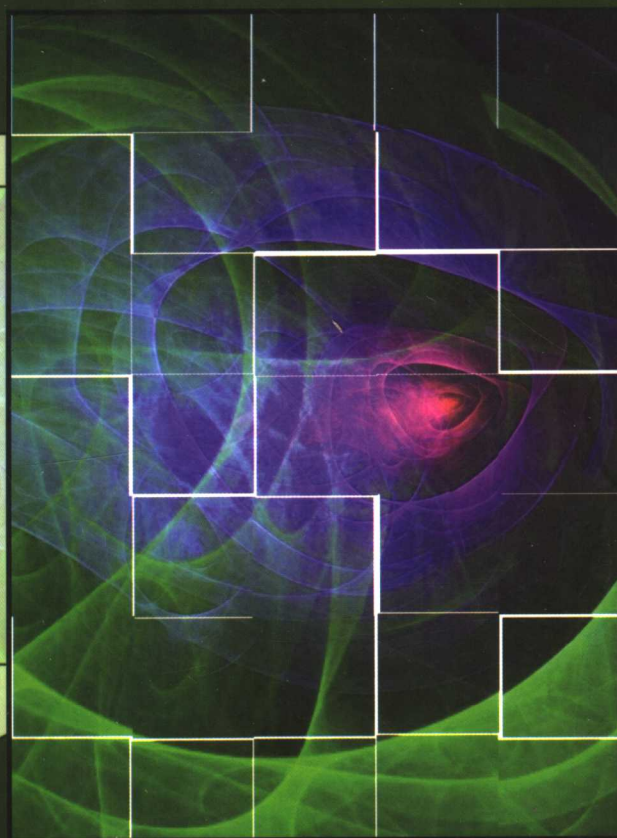


Delphi

深度探索 第二版



陈省 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONIC INDUSTRY
<http://www.phei.com.cn>

ROM Included

内 容 简 介

本书深入介绍了 Delphi 开发的各个方面的功能, 实例丰富。COM 开发部分深入讨论 ActiveX、Office 自动化编程、基于 COM 的拖放技术以及基于 COM 的数据库应用开发实例。Shell 编程部分叙述未经公开的 Windows 外壳扩展的编程技术, 将程序同 Windows 外壳无缝地结合, 展现操作系统内部的精彩世界。IDE 扩展部分叙述使用 Open Tools Api 开发接口开发出提高工作效率的强力辅助开发工具。VCL 深度探索部分讲述 Action、Drag Drop、Dock 等 VCL 界面库高级开发技巧, 面向对象的界面复用技术。深入探讨 VCL 容器类、强大的 Web 开发框架 IntraWeb 以及 OR Mapping 框架 Bold 的开发。经典模式部分结合 VCL 中的设计模式实例介绍 GOF 的 22 个经典设计模式之 Delphi 实现。IE 扩展部分讲述了 IE 界面扩展、协议扩展和 DOM 的高级用法。辅助工具部分讲述极限编程测试工具 DUnit、性能分析工具 GpProfile 等强大的第三方辅助开发工具的使用。通过阅读本书, 读者可以充分利用 Delphi 的强大功能, 开发出高效率的 Delphi/Kylix 系统。

本书适合中高级 Delphi 程序员、系统设计师、数据库程序设计师以及对 Delphi 感兴趣的人员阅读。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目(CIP)数据

Delphi 深度探索 / 陈省编著. —2 版. —北京: 电子工业出版社, 2004.1

(Borland In-Depth Series\Borland 大系)

ISBN 7-5053-9411-8

I.D... II.陈... 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 110151 号

责任编辑: 周 筠 张兴田

技术编辑: 韩 磊

印 刷: 北京增富印刷有限公司

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: 34.75 字数: 900 千字

印 次: 2004 年 1 月第 1 次印刷

印 数: 6 000 册 定价: 49.80 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 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

序

Delphi 的书籍在国内已经出版了不少，可是好的作品还是屈指可数。已出的书大致有以下三类：第一类是从入门到精通的系统讲解，内容覆盖面非常广，主要是外版图书，经典的作品有 Charlie Calvert 的《Delphi X Unleashed》，Steve Teixeira & Xavier Pacheco 的《Delphi X Developer's Guide》，Marco Cantù 的《Mastering Delphi X》(X 表示 Delphi 的不同版本)这三本都是大部头的经典著作，有不同版本的中译本，使用 Delphi 的人至少要拥有其中一本，写这种书籍对作者要求很高；第二类是专题领域的著作，李维先生的三本多层应用系列就是其中的代表，这种书籍要求作者在专门领域有很深的研究；第三类就是以应用为主的著作，相对系统性差一点，不过实用性更强，这方面最近出版的陈宽达的《Delphi 深度历险》即是代表作。

非常遗憾，这些好作品基本上都是引进国外和台湾地区的版权。大陆出版的原创 Delphi 书籍，不是作者本身水平有限，就是作者也许技术水平还不错，但却不能把自己的经验和心得很好地表述成文，或者没有花足够的时间去构思创作。

