



Borland C++ Builder 6

应用开发技术解析

程展鹏 编著



清华大学出版社

Borland C++ Builder 6 应用开发技术解析

程展鹏 编著

清华大学出版社

北 京

内 容 简 介

本书详细介绍使用 C++ Builder 6 进行程序开发的各个方面,尤其是 C++ Builder 6 中的 DataSnap、WebSnap 和 BizSnap 三大新技术。同时,给出大量实例程序,帮助读者理解和掌握相关的技术知识;本书还对和 C++ Builder 6 相关的未来技术发展作了展望和讨论,可以让读者从较高的层次来理解软件技术的发展。

本书第 1~4 章介绍使用 C++ Builder 6 进行软件开发的基础和背景知识;第 5 章介绍大量实用的开发技术;第 6~7 章介绍数据库开发技术;第 8~9 章介绍组件编写和线程使用;第 10~11 章介绍和 Internet 相关的开发技术,包括 Internet 组件、WebSnap 和 Web Services;第 12 章介绍利用 C++ Builder 6 和 Kylix 3 进行跨平台开发的内容。

本书附带一张光盘,内容为书中全部示例程序的源代码。

本书内容全面、实例丰富、文字流畅,有助于开发者快速掌握 C++ Builder 6 的核心开发技术。本书适合有一定程序设计经验的读者阅读。

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。

图书在版编目(CIP)数据

Borland C++ Builder 6 应用开发技术解析/程展鹏编著. —北京:清华大学出版社,2003

ISBN 7-302-07423-2

I. B… II. 程… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2003)第092898号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

责任编辑:马 丽

封面设计:钱 诚

版式设计:俞小红

印 刷 者:北京鑫丰华彩印有限公司

装 订 者:三河市金元装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:25.5 字数:586千字

版 次:2003年10月第1版 2003年10月第1次印刷

书 号:ISBN 7-302-07423-2/TP·5485

印 数:1~5000

定 价:36.00元(附光盘一张)

前 言

对于已经有较长程序开发经验的人来说，Borland 公司和 C/C++ 语言都具有特别的意义。Borland 公司在 20 年的发展历史中，为软件开发者提供了许多非常优秀的开发工具，包括 Turbo PASCAL、Turbo C/C++、Borland C/C++、Delphi 等。而 C/C++ 语言以它优美简练的语法和灵活强大的功能，曾经吸引了绝大多数程序员学习和使用。C++ Builder 作为 Borland 公司推出的 C/C++ 开发工具，无疑满足了许多程序员的梦想。

首先，它提供了强大的 RAD (Rapid Application Development, 快速应用程序开发) 功能，使得开发者可以节约大量设计界面和实现程序底层支持功能的时间，将精力主要用于程序核心逻辑的设计上。Borland C++ Builder 利用了 Borland 公司划时代的产品——Delphi 的技术，包括其类库 VCL (Visual Component Library)。VCL 体系结构完善而精致，功能丰富，易于使用，并且完美地体现了面向对象的设计思想。对于许多开发者来说，他们既喜欢 Delphi 的强大易用，又不愿意舍弃所钟爱的 C/C++ 语言，C++ Builder 正好满足了他们的需要。

其次，C++ Builder 依靠 C/C++ 语言和 Borland 公司的技术，提供了高级开发的能力。C++ Builder 在提供简单易用特性的同时，并没有为开发者增加限制。开发者可以利用 C++ Builder 来完成许多高级应用开发，扩展空间几乎是无限的。从许多优秀的第三方 VCL 组件中，可以看出 C++ Builder 强大的开发能力。所以，C++ Builder 不仅能够让初学者快速开始应用程序的开发，也可以让富有经验的开发者用来开发功能强大的软件。

最后，C++ Builder 提供了最广泛的支持技术。目前，开发者已经可以使用 C++ Builder 和 Kylix 3 来开发跨平台 (Windows/Linux) 的应用程序。相信随着 Linux 向桌面操作系统的发展，Kylix 能够得到更广泛的使用。另外，对于主流的数据库开发、Web 应用、Web Services 等，C++ Builder 都有很好的支持。未来，随着 Borland 公司 ALM (Application Lifecycle Management, 应用周期管理) 计划的发展，C++ Builder 中会加入更多的软件工程支持，包括建模、团队开发、配置管理、测试、优化等环节，都会无缝集成到优秀的软件工具中，配合 C++ Builder，可以在整个软件开发周期中为开发团队中的各个成员 (分析师、架构师、开发人员、测试人员、分发团队和管理者) 提供完备的支持。例如，将 UML 建模集成到 C++ Builder 中，提供 UML 和 C++ Builder 程序代码之间的双向生成，可以大大加快软件项目的开发过程。

