

计算机编程技术丛书

Delphi 5

实战与精通

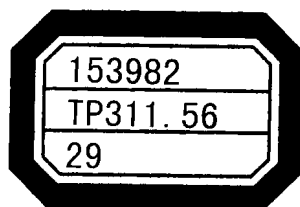
(实战篇)

唐 健 林苇茗 任雨生 编著

清华大学出版社
<http://www.tup.tsinghua.edu.cn>



计算机编程技术丛书



Delphi 5 实战与精通

(实战篇)

唐 健 林苇茗 任雨生 编著

（京）新登字 158 号

内 容 提 要

本书引导你进入著名开发工具 Delphi 5 的编程世界。书中采用循序渐进、深入浅出的讲解方法，运用翔实的实例，介绍了 Delphi 编程的各个侧面，并介绍了 Delphi 5 版本的许多新特点。本书注重开发实例、开发经验、开发技巧和 Windows 高级特性开发，适合于各个层次的 Delphi 用户。

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。

书 名：Delphi 5 实战与精通（实战篇）

作 者：唐 健 林苇茗 任雨生

出 版 者：清华大学出版社（北京清华大学校内，邮编：100084）

<http://www.tup.tsinghua.edu.cn>

责任编辑：徐培忠 林庆嘉

印 刷 者：北京市清华园胶印厂

发 行 者：新华书店总店北京发行所

开 本：787×1092 1/16 **印张：**20 **字数：**472 千字

版 次：2000 年 5 月第 1 版 2000 年 5 月第 1 次印刷

书 号：ISBN 7-302-01130-3/TP·373

印 数：0001~5000

定 价：28.00 元

《计算机编程技术丛书》出版说明

《计算机编程技术丛书》，包含了现今流行的开发工具系列软件，如 Authorware 5.0、AutoCAD 2000、Visual C++ 6.0、Visual Basic 6.0、Visual Foxpro 6.0、Delphi 5 等，是一套易学易用的编程指导书。读者既可循序学习，亦可随查随用，使您学有所依，用有所循，快速便捷地掌握软件的精髓和编程技术，得心应手地解决实际问题。

为了使本系列书有的放矢，我们力求使本系列书具有如下特点：

知识速查 我们将对编程的基本知识作一点说明，以备读者速查。没用过该软件但有一定 Windows 开发经验的用户也可以在较短时间内掌握如何编写基本的应用程序。

循序渐进 在编写本书过程中，尽量按照“由浅入深”的顺序来编写，以便于读者的学习。

深入浅出 作者站的角度较高，能深入探讨编程原理及技巧，而且又是本国作者，能以流利、规范、通俗易懂的语言和详尽的语言来说明。

示例详尽 书中的难点、疑点尽量利用实例来进行说明，以便于理解掌握。同时每个细节中都尽可能地给出详尽的示例，从而避免由于一些细微的错误而导致不能掌握某个知识点。

创新思维 以生动的、具有启发性的例子启发读者，且具有醒目的步骤和生动的画面，以及切中要害的经验技巧提示，让读者深刻领悟该软件的精华。

内容全面 本书所包含的内容丰富而全面，以适应开发不同应用程序的需要。

本系列书面向工作在各行各业的计算机编程人员和大学师生，为他们提供最新的编程技术与编程经验，使他们能尽快跟上技术发展，借鉴他人的经验，结合自身工作，编制出更先进实用的应用程序。

在本系列书的编写过程中，我们尽量注意减少冗长无味的说明，代之以具体实用的例子演示。通过例子，引导读者把握精髓所在。本系列书注重开发实例、开发经验、开发技巧和高级特性开发，适应用户的急需。