《Delphi 深度探索》属于第三类的作品，比起前两种类型，这种作品应该说容易写作一些，但对作者本身的要求并不低，陈省（网名 Hubdog）在 Delphi 上钻研颇深，国内多个技术论坛上都可以见到他的身影，难得的是他花了大量的时间去精心构思这样一本书籍，内容虽然看起来涉及领域不算广，但却包含了丰富的内涵，很多部分其实是独立成章的，本书有很高的实用价值，充分展示了 Delphi 的强大功能。

这本书对读者要求也不低，读者必须基本掌握 Delphi 编程，并具有一定实际经验。仔细体会本书其中的功能是如何实现的，将大大提高你应用 Delphi 的实力。

我个人还非常喜欢本书的“工具篇”这一部分，GExperts, CodeSite, MemProof 也是我喜欢的工具，Delphi 的强大还在于它丰富的第三方控件和工具。就像好酒的酿造需要好水一样，好的程序员需要最好的开发工具，现在的开发工具具有非常强大的功能，可极大地提高程序员的工作效率。好书的作用就更大了，我衷心希望今后能看到越来越多大陆原创的高质量书籍。

蒋涛

2000年12月于北京

前 言

从编程和软件开发图书市场上来看, Visual C++的书长期一直都占主流,大部头的和深入探索内部机制的书相对很多,而 Delphi 相关的中高层次的图书要少得多,而且精品比较少,大部分是老外写的,国人所写的值得收藏的无疑只有台湾著名的 Delphi 专家李维写的系列丛书了(由机工社出版,共三本)。在内容方面,国内所出版的 Delphi 图书也主要集中在数据库开发方面,这就难免给人以一种印象,好像 Delphi 就是一个简单易学的快速数据库开发工具,用来开发数据库非常方便,但用于其他关键任务就有点不堪重任了。其实不然, Delphi 早已经在应用软件等方面展现了它的强大功能,在整个软件开发领域中, Delphi 早就在网络、游戏、系统开发等各个方面大展拳脚了,最明显的就是 Foxmail、NetVampire 等程序上的应用,但无论是翻译还是国内编著介绍它的书都很稀少,即或稍有涉及,也是浅尝辄止,没有搔到痒处。本书期望能够尽量在适度的篇幅内,展现 Delphi 在一些应用开发方面的强大能力,查缺补漏,期望能让广大程序开发者看了这本书后能够得到一些另类的体验,发现原来 Delphi 还有这么强大的功能呀!

本书分成四个部分,第一部分是 COM 篇。毫无疑问,未来 Windows 操作系统将完全架构于通用对象模型 (COM) 的基础上,但已有的 Delphi 的书在这方面的论述显得很薄弱、零散,唯一一本老外写的介绍了 COM 的书,也是对于原理讲得比较多,对于应用则讲得比较少,给人以一种“雾里看花”的感觉,本书则从 COM 的更为偏向应用的角度讲述,期望读者通过书中的各个实例,能对 COM 的应用有更为清晰和感性的认识。

第二部分就是外壳 (Shell) 篇, Windows 之所以能够占有操作系统如此大的份额,其方便和漂亮的外壳操作界面无疑是一个重要的砝码,如能将我们的程序同漂亮外壳无缝地结合起来,无疑会让用户添加很多的印象分,但遗憾的是不知出于什么原因,微软在这方面的资料非常欠缺,这就为这方面的开发造成了很大的困难。本书期望能够通过揭示其冰山之一角,为大家展现操作系统内部更为精彩的世界。

如果说微软公司的文档有所欠缺, Borland 公司的文档就只能说是 very very 欠缺了。程序员们普遍认为 Borland 公司的 VCL 的架构绝对是超先进的,比微软的 MFC 领先了

一个时代，但由于其在经济实力上同微软无法相比，使得其无法完成特殊 VCL 的详细开发文档，这无疑极大地限制了程序员们对 VCL 架构的扩展。最具讽刺意味的就是它的集成开发环境（IDE）的扩展的实现 Open Tools API 了，名为开放工具接口，但只提供了一些超长的接口声明单元和简单的注释，一切开发都需要半靠注释，半靠猜测。针对这些问题，本书的第三部分特殊 VCL 篇期望能够揭开覆在其上的那层薄薄轻纱。

工欲善其事，必先利其器，好花还需绿叶扶持。Delphi 虽然很强大，但它也不是万能的，因此，还需要寻找和掌握那些其他人开发的强有力的工具来进一步提高工作效率，第四部分工具篇中提到的 GExperts、CodeSite、MemProof 等无疑是其中的佼佼者。

陈 省

2001 年 10 月于北京

hubdog@263.net

<http://hubdog.myrice.com>

《Delphi 深度探索》第二版序言

今年三月份，周筠老师与我商量，说《Delphi 深度探索》第一版的销售情况不错，希望我能够在第一版读者意见的基础上，对原书做一个比较大的修订，争取写一本比上一版更好的书。