笔者从 1.0 版本开始使用 C++ Builder，一直到最新的 6.0 版本，深深感觉到使用 C++ Builder 的方便之处。C++ Builder 非常容易上手，但是在它易于操作的界面之下，蕴含着无比强大的功能。对于各种层次的开发者，在大部分软件开发领域，C++ Builder 都能够提供强大的支持。Borland 公司总能为 C++ Builder 加入最新的技术，来满足开发者的需要，包括多层结构数据库开发 (MIDAS/DataSnap)、Web 开发 (WebBroker、InternetExpress 和



WebSnap)、Web Services (BizSnap)、跨平台开发 (Kylix/CLX) 等。从 Borland 公司公布的信息来看, Borland 公司未来会持续发展 C++ Builder, 并且提供更多更强大的功能。另外, 在本书完稿之时, Borland 公司已经发布了支持 Microsoft .Net Framework 开发的 C# Builder, 相信这又能为开发者们带来一个优秀的开发工具。

本书共有 12 章, 除了第 1、2 章用于介绍背景知识和基本技术之外, 其余部分均按照不同的功能和技术来划分章节。建议初学者按照章节顺序阅读, 对于已经有 C++ Builder 开发经验的读者, 可以直接阅读自己感兴趣的章节。

本书包含的示例程序源代码均包含在随书光盘中, 并且都经过仔细的调试。除了与数据库相关的程序, 需要读者进行必要的环境设置外, 大部分示例程序代码应该都可以在读者的电脑中直接编译和运行。本书示例程序使用的数据库文件也包含在随书光盘中。

本书由程展鹏编著。参加本书编写工作的人员还有程琨、李藜、周韬、纪文秀、何干等。尽管笔者花费了大量的时间来保证本书内容的正确性, 但是仍然可能存在疏漏之处, 请读者谅解, 并希望读者能够与笔者进行交流, 共同提高。

作 者

2003 年 7 月于西夏

目 录

第 1 章 了解 Borland C++ Builder	1
1.1 Borland 公司	1
1.2 BCB 的发展	2
1.2.1 Turbo 时代	2
1.2.2 Borland C++	3
1.2.3 Delphi/BCB	3
1.2.4 Kylix	4
1.3 BCB 和其他编程工具的比较	5
1.3.1 BCB 的定位	5
1.3.2 BCB 与 Delphi、PB 等 RAD 工具的区别	5
1.3.3 BCB 与 VC 的区别	6
1.4 学习 BCB 所需要的资源	6
1.4.1 基本知识	6
1.4.2 BCB6 软件包	7
1.4.3 其他软件工具	7
1.4.4 Internet	8
1.4.5 学习过程	8
1.5 BCB6 的更新	9
第 2 章 BCB6 基础	10
2.1 熟悉 BCB6 的界面	10
2.1.1 主窗口	11
2.1.2 窗体设计	11
2.1.3 对象检查器 (Object Inspector)	12
2.1.4 代码编写与类浏览器	12
2.1.5 对象树状层次观察器 (Object TreeView)	14
2.1.6 项目管理器 (Project Manager)	14
2.1.7 程序调试窗口 (Debug Windows)	14
2.1.8 辅助窗口	16
2.1.9 定制和保存桌面	17
2.2 C/C++ 语言概览	17
2.2.1 变量	17
2.2.2 流程控制语句	20



2.2.3	函数.....	22
2.2.4	面向对象编程.....	24
2.3	Windows 的运行机制.....	27
2.3.1	Windows 的元素——窗体.....	27
2.3.2	Windows 的动力——消息.....	28
2.4	第一个程序——Hello, world!.....	29
2.4.1	BCB6 一般编程步骤.....	29
2.4.2	Hello, world!	29
第 3 章	BCB6 常用组件使用.....	33
3.1	界面控件.....	33
3.1.1	界面框架组件.....	35
3.1.2	输入输出控件.....	41
3.2	对话框控件.....	61
3.2.1	文件操作对话框.....	61
3.2.2	字体 (FontDialog) 与颜色 (ColorDialog) 设置对话框.....	63
3.2.3	打印设置 (PrintSetupDialog) 与打印 (PrintDialog) 对话框.....	64
3.2.4	查找 (FindDialog) 与替换 (ReplaceDialog) 对话框.....	64
3.3	定时器控件.....	64
3.4	重要的不可视组件与 BCB 类.....	65
3.4.1	字符串类.....	65
3.4.2	时间类 TDateTime.....	68
3.4.3	INI 文件类与注册表类.....	70
第 4 章	图形与多媒体.....	76
4.1	图形编程.....	76
4.1.1	显示简单的图形.....	76
4.1.2	使用图形文件.....	77
4.1.3	使用画布 (TCanvas) 对象.....	78
4.1.4	动画实现.....	82
4.2	多媒体编程.....	88
4.3	DirectX.....	91
第 5 章	应用程序开发技巧.....	94
5.1	编写 DLL.....	94
5.2	异常处理.....	103
5.3	发布程序.....	109
5.3.1	包.....	109
5.3.2	安装程序的生成.....	111
5.4	一些编程技巧.....	129
5.4.1	使用 TrayIcon (系统托盘)	129



