

Delphi 8

程序设计经典解析

黄文钰 编著

清华大学出版社

Delphi 8 程序设计经典解析

黄文钰 编著

清华大学出版社

北 京

内 容 简 介

Delphi 在程序开发中占有相当大的优势,主要是它的核心在一开始就设计得很好,现在不但可以同时多个来源的程序语言进行编译,而且不失去它原有的特性,让设计师开发系统更加快速,且学习更容易,对于正在选择何种开发工具的人有相当大的吸引力。

本书主要介绍 Delphi 本身程序的制作原理及概念,示范 Delphi 常用的程序编写及应用,说明 Delphi 比较高级的应用功能,剖析 .NET Framework 在 Delphi 中的应用,并简单介绍 Delphi 中的函数。

假如您是初学者,在选定 Delphi 作为开发工具时,本书会告诉您如何利用 Delphi 来开发程序;若您已是 Delphi 的老用户,则可以从书中了解程序的开发技巧。本书主要适于初、中级的 Delphi 用户,也可作为程序设计师的参考用书。

本书繁体字版书名为《Delphi 8 程式设计经典学习》,由文魁资讯股份有限公司出版,版权属黄文钰所有。本书简体字中文版由文魁资讯股份有限公司授权清华大学出版社独家出版。未经本书原版出版者和本书出版者书面许可,任何单位和个人不得以任何形式或任何手段复制或传播本书的部分或全部内容。

北京市版权局著作权合同登记号 图字:01-2006-2345

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Delphi 8 程序设计经典解析/黄文钰编著. —北京:清华大学出版社,2006.5

ISBN 7-302-12861-8

I. D… II. 黄… III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字(2006)第 034350 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦
http://www.tup.com.cn 邮 编: 100084
社 总 机: 010-62770175 客户服务: 010-62776969

责任编辑: 李春明
排版人员: 李月菊
印 刷 者: 北京市清华园胶印厂
装 订 者: 三河市春园印刷有限公司
发 行 者: 新华书店总店北京发行所
开 本: 185 × 260 印张: 31.75 字数: 753 千字
版 次: 2006 年 5 月第 1 版 2006 年 5 月第 1 次印刷
书 号: ISBN 7-302-12861-8/TP · 8176
印 数: 1 ~ 4000
定 价: 46.00 元

前 言

Delphi 的版本更新很快, 这对于一个程序语言来说, 我们乐见于其不断发展, 不断地加强和扩充。而对于程序设计师来说, 虽然程序语言的功能增强, 有助于其更容易地设计程序, 但这对我们适应不断发展的程序语言提出了很高的要求。

现在新一代的程序语言已经有了一些的变化, 我们不再需要“单恋”一种程序语言, 因为在网络的架构下, 在一个时间点可能需要运行的语言就有好几种, 我们不能只有一个“招式”, 或是只有一种“兵器”。Delphi 在进入 .NET Framework 后, 不但可以进行团队合作, 而且每个成员都可以使用不同的功能模块, 发挥更强大的功能。

在以前, 写程序的人总是具有相当高的地位, 因为在纯文字的时代, 要看懂程序需要相当的功力。但在进入 Windows 时代后, 许多写程序的人会感到惶恐, 因为在 Windows 下编写程序确实是轻松许多。

我们在学习 Delphi 8 时, 可能不能再像 Delphi 7 之前的版本一样。以前的程序拿来再重新编译, 不一定就可以用。因为在 Delphi 8 中, 不只是版本的升级, 而是 Delphi 一个新的蜕变, 原先的功能都会存在, 但是在使用上将会有所不同, 因此如果要重新编译, 对程序进行少量的修改可能是免不了的。

Delphi 的“老传统”仍然没变, 在 Help 的使用上, 仍然是相当困难。在查询数据时, 同时可以查到 Delphi 原有的功能, 也可以查到 .NET Framework 的功能, 有时命令相同, 很容易搞混, 而且说明上也不太详尽, 在非英语系国家这一点仍然是 Delphi 较弱的一环, 不过似乎在新的版本, 仍没有加强的感觉。

写程序是一个非常有趣, 但也可能是非常无聊的工作, 可能和业务员面对“新鲜事物”的情况类似, 因为时代的巨轮一直在转动, 要拥有最新的技术就必须不断地学习。有人会觉得这样的工作太辛苦了, 但笔者认为有适当的新鲜感, 才是让人可以继续努力工作的动力来源之一。因为每天都重复做着不变的工作, 这才是真正让人难以忍受的事, 有新事物才能让精神振奋, 向同行炫耀自己领先的能力。