就我而言，《Delphi 深度探索》第一版还是有不少令我不甚满意的地方的。这本书第一版的写作是从 2001 年的 2 月开始的，大约花了 5 个月的时间，完成于在 2001 年 7 月。当时我还在北大读书，写作的内容以我当时的个人兴趣为主，主要包括 COM、Shell 和 Open tools api 的内容，应该说这部分内容同社会上主流 Delphi 程序员的开发需求是有一定程度的偏差的，有些内容显得不是很实用。此外，第一版的某些部分相互之间联系不是很紧密，稍嫌零乱，还有部分内容缺少基本原理的铺垫，单刀直入就进入了技术细节的描述，对于一些基础不够的读者来说，就会产生理解上的障碍。

所以接到再版修订的任务，对我来说也是一个很好的机会。与此同时，考虑到广大读者的经济承受能力，为了不过多的增加读者的负担，我采取以删节为主，修订为辅的原则，对旧版进行了比较大的调整。下面是一个旧版内容的删节修订列表：

章 节	修订情况	修订原因
1.2.2 创建 Office 2000 插件		原文中遗漏了插件注册部分，导致注册无效，无法使用
1.3 基于 COM 的拖放技术	删除 42 页	这部分内容比较乱，同时不是很实用
1.4 Web 相关技术	删除 25 页	XML 技术发展非常迅速，从今天来看原书的这部分内容有些过时，同时 Keith wood 和李维的 XML 以及 Soap 的专著都陆续出版，使得这部分内容显得不是很必要了
1.6 其他	删除 39 页	这部分内容相互之间比较独立，不是很系统
2.1.6 属性页扩展		原文的例子在 Win9x 上可以正确运行，但是在 Win2K 下无法正确运行，对此进行了修正。
2.2.3 外壳对话框	删除 25 页	这两部分内容同本篇其它部分不是很相关
2.2.4 外壳事件通知		
3.1 属性分类及其实现	删除 6 页	主要是这部分内容实用意义不大，同时我预计未来的 Delphi8 中实现会有改变，很可能会向 .net framework 的实现方式靠拢
3.2.3 Open Tools Api 主要接口概述	删除 6 页	从 Delphi7 开始，Borland 终于完成了 Open Tools Api 的文档，因此这部分不再重要，直接查帮助文档就可以了
3.2.6 编辑器增强功能	删除 8 页	这部分内容没有足够的原理叙述，同时例子本身显得过于琐碎

续表

章 节	修订情况	修订原因
3.2.9 其他 OTA 相关问题	删除 15 页	同样, 缺少原理的叙述, 有过多的代码示例
3.4 自绘画的属性编辑器	删除 7 页	这部分内容在 Delphi6、7 中变化比较大, 同时实用价值不高, 所以删除
4.1 No.1 之 GExperts	删除 17 页	这些内容与提升编程技术关系不大, 因此删除
4.4 异常杀手— ExceptionalMagic	删除 2 页	
4.5 代码格式修正专家	删除 10 页	

修订情况是总计删除了 202 页, 修订了两个小节的内容, 修订总量占原书内容的 40% 以上。同时考虑到虽然删除的内容不是很重要, 但是可能仍然会有一些新的读者希望阅读这些内容, 所以这些删除的部分我都整理成了电子文档, 公开在我的网站 <http://hubdog.csdn.net> 上了, 感兴趣的朋友可以下载。

另外, 在过去的两年中, 我离开了校园, 投身到了实际开发过程中, 对于 Delphi 的技术发展有了更多的新的认识, 因此除了修订旧版内容外, 在新版中, 我又增加许多新内容, 希望能与大家分享过去两年来我学习的小小心得。

首先, 在实际开发过程中, 我越来越感到软件作为一种商品, 其质量以及性能的重要性, 为此我在工具篇中增加了基于测试先行理念来保证软件质量的极限编程测试工具-Dunit 的使用指南, 以及免费开源的性能分析工具 GpProfile 的使用介绍。

此外, 考虑到在 Internet 时代, IE 浏览器的影响越来越大, 和以 IE 为平台的应用越来的越多的趋势, 因此本次修订增加了 IE 扩展篇, 基本涵盖了 IE 扩展二次编程的主要方面, 包括 IE 界面扩展、协议扩展、DOM 的高级用法等等内容。

还有, Delphi 的精髓之一就是它的强大的 VCL 类库。只有对 VCL 类库有了深入清晰的认识, 才能写出高效、高质量的应用程序。特别是从 Delphi7 开始, Borland 提供了新一代的所见即所得的 Web 开发框架 Intraweb, 以及模型驱动的 MDA 框架 Bold, 这两个产品使得基于 Delphi 的 Web 程序开发以及完全面向对象的基于 OR Mapping 的数据库应用的开发实现了质的飞跃。为此我在新版中增加了 VCL 深度探索篇, 主要包括: VCL 高级拖放及停靠技术的实现, 面向对象的界面重用技术, VCL 容器类的使用, 以及新一代的 Web 开发框架 Intraweb 和基于 Bold 的 UML 模型驱动的数据库应用开发的内容。

