只需要记住一点：这几个窗口中的每一个都能够单独完成不同的任务。我们在稍后的示例中将把对象观察器和代码编辑器放大，以显示更多的内容。

## 1-1-1 工具栏

工具栏提供了使用鼠标快速访问许多常用功能的机制，这些功能当然也可以使用菜单选项来实现，如图 1-2 所示。

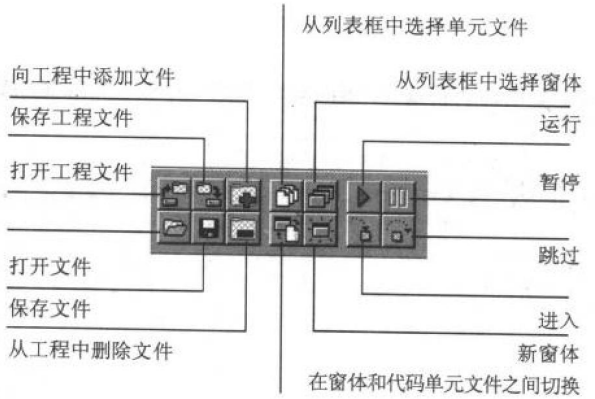



图 1-2　工具栏提供对菜单选项的快速访问

下面介绍工具栏上各个按钮的具体功能。


该按钮上有一个从硬盘到文件夹的箭头，这表示该按钮的任务是打开一个工程文件，它对应于选择“File | Open Project”菜单项。


该按钮上有一个从文件夹到硬盘的箭头，这表示该按钮的任务是保存工程文件，它对应于选择“File | Save All”菜单项。


该按钮上显示一个带有加号的文件夹图符，这表示该按钮的任务是向工程中添加文件，它对应于选择“Project | Add to Project”菜单项。


该按钮上显示一个带有减号的文件夹图符，这表示该按钮的任务是从工程中删除文件，它对应于选择“Project | Remove from Project”菜单项。


这个看似打开文件夹的按钮可以执行装载单元文件的任务，它等同于选择“File | Open”菜单项。


这个看似磁盘的按钮可以执行保存当前单元文件的任务，它等同于选择“File | Save”菜单项。


在接下来的一组按钮中，最左上角的按钮是一幅堆放在一起的多个文档的图案，它表示该按钮的任务是从列表框中选择库单元文件。该按钮允许用户从所有代码单元的列表中选择一个进行编辑。它对应于选择“View | Units”菜单项。


该按钮的图案是一堆叠放在一起的窗体。该按钮允许用户从列表框中选择一个窗体进行设计。它对应于选择“View | Forms”菜单项。


该按钮的图案是一个发光的窗体，它的任务是打开一个新的窗体进行设计。它对应于选择“File | New Form”菜单项。


该按钮是一个窗体和代码文件的切换按钮，它对应于选择“View | Toggle Form/ Unit”菜单项。


该按钮表示运行该工程，它等同于选择“Run | Run”菜单项。


该按钮表示暂停过程的运行，它对应于选择“Run | Program Pause”菜单项。


该按钮表示运行到函数体的内部，它对应于选择“Run | Trace Into”菜单项。


该按钮表示跳过函数体的运行，它对应于选择“Run | Step Over”菜单项。

## 1-1-2 对象观察器

位于窗体窗口左边的对象观察器给出了当前选中对象的详细信息，如图 1-3 所示。

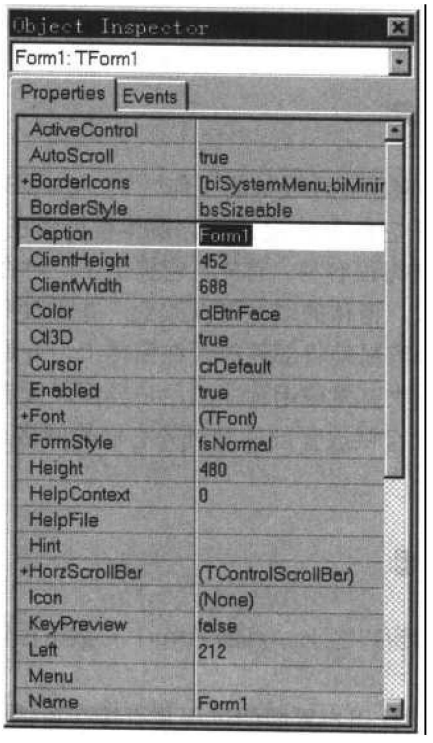



图 1-3 对象观察器显示当前对象的所有属性和可响应的事件