有句话说得好, “教学相长”, 相信有许多读者必定是使用 Delphi 的高手, 如果读者不吝赐教, 可以通过电子邮件与笔者进行学术上的交流, 感谢您的支持。

作者 黄文钰

E-mail:huangcp@ms19.url.com.tw

目 录

第 1 章 Delphi 简介.....1	2.3 创建 Package 35
1.1 什么是 Delphi.....1	2.4 创建 ASP.NET Web Application..... 36
1.1.1 什么是.NET.....1	2.5 创建其他项目..... 37
1.1.2 什么是 Delphi .NET.....2	2.6 自定义下拉菜单..... 40
1.2 Delphi 8 新特色.....3	第 3 章 Object Pascal 42
1.3 安装 Delphi 8.....4	3.1 Unit 与 Nfm 42
1.3.1 Delphi 8 安装界面.....4	3.1.1 Nfm 文件..... 42
1.3.2 安装 Microsoft .NET Framework SDK 1.15	3.1.2 Windows 窗体设计器 生成的代码..... 43
1.3.3 安装 Microsoft .NET Framework 1.16	3.2 基本语法..... 44
1.3.4 安装 Microsoft Visual J# .NET.....6	3.2.1 特殊字符 44
1.3.5 安装 Delphi 87	3.2.2 字符串表示 45
1.3.6 安装 ComponentOne Studio.....10	3.2.3 Delphi 8 可用 Unicode 45
1.3.7 启动 Delphi 811	3.2.4 空格和 Tab、begin...end..... 46
1.4 如何使用 Delphi 帮助文件.....12	3.2.5 关键字 48
1.5 Delphi 8 环境介绍.....13	3.2.6 标明注释 49
1.5.1 Welcome Page14	3.2.7 命令的操作 50
1.5.2 窗体.....14	3.3 变量类型..... 51
1.5.3 工具栏窗口17	3.3.1 整型 51
1.5.4 组件面板 Tool Palette20	3.3.2 浮点型 52
1.5.5 Model View21	3.3.3 布尔型 52
1.5.6 用于窗口开发设计的窗体.....22	3.3.4 字符型 53
1.5.7 程序代码编辑窗口24	3.3.5 字符串类型 53
1.5.8 改变当前的环境设置26	3.4 变量声明..... 55
1.5.9 视觉组件的对齐28	3.4.1 变量的声明方法..... 55
1.5.10 项目选项29	3.4.2 变量的命名规则..... 57
第 2 章 Delphi 操作基础—— 创建项目32	3.4.3 常量声明 59
2.1 创建 VCL Forms Application.....32	3.4.4 数组声明 60
2.2 创建 Windows Forms Application33	3.5 程序流程控制..... 61
	3.5.1 变量赋值 61
	3.5.2 With 语句 61
	3.5.3 比较语句: If..... 63

3.5.4 多选一语句: Case.....	64	4.4.5 选择代码块	101
3.5.5 循环语句(一): Repeat.....	66	4.4.6 找寻最初的声明位置.....	101
3.5.6 循环语句(二): While	67	4.5 调试程序.....	102
3.5.7 循环语句(三): For	67	4.5.1 中断程序	102
3.5.8 继续(Continue)和停止(Break) 语句	69	4.5.2 逐行检查	103
3.5.9 停止执行语句: Exit.....	69	4.5.3 查看变量值	105
3.6 使用子程序	70	4.5.4 编译消息	105
3.6.1 声明过程: Procedure	71	4.6 执行外部程序.....	106
3.6.2 声明函数: Function	72	4.6.1 命令说明	106
3.6.3 声明方法	73	4.6.2 界面说明	107
3.6.4 跳出程序语句: exit	74	4.6.3 程序说明	108
4.6.4 运行结果	108		
第 4 章 程序编写基础	75	第 5 章 键盘及鼠标的使用	110
4.1 如何执行程序	75	5.1 键盘及鼠标简介.....	110
4.1.1 运行程序的方式	75	5.1.1 关于键盘	110
4.1.2 程序的运行步骤	77	5.1.2 关于鼠标	111
4.1.3 使用项目组	78	5.1.3 界面说明	113
4.1.4 程序运行的暂停或终止.....	79	5.1.4 程序说明	113
4.1.5 设置运行时选项	80	5.1.5 运行结果	114
4.1.6 认识类(Class).....	81	5.2 按键触发的事件.....	116
4.1.7 TForm	82	5.2.1 界面说明	116
4.2 项目程序	83	5.2.2 程序说明	116
4.2.1 窗体类型	84	5.2.3 运行结果	117
4.2.2 创建项目	85	5.3 按键的焦点.....	118
4.3 设计窗体	88	5.3.1 界面说明	119
4.3.1 创建窗体	88	5.3.2 程序说明	120
4.3.2 使用 Tool Palette.....	91	5.4 鼠标触发的事件.....	123
4.3.3 Object Inspector 的 Properties 选项卡	92	5.4.1 界面说明	123
4.3.4 Object Inspector 的 Events 选项卡	95	5.4.2 程序说明	124
4.3.5 Windows Forms Designer.....	96	5.4.3 运行结果	125
4.4 编写程序	97	5.5 用鼠标选择颜色.....	126
4.4.1 使用 Tool Palette.....	97	5.5.1 界面说明	127
4.4.2 自动显示代码 (Code Insight).....	98	5.5.2 程序说明	127
4.4.3 设置 Code Insight.....	99	5.5.3 运行结果	128
4.4.4 设置代码环境	99	5.6 实现鼠标的拖放操作.....	128
		5.6.1 界面说明	128
		5.6.2 程序说明	130
		5.6.3 运行结果	131