5.4.2	如何防止程序被多次运行	129
5.4.3	文件操作相关函数	131
5.4.4	消息的使用	132
5.4.5	鼠标拖放	136
5.4.6	使用剪贴板	140
第 6 章	数据库程序开发基础	142
6.1	BCB6 中的数据库程序开发技术	142
6.1.1	数据库技术的发展	142
6.1.2	Borland 数据库访问技术概述	143
6.1.3	选择合适的数据库访问技术	144
6.1.4	BCB6 中的数据库开发技术架构	145
6.1.5	BCB6 中的数据库开发组件	146
6.2	创建数据库	147
6.2.1	使用 Database Desktop 创建数据库表	149
6.2.2	定义数据库表的结构	149
6.2.3	保存数据库表和设置数据库别名	151
6.3	使用数据库向导	153
6.3.1	生成单一数据库表程序	153
6.3.2	生成使用“主/明细”表的应用程序	158
6.4	基本数据库相关组件的使用	163
6.4.1	TTable 的使用	163
6.4.2	DataSource 的使用	177
6.4.3	数据感知控件的使用	177
6.5	使用 SQL 连接	183
6.5.1	SQL 语言概述	184
6.5.2	使用 TQuery 组件	185
6.5.3	和 SQL 相关的数据库辅助工具	189
6.6	连接网络数据库	193
6.6.1	连接网络数据库	193
6.6.2	事务	195
6.6.3	存储过程	197
第 7 章	数据库程序开发高级技术	201
7.1	多层数据库技术	201
7.1.1	多层数据库技术概述	201
7.1.2	DataSnap	205
7.1.3	编写应用服务器程序	206
7.1.4	编写客户端程序	209
7.1.5	多层数据库应用程序的实现技术	211



7.1.6	客户端数据集 (TClientDataSet) 的应用	230
7.2	dbExpress	245
7.2.1	dbExpress 概述	245
7.2.2	dbExpress 组件	247
7.2.3	建立 dbExpress 连接	248
7.2.4	开始第一个 dbExpress 程序	252
7.2.5	dbExpress 与 DataSnap	255
7.2.6	dbExpress 组件的使用	258
7.2.7	dbExpress 使用总结	259
7.3	ADO	260
7.3.1	BCB6 中的 ADO 组件	260
7.3.2	ADO 组件特性及使用	261
7.3.3	实例	266
第 8 章	编写 VCL 组件	269
8.1	VCL 组件编写技术基础	269
8.1.1	VCL 结构	269
8.1.2	属性、事件与方法	270
8.1.3	属性与事件的语法	272
8.1.4	使用 VCL 组件父类的成员	274
8.2	编写 VCL 组件的实例	276
8.2.1	创建 VCL 组件框架	276
8.2.2	编写属性	279
8.2.3	测试 VCL 组件	282
8.2.4	安装组件	284
8.2.5	消息映射的处理	286
第 9 章	使用线程	288
9.1	使用线程对象	288
9.1.1	创建线程	289
9.1.2	使用线程	292
9.2	线程同步	294
9.2.1	与主 VCL 线程同步	294
9.2.2	锁定对象	295
9.2.3	使用临界区域	295
9.2.4	使用多重读、独占写的同步器	300
9.2.5	使用线程对象的 WaitFor 方法	301
9.2.6	使用事件	302
9.2.7	使用互斥 (Mutex) 对象	303
9.3	调试线程	305



9.4 线程使用中应当注意的问题	305
第 10 章 Internet 开发技术	307
10.1 FastNet 系列 Internet 控件	307
10.2 Indy	308
10.2.1 什么是 Indy	308
10.2.2 控件使用	308
10.2.3 实例	309
10.3 使用 Web Browser	314
10.4 WebSnap 技术	315
10.4.1 WebSnap 实例	317
10.4.2 用户管理	340
第 11 章 编写 Web Services 应用程序	346
11.1 Web Services 概述	346
11.1.1 Web Services 的概念	346
11.1.2 Web Services 结构	349
11.1.3 BCB6 中的 BizSnap	351
11.2 使用 BCB6 开发 Web Services 应用程序	354
11.2.1 编写第一个 Web Services 程序	354
11.2.2 调用 Web Services	363
11.2.3 利用 WAD 来了解 Web Services 的调用过程	367
11.2.4 利用 Web Services 传递复杂类型数据	368
11.2.5 调用 Internet 上的 Web Services	375
11.3 SOAP 与数据库	379
11.3.1 建立 SOAP 远程数据模块	379
11.3.2 利用 SOAP 连接数据库应用服务器	381
第 12 章 跨平台编程 (Windows/Linux)	384
12.1 Kylix	386
12.1.1 Kylix 的由来	386
12.1.2 Kylix 技术	387
12.1.3 Kylix、Kylix2 与 Kylix3	387
12.2 CLX	387
12.2.1 CLX 结构	387
12.2.2 CLX 程序	388
12.2.3 编写跨平台程序	389
12.2.4 跨平台应用程序开发实例	390