最后, 在本书修订之前, 我在大富翁论坛 (<http://www.delphibbs.com>) 上发了一个帖子, 征求大家对于新版的修订意见以及期望, 有很多热心的朋友建议我写一些 Delphi 与设计模式的内容, 因为市面上关于设计模式的书大都以 java 和 C++ 的示例为主, 对于 Delphi 程序员来说有学习上的障碍。从我个人学习设计模式的体会来看, 其实在 VCL 的源代码中有着大量的设计模式的应用, 作为 Delphi 程序员却要

通过 Java 与 C++ 例子来学习设计模式，在我看来就是骑驴找驴，简直是一个莫大的讽刺。因此，我最终决定写一章以 VCL 中的示例来介绍如何使用 Delphi 应用设计模式的内容。应该说，这一章对我来说是整个写作过程中最困难的一章了，有时为了想一个简单而清晰的模式例子，经常要花去我半天的时间，别的章节一般是一周能写 30 页，而设计模式这一章的写作进度通常一周只有 10—15 页。虽然花了很多的时间，但是我觉得写作这部分内容也帮助我理清了以前头脑中的一些混乱概念，让我自身对于设计模式也有了更加清晰的认识，同时也是这次写作中最开心的收获，我也衷心地希望大家在阅读这章内容的时候能够获得同我类似的收获。

最后，我要感谢《程序员》杂志社的蒋涛社长以及电子工业出版社博文视点资讯有限公司的周筠老师、韩磊编辑和张兴田编辑，没有他们的大力协助和支持，本书不可能这么快就能同读者见面。此外，还要感谢我的父母，他们向我许诺，虽然他们看不懂这书的好赖，但是不管书有多烂，他们一定会买 10 本来支持我。还要感谢我亲爱的夫人，是她在我写作期间独自一人承担了所有的家务，并在此期间把我的腰围从原来的二尺三寸饲养到了二尺五寸☺。