5.6.4 停靠式拖放	132	8.1.3 运行结果	200
第 6 章 对话框	134	8.2 目录的使用	201
6.1 对话框简介	134	8.2.1 界面说明	202
6.1.1 使用对话框的时机	134	8.2.2 程序说明	203
6.1.2 对话框的使用方法	135	8.2.3 运行结果	204
6.2 显示、输入及确认对话框	135	8.3 文件的属性	205
6.2.1 ShowMessage	135	8.3.1 界面说明	207
6.2.2 InputBox	137	8.3.2 程序说明	208
6.2.3 MessageDlg	138	8.4 文件的信息	213
6.2.4 打开文件对话框	141	8.4.1 界面说明	216
6.2.5 保存文件对话框	144	8.4.2 程序说明	216
6.2.6 打开图片文件对话框	146	8.5 文件的创建与删除	218
6.2.7 字体对话框	148	8.5.1 界面说明	218
6.2.8 颜色对话框	150	8.5.2 程序说明	219
第 7 章 窗体控制与菜单	153	8.6 读取文件	221
7.1 窗体简介	153	8.6.1 界面说明	222
7.1.1 程序文件结构	153	8.6.2 程序说明	222
7.1.2 窗体中的组件	156	8.7 写入文件	226
7.1.3 主窗体(Main Form)	157	8.7.1 界面说明	226
7.1.4 如何使用窗体	158	8.7.2 程序说明	226
7.2 Form.Show	161	8.8 文件群体的文件名更改	229
7.3 Form.ShowModal	164	8.8.1 界面说明	230
7.4 Application.CreateForm	166	8.8.2 程序说明	231
7.4.1 Unit1 程序说明	169	8.9 查找文件	237
7.4.2 Unit2 程序说明	170	8.9.1 界面说明	239
7.4.3 运行结果	171	8.9.2 程序说明	240
7.4.4 MDI 窗口的使用	171	8.9.3 执行结果	242
7.5 Frame 窗口的应用	175	第 9 章 INI 文件及注册表的使用	243
7.6 窗体的外观	180	9.1 INI 文件的设计结构	244
7.7 主菜单(Main Menu)	183	9.1.1 个案研究	244
7.8 快捷菜单(Popup Menu)	187	9.1.2 INI 的数据类型	244
7.9 MDI 与 Menu 的应用	189	9.2 读取与写入 INI 文件	245
7.10 如何用子程序产生窗体	192	9.2.1 界面说明	246
第 8 章 文件及目录的使用	198	9.2.2 程序说明	246
8.1 磁盘的使用	198	9.3 使用 INI 文件来保存	
8.1.1 界面说明	199	窗体信息之一	248
8.1.2 程序说明	199	9.3.1 安装 Sample 的 VCL 组件	248
		9.3.2 界面说明	250

9.3.3 程序说明	250	10.5.1 界面说明	298
9.3.4 第一次执行	254	10.5.2 程序说明	299
9.3.5 第二次执行	255	10.5.3 执行结果	300
9.4 Delphi 中的注册表	255	10.6 在 DBGrid 组件上画图	300
9.4.1 注册表的设计结构	255	10.6.1 界面说明	301
9.4.2 注册表的数据类型	256	10.6.2 程序说明	302
9.5 使用 RegEdit 注册表编辑器	257	10.6.3 执行结果	304
9.6 读出与写入注册表	259	第 11 章 多线程	305
9.6.1 界面说明	260	11.1 何谓多线程	305
9.6.2 程序说明	260	11.1.1 多线程的问题	305
9.7 使用 INI 文件来保存		11.1.2 非多线程和多线程	
窗体信息之二	264	的差别	306
9.7.1 界面说明	264	11.1.3 多线程原理	307
9.7.2 程序说明	264	11.2 非多线程程序	308
9.8 记录应用程序的序列号	271	11.2.1 界面说明	308
9.8.1 界面说明	271	11.2.2 程序说明	308
9.8.2 程序说明	271	11.2.3 执行结果	309
9.8.3 执行结果	275	11.3 创建一个多线程单元	309
第 10 章 开发绘图应用程序	277	11.4 执行一个多线程程序	310
10.1 看图程序的制作	277	11.4.1 界面说明	310
10.1.1 Win 3.1 VCL 对象	278	11.4.2 程序说明	311
10.1.2 界面说明	278	11.4.3 执行结果	314
10.1.3 程序说明	282	11.5 多线程的同时访问	314
10.1.4 执行结果	283	11.5.1 程序说明	314
10.2 制作游戏动画	285	11.5.2 执行结果	316
10.2.1 界面说明	285	11.6 多线程的实例研究——	
10.2.2 程序说明	286	跑步比赛	316
10.2.3 执行结果	288	11.6.1 界面说明	317
10.3 自定义一个画图程序	289	11.6.2 程序说明	317
10.3.1 Canvas 简介	289	11.6.3 执行结果	320
10.3.2 界面说明	290	第 12 章 异常处理	322
10.3.3 程序说明	291	12.1 异常处理	322
10.3.4 执行结果	293	12.1.1 何谓异常处理	322
10.4 文字分解	293	12.1.2 没有异常处理时的流程	323
10.4.1 界面说明	294	12.1.3 加上异常处理的流程	323
10.4.2 程序说明	295	12.2 try...except...end	324
10.4.3 执行结果	296	12.2.1 界面说明	326
10.5 条形码的制作	297		

