

Delphi 6.0

高级编程

傅红波 编著



多媒体教学盘

Delphi 6.0 高级编程

傅红波 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

本书以实例为主线,辅以必要的技术背景要点来讲解 Delphi 6.0 编程的精髓。本书实例偏重于多媒体和网络程序开发,其中涉及到 OpenGL、Indy 组件、XML、Web Broker 和 WebSnap 等技术。本书还讲述了 Delphi 中的 SQL 编程,在两个短小精练的程序中充分运用了 Delphi 数据开发技术,包括 BDE 组件的 ADO 组件的使用。通过本书,让读者在编程过程中充分享受编程的乐趣,同时掌握最新的编程思想。

本书在选材和编写中主要顾及具有一定 Delphi 编程经验的读者。由于本书以实例形式讲解 Delphi 6.0 的使用,所以对于初学者理解 Delphi 6.0 的应用程序开发也是很有益处的。

为了配合读者更好的领会书中内容,我们制作了多媒体教学光盘。

未经许可,不得以任何方式复制或抄袭光盘及配书之部分或全部内容。

版权所有,翻版必究。

书 名: Delphi 6.0 高级编程

总 策 划: 北京暴风雪科技有限公司

创 作: 傅红波

电脑制作: 北京暴风雪科技有限公司

出 版: 中科多媒体电子出版社

印 刷: 北京安泰印刷厂

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 21

版 次: 2002 年 3 月第 1 版

本 版 号: ISBN 7-900084-62-2

定 价: 38.00 (本版 CD)

凡有质量问题的,请向购买处调换。

电话: (010) 62480269、13801352164

前言

Delphi 是 Borland 公司开发的全新可视化编程环境。Delphi 提供了新颖的可视化设计工具,采用面向对象的方法把 Windows 编程的复杂性封装起来,实现了可视化编程与 Object Pascal 语言的有机集成。可以说,Delphi 既具有 Visual C++ 的强大功能,又具有 Visual Basic 易于使用的特点,无怪乎倍受程序员的青睐。使用 Delphi,无疑会大大提高编程效率。如果你选择 Delphi,那你就应该庆幸,因为从此编程将不再是枯燥乏味的工作了。

Borland 公司推出 Delphi 6.0 之后,很多人都迷恋于它的强大功能。Delphi 6.0 除扩充了它对数据库应用程序开发的支持,又增加了对当今流行网络技术的支持,例如 XML、SOAP 等等。可以说 Delphi 6.0 的推出一方面是 Borland 公司自己产品的进步,另一方面也是向 Microsoft 的 VS.Net 的一种挑战。随着本书学习的深入,读者将会发现 Delphi 6.0 有很多功能强于 VS.Net,这不仅体现在 Delphi 6.0 操作简便上,也体现在 Delphi 6.0 的功能强大上。

鉴于 Delphi 的优良特性和顺应广大用户的要求(目前国内还无正式介绍 Delphi 6.0 新增功能的书籍),我在积累了大量开发经验的基础上写了本书,希望读者能够通过本书掌握 Delphi 6.0 的使用。

本书的主要目的是帮助具有一定 Delphi 编程经验的程序员熟悉 Delphi 6.0 的程序开发,以及如何在 Delphi 6.0 中开发功能强大网络应用程序以及 Web 服务器应用程序。本书以实例为主线,读者在使用本书时,既可以循序渐进地阅读,也可以直接查阅自己感兴趣的部分。

第 1 章:简单介绍了 Delphi 6.0 的新增功能,使得读者对于 Delphi 6.0 所能做的事情有大致了解。

第 2 章:通过一个媒体播放器和一个弹球游戏程序的开发,使读者初步领略到 Delphi 在多媒体编程方便的强大功能。

第 3 章:数据库编程是 Delphi 最值得称道的,本章详细讲述了 Delphi 中的 SQL 编程,在两个短小精练的程序中充分运用了 Delphi 数据开发技术,包括 BDE 组件的 ADO 组件的使用。

第 4 章:OpenGL 是目前比较流行的游戏程序编程语言,结合 Delphi 6.0 对窗口设计的强大功能,可以使程序员快速开发具有交互式的 OpenGL 程序。本章通过实例演示了如何使用 Delphi 开发 OpenGL,并简单介绍了 OpenGL 中的相关概念。

第 5 章:Internet 技术无疑是当今计算机的最大热点。本章使用 Delphi 提供的组件开发了目前较为流行的聊天室程序和邮件收发程序。

第 6 章:Delphi 6.0 新增了 Indy 组件,为开发 Internet 应用程序提供了另一途径。本章通过 Gopher 程序和 FTP 程序的开发,让读者领略 Indy 组件的奇妙。

第 7 章:XML 是当今流行的传输数据的格式,Delphi 6.0 中增加使用 XML 数据源的支持。通过实例演示了如何使用 Delphi 6.0 提供的封装了 XML 技术的组件来开发 XML 和数据集之间的转换程序。

第 8 章:开发 Web 服务器应用程序已经成为企业目前最流行,也是最重要的工作。Delphi 早在 5.0 版本中引入 Web Broker 技术,使得可以快速开发 Web 服务器应用程序。在 Delphi 6.0

中又引入了 WebSnap 技术，它是基于 Web Broker 框架的，但同时又增加很多功能，包括服务器端脚本语言的使用等等。本章首先比较了这两种技术，然后通过实例演示了如何使用 Web Broker 技术开发 Web 服务器应用程序。