陈 省
2003 年 9 月于北京

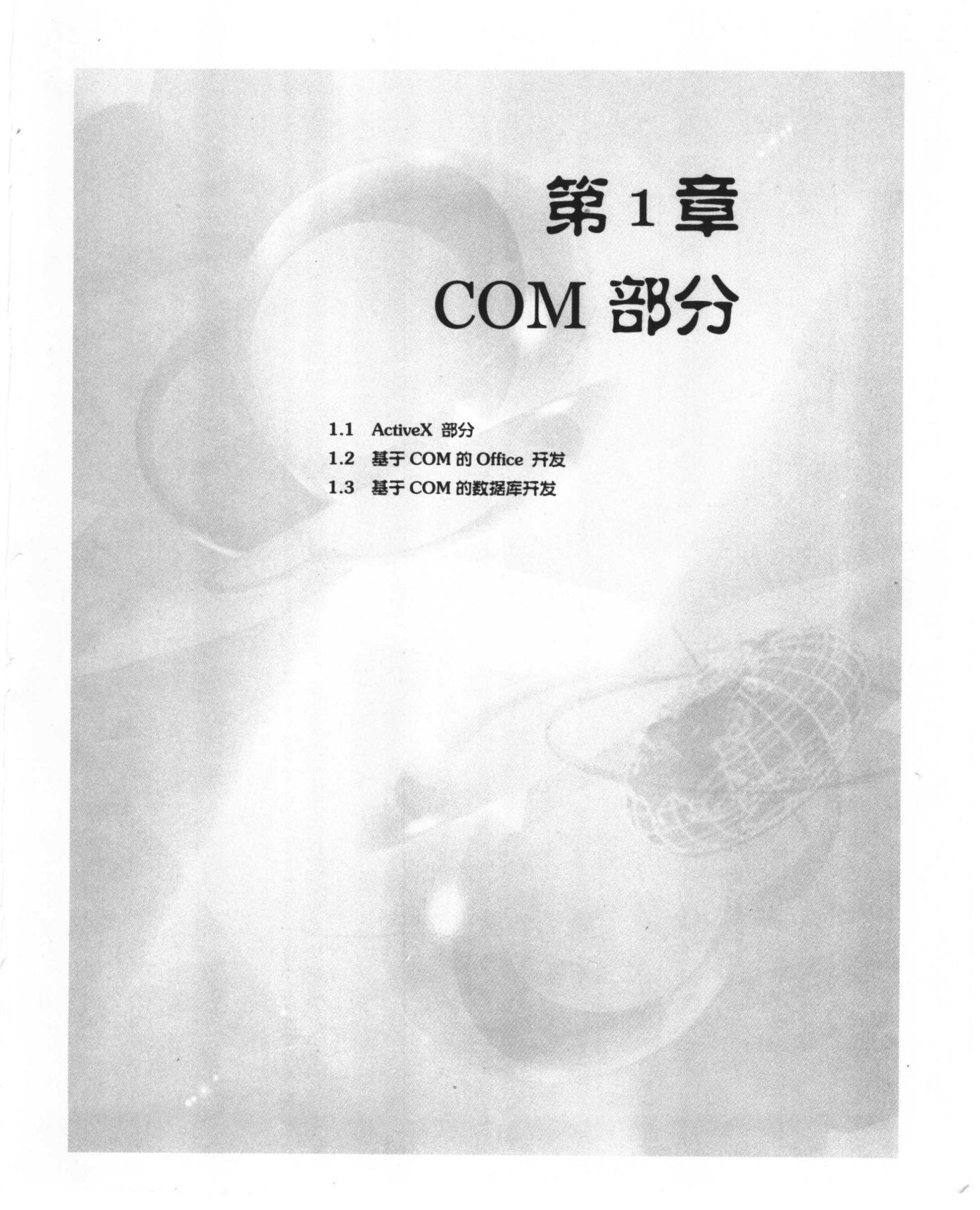
目 录

第 1 章 COM 部分	(1)
1.1 ActiveX 部分	(3)
1.1.1 ActiveX 控件之消失的事件	(3)
1.1.2 创建 ActiveX 控件之高级编辑界面	(10)
1.1.3 数据库明了的 ActiveX 控件	(19)
1.2 基于 COM 的 Office 开发	(22)
1.2.1 Office 自动化编程	(22)
1.2.2 创建 Office 2000 插件	(53)
1.3 基于 COM 的数据库开发	(60)
1.3.1 ADOX 的数据库开发	(60)
1.3.2 SQL Server - DMO 数据库开发	(78)
1.3.3 OLE 结构化存储及其在公文包型数据库中的应用	(95)
第 2 章 Shell 部分	(113)
2.1 外壳扩展	(115)
2.1.1 搜索扩展	(115)
2.1.2 文件飞跃提示扩展	(121)
2.1.3 拖放控制扩展	(124)
2.1.4 命名空间扩展	(130)
2.1.5 实现 AutoComplete	(154)
2.1.6 属性页扩展	(159)
2.1.7 外壳执行操作记录器	(168)
2.2 未经公开的外壳奥秘	(172)
2.2.1 PItemIDList 的基本概念	(172)
2.2.2 用外壳接口对系统进行管理	(184)
第 3 章 IDE 扩展部分	(193)
3.1 Open Tools API 简介	(195)
3.2 Hello World	(196)
3.3 消息通知器	(198)
3.4 将 Winamp 集成到 Delphi 中	(201)

3.5	自动规范控件前缀命名的专家	(216)
3.6	To-Do List 的增强	(223)
第 4 章	IE 扩展部分	(231)
4.1	IE 菜单扩展	(233)
4.2	IE 按钮扩展	(236)
4.3	阻断弹出式广告的 BHO	(241)
4.4	IE 右键菜单扩展	(247)
4.5	IE 下载扩展	(251)
4.6	创建 3721 样式的 Google 搜索扩展	(255)
4.7	IE 异步可插入协议扩展	(258)
4.8	创建可自动填表的 IE 面板插件	(268)
第 5 章	VCL 深度探索	(283)
5.1	Action 开发之七种武器	(285)
5.2	VCL 拖放高级应用	(296)
5.3	高级停靠技术的实现	(308)
5.4	面向对象的界面复用技术	(326)
5.5	VCL 中的容器类	(336)
5.6	所见即所得的 Delphi Web 开发利器 —— IntraWeb 开发指南	(357)
5.7	基于 Bold 的 UML 模型驱动的数据库应用开发	(382)
5.7.1	Object RDBMS Mapping 原理简介	(382)
5.7.2	使用 Bold 开发数据库应用	(383)
5.7.3	Bold 的代码迭代开发	(396)
第 6 章	Delphi 与设计模式	(417)
6.1	使用 Delphi 实现设计模式	(419)
6.1.1	工厂模式	(419)
6.1.2	单例模式(singleton)	(421)
6.1.3	建造模式(Builder)	(424)
6.1.4	原型模式(Prototype)	(427)
6.2	结构模式	(431)
6.2.1	适配器模式(Adapter)	(431)
6.2.2	合成模式(Composite 模式)	(434)
6.2.3	装饰模式(Decorator 模式)	(437)