12.2.2 程序说明	327	14.1.1 .NET 的影响力	353
12.3 try...finally...end	328	14.1.2 .NET 的结构	355
12.3.1 界面说明	330	14.2 .NET Framework	355
12.3.2 程序说明	330	14.2.1 何谓 .NET Framework	356
12.4 嵌套的异常处理	332	14.2.2 从 Windows API 到 .NET Framework	357
12.4.1 界面说明	334	14.2.3 从程序语言谈起	360
12.4.2 程序说明	335	14.2.4 不同程序语言的互通	361
12.5 文件异常处理的使用	337	14.3 .NET 的语言	362
12.5.1 界面说明	337	14.3.1 Visual Basic	362
12.5.2 程序说明	338	14.3.2 C#	362
12.5.3 执行结果	338	14.3.3 C++	362
12.6 文件绝对处理的使用	339	14.3.4 JavaScript	363
12.6.1 程序说明	339	14.3.5 Web 窗体	363
12.6.2 执行结果	339	14.3.6 Windows 窗体	363
第 13 章 DLL 的设计方式	341	14.3.7 XML Web 服务	363
13.1 DLL 的设计结构	341	14.3.8 XML 支持	363
13.1.1 DLL 的结构	341	14.4 Windows 窗体简介	364
13.1.2 如何制作一个 DLL 文件	342	14.5 打开 WinForm	367
13.2 快速创建 DLL 文件	343	第 15 章 Windows 窗体	369
13.2.1 生成 DLL 项目	344	15.1 ShowMessage	369
13.2.2 Library1 程序说明	344	15.1.1 界面说明	370
13.2.3 Unit1 程序说明	345	15.1.2 程序说明	370
13.2.4 编译 DLL 文件	346	15.1.3 执行结果	371
13.2.5 界面说明	346	15.2 Keyboard	371
13.2.6 程序说明	347	15.2.1 界面说明	373
13.2.7 执行结果	348	15.2.2 程序说明	375
13.3 创建 DLL 窗体	348	15.2.3 执行结果	376
13.3.1 制作结构	349	15.3 Mouse	376
13.3.2 界面说明	349	15.3.1 界面说明	377
13.3.3 Library1 程序说明	350	15.3.2 程序说明	378
13.3.4 Unit1 程序说明	351	15.3.3 执行结果	379
13.3.5 界面说明	351	15.4 Dialog	379
13.3.6 程序说明	351	15.4.1 加入 Microsoft. VisualBasic	380
13.3.7 执行结果	352	15.4.2 界面说明	381
第 14 章 .NET 基本介绍	353	15.4.3 程序说明	382
14.1 .NET Framework 简介	353	15.5 Menu	386

15.5.1 界面说明	386	17.2.1 打开 ASP.NET 应用程序	412
15.5.2 程序说明	387	17.2.2 ASP.NET 项目的选项	413
15.6 MDI	393	17.3 设计 WebForm	415
15.6.1 界面说明	393	17.3.1 设计 HTML	415
15.6.2 程序说明	395	17.3.2 两个 HTML 文件	417
第 16 章 .NET 的注册表和文件	398	17.3.3 执行 WebApplication	419
16.1 使用 WinForm 控制注册表	398	17.3.4 HTML 组件	419
16.1.1 界面说明	398	附录 Delphi 常用函数简介	421
16.1.2 程序说明	398	A.1 字符串处理函数(1)	421
16.2 使用 Visual Basic 控制 Registry	402	A.2 字符串处理函数(2)	427
16.2.1 界面说明	402	A.3 文件相关函数	435
16.2.2 程序说明	403	A.4 不同格式之间的转换	444
16.3 使用 Visual Basic 写入文件	405	A.5 浮点数格式的转换	452
16.3.1 界面说明	406	A.6 日期时间函数(1)	460
16.3.2 程序说明	406	A.7 日期时间函数(2)	470
16.3.3 执行结果	407	A.8 系统内定的颜色值	483
16.4 使用 Visual Basic 读取文件	408	A.9 颜色的值	483
16.4.1 界面说明	408	A.10 颜色相关函数	484
16.4.2 程序说明	409	A.11 指针的代号	486
16.4.3 执行结果	410	A.12 指针相关函数	486
第 17 章 ASP.NET Web Framework	411	A.13 系统信息变量	489
17.1 ASP.NET	411	A.14 日期变量名称(1)	490
17.2 打开 ASP.NET 项目	412	A.15 日期变量名称(2)	491
		A.16 系统的错误信息	492

第 1 章 Delphi 简介

Delphi 是由 Borland 公司所推出的程序开发工具，历经了十个年头以后，当前已经是第 8 个版本，这个版本是 Delphi 有史以来最大的改变，Borland 不再坚持孤军奋战，而是选择与微软合作，使用了与 .NET 兼容的开发方式，无论是 Delphi 的界面风格、操作方式，还是开发方式，都有全新的改变，而且完全向下兼容，可说是具有 Delphi 的强大功能，且兼容于 .NET 的各项标准。

