计算机最新技术丛书

PEKING UNIVERSITY PRESS

黄庆生 策划
郭迅华 编著

Borland C++ Builder3 应用程序开发学习教程

How do you collapse four weeks of C++ programming into a minute and a half?

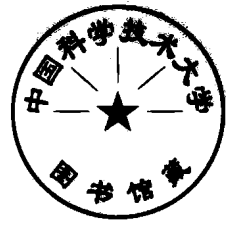
- RAD Productivity through 130 reusable components
- Code Insight, Wizards, Tooltip Expression Evaluation
- No Compromises C++ Compiler with Incremental Linking

北京大学出版社

计算机最新技术丛书

Borland C++ Builder 3
应用程序开发学习教程

黄庆生　策划
郭迅华　编著



北京大学出版社
北　京

内 容 简 介

Borland C++ Builder 3 是著名软件公司 Inprise International 于 1998 年 2 月推出的快速应用程序开发工具，它不但继承了 C++ 语言的全部优点，而且与当今最优秀的可视代交互式界面 Windows 95 紧密地结合在一起，使得应用程序的开发只需移动鼠标即可在 Windows 的窗口上进行；它还提供多种强大的数据库开发工具、丰富的组件资源，可同时创建 Web 客户机和服务器的应用程序、支持已有代码和类库的应用，并提供 COM，ActiveX 等的设计接口，可与其他软件工具交互应用；同时，还采取多层化数据库途径，并提供 Access 数据库支持；……。

总之，C++ Builder 3 的成功，实现了该领域内的革命性突破，使其成为优秀的可视化的应用程序开发新工具，无论是专业技术人员或是初学者，都能从本书中获益。

图书在版编目(CIP)数据

Borland C++ Builder 3 应用程序开发/郭迅华编著. 北京：北京大学出版社，1998.11
ISBN 7-301-03928-X

Ⅰ.B… Ⅱ.郭… Ⅲ.C 语言-程序设计 Ⅳ.TP312

TS404/14

书　　名：Borland C++ Builder 3 应用程序开发学习教程
著作责任者：郭迅华
责 任 编 辑：杨锡林
标 准 书 号：ISBN 7-301-03928-X/TP·0424
出　版　者：北京大学出版社
地　　址：北京市海淀区中关村北京大学校内　100871
网　　址：http://cbs.pku.edu.cn/cbs.htm
电　　话：出版部 62752015　发行部 62754140　编辑部 62752037
电 子 信 箱：zpup@pup.pku.edu.cn
排　版　者：兴盛达激光照排中心
印　刷　者：国防科工委印刷厂印刷
发　行　者：北京大学出版社
经　销　者：新华书店
787 毫米×1092 毫米　16 开本　23.875 印张　595 千字
1999 年 1 月第一版　1999 年 1 月第一次印刷
定　　价：36.00 元

前　言

Borland C++ Builder 3 是著名的软件公司 Inprise International(原名"Borland International",更名后产品标识仍为"Borland",本书中两个名字是等价的)于 1998 年 2 月推出的快速应用程序开发工具。Inprise 公司在其十余年的成长历程中,始终如一地致力于为软件开发人员提供友好高效的开发环境,Turbo C,Turbo Pascal,Borland C++ 等经典作品久负盛名。近年来,Inprise 苦心培育的"Delphi"(现已推出第四版)快速开发工具更是捷报频传、连创辉煌,不仅赢得了众多软件开发人员的青睐,还在各种著名的评选中屡获大奖。

作为快速开发工具(RAD,Rapid Application Development)的一种,"Delphi"与微软公司的"Visual Basic"分庭抗礼,取得了令人瞠目的成功。然而,Delphi 所采用的 Object Pascal 语言对许多人来说都显得较为陌生。另一方面,在软件开发领域,C++语言的至尊地位事实上从未受到过动摇,人们依旧倾心于 C++强大的控制力和高度的灵活性,C++语言的使用,几乎成了高水平专业程序员的标志。在这种情况下,尽管 RAD(以 VB 和 Delphi 为代表)以其易学易用、生动形象的迷人特性如风一般地卷过,"Visual C++","Borland C++"仍以阳春白雪的身份存在着并发挥着不可替代的作用,120 万程序员执著地在繁琐的 C++程序上匍匐前进。