6.2.4	门面模式(Façade 模式)	(441)
6.2.5	桥模式(Bridge 模式)	(447)
6.2.6	代理模式 (Proxy)	(450)
6.2.7	享元模式(Flyweight 模式)	(453)
6.3	行为模式	(458)
6.3.1	策略模式 (Strategy 模式)	(458)
6.3.2	模板模式(Template Method 模式)	(461)
6.3.3	责任链模式(Chain of Responsibility 模式)	(464)
6.3.4	观察者模式(Observer 模式)	(466)
6.3.5	命令模式(Command 模式)	(470)
6.3.6	备忘录模式(Memento 模式)	(472)
6.3.7	迭代子模式(Iterator 模式)	(476)
6.3.8	访问者模式(Visitor 模式)	(481)
6.3.9	调停者模式(Mediator 模式)	(485)
6.3.10	状态模式(State 模式)	(490)
第 7 章	工具篇	(499)
7.1	极限编程的集成测试工具-DUnit	(501)
7.2	性能分析工具 GpProfile	(518)
7.3	内存泄漏清道夫——MemProof	(525)
7.4	CodeSite 应用指南	(533)



第 1 章

COM 部分

- 1.1 ActiveX 部分
- 1.2 基于 COM 的 Office 开发
- 1.3 基于 COM 的数据库开发

1.1 ActiveX 部分

1.1.1 ActiveX 控件之消失的事件

► Delphi ActiveX 框架

Delphi ActiveX(DAX)框架使我们能够很容易使用转换向导从 Delphi 本身的控件 (VCL) 出发创建 ActiveX 控件,而这一切不需要我们具备很多的 COM 知识,生成的 ActiveX 控件可以应用在 VC、VB 等不同的开发环境中,这就可以避免在不同的开发环境中重复开发组件,实现真正的代码复用。

创建 ActiveX 控件涉及到生成一个 VCL 的 COM 封装,封装类继承于 TActive-XControl 实现了一个对 Delphi 控件的引用参考。Delphi 的 VCL 控件要想能够转换为 ActiveX 控件,必须继承于 TWinControl。这是因为 COM 封装和 VCL 控件之间的通信需要一个窗口句柄,而只有 TWinControl 的子类才有句柄。

封装类通过属性和方法向其他的对象公开了 Delphi 控件的功能。同时它也负责创建管理和销毁嵌入的 VCL 控件。COM 封装类通常还负责响应控件产生的事件,并提交给 ActiveX 控件容器。

听起来好像有点复杂,幸好 Delphi 提供的 ActiveX 控件向导可以极大地简化这一过程。下面让我们把一个最常见的 TListBox 控件转化成 ActiveX 控件。首先选 File/New/ActiveX 页,如图 1.1 所示。然后选择 ActiveX Control 项,显示一个对话框,第一步是选择 VCL 的类名,VCL 类名下拉框中列出了全部 TWinControl 的子类(不包括使用 RegisterNonActiveX 函数注册的控件),从列表可以看到 TListBox,但找不到 TLabel,因为 TLabel 继承于 TGraphicControl(它是没有窗口句柄的),自然像那种 TTimer 的非可视控件就更是没有了,如图 1.2 所示。

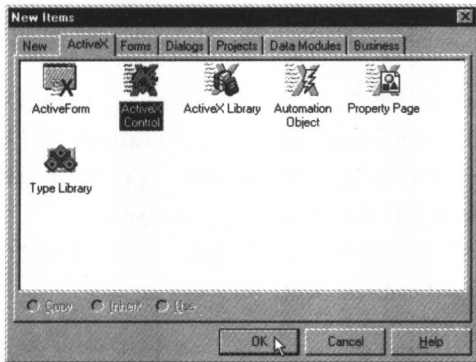


图 1.1

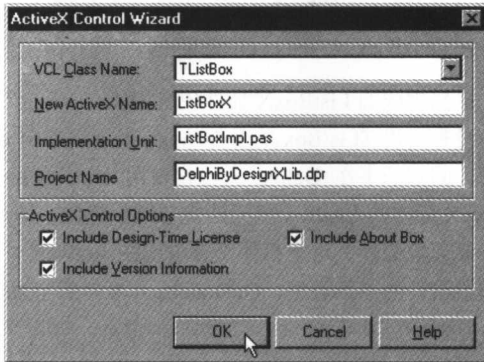


图 1.2

当选好了类名后,向导会自动给其他编辑框添加缺省值。这里使用它的缺省值,同时选中 ActiveX



Control Options 的三个选项，然后点 OK 按钮。

➤ ActiveX 控制选项

图 1.2 显示了全部可选的 ActiveX 控制选项，Include Design-Time License 选项表明设计时许可证将被创建，它可以防止控件在设计环境中被非法使用，如果选择了这个选项，向导会生成一个 LIC 文件，里面包含了许可证信息。用户必须有这个 LIC 文件才能在开发环境中使用相应的 ActiveX 控件。Include Version Information 选项使我们可以在项目中添加版权说明、版本号等版本信息。具体版本信息的设定可以在创建好项目后，选 ProjectOptions 然后切换到 Version-Info 页面来进行(注意：尽管版本信息是一个可选的选项，但如果打算在 Visual Basic 4.0 及以上版本中使用 Delphi 生成的 ActiveX 控件，则必须选定它)。Include About Box 选项表明是否为 ActiveX 控件创建一个“关于对话框”。“关于对话框”是一个独立的窗体单元，可以通过编辑它来显示所需要的版本信息或其他信息。关于信息可以通过在开发环境中点击 About 属性来调用。