1.1 什么是 Delphi

Windows 上的程序开发工具虽然相当多，但是有名、好用并且确实具有强大功能的，只有少数几种，微软本身是 Windows 的生产厂商，所提供的开发工具自然比较受到欢迎(如 Visual Basic、Visual C++)，其次就是 Borland 的 Delphi、C++ Builder、J Builder。

在 Visual Basic 刚面世时，Delphi 也随之推出，一推出就引起轰动，不但有银色子弹的美誉，更号称 VB Killer。其他开发工具能实现的功能，Delphi 几乎都能实现；其他开发工具能快速实现的功能，Delphi 甚至能够更快；其他开发工具可以支持的，Delphi 全部可以支持。当然，Delphi 也并不是十全十美，但以开发工具来说，选择 Delphi 不会让您后悔。

Delphi 是一种 Windows 平台上的程序开发工具，使用的是 Object Pascal 语法，这是一种面向对象的开发语言，可以利用面向对象的方式延伸出更多的对象，供开发者使用。Delphi 还使用了大量 VCL(Visual Component Library，可视化组件库)，使开发者能更快地开发出程序，并且还可以自创 VCL，使所开发的程序更具灵活性。

使用 Delphi 编写的程序，经过编译后形成的可执行文件可以独立运行，虽然容量会比较大一点，但运行效率比较高，而且可以直接复制到其他计算机上，不需安装即可执行(使用数据库的程序除外)，因此使用起来相当方便。

1.1.1 什么是 .NET

.NET 是微软公司所推出的程序开发工具，它涵盖了许多语法，但不管使用哪种语法，最后所编译出来的可执行文件都是相通的，甚至可以在不同的平台上执行。.NET 并非原创，但它会彻底改变你的生活，这其中包含正面的意义和反面的意义。

正面意义：

- 函数库的内容将全面统一。
- 不用担心内存的冲突与错误。
- 真正解决 COM 的 DLL Hell 问题。

- 以微软为中心的开发平台。

反面意义：

- 函数库的规模将过于庞大。
- 程序与硬件之间多了一层沟通接口。
- 需要学习更多的语法和技术。
- 许多继承代码将不会在.NET 上执行。

.NET Framework 有两个主要的组件，一个是 CLR(Common Language Runtime, 公共语言运行库)和 .NET Framework class library(.NET Framework 类库)。CLR 是.NET Framework 的主要基础，编写好了程序，编译后要执行，就要通过 CLR 才能运作，CLR 就像是 Java 的虚拟机，它使程序能够运行，并且自动管理内存的问题。

1.1.2 什么是 Delphi .NET

Delphi .NET 也就是 Delphi 8，它不但支持 Microsoft Win32，而且兼容于.NET Framework，开发人员可利用 Delphi 程序语言和可视化组件函数库(VCL)来进行程序开发。可以用 Delphi for .NET 来建立在.NET 平台下新的 Delphi 应用程序，并将以前就开发完成的 Win32 应用程序移植到.NET 之中。Delphi 开发者若要转换到.NET 平台，只要使用现有 Delphi 的技巧，就能轻松将其现有代码移植到.NET 中。

早期的 Delphi 就已经有许多成熟的 VCL，这些 VCL 在.NET 环境中同样可以顺利使用，最重要的是，这些 VCL 并不需要进行太多修改，其中大部分在不需进行任何修改的情况下，就能经过重新编译，快速移植到.NET 的 Win32 平台中来使用，Borland 尽一切可能让原有的 VCL 和 VCL for .NET 能够并存，并且保留原有的风貌。图 1.1 所示为 Delphi 的发展简介。

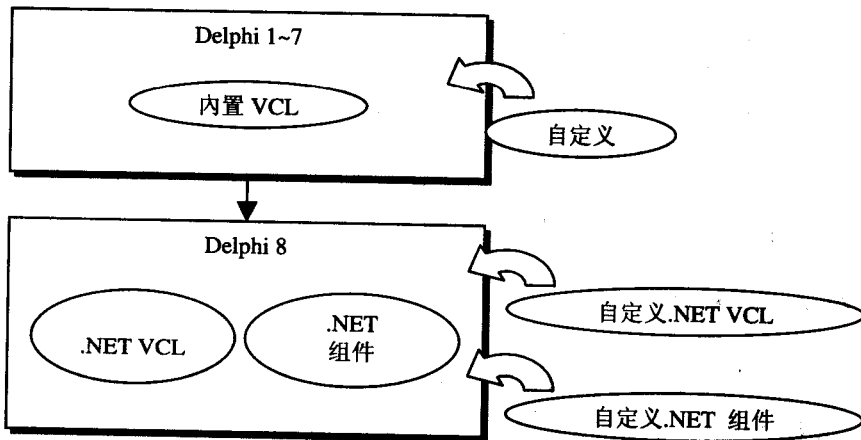


图 1.1

在 Delphi 8 中，VCL for .NET(.NET VCL)是 VCL 的最大子集，原有 VCL 都以全新的方式置于 VCL for .NET 之中，例如原先的 ComCtrls.dcu，现在改用 Borland.Vcl.CommCtrl.dcu 全新的文件名和格式，来与 Borland 的 VCL 区分，在.NET 环境中，更要清楚地分辨其中的差异。