第 9 章：WebSnap 技术是 Delphi 6.0 的一大卖点。WebSnap 功能强大，但在理解上有一定程度的难度。本章使用两个实例详细阐述了 WebSnap 技术的功能。读者应该细细体会 WebSnap 技术的强大功能。

第 10 章：Delphi 6.0 提供了对交叉式平台应用程序开发的支持。在 Delphi 6.0 中开发的应用程序可以在 Linux 下重新编译然后运行。不过这个应用程序是建立于 CLX 组件结构之上的。本章演示了如何使用 CLX 组件开发交叉式平台应用程序的开发。

在本书的编写过程中，我们尽量注意减少冗长无味的说明，代之以具体使用的实例演示。本书注重开发实例，通过实例，引导读者把握 Delphi 6.0 的精髓所在。通过本书，初学者可迅速加入 Delphi 高级用户的行列，有一定使用经验的读者也可通过本书掌握 Delphi 深层次的开发方法，学会用更有技巧性的方法开发出高水平的 Delphi 应用软件。

在读者阅读本书之前，需要掌握 Object Pascal 语法，有一定编程基础，对 Windows 可视化编程有所了解，有一定的 Delphi 编程知识。

在教学光盘中含有书中的全部源程序，这些程序都是经过严格的调试和测试的。读者可以对照本书，查看程序运行结果，这样有助于对复杂程序的理解。

限于作者水平以及由于编写时间紧促，难免在内容选材和叙述上有不当之处，希望读者谅解并批评指正。

编者

2002 年 1 月

目 录

第 1 章 Delphi 6.0 新特性.....1

- 1.1 Delphi 基本特性.....1
- 1.2 Delphi 新特性.....2
 - 1.2.1 IDE 2
 - 1.2.2 Internet6
 - 1.2.3 XML 支持（企业版） 9
 - 1.2.4 Compiler..... 10
 - 1.2.5 COM/Active X（专业版和企业版） 10
 - 1.2.6 Database support（专业版和企业版） 11
 - 1.2.7 CORBA（企业版）12
 - 1.2.8 Custom Variants 12
 - 1.2.9 Cross-platform development（专业版和企业版） 12
 - 1.2.10 Translation tools（企业版） 12
 - 1.2.11 Deployment change..... 13
 - 1.2.12 XML/XSL 的支持.....13
 - 1.2.13 Web Service 的向导和组件 14
 - 1.2.14 执行时函数库..... 15
 - 1.2.15 第 14 版的 Object Pascal 编译器 15
- 1.3 本章小结.....16

第 2 章 多媒体程序设计.....17

- 2.1 媒体播放器.....17
 - 2.1.1 技术背景.....17
 - 2.1.2 新建项目.....19
 - 2.1.3 界面设计.....19
 - 2.1.4 代码编写.....21
 - 2.1.5 程序运行.....24
- 2.2 弹球游戏.....25
 - 2.2.1 技术背景.....26
 - 2.2.2 新建项目.....27
 - 2.2.3 界面设计.....27

2.2.4 代码编写.....30

2.2.5 程序运行.....37

2.3 本章小结.....38

第 3 章 SQL 编程.....39

- 3.1 SQL 的语言简介.....39
 - 3.1.1 SQL 特点.....39
 - 3.1.2 数据定义.....40
 - 3.1.3 数据查询.....42
 - 3.1.4 数据更新.....43
 - 3.1.5 数据控制.....44
- 3.2 本地数据库编程.....44
 - 3.2.1 技术背景.....44
 - 3.2.2 创建本地数据库 Song 51
 - 3.2.3 新建项目.....54
 - 3.2.4 界面设计.....55
 - 3.2.5 代码编写.....58
 - 3.2.6 程序运行.....61
- 3.3 远程数据库编程.....62
 - 3.3.1 技术背景.....64
 - 3.3.2 存储过程和触发器建立.....66
 - 3.3.3 新建项目.....70
 - 3.3.4 界面设计.....70
 - 3.3.5 代码编写.....73
 - 3.3.6 程序运行.....74
- 3.4 本章小结.....76

第 4 章 OpenGL 编程.....77

- 4.1 OpenGL 简介..... 77
 - 4.1.1 OpenGL 基本操作..... 77
 - 4.1.2 OpenGL 语法规则..... 78
 - 4.1.3 Delphi 中的 OpenGL 语法... 78
- 4.2 简单的 OpenGL 动画程序..... 79
 - 4.2.1 技术背景.....79
 - 4.2.2 新建项目.....89
 - 4.2.3 界面设计.....90