在这样的形势下,Inprise 公司开始尝试将 C++真正地与可视化快速开发相结合。C++的特殊性决定了这项工作的特殊性,幸运的是,Inprise 公司同时拥有"Delphi"和"Borland C++"这两种分别在 RAD 和 C++两方面居于顶尖地位的产品。于是 C++ Builder 便顺理成章地成为这对"金童玉女"天作之合的产物。在解决了一系列问题之后,C++ Builder 1.0 作为 Turbo C++ Suite 的一个组成部分试探性地推向市场并获取了一系列的好评,在并不太长的时间里就卖出了 150000 套软件。近两年来,Delphi 3,Delphi 4 扶摇直上,C++ Builder 也跟紧了步伐。终于,在 1998 年 2 月,有了 C++ Builder 3。虽未能像 Delphi 一样火暴,但也足以让 Inprise 欣喜依旧了。况且,C++ Builder 相对于 Delphi 的最大优势,存在于它的 C++语言基础中,这一点,给了它更为强大的生命力。

可以说,C++ Builder 3 已经拥有了 Delphi 3 的全部优点,Delphi 3 中那些令人心醉的特性无一例外地可以在 C++ Builder 3 中找到。Borland C++强大的功能经过改造之后,也以一种无缝的形式被结合进去。从这种意义上说来,C++ Builder 3 已经是一个相当成熟的产品,将它称为迄今为止最丰富最强劲的 C++开发工具,丝毫不为过。Inprise 实现了这个领域中的一个革命性突破。将以往 C++所建构起来的世界,溶解在可视化快速开发的环境中的梦想,已经成为可能。无论是专业还是业余的开发者,都可以从各自的角度,发现它的魅力。

在国外,C++ Builder 已经成为开发工具市场上的热点,并呈显著的上升趋势。反应敏锐的国内的程序员也已经逐渐开始注意到了这一新兴的宠儿。于是,我萌发了创作这样一部书的想法,希望它能对国内的开发人员有所帮助,同时也为 C++ Builder 的进一步推广起到或多或少地促进作用。

如今,这本书终于以一种相对完整的姿态来到了我们的视线里。我回味着写作过程中的一

切喜悦与艰辛，回味着手指在计算机键盘上往复蹒跚的单调与凝重，回味着一道道沟坎牵绊带给我的那些步履维艰的时刻，身心的疲惫和完成任务的轻松交织成一种难以名状的感受。伴随着这种感受而来的是另一番深深的惶恐，我不知这本书是否真的能够具有我所期待的价值，不知它是否真的能够对使用者有所裨益，不知我这数十天来的殚精竭智焦思苦虑是否会有足够的成效来补偿。

然而我安慰自己，我相信我的努力不会是竹篮打水。在全书的编写过程中，我始终坚持着这样一个信念，那就是将我作为一个学习者和开发者的全部体验和感受融于字里行间，介绍和交流并重，以期使本书达到应有的效果。在长期的学习和工作中，我积累了经验，也产生了许多自己的观点想法，我力图把这些拿出来与人共享。同时，我也要求自己以严谨的态度去对待书中的每一个章节，广泛地查询资料，严格地进行试验。我深深地知道，C++ Builder 3 所涵盖的内容是如此地丰富，尽管历尽艰辛，加倍努力，也难尽显其风采，疏漏和错误实为难免。然而我毕竟已经用心地叙述了我的所知与所想。于是，一切的成败得失，就留待读者去评判。

最后，我得向每一位帮助过我的人致以深深的谢意；感谢倚天图书创作室的支持，是你们提供的便利条件使本书的诞生成为可能；本书在编写过程中，还得到了黄惠卿、任雪荫、林绮文、余超、肖铭、黄挺、黄进等同志的大力协作，在此向他们表示感谢。

郭迅华

一九九八年九月

目　录

第一部分　C++ Builder 3 概览

第二部分　C++ Builder 3 的语言

第三部分 用 C++ Builder 3 开发应用程序

第四部分 数据库及其他

第一部分 C++ Builder 3 概览

第一章 C++ Builder 3 的出现

对于 120 万 C++程序员来说，Borland C++ Builder 3 无疑是一次革命性的飞跃。威力无比的 C++语言和快捷便利的 RAD 环境相结合，孕育出一个程序开发者梦寐以求的编程工具。在这个编程环境中，强劲高效的 C++代码和随心所欲的可视化开发得到了前所未有的和谐统一。在这一章中，我们将粗略地巡视这一令人心醉神迷的软件的出现背景，并大致地窥视它的风貌。

1.1 软件开发工具的发展

计算机发展的历史同时也是软件发展的历史。从 1946 年的 ENIAC 出现开始，就有了计算机程序。然而，软件产业的真正兴起却是在高级语言的广泛应用之后。PC 和 DOS 的出现为计算机软件的开发打开了全新的天地。回顾软件开发工具的发展，真可谓是白驹过隙、沧海桑田。