Delphi 8 的第三方组件和 Win 32 组件, 需要利用 .NET 的 CLR 来在 Windows 中执行, 假如我们想要在 Linux 中执行 Delphi 8 的程序, 就要使用 Delphi 的 CLX 组件。当前 Delphi 的 VCL 都使用符合 .NET 的原则, 一旦在 Windows 下编译程序, 就可以生成 .NET 环境的源程序代码, 该代码与在 .NET 中所开发出的结果所产生的源程序代码是兼容的, 所以 Delphi 可以开发 .NET 本身的所有功能。

1.2 Delphi 8 新特色

通常一个软件的新版本与旧版本相比只会新增一些简单的功能, 但 Delphi 并非如此, 它的每一个新版本, 都有其重要意义, 从 Delphi 3 开始, 每一次的新版本都设置一个目标, 例如 Delphi 7 可以在 Kylix 上开发及执行程序, 而 Delphi 8 更是支持 .NET Framework 的开发功能, 使得原本使用 Delphi 的开发人员拥有更多的开发方式, 并且使习惯于 .NET 开发的开发人员也可很快学会使用 Delphi 8 开发程序。

Delphi 8 集成了原有 Delphi 的设计环境, 并且首次改变原有的开发界面, 使 Delphi 和 .NET 的开发环境得以并存, .NET 的开发环境以 Visual Basic 的开发环境为基础, 使用户可以选择要用 Delphi 旧版本的开发环境, 也可以选择 .NET 开发环境。

虽然 Delphi 8 的开发环境可以自行设置, 但也只是部分而已, 主菜单功能还是以原有 Delphi 的主菜单为主, 快捷键也是按照原 Delphi 的设置来使用, 只有属性窗口和组件窗口略作调整, 虽然整体感觉是全新的, 但是 Delphi 原来的操作界面就很实用, 所以仍可看到原来的影子。

在主菜单的右方可以看到一个如图 1.2 所示的下拉列表框, 其中的 Classic Undocked 选项就是 Delphi 原有环境的使用方式, 而 Debug Layout 是适合对程序进行调试时所采取的环境, Default Layout 是新的环境样式, 它比较像 .NET 的环境样式, 也可以在菜单 View | Desktops 中, 看到相同的功能。

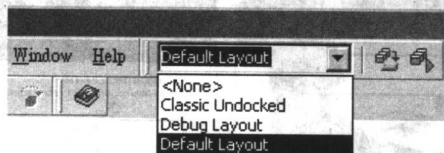


图 1.2

(1) 新的需求(Defining Requirements)

Delphi 8 提供了一个高品质的设计环境, 无论是使用 Delphi 的开发环境开发, 还是使用 Web 的环境来开发, 在经由 Delphi 编译后, 都能依照用户的需求, 生成恰当的结果。

(2) 模块化应用程序(Modeling Applications)

程序开发已经走向团队化的开发方式, 使用 UML 方式开发程序可以使程序更为模块化, Delphi 提供了以 UML 为基础的开发环境, 使开发程序模块化, 便于开发和修改。

(3) 新的用户界面(Designing User Interfaces)

Delphi 8 在保留原有优良开发界面的基础上, 提供了更多全新的操作方式, 让开发人

员可以更轻松地设计程序,利用更简单的方式设计出应用程序或网页。

(4) 自动生成代码(Generating and Editing Code)

无论在设计画面或页面中,Delphi 都能快速地自动生成合适的程序代码,为开发人员减轻许多的负担,其优秀的程序编辑器使用户在撰写或调试程序时更加得心应手。

(5) 编译到部署程序一体化(Compiling, Debugging, and Deploying Applications)

Delphi 的编译速度相当快,在设计程序完成后编译,Delphi 提供了一个良好的调试环境,使开发人员可以随时掌控程序运行时的变量内容,帮助开发人员确认程序运行正常,并且可利用 InstallShield Express 创建安装程序。

(6) 控制开发制度与监控代码变化(Controlling Access and Tracking Changes to Code)

利用团队工具可以掌控开发进度,并且随时监控程序代码,确保在团队开发时,使每个开发小组成员的任务进度都能与项目管理者显示结果相同。

(7) 支持 .NET 框架

提供 .NET 框架的 CLR,让我们可以使用 Delphi 的程序代码,编译成和 .NET 相同的代码,而且也可以使用 .NET 的各种库,使程序设计更为灵活。

1.3 安装 Delphi 8

1.3.1 Delphi 8 安装界面

Delphi 8 安装光盘插入光驱后会自动运行,主界面的左边有 5 个选项,单击 Install Products 来安装 Delphi,选择 Install Products 后,出现第 2 个界面,其中包括 6 个选项,如图 1.3 所示。