本系列书是集体智慧的结晶，而且每本书都是由最擅长该开发工具的开发人员执笔。

前 言

“真正的程序员用 C，聪明的程序员用 Delphi”。在程序员的队伍中，流行着这么一句话，由此可见 Delphi 的巨大魅力。

Delphi 是 Borland 公司（现已更名为 Inprise）为基于图形界面的多任务、多窗口操作系统 Microsoft Windows 推出的基于 Object Pascal 语言的第一个可视化开发平台。它集众多优越特性于一身，因而被人们称为“第四代编程语言”。Delphi 提供了新颖的可视化设计工具，采用面向对象的方法将 Windows 编程的复杂性封装起来，实现了将可视化设计与 Object Pascal 语言的有机集成，并配有 Database Engine（数据库引擎），可通过 SQL Links（结构化查询语言链接）、ODBC（开放数据库互连）访问多种数据库，并且提供了强大的开发基于客户/服务器模式的数据库应用的能力。Delphi 本身符合 Microsoft Windows95 Logo 的标准，并全面支持 Windows 95&NT 的 OLE Automation（对象链接与嵌入自动化）、OCXs（OCX 控件）、多线程、Unicode（统一代码）和 MAPI（消息应用程序编程接口）等功能。可以说，Delphi 既具有 Visual C++ 的强大功能，又具有 Visual Basic 易于使用的特性，这也就难怪其倍受编程者的青睐了。

Delphi 4 在以前版本的基础上已增加了不少新功能、新特性，首先，它对 Object Pascal 语言进行了进一步的扩展，使其支持动态数组、方法的重载及默认参数。其次在界面上提供了一些浮动的工具窗口。更重要的是它改进了其调试器，并增强了对 ActiveX 控件、VCL 及数据库的支持。

Delphi 5 又在 Delphi 4 的基础上大大地增强了功能。例如：ADO 数据集、数据模块设计器、InterBase 数据库的包装、MIDAS 技术的增强、新的调试特性、VCL 库的增强、可自定义的环境设置、对象检视器功能的增强、ActiveX 支持的增强等等。

鉴于 Delphi 的优良特性和广大读者的需要，我们在积累大量开发经验的基础上编写了这本书。初学者学习了本书，可迅速加入 Delphi 高级用户的行列；对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办法开发出高水平的 Delphi 应用软件。

书中实例所涉及的人名、电话号码、通信地址和邮件地址等均为虚构，如有雷同，纯属巧合。

本书由方可燕主编，唐健、林苇茗、任雨生编著，韦思亮审稿。参与编写工作的还有 Wizard Group 的方可军、王东辉、周兆确、毛诗华、方海英、唐家才、段小华、许刚、侯志东、张双民等。

本书提供的实例程序除个别已经在书中注明出处之外，其余均由本书作者自己编写，并且全部在 Windows 98 操作系统中调试通过。

限于作者水平，难免在内容选材和叙述上有不当之处。欢迎广大读者对本书提出批评和建议。

编 者

目 录

第 1 章 功能强大的 Delphi 5	1
1.1 图形化的“Hello,Delphi”	1
1.2 Delphi 的主要特点	4
1.2.1 友好方便的可视化设计工具	4
1.2.2 面向对象的程序设计方法	4
1.2.3 完整的 Windows 应用程序框架	5
1.2.4 Delphi 的版本演变与功能的改进	5
1.3 Delphi 5 的新特征	9
1.4 本书的阅读方法	10
第 2 章 Delphi 5 的集成开发环境	12
2.1 Delphi 5 的 IDE	12
2.1.1 主窗口	13
2.1.2 菜单栏	14
2.1.3 快捷工具栏	19
2.1.4 组件板 (Component palette)	20
2.1.5 对象检视器 (Object Inspector)	22
2.1.6 代码编辑器 (Code Explorer)	23
2.1.7 对象库 (Object Repository)	25
2.1.8 项目管理器 (Project Manager)	26
2.1.9 项目浏览器 (Project Browser)	28
2.2 Delphi 5 的文件类型	28
2.3 Delphi 5 的帮助系统	29
第 3 章 Object Pascal 的面向对象特性	31
3.1 类与对象	31
3.1.1 类的封装性	31
3.1.2 类的继承性	32
3.1.3 类的多态性	32
3.1.4 对象的创建、销毁与赋值	36
3.1.5 类的作用域 (Scope)	37
3.1.6 访问类的域和方法	37