第 1 章 了解 Borland C++ Builder

Borland C++ Builder (以下简称 BCB) 是美国 Borland 公司推出的 Windows (6.0 版本中加入了 CLX 控件支持, 因此也正式将 BCB 扩展到了 Linux 窗口开发领域中。) 可视化开发工具, 目前最新版本为 6.0。顾名思义, BCB 是以 C/C++ 语言为核心的编程工具。在桌面程序开发和数据库开发等领域中, BCB 得到了广泛的应用。本章将对 BCB 以及其开发公司 Borland 做一些背景介绍; 最后, 将描述读者阅读本书应当注意的几个方面。

1.1 Borland 公司

Borland 公司于 1983 年由 Philippe Kahn 在美国加利福尼亚州 Scotts Valley 创立, 同时加入公司的还有编程开发工具领域的大师级人物 Anders Hejlsberg。从推出的第一个产品 Turbo PASCAL 1.0 开始, Borland 公司不断生产出许多高品质和富有革新精神的编程工具。其革命性的 Turbo PASCAL, Turbo C 2.0, Turbo C++ 2.0, Borland C++ 3.0/4.0, Delphi 1.0 等都给编程工具领域带来了深刻和长远的影响, 也奠定了其在编程工具领域第一独立开发商的地位。

但是, 在 1995 年, 由于 Borland 的产品在市场上接连失利, 尤其是主力的 Borland C++ 被微软的 Visual C++ 打得溃不成军, 加上 Borland 在数据库等领域盲目扩张的失败, Borland 公司陷入低潮, 其创始人 Philippe Kahn 不得不辞职并离开了 Borland。1996 年, 新总裁 Yocam 上任, 为了适应当时 Internet 迅速发展的形势, 也为了避免在 Windows 桌面程序开发这一市场上与微软直接冲突, 全面调整公司战略, 转向企业电子商务开发工具提供商, 也因此将公司名称改为 Inprise (代表 Integrate-to-enterprise)。这一改名的决定今天看来十分愚蠢, 首先是丢失了 Borland 这一金字招牌, Inprise 公司为了保持老客户, 其产品仍然以 Borland 命名, 造成了公司品牌与产品品牌的冲突; 其次是一个简单的名称改变并不能代表其战略转向, 也未能在企业电子商务领域获得有效的品牌认知。Yocam 虽然在企业级用户市场为 Borland 打开了一片天地, 但是他独断专行的作风和大规模裁员的做法都伤害了 Borland 原有一大批优秀程序员的感情。不少业内顶尖的程序员都离开了 Borland, 包括曾经领导开发出 Turbo PASCAL 和 Delphi 的 Anders Hejlsberg。在伤筋动骨后, Borland 的经营状况却仍然没有起色。

Yocam 之后, 于 1999 年 4 月上任的总裁 Dale L. Fuller, 于 2001 年 1 月, 将公司名称从 Inprise 改回 Borland, 令市场、消费者、软件开发业界和公司员工莫不拍手叫好。Fuller 曾在苹果公司复兴中起过重要作用, 此次在 Borland 任职后, 他还为 Borland 做了一件贡献



巨大的事，那就是运用法律手段从微软手中要了一亿多美元的专利权利金。这笔钱大大缓解了 Borland 的财务压力，支撑 Borland 推出了许多至关重要的新产品，加上 Fuller 其他一系列的改革措施，终于使得 Borland 走出了困境。目前 Borland 公司运营状况良好，盈利稳步增长。但是 Fuller 在推行其管理政策时，解雇了大批不接受他管理理念的程序员，Borland 公司再次遭受技术人员流失的打击，对 Borland 前景的影响还未可知。