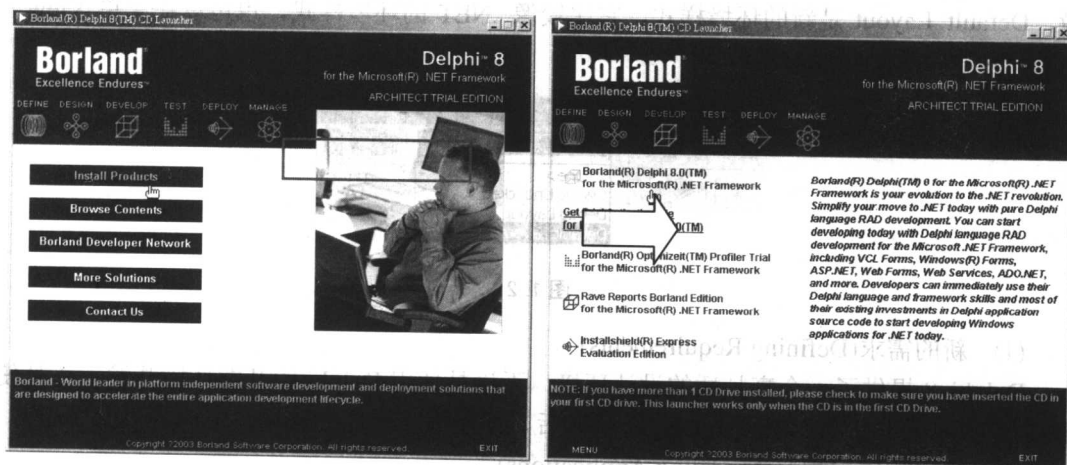


图 1.3

单击如图 1.4 所示的 Borland Delphi 8.0 for the Microsoft .NET Framework 即可安装 Delphi 所需的所有程序,包含 Microsoft .NET Framework SDK 1.1、InterBase、Rave Reports 等。



图 1.4

1.3.2 安装 Microsoft .NET Framework SDK 1.1

接下来需要安装 Microsoft .NET Framework SDK 1.1, 因为 Delphi 8 支持 .NET Framework, 因此这是必装的程序, 安装 SDK 后, Delphi 才会具备 .NET 所需的功能, 在光盘的 DOTNETSDK 文件夹中可以找到 setup.exe 文件, 双击执行, 如图 1.5 所示。