4.2.4 代码编写.....	90	6.3 本章小结.....	166
4.2.5 程序运行.....	95		
4.3 奥运五环程序.....	96	第7章 XML 技术.....167	
4.3.1 技术背景.....	96	7.1 XML 技术.....	167
4.3.2 新建项目.....	103	7.1.1 XML 历史.....	167
4.3.3 界面设计.....	103	7.1.2 XML 下的 Web 体系结构	169
4.3.4 代码编写.....	103	7.1.3 文件结构.....	170
4.3.5 程序运行.....	110	7.1.4 XML 语法.....	171
4.4 本章小结.....	110	7.2 XML Transform.....	171
第5章 Internet 应用程序.....111		7.2.1 技术背景.....	172
5.1 聊天室.....	111	7.2.2 新建项目.....	180
5.1.1 技术专题.....	111	7.2.3 界面设计.....	180
5.1.2 新建项目.....	115	7.2.4 代码编写.....	185
5.1.3 界面设计.....	115	7.2.5 程序运行.....	186
5.1.4 代码编写.....	118	7.3 XML Provider.....	187
5.1.5 程序运行.....	123	7.3.1 技术背景.....	187
5.2 Email 收发系统.....	124	7.3.2 新建项目.....	188
5.2.1 技术背景.....	124	7.3.3 界面设计.....	188
5.2.2 新建项目.....	126	7.3.4 代码编写.....	191
5.2.3 界面设计.....	126	7.3.5 程序运行.....	192
5.2.4 代码编写.....	132	7.4 XML Client.....	194
5.2.5 运行程序.....	137	7.4.1 技术背景.....	194
5.3 本章小结.....	137	7.4.2 新建项目.....	195
		7.4.3 界面设计.....	195
第6章 Indy 组件的使用.....138		7.4.4 代码编写.....	197
6.1 用 Indy 组件建立 Gopher 服务器和客户		7.4.5 程序运行.....	198
端.....	138	7.5 本章小结.....	199
6.1.1 技术背景.....	138		
6.1.2 新建项目.....	139	第8章 Web Broker.....200	
6.1.3 界面设计.....	139	8.1 Web 服务器应用程序.....	200
6.1.4 代码编写.....	142	8.1.1 关于 Web Broker 和	
6.1.5 程序运行.....	149	WebSnap.....	200
6.2 用 Indy 组件开发 FTP 客户端应用程		8.1.2 Internet 术语 (Terminology)	
序.....	150	和标准 (Standards)	201
6.2.1 技术背景.....	151	8.1.3 HTTP 服务器.....	202
6.2.2 新建项目.....	152	8.1.4 Web 服务器应用程序类型	203
6.2.3 界面设计.....	152	8.1.5 调试服务器应用程序.....	204
6.2.4 代码编写.....	157	8.2 Web Broker 技术.....	207
6.2.5 程序运行.....	166	8.2.1 建立 Web Broker 结构的 Web	
		服务器应用程序.....	207

8.2.2 Web Broker 应用程序的结 构.....	209	10.3 文本浏览器.....	294
8.2.3 Web Dispatcher.....	209	10.3.1 技术背景.....	295
8.2.4 动作项 (Action Item) ...	210	10.3.2 新建项目.....	295
8.2.5 产生响应消息的内容.....	215	10.3.3 界面设计.....	296
8.3 AlbumSearch.....	219	10.3.4 代码编写.....	297
8.3.1 新建项目.....	220	10.3.5 程序运行.....	299
8.3.2 界面设计.....	220	10.4 本章小结.....	300
8.3.3 代码编写.....	221		
8.3.4 程序运行.....	223		
8.4 本章小结.....	225		
第 9 章 WebSnap.....	226		
9.1 WebSnap 技术.....	226		
9.1.1 用 WebSnap 创建 Web 服务器 应用程序.....	226		
9.1.2 Adapter 组件.....	231		
9.1.3 Page producers.....	232		
9.1.4 WebSnap 中的服务器端的脚 本.....	232		
9.1.5 响应请求.....	235		
9.2 创建简单的 WebSnap 程序.....	239		
9.2.1 新建项目.....	240		
9.2.2 界面设计.....	241		
9.2.3 运行程序.....	249		
9.3 Photo Gallery 程序.....	250		
9.3.1 新建项目.....	252		
9.3.2 界面设计.....	253		
9.3.3 代码编写.....	259		
9.3.4 程序运行.....	286		
9.4 本章小结.....	288		
第 10 章 CLX 编程.....	289		
10.1 CLX 和 VCL 组件比较.....	289		
10.2 LCD 秒表.....	290		
10.2.1 技术背景.....	290		
10.2.2 新建项目.....	291		
10.2.3 界面设计.....	291		
10.2.4 代码编写.....	292		
10.2.5 程序运行.....	294		



第1章 Delphi 6.0 新特性



我们没有必要详细研究 Delphi 的发展历史，显然也不会有人对此感兴趣，但当我们面对一个比较完美和成熟的软件时，却自然会提出这样的要求。就其发展历史来说，Delphi 的发展基础能够给人以足够的信心。Turbo Pascal 编译器是 Delphi 的鼻祖，甚至是 Turbo C/C++、Borland C++ 与 Delphi 也存在不可分隔的关系。实际上，Delphi 就是这些成熟技术的必然结果。

Borland 公司的 Delphi 不仅在支持 Pascal 语言方面做的非常出色，甚至还在此基础上对 Pascal 做出了非常有益的改进，从而使 Delphi 所支持的 Object Pascal 语言可以与 C/C++ 媲美，其高效性也得到根本改善。