1.2 BCB 的发展

1.2.1 Turbo 时代

1983 年 11 月，创立不久的 Borland 推出了第一个产品——Turbo PASCAL 1.0。Turbo PASCAL 1.0 具有许多革命性的特点，比如它首次在 PC 机中实现了编程工具 IDE(Integrated Development Environment，集成开发环境)，将源代码的编辑、编译和运行集成在一个环境中，大大方便了程序员的开发。Turbo PASCAL 1.0 只有几十千字节大小，却在当时性能不佳的 PC 机上(当时还是使用 Intel 8088 的 IBM PC 的时代，内存都是以千字节为单位的。)编译速度飞快，有点类似后来在 386 上跑得很流畅的 DOOM，极大地发挥了软硬件的潜力。这些在业界领先的优点，使得 Turbo PASCAL 1.0 极为畅销。当时号称全世界 90% 的 PC 程序员都接触过它，包括很多不使用 PASCAL 的程序员，可以说 Turbo PASCAL 系列为 PASCAL 语言的普及做出了巨大贡献。Anders Hejlsberg 作为 Turbo PASCAL 的主设计师，也成为编程界顶尖的人物。当然 Turbo PASCAL 1.0 最重要的作用就是使得 Borland 公司得以迅速发展，并终于在日后成为 PC 软件开发工具领域的霸主。另外，在苹果机上，还出现过一款由 UCSD 开发的 PASCAL 产品，它甚至包括了一个操作系统(UCSD P-System)，不需要 Apple DOS 3.3 的支持。它的特点是将 PASCAL 代码编译成统一的 P 代码(p-code)，然后在 UCSD P-System 上运行，而不依赖具体的硬件环境。在当时，它可以支持 6502(Apple II 的 CPU)，Intel 8080(许多 PC 的 CPU，当然是在 IBM PC 出现之前)，Z80(Intel 8080 的天才设计师自创公司开发的 Intel 8080 兼容产品，完全兼容 Intel 8080，价格却非常便宜，因此曾经广为流行)和 PDP-11(DEC 最畅销的微机 PDP-11 的 CPU)。但是也许是作为一个教学产品，后来就再没有它的消息了。

之后，Borland 除了 SideKick 这款实用工具产品，将主力全部放在了编程工具的开发上，并都统一使用了“Turbo”这一品牌，比如 Turbo PASCAL、Turbo C 和 Turbo BASIC。除了 Turbo PASCAL 继续其无人可抵的辉煌外，Turbo C 也凭借其方便的 IDE 在市场上十分畅销。记得笔者当时在中学里主要使用的是 Turbo PASCAL，同时出于兴趣，也使用过 Turbo C。令我惊讶的是，直到现在，许多人在学校里还是使用 Turbo C 2.0 来完成他们的 C 语言课程作业。

之后，随着 C++ 的风行，Borland 将 Turbo C 2.0 升级为 Turbo C/C++ 2.0。这时，Borland 的核心产品已经渐渐从 PASCAL 开发工具转向了 C/C++ 开发工具。大约是 1993 年的夏天，



笔者在中国科技大学第一次见到了 Turbo C/C++ 2.0, 用 8 张软盘复制了下来。记得最清楚的就是大家都在热烈讨论 Turbo C/C++ 2.0 中最主要的新特性——支持面向对象编程。之后, Turbo PASCAL 5.5 引入了 Object PASCAL, 也开始正式支持面向对象编程。Object PASCAL 虽然不是由 Borland 公司提出, 但是它后来的发展则完全是由 Borland 公司主宰的。或者说, 没有了 Turbo PASCAL/Delphi, 也许 PASCAL 只能在课堂中找到使用者了。

在 Turbo 时代, Turbo PASCAL 几乎完全占据了 PASCAL 开发工具领域, 为 Borland 贡献了大量收入; Turbo C/C++ 系列虽然也十分成功, 但真正做到市场领先, 还是在 Borland C++ 推出后。

1.2.2 Borland C++

1992 年, Borland 将 Turbo C/C++ 系列正式升级为 Borland C++ 3.0。从这时起, Borland 奠定了它在编程工具界的霸主地位, 并且正式将公司核心产品从 PASCAL 系列转移到了 C/C++ 系列。Borland C/C++ 3.1 是一款非常成熟和成功的产品, 直到现在, 它还被用于一些软件的开发。

有关 Borland C++ 与 Microsoft Visual C++ 以及其他 C++ 产品的竞争历史, 参见台湾李维先生的文章《我的回忆和有趣的故事》以及《Borland 传奇》, 这里就不班门弄斧了。