3.1.7 强制类型转换	38
3.2 类的引用	39
3.3 异常处理	40
第 4 章 应用程序的开发	41
4.1 窗体与菜单	41
4.1.1 可视化编程的基本思想	41
4.1.2 窗体的各种属性和事件	42
4.1.3 在窗体中响应键盘和鼠标的输入	52
4.1.4 窗体的绘制	58
4.1.5 菜单的设计	67
4.1.6 窗体与菜单的重用	71
4.1.7 菜单命令的响应	75
4.1.8 图形菜单与系统菜单的定制	83
4.2 使用 Delphi 的组件	90
4.2.1 使用可视化控件	90
4.2.2 使用非可视化组件	108
4.2.3 控件的拖放编程	114
4.2.4 使用 Win32 控件	120
4.3 不同风格的应用程序	138
4.3.1 SDI 应用程序的一个有趣的实例	138
4.3.2 控制台应用程序	139
第 5 章 动态链接库 (DLL) 编程	142
5.1 什么是 DLL	142
5.2 在 Delphi 5 中使用 DLL	144
5.2.1 使用 DLL 可以获得的益处	144
5.2.2 可以在 Delphi 中使用的 DLL	144
5.2.3 使用 DLL 的两种方法	148
5.3 用 Delphi 5 创建 DLL	151
5.3.1 动态链接库编写中应注意的问题	151
5.3.2 DLL 的项目文件	152
5.3.3 编写 DLL 的一般步骤	154
5.3.4 编写 DLL 的一个应用实例	157
5.4 创建含有 Delphi 资源的 DLL	161
5.4.1 利用 DLL 实现窗体重用的步骤	161
5.4.2 窗体重用实例	162
5.4.3 在 DLL 中存放其他的资源	165

5.5 创建控制面板小应用 (Control Panel Applet)	168
第 6 章 Delphi 的文件	172
6.1 文件管理的基本概念与标准过程	172
6.1.1 Object Pascal 的文件类型	172
6.1.2 对文件的基本操作	173
6.1.3 Delphi 的标准文件过程	187
6.2 磁盘文件管理器的设计	191
6.2.1 程序界面设计	191
6.2.2 磁盘文件管理功能的实现	194
6.2.3 程序辅助功能的实现	207
6.3 小结	209
第 7 章 异常处理	210
7.1 Delphi 的异常处理体系	210
7.1.1 运行时间库异常类(RTL Exception)	212
7.1.2 通用对象异常类	217
7.1.3 组件异常类	218
7.2 处理异常	220
7.2.1 用 try...except 结构响应异常	220
7.2.2 用 try...finally 结构保护资源	222
7.3 自定义异常类的使用	226
7.3.1 自定义异常类	226
7.3.2 引发自定义异常	227
7.3.3 一个简单的实例	228
7.4 屏蔽 Delphi 的异常处理器	230
第 8 章 数据库应用程序开发概述	234
8.1 数据库系统的基本概念	234
8.1.1 数据库的含义	234
8.1.2 数据库系统	234
8.1.3 本地数据库和远程数据库	236
8.1.4 数据完整性、存储过程和触发器	236
8.1.5 事务	237
8.1.6 数据库应用程序的体系结构	237
8.2 Delphi 5 的数据库特性及功能简介	238
8.2.1 数据库连接工具	238
8.2.2 数据库相关组件	239

8.2.3 数据库工具	239
8.3 使用 Delphi 5 的数据库工具	240
8.3.1 数据库引擎 (Borland Database Engine)	240
8.3.2 数据库桌面 (Database Desktop)	243
8.3.3 数据库浏览器 (Database Explore)	245
8.3.4 数据字典 (Data Dictionary)	247
8.3.5 SQL 监视器 (SQL Monitor)	250
8.4 Delphi 5 的数据库组件	251
8.4.1 数据访问组件	253
8.4.2 数据控制控件	255
8.4.3 数据库窗体向导	257
8.4.4 ADO 数据库组件	257
8.4.5 InterBase 数据库组件	259
8.5 Delphi 数据库应用程序的基本开发步骤	259
第 9 章 数据库应用程序的开发	262
9.1 简单的单表数据库应用程序	262
9.2 Master/Detail 数据库应用	264
9.3 数据模块的使用	266
9.4 再论 TTable 组件及其他的数据库组件	267
9.4.1 TTable 组件的重要属性	269
9.4.2 TTable 组件的重要方法	272
9.4.3 数据控制控件的使用	280
9.5 数据库中记录的插入、修改、删除与范围界定	285
9.6 管理与数据库的连接	289
9.6.1 TDatabase 与 TSession 的关系	289
9.6.2 TDatabase 的创建	290
9.6.3 TDatabase 的使用	291
9.7 数据库的查询	294
9.7.1 SQL 概述	295
9.7.2 TQuery 组件的使用	296
9.7.3 数据库查询的一个实例	303