更令人高兴的是，自 Borland 公司发布 Delphi 1.0 以来，Delphi 版本的更新速度和功能的改善程度几乎令人吃惊，这足以说明 Borland 公司对 Delphi 的信心。而这同时带来的另一个非常重要的结果是 Delphi 的基础——VCL 得到极大改善，即使专业程序员也很难对 VCL 的改进提出任何意见。然而在这样的情况下，Delphi 的版本还得到不断更新，其 VCL 得到的改进也非常令人惊叹。继 Delphi 5.0 之后，Borland 公司终于在 2001 年推出了 Delphi 6.0。Delphi 6.0 不但对于 Borland 来说是一个重要的产品，而且对于现在的视窗平台应用程序的开发也是一个意义重大的产品。因为当 Microsoft 全力发展 .NET 虚拟环境，而 Borland 本身也往 Linux 平台上开发 RAD 工具——Kylix 的时候，由 Borland 窗口开发小组所研发的原生视窗开发工具正好填补了仍占多数窗口程序员的需要。

1.1 Delphi 基本特性

在介绍 Delphi 6.0 的新特性之前，先来简单介绍一下 Delphi 自身具有的特性。Delphi 作为面向对象的可视化开发工具，具有很多令人称道的特性。

首先 Delphi 是一个可视化、快速的应用程序开发工具，它具有高效、优化的源代码编辑器。可视化的集成开发环境和可扩展的数据库技术，适用于各种类型的可视化程序开发。

Delphi 使用的语言是 Object Pascal，即面向对象的 Pascal 语言。Object Pascal 是一种面向对象的、结构化的编程语言。Inprise 公司在 Pascal 方面所做的大量工作使之可以产生紧凑、高效的执行代码。同时，Pascal 编译语言的运行速度远远优于 C++ 编译器所生成代码的运行速度，更不要提及解释性语言 Visual Basic 了。Delphi 编译程序的速度之快，是大家有目共睹的。

Delphi 具有完善的异常处理机制，可自动处理各种异常。它的处理异常的结构是 try...except...finally，这是目前处理异常的标准结构。例如 Java 语言，甚至目前较为流行的 C# 都采用了此结构处理异常。Microsoft 公司在 VS.net 中对 Visual Basic 引入了该异常处理结构。种种迹象表明，Delphi 中的技术是处于领先水平的。

Delphi 最让人称道的就是它简单易用，并且功能强大数据库应用程序开发技术。它支持各种类型的数据库（桌面数据库和关系数据库），如 Paradox、Access、Oracle、SQL Server、

Sybase 等。

Delphi 具有操作系统无关性。这里所指的“操作系统无关性”不是指像 Java 语言那样一次编译后即可在 Windows、Mac、Linux 和 Unix 等操作系统中的 Java 虚拟环境中运行,而是指在 Windows 95/98 环境下开发的 Delphi 应用程序可以直接在 Windows NT 环境下运行,反过来也可以。另外 Delphi 6.0 中提供了 CLX 技术使得在 Windows 环境下开发的应用程序能够在 Linux 环境下使用,这在下面会详细解释。

Delphi 全面支持 Windows 95/98 及 Windows NT 的 OLE、ActiveX、Multiple Thread、MAPI 及 Unicode 等功能。它可调用 Win32 API,所以从原则上讲它可以完成 Visual C++ 能够完成的任务。

Delphi 具有强大的 Internet 开发功能,为网络应用程序开发提供了大量的组件。不管是开发可执行的网络应用程序,还是基于浏览器的 Web 服务器应用程序,Delphi 都提供了强大的支持。Delphi 对网络程序的开发和对数据库应用程序的开发都是值得称道的技术,它们可以互相融合,从而开发出功能无比强大的应用程序系统。而且只要掌握了开发此类应用程序的基本原理,它的使用就不是那么复杂。

1.2 Delphi 新特性

信息时代发展是迅速的,新技术层出不穷,要求程序开发人员在应用系统中加入这些新技术,或者用这些新技术开发下一代应用系统。例如 XML (Extensible Markup Language, 可扩展标记语言)/XSL (Extensible StyleSheet Language, 可扩展样式表单语言) 的盛行,让程序员对于如何处理和发挥 XML 的技术有强烈的需求;例如 SOAP (Simple Object Access Protocol) 逐渐成为远程和分布式计算的标准,因此程序员也需要使用 SOAP 来开发或是整合应用系统。更别说由 SOAP 和 WSDL (Web Service Definition Language) /WSML, UDDI (Universal Description, Discovery, and Integration) 等技术所衍生出的最新 Web 应用系统标准 Web Service 等。对于窗口平台上的程序员而言,即使是使用现在的开发工具,例如 Delphi 4.0/5.0 或是 Microsoft 的 Visual Basic 6.0 等也仍然可以利用这些新的信息技术开发应用程序,但是如此做不但有一定的难度,而且可能荒废时日。

Delphi 6.0 刚好可以帮助企业和程序员解决这些信息技术上的困扰,Delphi 6.0 不但在传统的开发能力方面有持续的进步,让程序员能够享有更具生产力的开发环境,对于最新的信息技术,例如 XML/XSL、SOAP、Web Service 等都有非常良好的支持,让程序员能够快速的使用这些新的技术。

此外,Delphi 6.0 在核心执行时期函数以及编译器方面也有了幅度不小的改善,不但让程序员可以开发功能更强的窗口应用程序,也可以更平顺的移植到 Linux 平台,再搭配跨 2 平台的新可视化组件 Framework CLX,即使是 Windows 窗口的程序员也可以快速的开发 Linux 平台上的应用系统。