1.2.3 Delphi/BCB

1995 年 2 月 14 日, Borland 公司推出了石破天惊的新一代 Windows RAD (Rapid Application Development) 工具——Delphi。记得当时我刚在 Windows 3.1 中装上 Delphi 1.0, 靠着以前使用 Turbo PASCAL 和 Visual Basic 的经验试着写了一个小程序, 结果立刻被它的品质所折服。Delphi 并不是第一个, 甚至也不是 Windows 平台上第一个 RAD 工具, 在它之前有 Visual Basic 这一背景强大的产品。但是 Delphi 具有许多超越以前产品的特性, 比如它优秀的 IDE (继承了 Turbo 系列的优点) 使用非常方便; 它的代码执行速度远远超过了 Visual Basic, 接近 C/C++ 编译代码的效率, 同时具有惊人的编译速度, 不像 C/C++ 都避免不了由于头文件带来的编译速度缓慢; 它还引入了 VCL (Visual Component Library) 作为基本的控件库标准, VCL 不但具有 VBX 和 ActiveX 的基本优点, 还具有代码精简、执行效率高和可编译成执行文件的特点; 此外, Delphi 凭借强大的 PASCAL 语言支持 (Anders Hejlsberg 经过多年开发 Turbo PASCAL, 在 PASCAL 语言上具有十分精深的造诣, 也因此挖掘出 PASCAL 语言的不少潜力), 在功能上十分强劲, 除了驱动程序编写外, 几乎可以完成 Windows 平台下所有程序的开发。最后这一点十分重要, 因为它第一次给程序员一个完全可视化的 Windows 快速开发环境, 同时又不以牺牲性能和功能为代价。要知道 Visual Basic 一直被当作 Bill Gates 的玩具, 好看好玩但是不管用, 其弱点在于代码执行效率低下和功能薄弱。

Borland 公司在 C++ 领域惨败后, 面临灭顶之灾。但是 Anders Hejlsberg, 还有 PASCAL 语言, 像当初赋予 Borland 以初期发展的生命力, 再一次支撑了 Borland。正如 Lotus 公司凭借 Notes 得以生存, 在微软巨大的阴影之下, Borland 依靠 Delphi 开辟了一条新生之路,



暂时收缩产品线，慢慢积蓄力量，等待东山再起的机会。

在使用过 Delphi 后，笔者的第一个感觉就是如果有类似 Delphi 的 C/C++ 快速开发工具该多好，因为毕竟使用了 C/C++ 多年，习惯了其代码简洁和给程序员的极大自由度。另外，由于 Windows API 完全使用 C 接口，使用 Delphi 还需要转换其 API 的函数接口，比较麻烦。相信使用过 Delphi 的众多 C/C++ 程序员也有此想法。这一来自开发者的需求如此明显，Borland 公司也不可能不了解。但是也许是由于 C++ 领域的惨败而导致的对微软的恐惧，也许是人才和财力的缺乏，直到 1996 年，Borland 公司才推出了 Delphi 的 C++ 版本——Borland C++ Builder 1.0。Borland C++ Builder 采用了 Delphi 的 VCL 作为其基本控件库标准，通过远指针调用来直接使用 Delphi 的 VCL，但是也还需要进行一些语言转换的调整，这也是为什么 BCB 总比 Delphi 晚些推出的原因。同样也是基于这个原因，在 Delphi 3.0 推出后，为了保持和 Delphi 的同步，BCB 直接从 1.0 版本升级到了 3.0 版本，从此便紧跟 Delphi 的版本变化而变化。对于相同版本的 Delphi 和 BCB 来说，BCB 中的部分功能会得到改进，相应的控件版本要略新一些。

1.2.4 Kylix

大约在 2000 年底到 2001 年初，在 Borland 公司（当时还叫 Inprise）的网站上开始进行一项面向全球软件开发者的调查，内容包括询问被调查者是否有兴趣在 Linux 中使用类似 Delphi 的快速开发工具以及会使用这一工具开发哪些应用等。而在这之前，互联网上就已经有大量关于 Borland 正在进行一项秘密的开发计划的传言，这项计划很有可能就是 Linux 环境中的 Delphi！当笔者看到 Borland 的调查时，就明白 Borland 真的要开始这一传说中的计划了，这无疑会让无数软件开发者为之一激动和期盼。实际上，在 1999 年 9 月 28 日，Inprise/Borland 便已经要开始开发一个“支持 C、C++ 和 Delphi 的高性能 Linux 应用程序开发环境”，也就是 Kylix 的雏形。2001 年 1 月，Borland 正式启动代号为“Kylix Project”的开发计划，很多人也将它非正式地称为“Delphi for Linux”。

2001 年 7 月，在全球众多软件开发者的关注之下，Borland 正式在 Linux 社区发布 Kylix。它具有和 Delphi 几乎一样的集成开发环境，使用的是类似 VCL 的 CLX 控件库标准，在第一个版本中只支持 Pascal 语言。