1.1.1 从 DOS 到 Windows 95：开发理念的变迁

众所周知，DOS 曾经是最为成功的个人计算机操作系统，几乎可以说是一个时代的标志。不过它采用的是命令行的用户界面，这足以让许多使用者望而生畏。百余条操作命令掌管了计算机的运作，计算机的一切行动几乎都在使用者一字一句的催促下进行。那时，计算机程序必须相对直接地对系统进行管理，曾一度是软件屈从于硬件的时代。最初的程序设计差不多成了一门艺术，人们所关心的，是如何用最少的资源，实现最多的功能。随着软硬件的发展，软件开发的效率开始受到广泛的关注，然而，程序的立足点，仍然是直接对机器的控制。

后来，Windows 出现了，并随着 3.0 版的推出，令人耳目一新的 GUI(图形用户接口，Graphic User Interface)设计擦去了阻隔在用户和机器之间的纱帘，使计算机以一种更为平易亲切的面貌出现在使用者面前，无数人被它漂亮的交互式界面所折服。从此，Windows 成为 Microsoft 倾尽心血培育的一代天骄，它的目标不仅仅是用户界面上的革新，而是要推动一种计算机软件开发和应用理念上的革命。这种理念的一个关键是要实现比 DOS 更好的有序性。这种有序性的根本是要进一步地明确开发使用各个层次上的分工，把人们从往往需要为多余的事情操心的困境中解脱出来，从而更集中地致力于自己所关注的环节。

Windows 的有序性和 DOS 的无序性最明显的对比莫过于图形的编程了。大多数 DOS 下

的应用程序不得不采取直接读写硬件的方式进行各种低级的、超常规的操作。这种情况在Windows 中得到了彻底的改变，其应用程序接口(API)的核心就是要使用高级编程，这是一次工程革新。在这个新的世界里，重要的是硬件的性能和创新，而不是硬件的一致性。同时，事件驱动的程序模型，使软件摆脱了顺序性的框框，更灵活地适应于新的用户、新的要求。

Windows 95 的出现，刮起了又一阵旋风。软件工程的革新被它卓越的性能进一步地推向深入，Windows 95 力图割断与 DOS 千丝万缕的联系，干干净净地走向新的编程时代。这时，快速高效并且精彩得令人眼花缭乱的软件开发成为可能。众多美妙的工具出现了，我们捧在手上的 C++ Builder，就是其中一颗耀眼的新星。

1.1.2 RAD 的产生

快速应用程序开发(RAD, Rapid Application Development)是随着新一代软件开发环境的出现而产生的一个术语。在这个编程世界里，程序员使用直观可视的开发手段，而不再需要为 Windows 繁琐的环境设置费尽苦心。其基本原理，是服务程序的大量封装继承。换句话说，Windows 将种种低级艰深的操作包装在 API 中，RAD 则进一步将 Windows 的开发接口包装成直接的建筑材料，程序员所要做的，不过是要将这些烧好的砖瓦，搭建成高楼大厦。

在搭建过程中，界面的设计几乎已经成了完全的拼装，在此过程中使用的拖放技术(Drag-and-Drop)着实是赏心悦目的。在用户界面上，已经没有多少代码需要写了，美观与舒适的设计是首要的问题。于是，程序员就能够更为完全地投入到软件功能的内部实现之中去。

1.1.3 BC,VC,VB 和 Delphi

BC(Borland C++)和 VC(Visual C++)不是 RAD 工具，之所以提到它们，无非是因为它们凭借 C++的八面威风而一度在 GUI 软件开发中占据着举足轻重的地位。

第一个真正意义上的 RAD 工具是 Microsoft 的 VB(Visual Basic)。这个开山之作十分出色并且取得了巨大的成功，其最大的功绩是帮助一般人开辟了一个崭新的编程世界，开始引入了窗体、控件等概念。但其不足从一开始便暴露得和它的优点一样明显，那就是语言设计上的不完善所带来的程序的草率和疏松，甚至会促使编程人员养成不好的编程习惯。此外，最初的伪编译方式大大影响了代码运行的速度。

几乎是针锋相对的，Borland 公司推出了 Delphi，它具有 VB 的多数优点，同时以 Object Pascal 语言克服了 VB 的致命不足。由于 Borland 拥有历史上最为成功的 Pascal 语言工具，在 Pascal 渐渐成为一种被弃之不顾的语言的时候，Borland 将其改头换面，使其焕发生机，构建出一种新的语言标准。Delphi 出尽了风头。