下面来看看 Delphi 6.0 增加了哪些新的功能使得应用程序开发更加容易。由于篇幅关系没有列出 Delphi 6.0 新增的所有特性,只是选择了对编程人员比较重要的新特性。另外,这里要注意的是新增功能并不是在 Delphi 6.0 的任何版本中都支持的,下面若没说明这表明它支持任何版本。

1.2.1 IDE

【Object TreeView】窗口

在 Delphi 6.0 中新增了【Object TreeView】窗口,如图 1-1 所示:

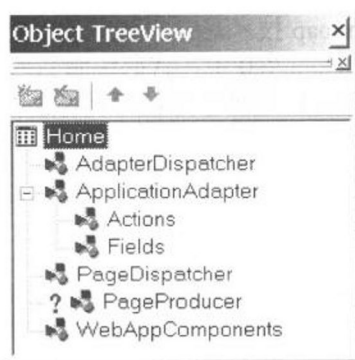


图 1-1 【Object TreeView】窗口

【Object TreeView】窗口缺省情况下位于 IDE 左上角。它除了可以显示当前活动窗口或者数据模块中所有的组件之外，也以层级的方式表达组件之间拥有者的关系。【Object TreeView】窗口会以特定的图标显示组件的状态，例如图 1-1 中，PageProducer 组件状态不明，这可能是没有设置正确属性造成的。

而当程序员在【Object TreeView】窗口中选中了一个组件之后，这个组件也会立刻出现在【Object Inspector】窗口之中，让程序员可以改变这个对象的属性值和事件处理函数。

也许对于有些 Delphi 6.0 使用者，可能上来就把【Object TreeView】窗口关闭，总觉得它占据了整个开发环境空间，又没有什么作用。但是用久了就会发现它的好处，特别是当窗口或者数据模块比较复杂，其中含有大量的组件时，【Object TreeView】窗口可以让你立刻找到这个组件，并且知道和它相关的组件，非常的方便。这比以往在【Object Inspector】窗口直接搜寻组件高效的多。Delphi 6.0 的【Object TreeView】窗口可以让程序员快速找到窗体中的任何对象。

代码编辑器（Code Editor）窗口

Delphi 6.0 的代码编辑器中的【Diagram】标签页（专业版和企业版）代替了 Delphi 5.0 中的【Data Diagram】。它提供了一种可视化的途径来表示可视化和非可视化组件之间的联系，如图 1-2 所示为【Diagram】标签页，它显示了数据表 and 数据库以及字段的联系：

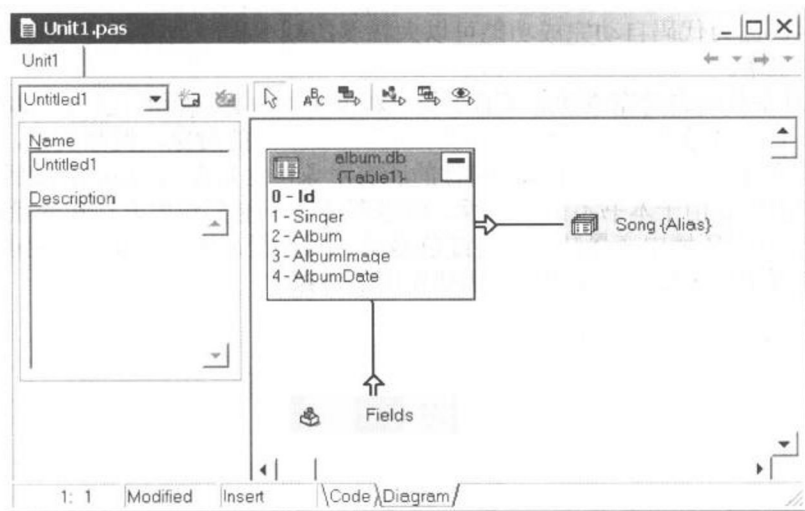


图 1-2 代码编辑器中的【Diagram】标签页

在 Delphi 6.0 中新增了 WebSnap 技术,它主要用于创建 Web 服务器应用程序。为了方便 WebSnap 应用程序的开发,Delphi 6.0 中在代码编辑器内嵌入了 HTML 编写标签页、脚本语言查看标签页、Web 页面预览标签页等等,使得不用运行程序就可以在代码编辑器内部查看网页设计结果,并且可以修改 HTML 代码。

如图 1-3 所示,熟悉 Delphi 开发环境的程序员应该会注意到代码编辑器的下方多了数个标签页,例如【HTML Result】、【Preview】标签页等,这些标签页就是为了方便开发 Web 服务器应用程序而新增的:

