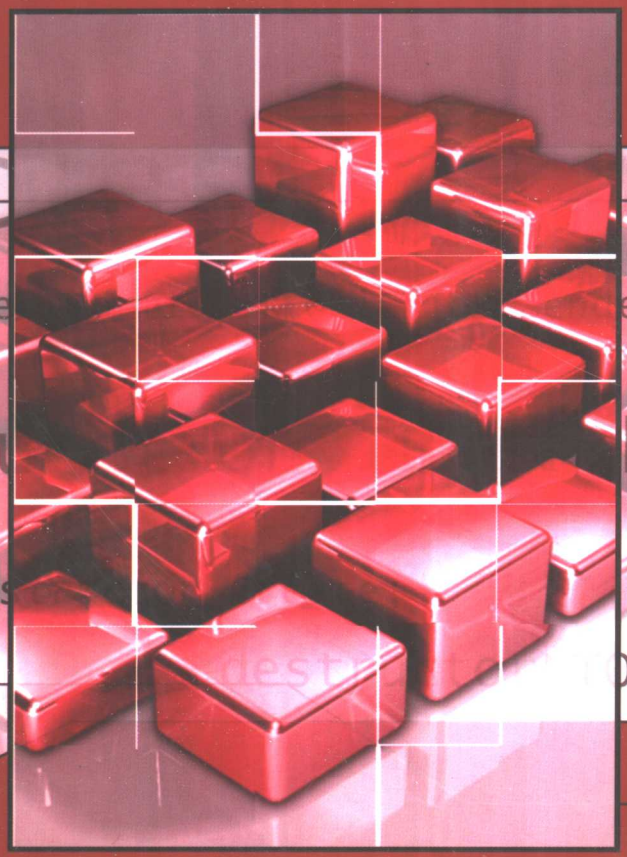
Broadview
WWW.BROADVIEW.COM.CN

全世界惟一深入讨论VCL Framework的技术书籍

Inside 深入核心 VCL架构剖析

掌握Framework之设计架构　　学习VCL之精湛实现技巧



李维 著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONIC INDUSTRY
http://www.phei.com.cn

Borland In-Depth Series\Borland 大系

Inside 深入核心 VCL架构剖析

李 维 著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书包括 10 个章节，从最基础的 Windows 操作系统原理讲起，回顾 Windows Framework 和 VCL Framework 的发展历史，介绍 Object Pascal 在 OO 方面对 VCL 的支持，描述 VCL Framework 与 Windows 消息体系的集成，列举 VCL 组件与 Windows 组件的结合，探究基于接口的程序设计，指明 VCL Framework 设计 COM 架构的方法，探讨 VCL Framework 的永续储存，还以一章的篇幅专门讨论了 VCL Framework 中的设计模式，最后对下一代 VCL Framework——VCL.NET 作了前瞻式的研究。

单看目录就可以知道，这本书不但涉及 VCL Framework 本身，还旁及 Windows Framework、COM、设计模式等相关技术。读者从中获得的，也不仅只是 VCL 架构知识，更会在整个阅读和实作过程中极大地拓宽自己的开发眼界，形成在系统设计方面的大局观，追寻大师级的 Framework 设计思路，提升整体开发素质。

图书在版编目(CIP)数据

深入核心——VCL 架构剖析 / 李维著. —北京：电子工业出版社，2004.1
（Borland In-Depth Series\Borland 大系）
ISBN 7-5053-9489-4

Ⅰ.深... Ⅱ.李... Ⅲ.软件工具—程序设计 Ⅳ.TP311.56
中国版本图书馆 CIP 数据核字（2003）第 117294 号

责任编辑：周 筠 方 舟
技术编辑：韩 磊
印 刷 者：北京市增富印刷有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
经 销：各地新华书店
开 本：787×980 1/16 印张：44 字数：960 千字
印 次：2004 年 1 月第 1 次印刷
印 数：8000 册 定价：80.00 元(含光盘 1 张)

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：(010)68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

旁征博引　丝丝入扣

在与李维同事之前，我早已是他忠实的读者。

IT 领域技术变迁甚快，在众多令人目眩的新技术中，李维成功扮演了导航者的角色。从李维的书中，我可以察觉这位吸收新知速度异于常人的技术作家已读过了哪些国外著作为基础，再酝酿出独有的经验与见解，看完李维的书后，我即知道自己该先看些书，以补所学不足，亦了解到书中有多少内容，是言人所未言之精华。