第 1 章 功能强大的 Delphi 5

Delphi 的每一个新版本都能给用户提供更新的功能，Delphi 5 也不例外，它比以前的版本提供了更多、更完美的特性，是 Borland 公司更名为 Inprise 后推出的功能最为强大的可视化编程工具。下面让我们从一个简单的但几乎为所有的编程类书籍所采用的 Hello 实例开始我们的“Delphi 之旅”。

1.1 图形化的“Hello, Delphi”

创建“Hello, Delphi”的程序十分简单，简单得令你怀疑以前所听说的 Windows 编程如何如何难之类的话是否正确。遗憾的是，这种话大体是对的。此处的简单只是因为 Delphi 替我们做了许多幕后的工作，由此也可见 Delphi 的强大。

好了，言归正传。第一步，启动 Delphi 5，Delphi 将自动为我们创建一个新窗体。如图 1-1 所示。

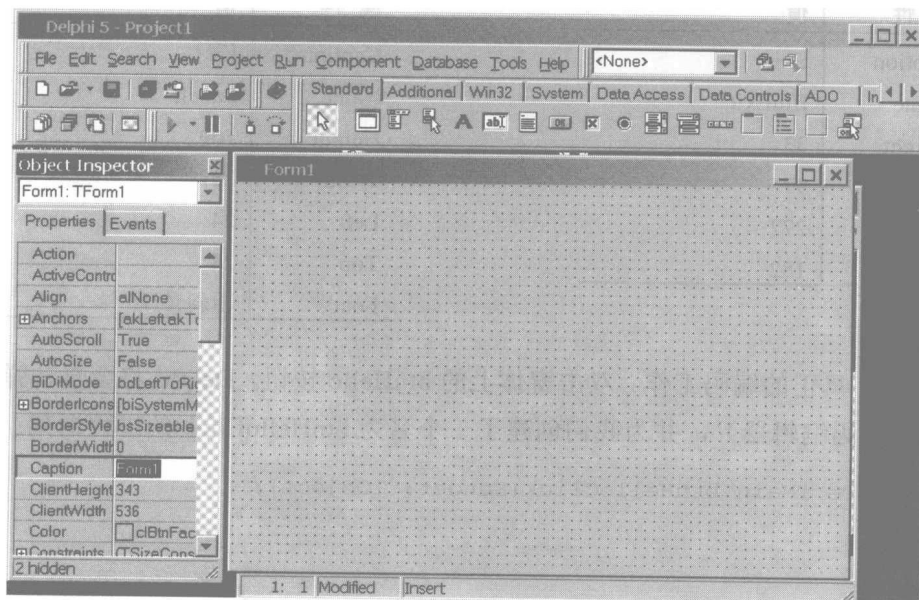


图 1-1 Delphi 5 的编程界面

利用窗体（Form）左边的对象检视器（Object Inspector，将在后面介绍）改变该新窗体的部分属性，如表 1-1 所示。

第二步, 给窗体添加两个按钮控件 (TButton) 和一个画板控件 (TPanel), 它们都位于窗体上方组件板 (component palette) 的标准控件页中, 然后如表 1-2 所示设置按钮 1 的属性。

表 1-1 需改变的新窗体的属性

属 性	值
Caption	Delphi Demo
Name	Main
Height	251
Width	402

表 1-2 按钮 1 的属性

属 性	值
Caption	Say Hello
Name	BtnHello
Height	33
Width	97
Left	272
Top	96

如表 1-3 所示设置按钮 2 的属性。

如表 1-4 所示设置画板的属性。

其中, 设置画板的字体属性 (Font) 时将会弹出一个字体对话框, 如图 1-2 所示, 在该对话框的字形下拉列表中选择粗斜体, 大小为 18 磅。

表 1-3 按钮 2 的属性

属 性	值
Caption	Exit
Name	BtnExit
Height	33
Width	97
Left	272
Top	152

表 1-4 画板的属性

属 性	值
Caption	[空]
Name	Panel1
Height	169
Width	217
Left	24
Top	24
Font	粗斜体 18 号

第三步, 真正的编码工作。双击窗体上的 btnHello 按钮, Delphi 将会自动弹出 Code Explorer (代码编辑器), 并为我们创建了一个名为 btnHelloClick 的过程框架:

```
procedure TMain.btnHelloClick(Sender: TObject);
begin

end;
```