然而，C++的魔力是神奇的，百万程序员执着地拥抱着它，任凭 VB 和 Delphi 如何散发着诱惑，他们始终不愿放下手中千锤百炼的 C++神剑，走进 RAD 的乐土。

1.1.4 Inprise 公司的历程

Borland 公司成立于 80 年代上半期，十余年中，它曾一度闯入全球十大软件开发商的行列。基于 DOS 的 Turbo 系列软件给无数人留下了美好的记忆。更名 Inprise 后，它似乎将注意力转向了企业。但它已经无可辩驳地在 RAD 和其他开发工具领域，都占据了不容忽视的地位。

在 C++ Builder 3 推出之前，Inprise 先后拥有了下列相关产品：

- Turbo C++ Visual Edition
- Turbo C++ 4.5
- Borland Delphi 1.0
- Borland C++ 4.5
- Borland C++ 5.0 & 5.01
- Borland Delphi 2.0
- Turbo C++ Suite
- Borland C++ 5.02
- C++ Builder 1.0
- Borland Delphi 3.0

1998 年 2 月，C++ Builder 3 水到渠成、瓜熟蒂落。从上面的产品中也可以看出，C++ Builder 很显著地带有 Borland C++和 Delphi 的烙印，或者可以说，它是二者结合的产物。

1.1.5　拖放技术(Drag-and-Drop)

拖放技术可以说是 RAD 界面设计中的关键，但可喜的是这是一项很容易掌握的技术，因为它的操作方法非常直观。你所要做的，不过是把现成的组件，用鼠标拉到你正在设计的窗体上，它就能为你所用了。在今后的学习中我们将会看到，这一功能是如何的强大和方便。用这种方式，你可以构建起绝大多数的用户界面，而且是又快又漂亮。

Borland C++ Builder 3 包含了先进的高度集成的可视化工具，以利于集成组件和开发代码。适时控制和代码维护是 C++开发中的重要方面，因此，可视化开发工具就必须致力于将程序分解成干净的、可同步更改的代码块。此外，在 C++ Builder 3 中，源程序的任何改变会同时为可视化设计工具和代码所接受，即两方面自动同步更新。更具体一点，举例来说，当你使用拖放技术改变一个窗体的外观设计时，有关这个窗体的数据和代码会自动地随之变动，而不用你再进行手工更改(见图 1-1)。相对于从前的单独更新的 CASE 开发工具和向导来说，这种同步更新方式不失为一个飞跃。因为你从此不必再担心可视化设计和代码设计的不统一可能带来的任何麻烦。

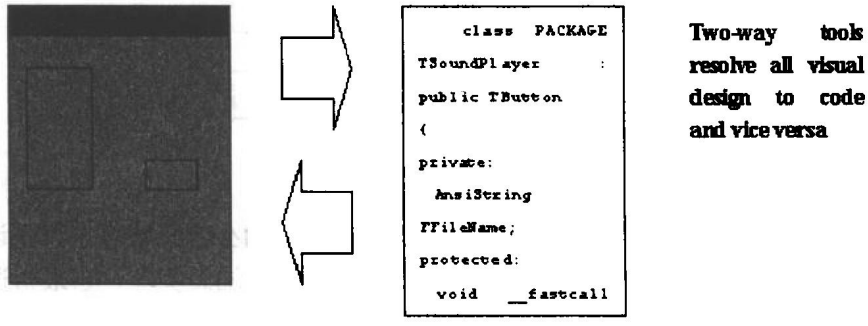


图 1-1　同步更新

1.1.6 Inprise 开发工具的传统特点

熟悉 Turbo 系列的读者都知道,Borland 公司开发工具的成功,很大程度上归功于它奇快的编译速度和极高的的代码效率。这一优势在 RAD 中得到了保持,Delphi 的编译速度能让开发者心满意足。由于C++语言的自身特点,C++ Builder 的编译速度必然要慢于 Delphi,但只要你的机器不是老得太过分,也足以让你满意了。在 Windows 95 下,一不留神就会写出臃肿的代码,这一危险在我们的 C++ Builder 中同样存在,但你会发现,这种情况并不会经常出现,尤其是在你有了一定的经验之后。

舒适的帮助系统曾经是 Turbo 系列的一大特色,可惜的是 Delphi 和 C++ Builder 的帮助系统并不出色,相反地,却是常为人所指责的缺陷。我们期待 Inprise 今后有所改进,而目前,解决这一问题的办法,似乎只有多动脑筋多琢磨了。