《Inside VCL》与李维过去的著作有着很大的不同，此书并非针对“应用”为出发点撰写，而是探索庞大的 VCL Framework，协助读者了解 Delphi 易学易用的背后，VCL 组件模型是如何运作的，为什么如此设计。最重要的是，本书让读者知道，这样的设计与技巧将如何运用于读者日后的工作中。过去 *Anders Hejlsberg* 以及 Borland 研发团队在 VCL Framework 投入相当心力，今日读者可通过李维的文字，抽丝剥茧地了解顶尖软件研发团队之设计结晶，也解答了许多我们知其然却不知其所以然的 Delphi 用法与语法。

《Inside VCL》共计十章，前八章涵盖了 VCL Framework 中最值得深入探讨的几项主题，包括 VCL Framework 与窗口消息间如何交互、VCL 组件和窗口控件间如何结合、Delphi 之 Interface 运作机制、COM 设计架构以及持久化等主题，内容由浅入深，循序渐进。在探索 VCL 设计的同时，旁征博引了许多 Java 与 Microsoft .NET Framework 设计方式，让读者了解其中之差异与奥妙。第九章中，李维首度在著作中安排章节讨论 Design Pattern 这项课题，并呈现 VCL Framework 中所运用之各式 Pattern，近年来敏捷方法论（Agile Methodology）对于软件设计中误用与滥用 Pattern 提出了许多省思，经由经历市场验证的 VCL Framework，让读者一窥如何正确地使用 Pattern，别具意义。当本书问世的同时，Delphi 产品即将迈入 Microsoft .NET 平台，最后一章中介绍了 VCL.NET 的设计，承先启后，引导读者进一步了解 Delphi 的未来。

由于共事多年，在旁的我也目睹了这本书诞生之曲折。李维有心撰写这本书至少可回溯至三年前，当时有感于台湾市场的生态，以及出版商忧心曲高和寡的态度，其中几度打算放弃此书，也多次重写书中章节。今日能看到此书顺利出版，深知其得来不易，亦盼此书能为华人 IT 技术著作市场，立下了新的里程碑。

李匡正

2003 年 10 月于台北

序

Delphi 已经推出了七个版本，在未来也会持续地推出新的版本，许多人可能也已经使用 Delphi 许多年，并且开发了各种不同的应用系统，但是不管我们使用了 Delphi 多久，我们真的已经了解 Delphi 而且发挥 Delphi 十成的功能了吗？

从 Delphi 1 推出以来，在每一个版本都加入了许多新的功能，融合的软件技术也一直在增加之中。从 RAD、Flat-File 数据库功能、VCL 组件，一直到主从架构、Web、COM/COM+、MIDAS、多层分布式应用系统，到现在的 SOAP/Web Service、dbExpress，DataSnap 等技术，程序员们不断地学习和使用新的技术，以便用来开发新的应用系统或是增加程序员个人的附加价值，以求在信息领域能够更上一层楼。不过这些技术大都是属于“应用类”。但是在这些技术之外 Delphi 亦拥有优美的 Object Pascal 程序语言，以及有若宝山般的 VCL Framework。Object Pascal 提供了典雅的面向对象程序语言的功能以及 Pascal 传统的严谨语法，值得程序员通过它来学习面向对象的观念和技巧。而 VCL Framework 更蕴藏了大量的软件技术宝藏，它充分地使用了 Object Pascal 的特性。巧妙的软件实现技巧，深入的语言/编译器技术，动态对象生命周期的管理以及对设计模式(Design Pattern)的结合，是许多 Borland 工程师多年软件开发智慧和技巧的结晶。

VCL Framework 这座宝藏公平地呈现在每一个 Delphi 程序员之前这么多年，然而大多数的我们，很久以来大都只是提取 VCL Framework 表面之一角就可以快速使用 Delphi 开发和完成我们的应用系统，只有极少数的程序员真的尝试开启 VCL Framework 宝藏的大门，进入另外一个宏伟的软件工程大殿，接受从未经历过的高等软件技术的洗礼。因此，长久以来 VCL 的深度知识也被封锁在少数程序员的圈子中，这些极为有用的软件技巧自然也只被少数程序员所拥有。

在笔者工作的这么多年中也不时地追踪 VCL Framework，因此学习到许多宝贵的软件知识，也解决了许多问题。笔者认识的许多朋友也或多或少都研究过 VCL Framework，许多人也因此成为佼佼者，在工作上表现非常杰出。笔者和这些朋友都知道一个要诀，那就是当有解不开的麻烦时看看 VCL Framework 大概就可以解决了；笔者也经常在网络上看到一些高手说学 Delphi 根本不需要看什么书籍，只要看看 Delphi 的在线帮助和 VCL Framework 源程序就可以解决所有的问题。虽然笔者不绝对赞成这句话，因为在属于 Delphi“应用层”的技术方面仍然需要经验才能够运用得很好，否则只是属于会用而已，但是笔者赞成：在偏向 Windows 程序设计、系统功能方面，如果 Delphi 程序员能够了解 VCL Framework，那么的确是非常容易解决这些问题的，不过前提是 Delphi 程序员必须能够掌握 VCL Framework，简单地说，也就是必须看得懂 VCL Framework 的源程序并且掌握其中的设计诀窍。