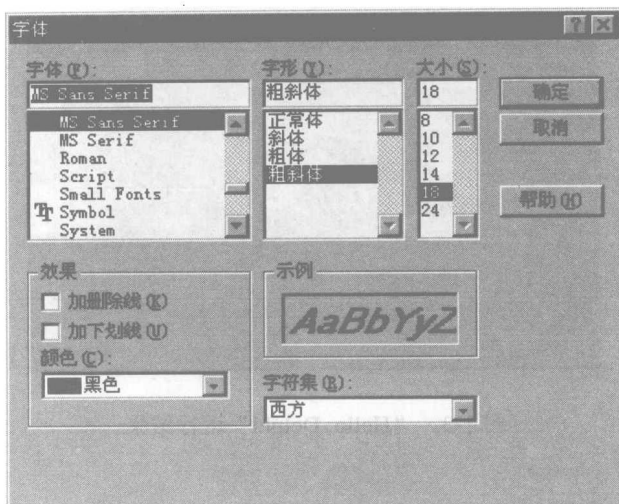


图 1-2 通过字体对话框选择 Font 属性

在框架中添加自己的代码使得整个过程如下：

```
procedure TMain.btnHelloClick(Sender: TObject);
begin
    Panel1.Caption := 'Hello, Delphi';
end;
```

语句 `Panel1.Caption := 'Hello, Delphi'` 用于在运行中当 `btnHello` 按钮被单击时修改面板的 `Caption` 属性，以显示文字“Hello, Delphi”。

用同样的方法为 `btnExit` 添加上相应的代码：

```
procedure TMain.btnExitClick(Sender: TObject);
begin
    Application.Terminate;
end;
```



专家指导

任一个 Delphi 创建的 VCL (可视组件库) 程序均包含有一个叫做 `Application` 的变量，它代表应用程序实例。`Terminate` 是 `Application` 提供的一个方法，用于结束应用程序。事实上，它调用 Windows API 中的 `PostQuitMessage` 函数以终止应用程序的运行。

好了，程序已经完成了。选择菜单栏中的 **Run/Run** 或直接按下快捷键 **F9** 以运行应用程序，单击 **Say Hello** 按钮后的程序界面如图 1-3 所示。

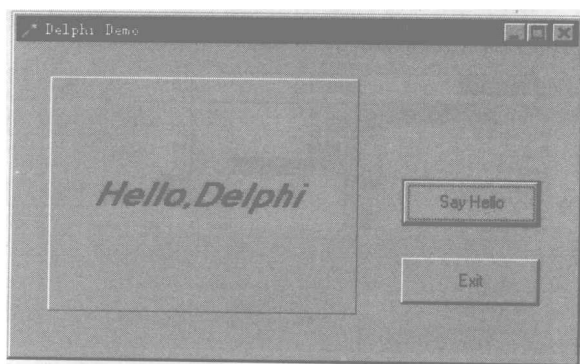


图 1-3 “Hello, Delphi” 运行结果

1.2 Delphi 的主要特点

Delphi 的主要特点, 从大处来说, 主要包括三个方面: 即友好方便的可视化设计工具、面向对象的程序设计方法和 Delphi 给每一个待开发的应用程序所提供的完整的 Windows 应用程序框架。正是具备了这三个特点, 因此可以说, Delphi 既具有 Visual C++ 的强大功能, 又具有 Visual Basic 易于使用的特性。另外, Delphi 的编译器也十分优秀, 产生的应用程序的性能很高。可以这么说, 选择 Delphi 意味着软件开发已经成功了一半。

1.2.1 友好方便的可视化设计工具

Delphi 是第一个提供可视化设计工具的编程环境。从上面简单的实例就可以看出可视化编程的好处。在程序设计中, 只需简单地从组件板中拖动组件到已经存在的窗体中即可在运行中看见它们。而传统的 Windows 编程则往往需要在界面问题上编写大量的代码、浪费大量的时间。因此可以说, 可视化的设计工具将程序员们从枯燥的 Windows 界面编程之中解脱了出来, 使他们可以把精力放在与应用相关的编程上。