1.2 C++的特色

C++为人所垂青不是没有理由的,这一语言的独特魅力使得人们格外地关注这样一个问题:C++ Builder 中的C++是否还是纯粹的C++?它是否依旧具有强大灵活的控制力?是否依旧具有高度的可移植性?是否……,C++ Builder 3 可以自豪地回答:是的,C++的优越性,已经全部溶解在 RAD 中了。C++ Builder 3 和 Borland C++ 5.02 一样的强劲刚勇。

1.2.1 从C语言到C++

图 1-2 示出了 C 语言的发展史:

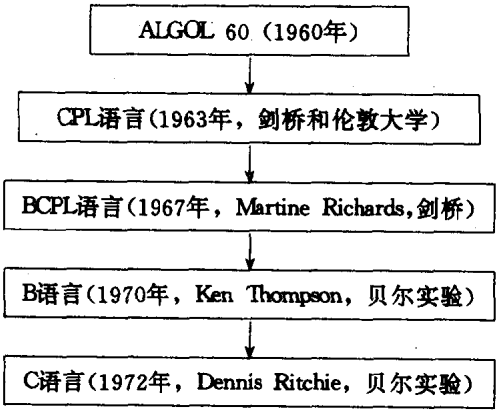


图 1-2 C语言的由来

C 语言风行于世已有 20 多年的历史了。它是国际上公认的少数几种通用程序设计语言之一,最早由 D. M. Ritchie 在贝尔实验室中为编写 UNIX 操作系统而实现。它最初适用于作为系统描述语言,即用来编写系统软件,但在应用软件的编写中也大有用武之地。

C++的名字表示了从 C 语言的进化,增量符号"++"明确地指示了其扩充方向。C++语言保持了 C 语言高效灵活、易于理解、可移植性好等优点,并包含了所有面向对象的特征。它在 1980 年产生于贝尔实验室,1983 年开始推广到外界,并具有了 ANSI 标准,在不长的时

间内,C++便完全取代了 C。

1.2.2 C++:永恒的魅力

相对于其他高级语言,C++的优点主要在于:

- 基本组成元素紧凑、简洁,具有编写结构良好的程序所需要的各种控制结构;
- 提供了某些接近于汇编语言的功能以及与地址密切相关的指针和有关运算符,能够用高级语言的形式对系统进行某些低级操作,具有强有力的软硬件接口以及低层系统设计能力;
- 预处理程序和预处理命令提高了程序的可读性和可移植性,也给程序调试带来了便利;
- 生成代码的质量高,运行效率仅比汇编语言代码慢 10%到 20%;
- 精致的面向对象特性,具有十分自然的可扩充性,可方便地构造出模拟现实问题的实体和操作。

风风火火的 C++时代令我们记忆犹新,它养育出数百万的程序员,如今,他们中还有着相当杰出的一部分,仍在用他们娴熟的专业技巧,对抗着"合者千万"的"下里巴人"——RAD。

1.2.3 Borland C++和 Visual C++

这是 Windows 下 C++语言的两位杰出代表,Microsoft 和 Borland 各自赢得了一批拥护者。它们的优点和缺点都是相似的。它们同样都是强有力的 C++开发工具,同样提供了极为出色的 Windows 程序库。然而,它们在可视化方面却都没能取得真正的成功。它们一直在努力,不断在改进,但界面的管理仍然需要消耗掉大量的精力。因此,这不是任何人都可以使用的工具,需要经过相对漫长的学习过程才能够登堂入室。不过,一旦你掌握了它们,那种优越感和成就感必然是无可比拟的。

1.2.4 C++进入 RAD 的瓶颈

使用 RAD 技术似乎会很容易地冲击 C++的原有优点。另一方面,RAD 软件作为一种新事物被推向市场,成功与否具有一定的不确定性,同时,开发者需对其使用的语言进行大面积的改动。因此,开发商自然更愿意选取一种不那么被人关注的高级语言来作为 RAD 语言基础。PASCAL 是如此,BASIC 更是如此。况且,这两种语言在 RAD 中也确实被改得面目全非了。

相反地,C++却不是一种可以随意修改扩充的语言,因为有太多的人使用着它,人们对它有着太高的要求。开发 C++的 RAD 不仅需要勇气,还需要审慎的思考分析。

1.2.5 C++Builder 的突破

我们援引两位著名分析员的话来说明这一问题。

Steve Garone,国际数据公司研究员:

"Borland 公司长期以来保持着这样一个传统,就是为专业程序员提供革命性的软件开发工具。Borland C++ Builder 3 发扬了这一传统,这是一个强大的 C++应用程序和系统软件开发工具,它增强了产品的开发效率和运行性能。此外,Borland 公司新的 ActiveForm 技术为 C++程序员开发 Internet 应用程序提供了足够的便利。"