之后，配合 Delphi 6 的发布，2001 年 10 月 Kylix 2 推出，主要是提供了 Linux 下 Web Service 的开发。而 2002 年 8 月，Kylix 3 发布，其中增加了重要的特性——支持 C++ 语言开发。随着 Kylix 3 的推出，在 BCB6 中增加的 CLX 支持不再只是摆设，开发者终于可以使用熟悉的 C++ 语言和 BCB 开发环境在 Linux 下进行应用程序开发了。

Kylix 系列产品的推出，也标志着 Borland 正式转向跨平台编程工具提供商。Borland 的目标是支持最流行的 Windows 和 Linux，支持统一的 Web Service 标准（包括 SOAP 和 XML），同时完全支持 Microsoft .NET。



1.3 BCB 和其他编程工具的比较

经常会看见网络上或者专业报刊杂志上有人询问该学习哪种编程工具，或者是疑惑流行的 Visual C++（简称 VC）、Visual Basic（简称 VB）、Delphi、BCB 等有何不同，答案也是五花八门，夹杂了不少回答者个人的偏好。对待这个问题，比较理性和统一的看法是，任何一种流行的编程工具都有其独到之处，学习哪一种都不会白白辛苦。但是要根据学习者本身的条件和目标，来选择适合自己的编程工具。

BCB 有众多优点，但并不是说它可以完全代替其他编程工具。下面先说明 BCB 的定位及它与其他编程工具的区别。了解了这些知识，读者再根据自己的情况，自然可以做出明智的选择。

1.3.1 BCB 的定位

BCB 定位在快速应用开发（RAD）工具层面上，也就是说，BCB 主要用来开发基于 Windows 桌面（目前也包括 Linux 下的桌面环境）的应用程序。由于 BCB 具备强大的可视化开发能力，并且含有众多可以直接使用的 VCL 控件，因此可以使得开发者专注于逻辑代码设计，而不需要花太多精力在图形界面设计上。另外，和 Delphi 一样，BCB 中特别增强了数据库连接和开发能力，这就使得 BCB 在数据库桌面程序开发上具备一定的优势。

所谓 Windows 应用程序，一般指由用户运行并带有操作界面（窗口或者命令行），并运行在 ring 3 级别的程序。驱动程序恰恰不具备这些特点。因此使用 BCB 来开发系统级程序（一般为设备驱动程序）是非常困难的。

1.3.2 BCB 与 Delphi、PB 等 RAD 工具的区别

BCB 与 Delphi 系出同门，而且使用几乎一样的集成开发环境（IDE）和 VCL 组件架构。它们最大的区别就是使用语言的不同。因此对编程语言的偏好往往决定了开发者使用 BCB 或者 Delphi。

另外，由于 Windows 是用 C 语言写成，Windows API 提供的都是 C 调用接口，许多的例程也大部分使用 C/C++ 语言。因此，使用 BCB 将给 Windows API 的调用带来方便，同时也可以利用大量的示例代码资源。当然，对一个熟练的 Delphi 程序员来说，转换 Windows API 是完全没有问题的。

熟练的 C/C++ 程序员会毫不犹豫地转向 BCB，他们的 C/C++ 基础也可以帮助他们了解 VCL 及 Object PASCAL。但是对于初学者来说，如果完全没有编程语言基础，那么直接学习 C 语言将是很困难的。这时，PASCAL 严谨的语法将带来方便，初学者能够比较容易学习 Delphi，并且不至于养成一些不好的编程习惯，而这些习惯是初学者使用 C 语言时很容易染上的。另外，Delphi 得到了 Borland 公司更好的支持，其资源也更为丰富。



BCB 与 Delphi 是密不可分的, 不仅因为它们使用同一个应用程序架构 (VCL), 而且因为它们在语言层面上的交融性。Delphi 程序员必须要了解 Windows API 及许多基于 C/C++ 的 SDK, 才能实现一些特定的功能; 而 BCB 程序员也必须了解 Object PASCAL, 能够阅读 VCL 源代码, 才有可能深入了解 VCL, 提高编程水平。

1.3.3 BCB 与 VC 的区别

BCB 与 VC 同样使用 C/C++ 语言来进行开发, 其最大的区别在于使用的应用程序架构 (Application Framework)。BCB 使用了 Delphi 中的 VCL, 而 VC 使用了 MFC。MFC 直接来自 Windows API, 早期的 Windows 开发者会感到 MFC 几乎和 Windows API 是一一对应的。这样的好处就是 MFC 紧密贴合 Windows 系统机制, 用做系统级开发十分便利; 缺点就是 MFC 过于庞杂烦琐, 初学者接触起来困难重重。VCL 成功地封装了基本 Windows API, 并且提供了强大的可视化开发能力。VC 程序员一直都耗费了大量精力来进行图形界面的设计, 而 BCB 程序员只需要花少很多的精力就可以设计好程序的全部界面。