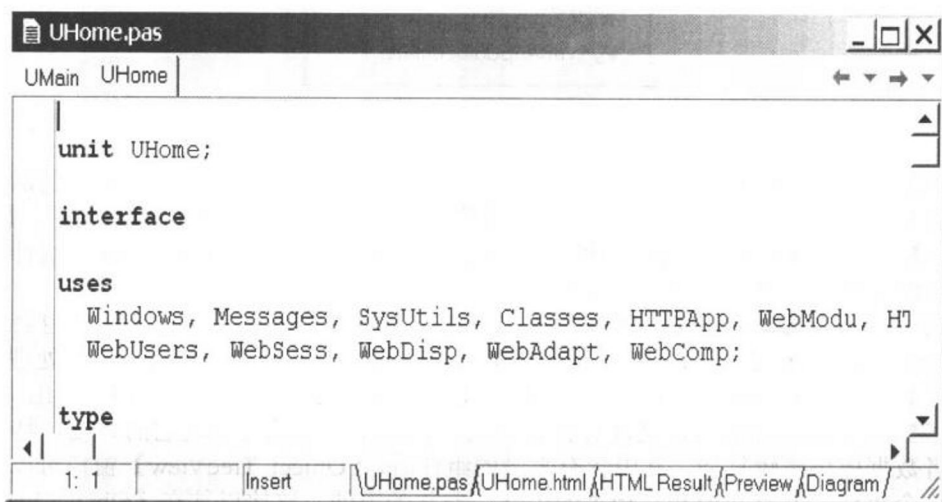


图 1-3 【Code Editor】窗口

除了上面提到的变化外,Delphi 6.0 的代码编辑器也可以让程序员直接查看 XML 和 XSL 的树状架构,更允许程序员自行开发定制化的视图窗口整合到 Delphi 的编辑器之中,可以说功能非常的强大。读者在以后的程序开发中会慢慢体会到这些新增的标签页的方便性。

代码自动完成功能 (Code Complete)

代码自动完成功能 (Code Complete) 在 Delphi 5.X 就已经存在了,这项功能非常受到程序员的欢迎。因为代码自动完成功能可以大幅度的减少程序员需要键入的程序代码,并且减少打错的机率。

Delphi 6.0 中代码自动完成功能又有了很大程度上的改善。新的代码自动完成窗口不但可以由程序员自行调整大小,以及以不同的颜色代表不同的对象,例如变量,方法和属性等。如图 1-4 所示,它显示了 TForm 组件的所有对象,这是在 TForm 对象后面键入“.”符号时自动弹出的。由于本书不是彩印版,所以图 1-4 不能分辨出方法和变量的不同颜色,其实在 Delphi 6.0 环境中变量缺省是用红色显示,而过程是用绿色显示。用颜色区分各种不同的对象有利于程序员快速定位自己需要使用的对象。

```

var      DataSource1 : TDataSource;
var      Table1 : TTable;
var      DBGrid1 : TDBGrid;
var      Table1Id : TFloatField;
var      Table1Singer : TStringField;
var      Table1Album : TStringField;
var      Table1AlbumImage : TGraphicField;
var      Table1AlbumDate : TDateField;
procedure FormCreate(Sender: TObject);
procedure ArrangeIcons;
procedure Cascade;
procedure Next;
procedure Previous;

```

图 1-4 代码自动完成窗口

最后，如果程序员在新的代码自动完成窗口中选择了使用的方法之后，代码自动完成功能会自动的在方法名称之后自动加入“(;)”字符，把光标自动定位在圆括号之中，并且自动显示这个方法需要的所有参数。

【New Items】对话框

【New Items】窗口增加了 3 个标签页，它们是：

- 【WebSnap】标签页：包含 WebSnap Application、WebSnap Data Module 和 WebSnap Page Module。
 - 【WebServices】标签页：包含 Soap Server Application、Soap Server Data Module 和 WebServices Importer
 - 【Corba】标签页：包含 CORBA Client Application 和 CORBA Server Application。
- 你也能够通过选择【Tools】⇒【Regenerate CORBA IDL Files】菜单项来访问这些向导。

【Internet】工具栏

Delphi 6.0 新增一个【Internet】工具栏来帮助你创建 WebSnap Web 服务器应用程序。可以选择【View】⇒【Toolbars】⇒【Internet】菜单项来显示这个工具栏。该工具栏包含三个新增的向导按钮：【New WebSnap Application】、【New WebSnap Page Module】和【New WebSnap Data Module】和一个激活外部编辑器的按钮。这些按钮提供了用 WebSnap 创建 Web 服务器应用程序最常用的功能。

组件面板的变化

Delphi 6.0 中组件面板做了不小的改动。除了增加了几个组件标签页外，对原来的组件位置也有了调整。

【Additional】标签页增加了 4 个组件：TActionManager、TActionMainMenuBar、TActionToolBar 和 TCustomizeDlg。

当我们创建 CLX 应用程序时，【Common Controls】标签页代替了【Win32】标签页（专业版和企业版）。创建 CLX 应用程序时，组件面板会有很大的改动，在编写 CLX 应用程序程序时会详细介绍。

新增了【WebSnap】标签页（企业版），包含用多模块建立 Web 服务器应用程序的组件。【WebSnap】标签页内的所有组件如图 1-5 所示：



图 1-5 【WebSnap】标签页中的组件

另外在 Delphi 6.0 中有新增了一组组件，它为开发 Web 应用程序提供了另一种途径。增

加了有关 Internet 协议的组件页（专业版和企业版），包括【Indy Clients】标签页（如图 1-6 所示），【Indy Servers】标签页（如图 1-7 所示）和【Indy Misc】标签页（如图 1-8 所示）。Internet Direct（Indy）组件是基于 blocking sockets 的开放 Internet 协议。



图 1-6 【Indy Clients】标签页中的组件



图 1-7 【Indy Servers】标签页中的组件



图 1-8 【Indy Misc】标签页中的组件

1.2.2 Internet

对 WebServices 的支持（企业版）

由于 Web Services 是一门新兴的技术，有待于发展。不过为了方便 Delphi 程序员也能体验这种新兴的技术，Delphi 6.0 中增加了相关支持。在 Delphi 6.0 组件面板的【WebServices】标签页中新增了一些组件，这些组件方便程序员开发 Web Services 应用程序。