1.2.2 面向对象的程序设计方法

面向对象的程序设计 (Object-Oriented Programming, 简记为 OOP) 是 Delphi 诞生的基础。OOP 立意于创建软件重用代码, 具备了更好地模拟现实世界环境的能力, 这使它被公认为是自上而下编程的优胜者。Delphi 通过 Object Pascal, 把 Windows API 函数封装进为 Windows 编程而抽象出的类中。

面向对象的程序设计方法使得复杂的工作条理清晰、编程容易, 且代码可重用性高。

这也正是 Delphi 编程相对于传统的 Windows 编程的一大优势。

1.2.3 完整的 Windows 应用程序框架

熟悉传统 Windows 编程的程序员一定知道,所有的 Windows 程序都有类似的结构,比如,都有一个 WinMain 函数,若干个响应窗口事件的窗口函数。WinMain 函数中又都需要注册窗口类,显示主窗口及提供消息循环等等。老是编写同样的代码实在令人厌烦。

Delphi 则给每一个待开发的应用程序提供了完整的 Windows 应用程序框架。只要一进入 Delphi,创建一个新项目,无需进行任何编程,该项目就可以编译运行了,而且是一个不折不扣的 Windows 程序。Delphi 为我们自动构筑了应用框架,只等着我们往里添加自己的内容了。

1.2.4 Delphi 的版本演变与功能的改进

Delphi 的强大功能并不是一蹴而就的,它的每一个新版本都给用户提供了新的功能,日积月累才变成今天的模样。

Delphi 1 版本是基于 16 位 Windows 的,它给用户提供了一种全新的可视化的编程环境,用于快速开发 Windows 3.1 的应用程序。

Delphi 2 版本改写了内核,为使用者提供了快速的 32 位开发环境,使用户可以很容易地编写符合 Windows 95 Logo 的应用程序。另外它还通过增加对象库 (Object Repository) 更好地体现了软件重用的设计思想。

更新的 Delphi 3 版本则提供了多层 Client/Server (客户/服务器) 的 32 位快速开发工具。另外,Delphi 3 在以下三方面采用了新技术:

1. 数据库的处理技术

Delphi 3 针对最新一代的数据库系统、即在传统的 Client/Server 结构中增加一个应用服务器的叫做 n-Tier 或 Multi-Tier 的新结构,提出了三种 Broker 和新一代的数据库引擎,来适应 n-Tier 的需求。

第一种叫做 RemoteDataBroker。RemoteDataBroker 结构用一个集中化的 BDE 取代每一个客户端的 BDE,以集中管理的方式降低每一个客户在 BDE 上调整的开销和复杂度。

第二种叫做 ConstraintBroker。它保证所有客户数据的一致性及数据的完整性。

第三种是 BusinessObjectBroker。BusinessObjectBroker 自动地将应用程序做适当的划分,并复制重要的业务规则到每一个区间,以给一些关键性的商业应用程序提供一个快速而且可信赖的使用环境。

2. Web 上的 Client/Server 技术

Delphi 3 提供了 WebBroker 和 ActiveForms,使开发人员可以结合原有的 Client/Server

技术开发出 Web-Enabled (支持 Web) 的 Client/Server 应用程序。

3. 客户端的应用程序开发的改进

(1) 组件制造中心

企业组件的制作基础是对 ActiveX 技术的运用。配合对 COM (组件对象模型) 的支持, 组件制造中心使用简便易用的 ActiveX 工具, 可以轻松地建立企业组件, 从而提供企业内代码的最大重用能力。用 Delphi 3 制作出来的组件可以运用在各种不同的开发工具中, 例如 C++ Builder、IntraBuilder、VB 等。

(2) 数据分析

Delphi 3 提供了三个重要的组件, DecisionCube、TeeCharts 和 QuickReport, 它们运用图形及报表的形式将现有的数据转换成有用的信息, 来辅助决策者做出最正确的抉择。

DecisionCube 让使用者能依据个人需求, 动态查询数据, 显示多维数据和动态图表, 产生不同性质的有用分析结果, 充分显示数据背后的相互关系, 给决策者提供最快最有用的信息, 帮助他作出最正确的决定。TeeCharts 是一个三维的可视化图表, 能使数据更容易被理解。QuickReport 是一个功能强大的报表生成组件群, 除了提供一般的报表外, 还可以产生 HTML 的报表格式, 让所产生的报表得以在 Web 上发布。

