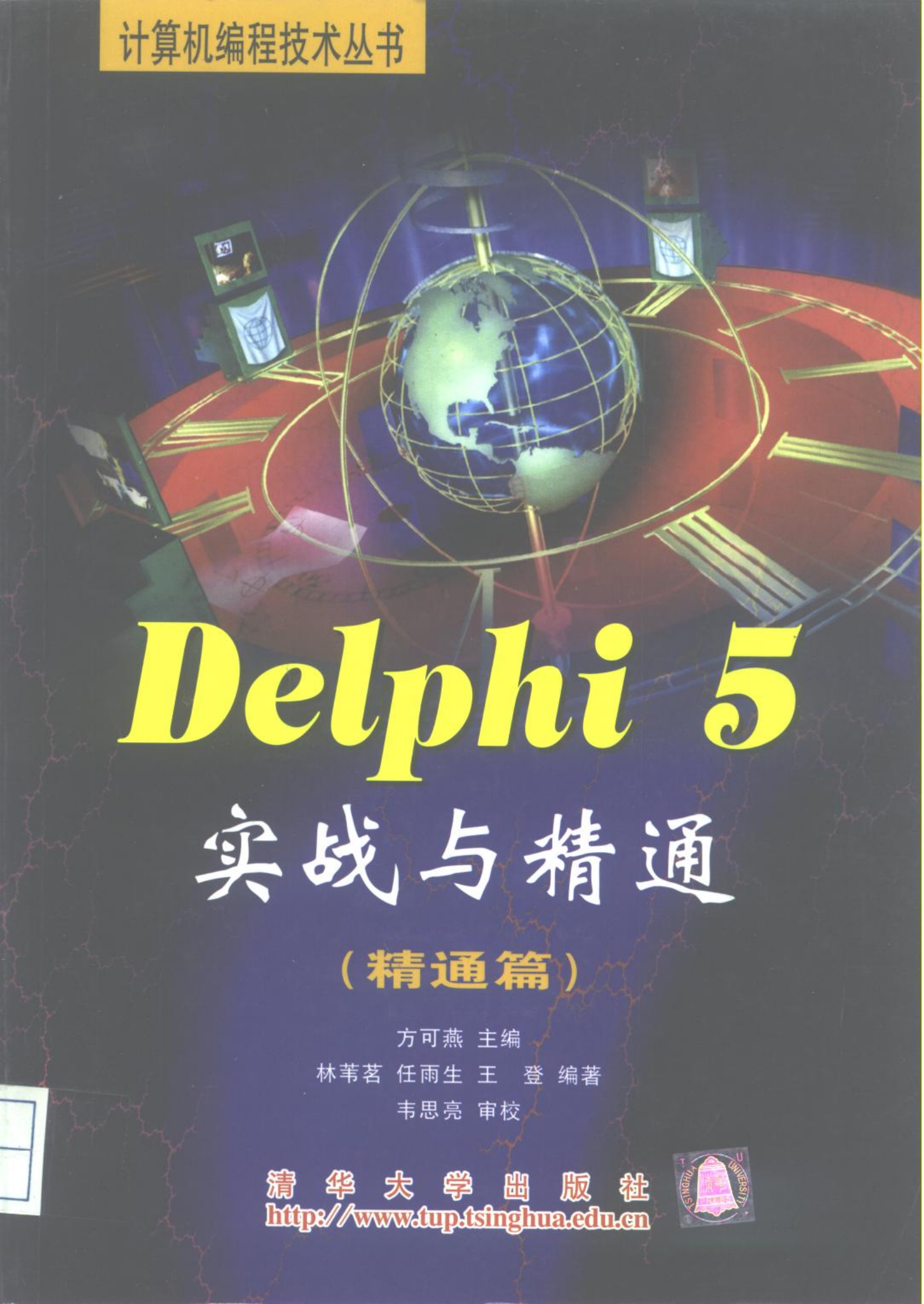


计算机编程技术丛书



Delphi 5

实战与精通

(精通篇)

方可燕 主编

林苇茗 任雨生 王 登 编著

韦思亮 审校

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



TP212
L.W.M/1
计算机编程技术丛书

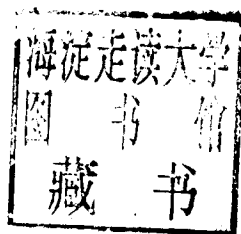
Delphi 5 实战与精通

(精通篇)

方可燕 主编

林苇茗 任雨生 王 登 编著

韦思亮 审校



海定走读 0057578

清华大学出版社

057578

(京)新登字 158 号

内 容 简 介

本书是一本极具参考价值的 Delphi 编程参考书。在本书中，我们力图以具体实用的样例来引导读者把握 Delphi 的精髓所在。注重编程知识、技巧和经验的结合是本书最大的特色。全书分为 11 章，每章之间相对独立，读者可以根据自己的需要有选择地进行学习。本书主要介绍程序员需要掌握但又相对比较难的一些关键性技术细节，有多层体系结构的构建、DDE 技术、控件的构造、OLE 和 COM 技术，还包括在程序设计中经常要涉及到的打印和多媒体的实现以及安装程序的创建等。