【WebServices】标签页内的所有组件如图 1-9 所示。



图 1-9 【WebServices】标签页中的组件

对 Web 服务器应用程序的支持（专业版和企业版）

Delphi 6.0 包括了一组新的 Internet 新特性来建立 Web 服务器应用程序，这些变化包括对原来存在的 WebBroker 特性的改变，还包括新增的一组特性 WebSnap。Delphi 6.0 中最重要的新增功能之一应该就要算是能够让程序员开发 Internet/Intranet 应用系统的新架构和组件，这个新的功能称为 WebSnap。

Delphi 6.0 的 WebSnap 允许程序员使用组件和可视化的方式快速开发复杂的 Internet/Intranet 应用系统。WebSnap 的特点是允许程序员直接在这些组件之中加入服务器端脚本语言的能力，并且能够直接在 Delphi 6.0 的整合发展环境中编写脚本语言，预览 Web 应用程序的输出结果，以及 Web 应用程序产生的 HTML 程序代码。

此外当程序员在开发 Web 应用程序时，经常需要使用许多的服务，例如处理图形的组件、上传文件的服务、Session 的服务、搜寻文件的服务、登录和注销的服务等，WebSnap 都已经制作成了组件可以让程序员直接使用，程序员再也不需要花费时间撰写这些程序代码。图 1-5 便是 WebSnap 的所有组件。

Delphi 6.0 的 WebSnap 提供了超过 10 个的组件，此外 Delphi 6.0 提供了 Web App Debugger 公用程序，允许程序员开发和除错 Web 应用程序，让开发 Web 应用程序再也不是痛苦的过程。而 WebSnap 向导更允许程序员建立不同的 Web 应用程序，如图 1-10 所示便是 WebSnap 向导的画面，从图中可以看到 Delphi 6.0 能够建立各种型态的 Web 应用程序。

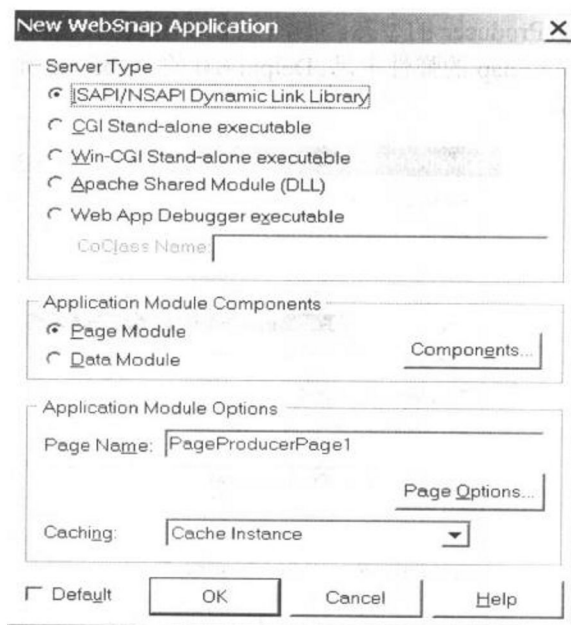


图 1-10 WebSnap 向导

在 WebSnap 向导除了可以选择 WebSnap 应用程序的服务器类型外，还可以单击【Components】按钮，在弹出的窗口中选择 WebSnap 向导在自动建立 Web 应用程序时使用的 WebSnap 组件。如图 1-11 所示显示了程序员可以选择使用的丰富的组件和服务。Delphi 6.0 的 WebSnap 提供了许多开发 Web 应用系统时需要使用的服务。



图 1-11 【Web App Components】窗口

由于 WebSnap 应用程序一般需要结合脚本语言的编写，因此程序员需要指定使用的脚本语言种类。可以在 WebSnap 向导中指定 WebSnap 在自动产生脚本语言时使用的脚本语言

种类, 以及产生的 PageProducer 的型态。如图 1-12 所示显示了程序员可以选择使用 JScript 或是 VBScript, 可见 WebSnap 的弹性十足。Delphi 6.0 的 WebSnap 允许程序员使用 JavaScript 或 VBScript 等脚本语言。

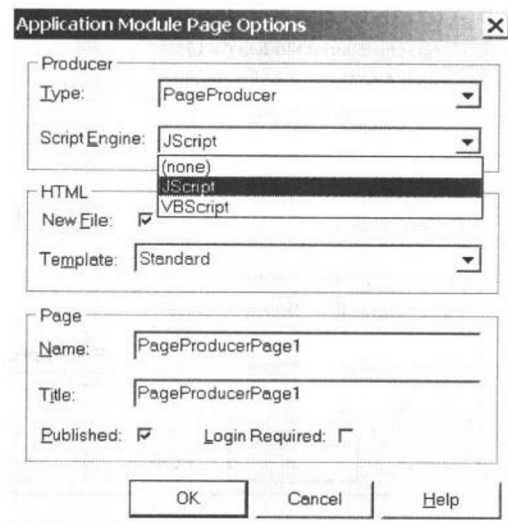


图 1-12 【Application Module Page Options】窗口

前面已经提到在 Delphi 6.0 中可用让让程序员直接在 Delphi 6.0 的代码编辑器窗口中就可以编写 Web 应用程序的 HTML 源代码, 预览 Web 应用程序的生成结果。如图 1-13 所示, 可以直接在 Delphi 6.0 的代码编辑器窗口中使用脚本语言修改 Web 应用程序, 而且脚本语言也具有和 Object Pascal 一样以不同颜色显示不同语言元素的效果。