➤ 生成单元文件

一切设定好后，点 OK 按钮。向导首先创建一个 ActiveX Library 项目。接着创建一个实现单元，来实现对原生 Delphi 控件的 COM 封装。封装类是 TActiveXControl 一个简单的子类，然后是类型库和类型库接口单元（类型库实际上是一个二进制文件，里面定义了数据类型、接口、方法以及 ActiveX Library 要公开的对象。类型库接口单元包含了对应于类型库中信息的 Pascal 声明）。最后是许可证文件，关于对话框和源码文件。

➤ 编译和注册 ActiveX 控制

现在可以编译 ActiveX Library 项目来生成包含 ActiveX 控件的 OCX 文件。我们可以直接调用 RunRegister Server 来注册 ActiveX Server。然而在这之前，建议保存全部的项目文件。如果没有先保存项目文件，那么储存在注册表中的路径就是当前的路径而不一定是项目的真正路径。

注册后就可以在 Visual Basic 等程序中使用 Delphi 生成的 ActiveX 控件了，如图 1.3 所示。

➤ ActiveX 转换过程中的问题

虽然生成的 ListBoxX 控件使我们能在 Visual Basic 中使用 TListBox 的多项功能。但 ListBoxX 控件并没有实现 TListBox 的全部功能。比如 TListBox 的 OnDrawItem 的事件定义在 ListBoxX 中是找不到的。回想一下在 Delphi 里经常使用 OnDrawItem 事件来支持用户自绘画功能。没有了这个事件，ActiveX 版本的 ListBox 就无法支持这一特性，即使 Style 属性设定为 lsOwnerDrawFixed。

要理解为什么这个事件没有出现在 ActiveX 中，我们需要研究一下事件类型，特别是要研究一下传递给事件的参数类型。OnDrawItem 事件声明为 TDrawItemEvent 类型，定义如下：

```
TDrawItemEvent = procedure( Control: TWinControl;  
                             Index: Integer; Rect: TRect;  
                             State: TOwnerDrawState) of object;
```

我们看到 OnDrawItem 事件处理函数接受 4 个参数。其中第三个参数阻止了事件出现在 ActiveX Control 中。因为 OLE 自动化不知道如何列集 (marshall) TRect 类型的数据。

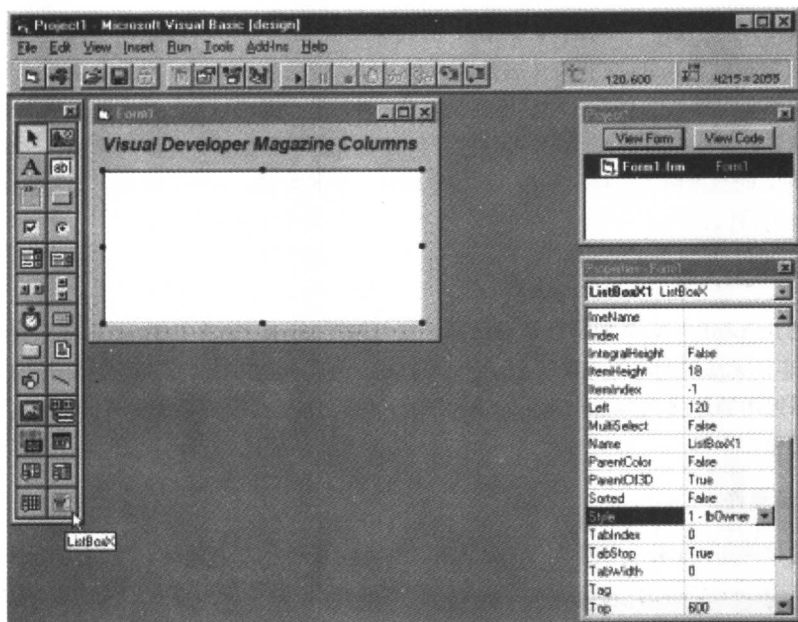


图 1.3

原因在于 ActiveX 控制实际上是小型的 OLE 自动化服务器。包含 ActiveX 控件的 ActiveX 容器使用自动化来和控件通信。因此要想使一个属性、方法或事件出现在 ActiveX 版本的控件中，全部参数和返回类型必须是自动化兼容的。图 1.4 列出了全部的兼容类型。

在这个表中虽没有列出，但 ActiveX 转换向导还可以正确处理 TColor、TFont、TPicture 和 TStrings 类型的属性。Color 属性也可以很容易地转换，因为它实际上就是一个整数类型。而对于其他属性类型，Delphi 提供了一个定制接口来处理对应的 Delphi 属性。比如 IStrings 接口提供了处理 TStrings 属性的途径。