本书适用于已经具备了 Delphi 编程基本知识，而又渴望扩充自己的知识、提高编程技巧的读者。有一定使用经验的读者可通过本书掌握 Delphi 深层次的开发方法，学会用更具技巧性的办法开发出高水平的 Delphi 应用。此书可与《Delphi 5 实战与精通（实战篇）》配套使用。

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。

图书在版编目 (CIP) 数据

Delphi 5 实战与精通，精通篇/林苇茗，任雨生，王登编著. —北京：清华大学出版社，2000
(计算机编程技术丛书)

ISBN 7-302-03872-4

I. D... II. ①林... ②任... ③王... III. Delphi 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2000) 第 27232 号

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>

印 刷 者：北京市丰华印刷厂

发 行 者：新华书店总店北京发行所

开 本：787×1092 1/16 印张：22.75 字数：564 千字

版 次：2000 年 6 月第 1 版 2000 年 6 月第 1 次印刷

书 号：ISBN 7-302-03872-4/TP · 2263

印 数：0001~5000

定 价：34.00 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 书中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，配有 Database Engine，可通过 SQL Links、ODBC 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、OCX、多线程、Unicode 和 MAPI 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，因而倍受编程者的青睐。

Delphi 5 在 Delphi 4.0 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、对象监视器功能的增强、ActiveX 支持的增强等等。

Delphi 5 支持 XML 标准，可以在 Web 上快速提交商业信息，从而简化了数据分布、优化了数据交换过程。对 XML 的支持，使开发人员快速构建的系统具有弹性和可扩展性，以适应新时代的 Internet 电子商务发展需要。在 Delphi 5 中，可以利用 DHTML 和 XML 开发超“瘦” (ultra-thin) 客户端，并在 Internet 上快速分发标准 Web 特性的客户端应用程序。

ADO (Active Data Object) 技术是 Windows 操作系统中一种基于 COM 的数据库连接技术。利用 ADO，Delphi 5 能够快速访问关系型和非关系型数据库，其中包括电子邮件和文件系统。开发人员可以对 Internet 信息进行直接访问，帮助用户更好地进行商业决策。

Delphi 5 中引进了最新的 Frame 功能，使用 Frame，开发者能够可视化地为商业应用创建组件。开发人员可以创建、定制和重用自己的组件，也可以从 Delphi 的 Visual Component Library 中的 200 多个组件里选择适用的组件。而且，各个 VCL 源代码，包括新增的 Decision Cube 组件和附加的 Property Editors，以及为支持 MS Office 而提供的最新 Automation 组件，也可以供开发人员访问。这种基于组件的技术大大简化了许多复杂的开发任务。

Delphi 5 提供的 InterBase Express 可用于构建性能优越的应用系统。Delphi 5 使 Delphi 和 InterBase 更紧密地结合，即将高效率、高性能的开发工具与快速、低维护量和优化配置的中小型关系型数据库等技术融于一体。

鉴于 Delphi 的优良特性和广大读者的需要，我们继编写《Delphi 5 实战与精通》（实

战篇)》之后,在积累大量开发经验的基础上编写了本书。在本书的编著过程中,我们尽量减少冗长无味的说明,代之以具体实用的样例。通过这些实例引导读者把握 Delphi 的精髓所在。本书注重开发实例、开发经验、开发技巧和 Windows 高级特性开发,适合于较高层次的 Delphi 用户。有一定使用经验的读者可通过本书掌握 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。

第 1 章介绍 C/S(客户/服务器)应用原理、多层体系的基本原理以及 MIDAS 页中各个组件的用法,并通过具体的例子向读者介绍如何建立多层体系应用程序。

第 2 章简单介绍 Delphi 中的打印机对象,以及利用打印机对象进行文本打印、图形打印的一般方法。

第 3 章围绕如何在应用程序间进行数据交换的问题,介绍了动态数据交换 DDE 的概念、Delphi 中提供的 DDE 组件,并给出了使用 DDE 组件进行数据交换的例子。

第 4 章介绍构造自定义控件的方法。

第 5 章介绍 OLE 及其基础 COM(Component Object Model)技术在 Delphi 中的实现。

第 6 章在第 5 章的基础上介绍了 COM 技术的扩展——ActiveX 技术。第 5 章是第 6 章的基础,建议读者在学习第 6 章之前先学习第 5 章的内容。

第 7 章介绍如何在应用程序中加入多媒体组件——Animate 动画控件和 MediaPlayer 多媒体播放控件。使用 Animate 控件显示动态图画或图像,以及在需要声音支持的时候,使用 MediaPlayer 控件。