John Milam,NationsBank 高级程序员,系统分析员:

"Borland C++ Builder 3 实现了 C++程序员的一个梦想,它提供了一个应用程序或是系统软件开发者所能够要求得到的一切。从低层跟踪调试、高级工程管理、MFC、OWL 和微软的 ATL 到强大的 Internet 开发工具和单步 ActiveX 创建,我现在已经能够在有限的时间和预算限制下轻松自如地开发和配置 Windows 的工程了。"

1.3 应用程序类库

前面曾经提到过 Windows 的应用程序接口(API),可以说,API 使得在 Windows 下编程成为可能,但对大规模软件生产来说,API 显然是不能满足要求的。

1.3.1 程序库、类库和组件库

当程序规模不断增大时,要想避免重复的调试过程,一个办法就是把程序分成多个可控制的片段。于是程序员就很自然地采用了"分而制之"的策略,这就是从前所谓的"结构化"和现在的"封装",这就有了程序库(Library)。

把数据局部地限制于某一个特定函数的办法并不能满足程序设计的要求,当程序变得很复杂的时候,程序员经常发现其他函数也需要访问这些数据,他们希望把程序的各个部分,包括数据和代码,都集中在一起形成一个对象,而不是简单的函数。

可以为这种情况打一个比喻作为例子。一个对象,犹如一台电视机,程序员并不关心它的内部数据和方法,他们只知道这是一台可以输出节目信号的电视机,其他的都不重要。这也正是程序设计技术已经达到的水平,程序员把数据和函数组装在一起,于是程序的复杂的内部细节对用户来说就是不可知的了。利用这种方法可以把一个很大并且难以控制的程序分解成许多相对简单、符合人们思维方式的工具和资源。C++语言支持这种分解的办法是类(Class),于是有了类库(Class Library)。

在使用类库之前,Windows 环境下编程工作的困难程度可想而知,即使只不过是在屏幕上生成一个简单的窗口也往往要编写几十行的代码,用到好几个文件,还需要对 Windows 有深入的了解。引入类库之后,情况便焕然一新了。有时,可能根本就不用编写几行代码,就能够顺利地产生多种标准窗口。

组件(Component)则是随着 RAD 而出现的,是完成指定 RAD 编程任务的对象。我们所要讨论的 VCL 组件是 Object Pascal 对象的封装。组件库(Component Library),将是我们的重点研究对象。

1.3.2 使用类库的效率

一定意义上,使用类库所编写的程序要比 C 所写的程序臃肿迟钝。如果用 C 语言编写的简单的 Windows 程序大小为 75KB,那么用类库编写的具备相同功能的程序大小可能会是 200KB。但这是就最差情况而言的,并不总是这样。对于较小的程序,用 C 语言和用类库编写的应用程序大小之间的差别是非常明显的。但随着程序越来越大、越来越复杂,它们在大小上的差别也越来越小。

1.3.3 MFC,OWL 和 VCL

微软公司的 MFC(Microsoft Foundation Classes)最初是作为 Visual C++软件包的一部分提供的,它是 Windows API 的简单包装。对于原先的 C 语言程序员来说,这不失为一个优点,但是,类库的目的是要让用户不需知道他们所不用知道的东西,从这一点看来 MFC 做得并不成功。另外,MFC 相对比较容易学习(仅仅是相对而已,使用基本类库进行 Windows 编程始终不是一件简单的事)。然而,MFC 的最大优势在于,它是微软公司的产品,它可以随着 Windows 的发展得到最及时地更新。

Borland 公司的 OWL(Object-Windows Library)从 1.0 版发展到了 5.0 版。它曾经是作为独立产品单独出售的,后来被集成进了 Borland C++。早期版本中存在着不少问题,但还是吸引了大量的用户。在后来的版本中添加了 OLE(对象链接和嵌入)以及 OCF(对象组件类库,Object Components Framework)支持。它可用于 16 位或 32 位的程序设计,甚至可以在 16 位程序中仿真 32 位定制控制。成熟的版本是非常适用于面向对象编程的,也完全遵守 OOP 设计准则。它的不足在于,它为封装 Windows 环境做了大量工作,于是就变得相对复杂,难以学习。

应当指出,OWL 和 MFC 分别作为 Borland C++和 OWL 的核心部分,各自打开了一片天地,并且在长达 10 年的对抗中,没有分出明显的胜负。

可视组件库(VCL,Visual Component Library)是 1995 年同 Delphi 一起诞生的革命性产品,是用 Object Pascal 进行 Windows 编程时使用的类库。需要指出的是,它并不与 MFC 和 OWL 兼容,其核心已经面目全非,因为它是通过属性、事件和方法等概念设计出来的。读者也许已经注意到,C++ Builder 的核心部分正是 VCL,尽管语言不同,但事实上我们不用担心,因为 VCL 在 C++ Builder 中运行得同样出色。

1.3.4 为什么选择 VCL?