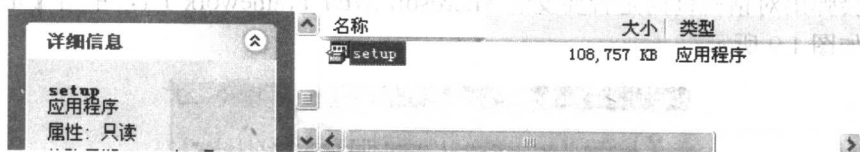


图 1.5

此时会解压 Microsoft .NET Framework SDK, 如图 1.6 所示。

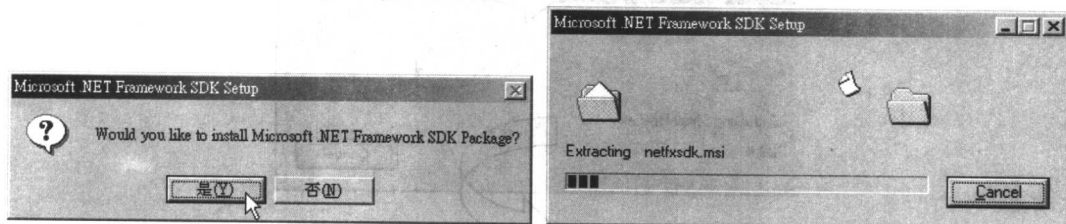


图 1.6

解压 Microsoft .NET Framework SDK 完成后就开始安装, 单击 Next 按钮进行下一步, 确定后开始安装, 如图 1.7 所示。

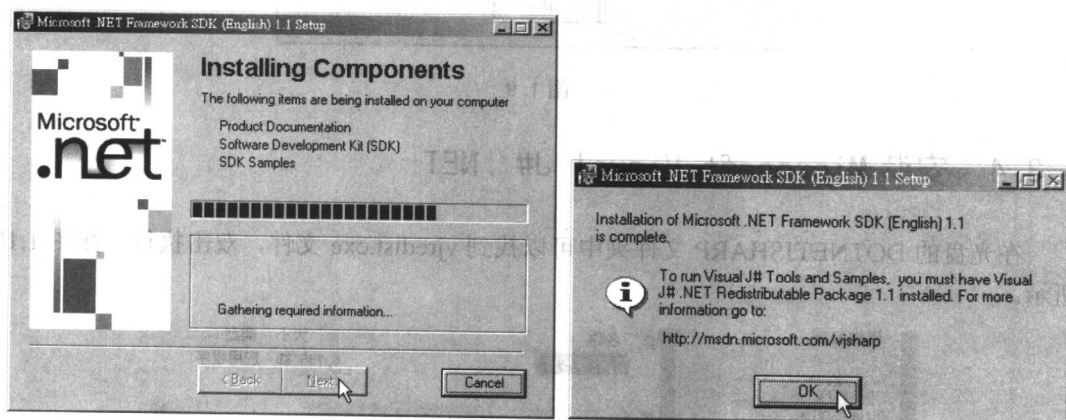


图 1.7

1.3.3 安装 Microsoft .NET Framework 1.1

在光盘的 DOTNETREDIST 文件夹中可以找到 dotnetfx.exe 文件，双击执行，如图 1.8 所示。

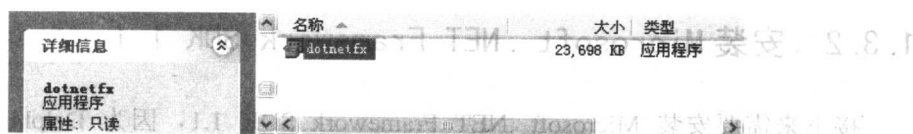


图 1.8

程序会弹出对话框询问是否要安装 Microsoft .NET Framework 1.1，单击【是】按钮进行安装，如图 1.9 所示。

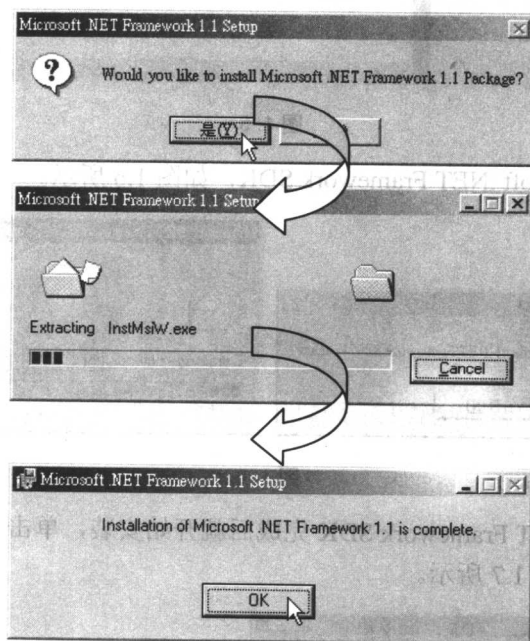


图 1.9

1.3.4 安装 Microsoft Visual J# .NET

在光盘的 DOTNETJSHARP 文件夹中可以找到 vjredist.exe 文件，双击执行，如图 1.10 所示。

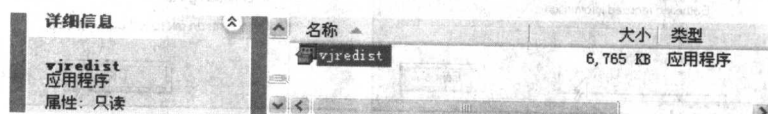


图 1.10

接着会询问是否要安装 Microsoft Visual J# .NET, 单击【是】按钮进行安装, 如图 1.11 所示。

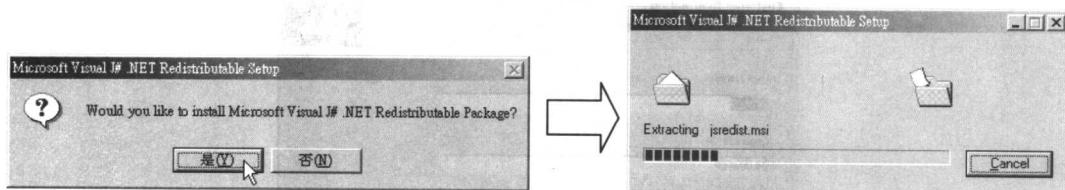


图 1.11

1.3.5 安装 Delphi 8

在安装 Microsoft .NET Framework SDK 1.1 后, 才能开始安装 Delphi 8, 而 Delphi 8 本身的安装也需要许多必备的软件, 因此整个步骤也需要装数个软件才能完成。

在 Installation Wizard 中进行到 Welcome to Borland Delphi 8.0 页面时, 就正式进入了 Delphi 8 的标准安装, 单击 Next 按钮, 开始安装, 如图 1.12 所示。

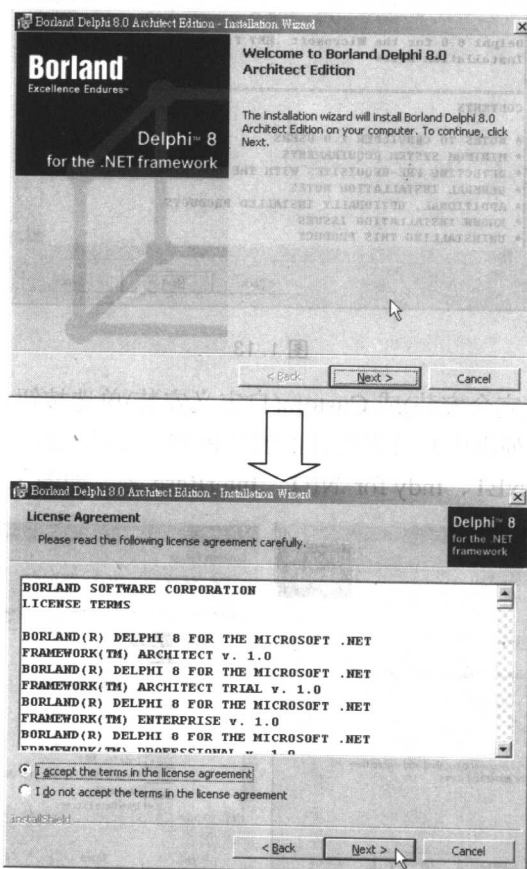


图 1.12