然而我们必须承认，在这么多的 Delphi 程序员中仍然只有少数人拥有掌握 VCL Framework 的优势。这是为什么呢？因为要了解 VCL Framework，需要具备许多基础的知识。如果没有其他人或是书籍来引导，那么就只有那些在软件开发上非常有经验的人才能够一窥全貌。

数年前笔者就知道当时 Delphi Informant 的主编表示有兴趣撰写一本 Inside VCL 书籍，准备剖析 VCL 的架构。当时笔者非常期待这本书，因为当时笔者也想了解 VCL Framework，以更上一层楼。不过笔者等了又等，到了现在仍然不见那本书的出现。为什么在 Delphi 领域中一直没有讨论 VCL Framework 的书籍出现呢？这是有许多原因的，最重要的两项因素可能是：一来这种书籍不易撰写，需要作者投入极大的心力和时间，此外也考验了作者本身在软件方面的素养，不是一般作者愿意或是能够撰写的；二来可能是因为出版商对于这种书籍的市场销量有考量，出版这种属于进阶技术的书籍通常需要比出版入门书籍做更多的考量。撰写 VCL Framework 的另外一个挑战是：VCL Framework 仍然在快速的演进之中，不时地有新的技术被加入，有时甚至是架构移植到新平台上去，不像其它的一些 Framework 几乎是处于停止发展的状态。因此，写这种书的作者必须不时地重新审视已经撰写的内容，看看 VCL Framework 是否又发生了变化？这对于书籍的作者而言是非常严峻的挑战。

笔者许多年来一直在写一些有关 VCL 的文章，但是这些内容从没有发表过，这是因为笔者在苦等不到 Delphi Informant 主编的书籍之后，便不自量力地想自己写一本有关 VCL 的书籍，但是由于笔者本人时间上的限制以及出版商的考量，因此后来就将计划暂搁了。直到最近，两岸的数家出版社接连都表达了出版此书的意愿之后，笔者才郑重考虑想要完成这本书，至此笔者开始认真地设计和收集这本书的内容。笔者知道，要完成这本书是一条漫漫长路，不过笔者决定一点一滴地试着完成这本书，希望 2003 年能够让这本计划多年的书籍问市，以达成笔者多年的心愿。

读者阅读本书时，需要知道本书并不是讨论特定的应用技术，也不会指导读者如何开发特定的应用系统。本书主要讨论的内容是 VCL Framework 设计和实现的技术，内容包含了：为什么需要 VCL Framework；VCL Framework 的设计思想；VCL Framework 的设计架构；VCL Framework 的实现技术；VCL Framework 的演进；VCL Framework 使用的软件技术；VCL Framework 使用和结合的 Design Pattern 以及 VCL Framework 的未来——VCL.NET。在这些被讨论的内容中，读者将可充分看到 VCL Framework 的各种实现技巧；此外笔者也会试着和读者一起探讨许多 VCL Framework 设计背后的理论和原因，试着让读者了解在设计一个 Framework 时牵涉到的东西，让读者不但能够知道 How，也能够知道 Why。当读者经过了 VCL Framework 的洗礼之后，相信在软件思想和实现方面将有脱胎换骨的感觉。

笔者要谢谢所有帮助本书出版的朋友，没有这些朋友的催促，这本书可能永无面市的机会。当然更要谢谢许多读者多年来对于笔者的支持，最后希望本书的内容真的能够帮助读者进入 VCL Framework 的软件知识宝山。在读者能够从其中学习到受用无穷的软件技术之后，这些软件技术在读者未来的信息领域生涯的发展中将呈现出其无价的珍贵。更重要的是，经此一窥 VCL Framework 的内部实现秘密，将会让读者至少在思想上能够拉近和世界一流软件工程师的距离，当然，读者如果在日常的实现中能够善用学习到的 VCL Framework 软件技术，那么也就表明读者正朝向一流软件人才的方向努力奋进。

李维

2003 年 10 月于台北，新店

目　录

It was there that we are coming from!