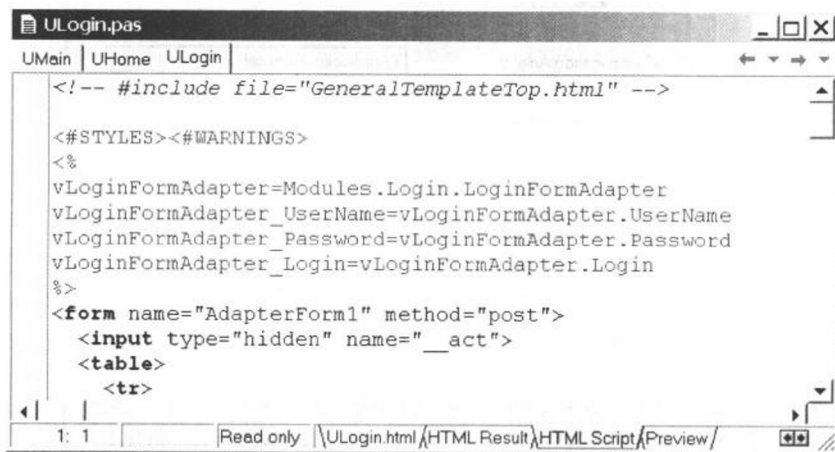


图 1-13 代码编辑器中的【HTML Result】标签页

在 Delphi 的代码编辑窗口中也可以直接查看 Web 应用程序的预览画面 (如图 1-14 所示), 可以让程序员知道 Web 应用程序将来在浏览器中执行的结果。

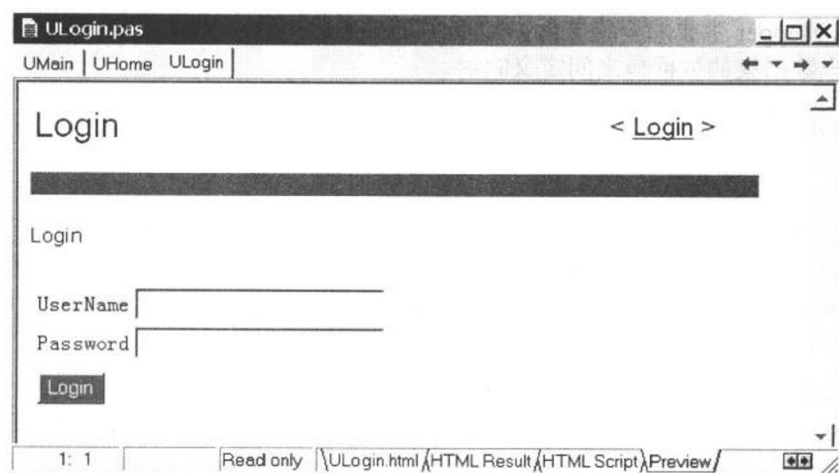


图 1-14 代码编辑器中的【Preview】标签页

总之, WebSnap 让 Delphi 6.0 在开发 Internet/Intranet 应用系统方面有长足的进步, 提供了非常好的整合开发环境。在未来 WebSnap 也将移植到 Linux 平台上, 让 Kylix 的程序员也能够使用相同的组件和技术开发 Internet/Intranet 应用程序。事实上程序员如果能够结合 WebSnap 和 Delphi 的数据库能力, 以及 COM+ 的功能, 将能够开发出威力十足的 Web 应用系统。

增强的 WebBroker 特性 (专业版和企业版)

允许你为 CLX 应用程序创建 Web 服务器应用程序。WebBroker 是向后兼容的, 不过这里有一些限制, 例如, 以前的项目可能需要增加或者改变单元名称。

1.2.3 XML 支持 (企业版)

XML Data Binding 向导

XML Data Binding 向导会自动生成 Object Pascal 代码, 为访问和更新 XML 数据文件提供了一个简单的途径。被生成的代码提供了一个自然的基于属性的程序模型, 它就像使用一般的 Delphi 组件那样使用 XML 数据。XML data bindings 向导代替了需要使用 XML Document Object Model (DOM) 接口的复杂代码, 它完全适合于那些需要操纵 XML 文档的应用程序。

XML document 编程

XML data bindings 是建立在新组件 TXMLDocument 和它相关的 IXMLDocument 接口/IXML 节点之上的, 这个组件提供了一个使用 XML 数据的简化的 DOM 接口版本。只要把 TXMLDocument 加入到窗口中, 设置 TXMLDocument 组件的 FileName 属性, 设置 Active 属性为 True, 就能访问全部 XML 数据了。

交叉平台/独立于开发商(vendor-independent)的 DOM 编程

在 XML 文档编程支持的最底层是新的 xmldom.pas 接口单元, 它为遵守 W3C DOM 标准 2 规则的编程提供了交互式平台和一套独立于开发商的接口。由于这些接口是开放式结构, 它们很容易同已经存在的基于 DOM 的 XML 方案整合。

在数据库应用程序中使用 XML

Delphi 6.0 新增了一套组件使你能够把 XML 文档集成到数据库结构中。这些组件利用