同时，一个属性是自动化兼容的并不意味着它就一定会出现在 ActiveX 控件中，向导不会转化对 ActiveX 控件无意义的属性、方法或事件。因而很多属性就没有被转化，比如：Height、HelpContext、Hint、Left、Name、ParentFont、ParentShowHint、PopupMenu、ShowHint、TabOrder、Tag、Top 和 Width 等。

此外，有两种特殊类型的属性向导也不会将其转化。这就是对象引用和数据明了属性，对象引用（就是指 Delphi 的某个属性对应于一个对象，比如 TTable 的 Datasource 实际上是引用了另外一个对象 TDataSource）在 Delphi 里是通过指针来实现的，因此是自动化不兼容的。此外，ActiveX 本身也没有

表 1

自动化兼容类型

预定义类型

SmallInt, Integer, Single, Double, Currency, TDateTime, WideString, IDispatch, WordBool, Variant, OleVariant, SCode, Byte, and IUnknown

Enumerations defined in a type library

Interface types defined in a type library

Dispatch interfaces defined in a type library

图 1.4



提供一个标准的方式来使一个控件同容器中的其他控件相关联。而数据明了属性 DataSource 和 DataField 没有被映射到 ActiveX 控件中去是因为 Delphi 实现的数据明了采用了完全不同于 ActiveX 的机制。当然如果非常想利用 Delphi 中好用的数据库控件,这也是可以的,但需要一些额外的工作,这将在后面介绍。必须指出的是 ActiveX 的数据控件和 Delphi 的数据控件是不一样的。这也就是为什么没有一个 Delphi 的数据明了控件出现在 ActiveX 转换向导的类名列表中的缘故,即使它可能是继承于 TWinControl 的控件。

➤ 正确转换的关键

到目前为止 ListBoxX 其他特性都没问题,也可以把 Style 变成 IsOwnerDrawFixed,但是没有 OnDrawItem 事件无法改变它的外观显示。让我们试用一种不同的方法,这就是定义一个自动化兼容的接口来实现自绘画的外观 (Owner-Draw)。例如我们可以创建一个新的事件 OnColorItem,来改变每一个列表项的颜色。

➤ 添加新事件

添加新的事件到 ActiveX 中需要修改类型库中的信息,修改信息有两种方法:通过选择 EditAdd To

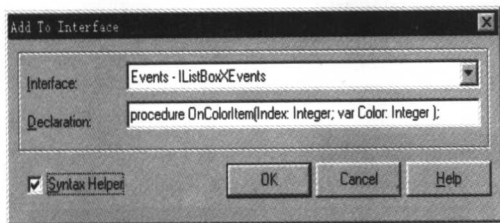


图 1.5

Interface 菜单或用内置的类型库编辑器都可以, EditAdd To Interface 菜单只有在当前文件为实现单元时才能使用。如图 1.5 所示。

图 1.5 显示了 OnColorItem 是如何加入到 IListBoxEvents 接口中的。定义一个 ActiveX 接口的新事件与 Delphi 中有点不同。不像 Delphi 中那样给事件声明一个属性类(比如 TNotifyEvent),必须写一个将会传递给事件处理函数的带参数的

过程。类似于写一个控件中的事件分配方法。OnColorItem 事件定义如下:

```
procedure OnColorItem(Index: Integer; var Color: Integer);
```

点 OK 按钮, Delphi 就把这个声明加入类型库。这使得我们可以给 OnColorItem 事件写处理函数了。但是在这之前,还必须实现产生事件的代码,这有点类似于在 Delphi 控件中定义新的事件。生成事件属性和事件分配方法仅仅是一部分工作,还必须在合适的时间调用事件分配方法来产生事件。最好的产生 OnColorItem 事件的位置就是在 Delphi 控件的 OnDrawItem 事件中,实现的方法见源码清单 1。

TListBoxX 封装类定义了一个事件处理函数 DrawItemEvent 来响应嵌入的 ListBox 的 OnDrawItem 事件。DrawItemEvent 函数在 InitializeControl 方法中进行初始化,同时 DrawItemEvent 函数负责产生 OnColorItem 事件。为了产生事件,要使用 FEvents 接口来激发 OnColorItem 事件过程。Index 参数来自于 DrawItemEvent 参数列表,而 Color 参数是在 DrawItemEvent 里定义的。当 OnColorItem 事件返回时,ItemColor 变量将包括缺省的颜色或用户定义的颜色,然后 ItemColor 变量就被用来画列表项。如图 1.6 所示。

图 1.6 显示了 ListBoxX 控件在 VB 中的运行状况。虽然 ActiveX 不可能像 Delphi 中的 VCL 控件那样可以随意使用 Canvas 来画图,但是这个例子证明了利用 ActiveX 控件的灵活性,我们还是可以作出一