(3) 提高程序员生产力的 Code Insight 技术

在程序开发过程中, 语言的语法、组件的方法和事件的名称, 或是已经声明过的变量、过程的名称往往很难准确地记住。Delphi 3 提供了 Code Insight 技术来帮助程序员, 使编写代码成为比较简单的工作。它解决了这一难题, 减少了程序开发所需的时间。Code Insight 技术包括四项功能, 分别是“类自动完成”、“类成员提示”、“代码模板”和“参数提示”。

此外, Delphi 3 还提供了新的编译技术, 即 VCLPackage (虚拟组件打包) 技术, 这种技术可以让程序开发者方便地将自己的应用程序拆解成小的程序片段。每一片段都被独立地编译过, 并且可重复使用, 其原有程序的执行效率不会因拆解而降低。

Delphi 3 已经是一个相当成熟的产品, Delphi 4 更是 Borland 更名为 Inprise 后推出的一个具有战略意义的产品。与前面的版本相比, Delphi 4 提供了一些更为强大的新功能。

1. Code Insight 技术

前面讲过, Delphi 3 已经有了 Code Insight 功能, 但 Delphi 4 又作了很多改进, 使这一组代码自动化功能更为强大, 能更好地帮助程序员快速生成代码, 从而减少语法错误并提高编程效率。

一是“类自动完成”。由于 Delphi 4 是一个完全面向对象的编程工具, 编程过程中经常要声明和实现类 (Class), 而类不同于一般的数据类型, 它具有特殊的语法, 很多初学者往往对类的使用缺少经验, 即使是 Delphi 的高手, 也会对过分严谨和古板的 Pascal 语法感到枯燥乏味。“类自动完成”向导可以帮助程序员快速生成有关类的代码。

二是“类成员提示”。当程序员键入一个类名或对象名, 再键入一个小圆点后, Delphi 4

估计程序员要访问类的成员，它就会自动弹出一个列表框，列出类的所有特性、方法和事件，这样程序员就不用记忆类有哪些成员，也不用查阅帮助或手册，更不用担心写错成员的名称。

三是“符号预知”。当处于设计期编写代码时，Delphi 4 在后台运行编译器，这样程序员无须显式地编译代码就能看到每个符号实际是怎样存储的，从而能够及时地帮助程序员消除错误。

四是“代码模板”。当程序员在设计期间编写代码时，经常要重复输入诸如 if...then...else 或 for...do 等语句，Delphi 4 把一些常用的代码结构预先做成模板，程序员只要选择一个模板，Delphi 4 就会自动把该模板的代码插入到代码编辑器中。代码模板也可以自定义。

五是“参数提示”。程序员只要键入实例名或方法名，再键入左圆括号，Delphi 4 就会自动显示该实例或方法的所有参数的名称及其数据类型，这样就能保证参数的个数、顺序和数据类型总是正确的。

2. Business Insight 技术

Business Insight 技术是前面所提到的数据分析功能的改进，主要包括前面所提到的三个组件，即 DecisionCube、TeeCharts 和 QuickReport。

3. Active Insight 技术

Delphi 4 通过 COM 接口和 DAX 技术完全支持 ActiveX，程序员可以很方便地创建、安装、注册、发布和使用 ActiveX 控件、ActiveForm 和 OLE（对象链接与嵌入）自动化对象。

Delphi 4 内建了对 COM 的支持，由于 COM 对象具有语言 and 平台无关性，因此用 Delphi 4 创建的对象可以与用 Visual Basic、Java、C++ 及其他语言实现的对象交互。在多层 Client/Server 环境中，COM 对象可以封装商业规则，为分布式的客户提供服务。此外，Delphi 4 用“包”技术减少冗余的 VCL 框架代码，从而使应用程序更精巧，更适合于在 Internet/Intranet 上传输和分发。Delphi 4 取消了原来的包编辑器，改用项目管理器对包项目统一进行管理。