随着 C++ Builder 3 的全面出击,Inprise 可能将停止 OWL 前进的步伐,因为只有 VCL 才代表了真正的趋势。程序开发将不再是普通人可望不可即的工作了。尽管 VCL 是针对 Delphi 开发的,但它在设计观念上已经具备了作为一个通用 RAD 类库的全部条件。

1.3.5 Delphi 对 C++Builder 3 的支持

Delphi 的成功无疑是 C++ Builder 的坚强后盾,C++ Builder 3 充分利用了这一资源,从多方面仿真源于 Delphi 的特性,使人们相信这两个环境可以兼容,并具有相同的魔力。

- 编译器:用一组新的有关 Delphi 的关键字来支持内部的属性和内部的封装,如_published,_property 等等;
- 组件:支持 Delphi 的 DCUs,可安装使用 Delphi 2.0 的组件,并可将其源代码安装于组件调色板中,以供使用和派生新类;
- 数组:接纳了 Delphi 的开放数组,即数组的大小和元素类型在编译中保持不确定,有一些辅助宏可支持这一功能,这些宏包括:ARRAYOFCONST,OPENARRAY,EXISTINGARRAY,SLICE 等;
- 类:增加了与 Delphi 兼容的 AnsiString,Variant,ShortString,TDateTime,Set 等;
- 开发环境:请比较 Delphi 4 和 C++ Builder 3 的集成开发环境(IDE),如图 1-3 和图