1

回到从前

"经过了这么多年发展的 VCL 到底是什么？ 我们为什么需要 VCL？VCL 和 Windows 程序设计有什么关系？Delphi 不使用 VCL 可以写 Windows 应用程序吗？VCL 和其它的 Windows Framework 有什么不同？Windows 上到底有多少 Framework？Framework 之间又有什么不同？"

"这么多的问题使我终于开始产生困惑，我真的了解 Windows 程序设计，真的了解 VCL 吗？"

本章讨论重点

- 多任务操作系统环境
- Windows 消息系统(Windows Messaging System)
- 系统消息队列(System Message Queue)和应用程序消息队列(Application Queue)
- 典型的 Windows 应用程序
- 典型应用程序设计的缺点
- 为什么我们需要 Windows Framework
- Windows 平台上 Framework 的发展简史

在您阅读本书时，笔者并不知道您的背景。不知道您是从 Windows SDK 一路走来的 Windows 程序员，就和笔者一样；或是属于比较新的世代，直接从使用 Delphi RAD 工具而进入 Windows 程序设计的程序员；甚至是纯粹喜欢设计程序的爱好者，不过笔者可以确定的一点便是您一定想要更了解 Borland 的 VCL (Visual Component Library) Framework，这本书的内容就是专门讨论 VCL 是如何设计和实现的技术书籍。

本书的目的就在于让您掌握 VCL 的精髓，让您了解 VCL 是如何运作的，如何影响您使用 Delphi 设计出来的 Windows 应用程序。更重要的是在您阅读完本书之后，您便可以掌握 VCL Framework 的设计原理，从此 Windows 的消息机制以及 VCL 如何封装和结合 Windows 消息机制将不再神秘，您也即将从 RAD 的程序员进入系统程序设计的领域，您也将会发现

Object Pascal 和 C/C++一样拥有强大的设计和表达能力(Expressing Power)，使用 Object Pascal 设计的 VCL 和 Windows 本身贴近的程度是和使用 C/C++程序语言设计的 Framework 没有什么分别的。

虽然本书即将讨论的内容非常丰富而且具有一定的深度，不过只要我们掌握学习的窍门，许多的事情和原理都非常的直觉，即使是如同 VCL 和 OWL/MFC 等的 Windows Framework，要了解和掌握它们也不困难。问题是如何掌握进入的窍门并且从何处开始着手。

1-1 角色扮演

笔者本身非常喜欢玩角色扮演游戏(Role Playing Game)，因为这类游戏总会让笔者宛如身历其境，也很容易融入游戏的精神并且了解为什么游戏的剧本会如此安排。角色扮演游戏这种精神也非常适合使用在许多的场景(Scenario)中，特别是在学习一些我们并不熟悉的东西时，如果我们能经由设身处地的方式来思考，那么便会发现许多的事情在观念的理解上并不困难，即使在实现上可能会有一定的难度。笔者现在就准备使用这种方式来讨论 Windows 程序的运作方式。

假设现在我们在设计一个多任务的执行环境，就像 Windows 一样。在这个环境中同时会有许多的应用程序在执行，在应用程序执行的过程中这个执行环境会不断地发生各种事件，例如用户可能点击了鼠标、从键盘输入了字符或是不同的应用程序之间可能产生相互沟通的事件。那么我们要设计什么样的架构来顺利地执行这类型的执行环境呢？

1-1-1 如何设计和驱动多任务执行环境

由于我们不知道每一个应用程序在何时会发生/触发事件，以及会触发什么事件，更不知道用户何时会点击鼠标、按下键盘。因此我们无法设计一个大型循环(Loop)，以便在这个循环中不断地检查每一个应用程序是否触发了特定的事件，如用户是否点击了鼠标等。而且即使我们能够在这个大循环中检查到所有可能发生的事件，这仍然是很愚蠢的设计，因为这不但会让整个系统不时地处于 100%的运作状态中，当循环检查一个特定的应用程序时，可能另外一个应用程序已经发生了事件，但是由于系统还在检查其它的应用程序，因此无暇来处理真正发生事件的应用程序，因而造成此时发生事件的应用程序的反应速度极其缓慢。

既然使用循环是不可行的设计方式，那么什么方式是比较适合的呢？我们不妨反过来想。既然我们无法预知何时以及会发生什么事件，那么何不让事件来驱动我们的系统？当特定的**事件**发生时，我们的系统只要知道是哪一个应用程序能够处理这个事件，就把此发生事件的**消息(Message)**分派给正确的应用程序来处理不就可以了吗？

消息驱动模型

上述以事件/消息处理模型的架构的确是现今许多操作系统、Framework 和 Library 使用的基础运作架构。因此我们设计的执行环境可以使用图 1-1 的架构来表示，在这个执行环境中会根据发生的事件转换事件为代表此事件的消息，最后再正确地分派这些消息给应该负责的应用程