4. 支持 Internet（因特网）应用

Delphi 4 包含了一组 Internet 控件，可以很方便地实现 WinSock 编程，以访问 HTTP、UDP、FTP、SMTP、POP3 和 NNTP 等服务。程序员可以利用 Delphi 4 方便地创建 Web 服务器应用程序，其中 WebBridge 简化了对 NSAPI 和 ISAPI 的访问，WebModules 封装了 Web 服务器的应用逻辑，WebDispatcher 实现了 HTTP 请求消息和 HTTP 响应消息的动态调度。运用 ActiveForm 技术和 RemoteDataBroker 技术则可以把分布式的 Client/Server 结构扩展到 Internet/Intranet（因特网）上。

5. MIDAS

MIDAS 是 Multi-Tiered Distributed Application Services 的简称，它实际上是一组分布式技术的统称，包括以下方面：

RemoteDataBroker 是多层 Client/Server 结构的关键技术，客户端只需要实现用户界

面,而应用逻辑(又称商业规则)则由应用服务器实现,这样就达到使客户端“减肥”的目的,并且“瘦”客户端的维护成本很低。**RemoteDataBroker** 技术的另一个作用在于它能显著地减少网络上的数据流量,这在带宽问题日益严重的今天具有重要的意义。

BusinessBroker 是一种分布式的对象名称注册服务,它能根据当前的负载情况动态地调整远程对象的投入和切除,从而保证关键性的商业应用永不中断。

OLEnterprise 是一个把传统的 **Client/Server** 应用程序向三层结构“迁移”的工具,它能够把应用逻辑和数据访问模块集中在应用服务器上完成。

MIDAS 除了能传递数据以外,还能传递表的结构,对于“瘦”客户端来说,程序员不用写任何代码,就能保证表与表之间的 **Master/Detail** 关系和嵌套关系得到维护和更新。为了保证系统的安全性,鉴别用户身份(登录)是一种最简单而又有效的手段。**Delphi 4** 内建了用户登录功能,程序员不必为此编写额外的代码。

Delphi 4 能够使“瘦”客户端显著地加快检索数据的速度,这是因为 **Delphi 4** 采用了一种“**DeferredBLOB**”的技术,它优先下载非 **BLOB** 型的数据,等非 **BLOB** 型的数据下载完后再根据需要下载 **BLOB** 型数据。

Delphi 4 在“瘦”客户端用 **TMIDASConnection** 与应用服务器连接,**TMIDASConnection** 支持三种连接方式:**TCP/IP**、**CORBA** 和 **DCOM**,其性能超过了 **Delphi 3** 的 **TRemoteServer**。

此外,**Delphi 4** 还支持 **MicrosoftTransactionServer(MTS)**。**MTS** 是 **Microsoft** 为实现分布式应用而设计的一个中间件,其基本组成部分是 **ActiveX**。**MTS** 中的“两阶段提交”技术使得应用程序能够跨服务器处理事务。**MTS** 中的“**X/OpenXA**”技术允许服务器与 **MTS** 的资源管理器通信,客户程序能够在异构环境下看到提交的数据包成功还是失败。

6. 开放性与灵活性

Delphi 4 的开放体系结构支持多个数据库引擎,任何一个数据库引擎都能与 **Delphi 4** 的数据显示控件如 **TDBGrid**、**TDBEdit** 等协调工作。**Delphi 4** 可以访问 **Oracle** 的 **BLOB(BinaryLargeObject)** 型字段,实际的数据存储在外部文件中。**Delphi 4** 支持 **Oracle 8** 的抽象数据类型(**ADT**),允许用户自己定义数据类型,自定义的数据类型可以基于真实的类型,也可以基于已定义的抽象类型。这个功能的意义是非常重大的,程序员可以把任何商业对象(**BusinessObjects**)存储到数据库中,或者把 **Oracle** 数据库中的商业对象放到 **Form** 上显示出来。

7. Advanced Debugging

Delphi 4 增加了“模块窗口”用以显示当前所有映射到应用程序地址空间的模块,包括应用程序自身、应用程序显式或隐式调用的 **DLL** 以及操作系统调用的 **DLL**,模块窗口可帮助程序员优化程序结构。此外,**Delphi 4** 还能自动记录在调试过程遇到的事件,如断点、**Windows** 的消息以及其他调试信息。**Delphi 4** 能够监视指针错误,如果某个指针试图非法访问内存的某个地址,程序就会暂时中断运行,由调试器接管控制权,就好像遇到断点一样。