1-4 所示；

• 数据库，Internet，ActiveX，包：所有这些，C++ Builder 3 都是通过和 Delphi 相同的途径实现的，兼容和移植相当完美。

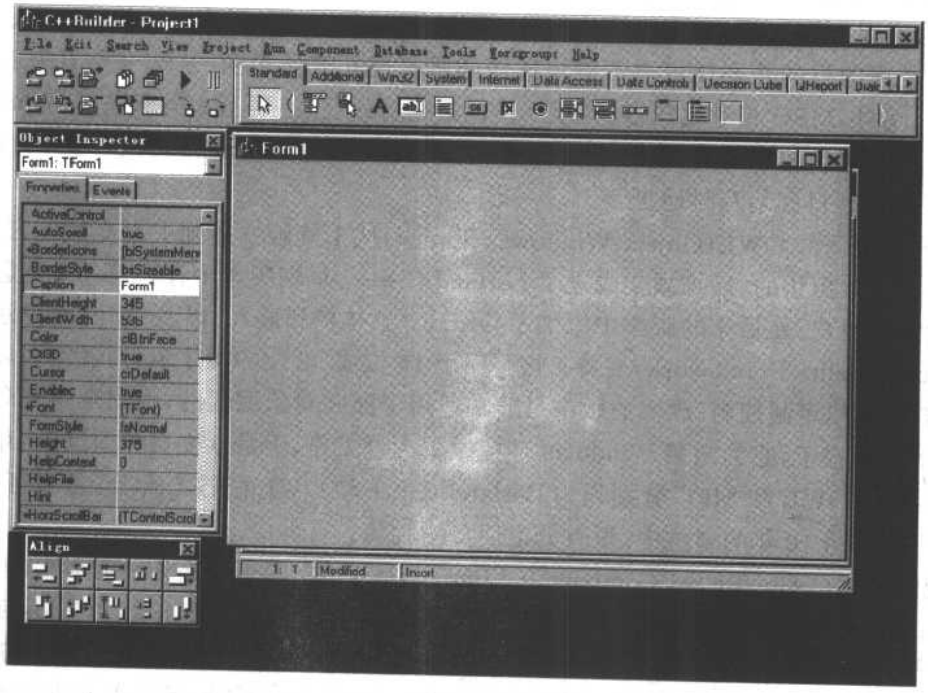


图 1-3 C++ Builder 3 的 IDE

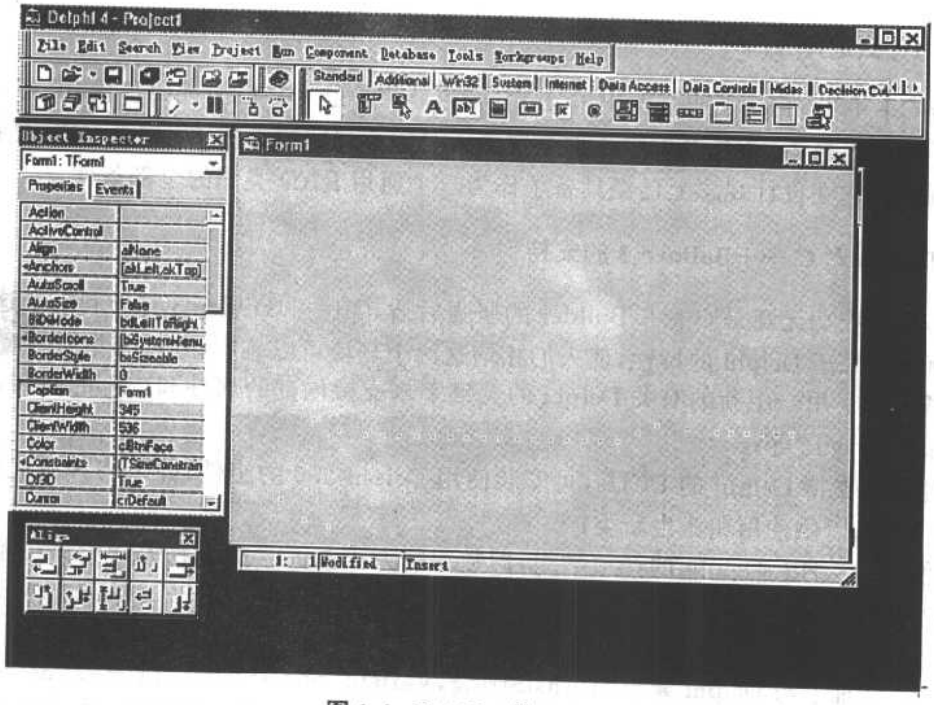


图 1-4 Delphi 4 的 IDE

1.4 C++ Builder 3 的功能和特性

现在，让我们开始体会 C++ Builder 3 的强大功能和特性。

1.4.1 能力(Capacity)、速度(Productivity)和效率(Efficiency)

让我们用 Inprise 公司自己提供的一个图形来展示 C++ Builder 的能力，如图 1-5 所示。在这个图形里，我们清楚地看到，Inprise 所孜孜以求的，就是要将 100%纯正的 C++语言和当今优秀的可视化开发工具的优点，组合成一个里程碑式的尖端产品。

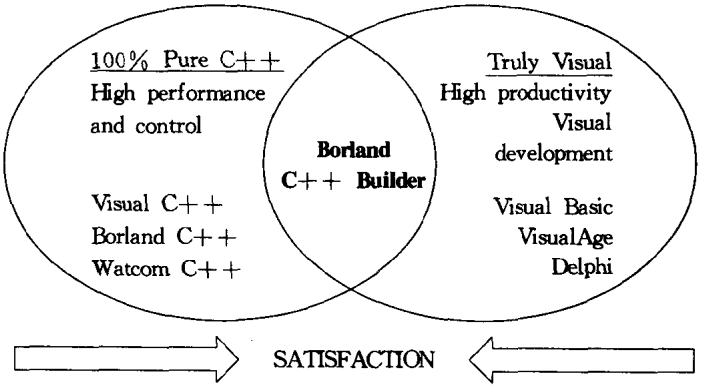


图 1-5 C++ Builder 3 的开发能力

1.4.2 以往 C++工具的不足

在 C++ Builder 之前的 Windows 环境下的 C++开发工具，包括 Borland C++和 Visual C++，由于 C++语言自身的限制，都存在着多方面的不足。这些不足使得它们难以和 VB，Delphi 等大众宠儿相提并论。若不是因为它们是 C++，恐怕其生命力就早已终结了。

总体说来，以往的 C++工具主要的缺陷在于：

- 开发周期过长；
- 可视化与软件内核的关联设计过于复杂；
- 可视化设计工具集成度太低。

1.4.3 程序员对 C++ RAD 的基本要求

程序员钟爱 C++的原因是多方面的，为了这种原因他们宁愿承受更长的开发周期和更繁重的工作。对于一个基于 C++的 RAD，他们至少会提出如下的要求：

- 保持 C++的强大、高效(代码紧凑、运行速度快)、严谨，以及全面的系统低层管理能力；
- 完全的可移植性；
- 完善强壮的调试、组织能力。