从封装的层次来看, BCB 采用的 VCL 应该是比 VC 的 MFC 要先进, 但是也有几个问题阻碍了 VCL 超越 MFC:

第一, MFC 是由微软开发的, 而伴随每次 Windows 的升级和附加 SDK 的推出, 总是可以使用 VC 来顺利进行开发。使用 BCB 则要多费一些力气。

第二, VC 对 ANSI C/C++ 的支持要比 BCB 好。在实际测试中, 一些跨平台的 C/C++ 代码很容易在 VC 下进行编译, 在 BCB 下则很难进行。需要指出的是, 使用 BCB IDE 环境很难编译一般的跨平台的 C/C++ 程序, 需要使用 BCC 命令行编译工具。跨平台编译问题也许是 VC 使用较为广泛的缘故, 大部分程序开发者在 Windows 平台下都会考虑使用 VC 来进行编译。在不同的测试中, VC6 和 BCB6 对 ANSI C/C++ 的兼容性各有缺陷, 这主要是为了提高性能和支持一些特性所导致的结果。

第三, 一个很实际的问题, 有关 VC 的资源远远多于有关 BCB 的, 甚至 BCB 的资源和 Delphi 比起来都十分稀少。所以, 一个优秀的 BCB 程序员需要了解 VC 和 Delphi! 这样才可以利用大量的编程资源来提升自己。

对于初学者来说, BCB 非常容易上手。但是要想提高自己, 学习 VC 和 BCB 都需要很多的努力。

1.4 学习 BCB 所需要的资源

1.4.1 基本知识

学习 BCB 当然首先需要了解一些 C/C++ 语言知识。如果用过 Turbo C++ 来编写过程序, 那么所具备编程语言知识已经基本足够使用 BCB 了。如果没有学习过 C/C++ 语言, 那么建议采用以下参考书籍:



- ◆ 谭浩强,《C 语言教程》
- ◆ 《C++ primer》,国内已有中文版,侯捷译的第三版
- ◆ Bjarne Stroustrup,《the C++ programming language》,国内有中文译本,北大裘宗燕译,《C++程序设计语言(特别版)》

当然,要想用好 BCB,还是应该对 C/C++ 有较深入的理解。深入理解 C/C++ 的最佳途径就是在使用中学习,通过实践掌握理论的精髓。读者使用本书,完整地学习 BCB 后,相信在熟悉 BCB 使用的同时,也提高了使用 C/C++ 的能力。

此外,基本的 Windows 操作和对 Windows 工作机制的简单了解也是必要的。对于那些掌握 Windows 操作但是不了解 Windows 的读者,本书中会穿插介绍有关知识,不需要再专门学习。

1.4.2 BCB6 软件包

Borland 公司推出的 BCB6 有四种版本,基本延续了以前的标准,那就是个人版(Personal)、专业版(Professional)和企业版(Enterprise),以及可以免费下载的试用版(Trial Version)。前面三个版本都是正式发售的版本,只是在所包含内容上有所区别,而高级版本包括了较低级版本的内容。

个人版除了基本的集成开发环境(IDE)之外,只具有基础的 Windows 程序开发能力,甚至不能用来开发数据库程序,只包含了基本的 85 个 VCL 控件。

专业版增加了创建 Windows 组件(COM、COM+和 ActiveX),编写 Web Service 客户端(Borland 统一将其开发工具中支持 Web Service 的功能命名为“BizSnap”),编写 Web 应用程序,CLX 支持(类似 VCL,可以跨平台使用于 Windows 和 Linux 的应用程序框架),增强的项目管理(使用通用脚本语言编写用户定制的编译连接过程)和基本的数据库支持(包括 Paradox, dBASE, FoxPro, Microsoft Access, InterBase 和 MySQL,此外还支持微软 ADO)等功能。相应地,专业版中包括了至少 225 个 VCL 控件。

企业版则进一步增加了扩展的 Web Service 开发(包括 Web Service 服务器端开发和其他 BizSnap 功能),Web 应用程序开发(Borland 将该功能命名为“WebSnap”),CORBA 支持和大型数据库支持(Oracle, DB2, Microsoft SQL, Informix, Sybase 等)等功能,同时包括了 300 多个 VCL 控件。

读者如果无法使用正式版本的 BCB6,可以从 Borland 公司网站(www.borland.com)下载试用版,以做学习之用。

1.4.3 其他软件工具

要进行 Windows 开发,MSDN Library 是必不可少的。要使用 Windows 基本 SDK 和扩展 SDK,在 MSDN 中都可以很详细地查到有关信息,另外大量的技术文档会对开发过程非常有益。

UltraEdit,非常好的文本编辑器,可以识别 HTML、C/C++、ASP、Java 等代码,还可