第 8 章是作者使用 Delphi 的经验汇集,内容相对比较庞杂。主要介绍了动态控件和控件数组,Sender 参数的使用技巧,以及注册表编程和使用多线程编程的方法。认真阅读本章将有助于开阔思路、提高技巧。

第 9 章结合当前热点,介绍了使用 Delphi 5 开发网络应用程序的方法,同时给出了一些有趣的例子。

第 10 章中主要介绍了如何在 Delphi 5 中调用 API 的方法。

最后,第 11 章介绍如何利用 InstallShield 制作一个符合商业标准的安装程序。

书中示例中所涉及的人名、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,林苇茗、任雨生、王登编著,韦思亮审稿。参与编写工作的还有 Wizard Group 的唐健、栗强、林国辉、方可军、王东辉、周兆确、毛诗华、方海英、唐家才、段小华、许刚、徐进明、施红芹、汪长春、叶青、王宇、邓增涛、谢威等。

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

2000 年 3 月于清华园

目 录

第 1 章 MIDAS 与客户/服务器编程	1
1.1 客户/服务器结构简介	1
1.1.1 客户/服务器体系结构	1
1.1.2 客户机、服务器以及中间件概述	2
1.1.3 客户/服务器型数据接口	3
1.1.4 Delphi 5 的解决方案	3
1.2 Delphi 客户/服务器应用开发环境的构造	5
1.3 多层体系简介	10
1.3.1 客户/服务器模式的不足	11
1.3.2 多层体系概述	11
1.4 Delphi 中的 MIDAS	12
1.4.1 Delphi 中的 MIDAS	12
1.4.2 MIDAS 中的组件	13
1.4.3 MIDAS 的应用程序工作流程	14
1.4.4 MIDAS 中连接方式的选择	15
1.4.5 MIDAS 多层数据库应用的建立步骤	16
1.5 MIDAS 在 Delphi 5 中的应用	17
1.5.1 如何创建应用服务器	17
1.5.2 建立服务器实例	22
1.5.3 建立客户应用	28
1.5.4 建立客户例程	29
1.5.5 客户端程序的完善和拓展	32
1.6 小结	36
第 2 章 实现打印功能	37
2.1 文本文件基本打印方法	37
2.2 利用打印对象 TPrinter 进行打印	42
2.2.1 TPrinter 对象及其打印操作	42
2.2.2 PrintDialog 和 PrinterSetupDialog 控件	45
2.2.3 利用 TPrinter 对象进行图像打印	47
2.3 强大的工具 QuickReport	54
2.3.1 QuickReport 组件	55

2.3.2 使用 QReport 建立基本报表	56
2.4 小结	61
第 3 章 动态数据交换 DDE	63
3.1 什么是动态数据交换	63
3.2 Delphi 5 提供的动态数据交换组件	64
3.2.1 TDdeServerConv 组件	64
3.2.2 TDdeServerItem 组件	65
3.2.3 TDdeClientConv 组件	66
3.2.4 TDdeClientItem 组件	69
3.3 动态数据交换例程	70
3.3.1 DDE 客户端程序的一个简单例子	71
3.3.2 DDE 客户服务器程序的编程	73
3.4 小结	79
第 4 章 自定义控件	81
4.1 创建一个最简单的控件	81
4.1.1 建立控件的准备工作	81
4.1.2 控件的类别	82
4.1.3 控件创建初步	83
4.1.4 编辑控件需要使用的资源	86
4.1.5 安装 MyBlank 控件	87
4.1.6 建立自己的组件页	89
4.2 创建输入电话号码的 Edit 控件	91
4.2.1 创建一个电话 Edit 框架并添加属性	92
4.2.2 为 TelEdit 添加事件	95
4.2.3 TelEdit 控件的完善	97
4.3 创建骰子控件	99
4.3.1 骰子控件设计构思	100
4.3.2 创建骰子控件框架	100
4.3.3 绘制控件外观	102
4.3.4 让骰子滚动	107
4.3.5 为控件添加事件	110
4.3 小结	114
第 5 章 OLE 与 COM	116
5.1 OLE 容器	116
5.1.1 什么是 OLE	116

5.1.2 Delphi 中的容器组件 OLEContainer.....	118
5.1.3 使用 OLE 容器组件的小例子	120
5.2 COM 基础	126
5.2.1 通用的软件组件	126
5.2.2 服务器	126
5.2.3 COM 接口	129
5.2.4 通用数据类型	130
5.2.5 COM 对象的生存期管理	132
5.2.6 COM 对象的内存管理	134
5.2.7 双接口	135
5.3 COM 技术在 Delphi 中的实现	136
5.3.1 使用 Type Library Editor	136
5.3.2 创建 COM 对象实例的方法	141
5.4 一个 Automation Server 例子	143
5.4.1 创建一个工程	143
5.4.2 为 Automation Object 编写框架	145
5.4.3 编写 COM 对象的程序代码	146
5.4.4 编写主窗体的代码	149
5.4.5 示例程序的补充说明	151
5.5 小结	152
第 6 章 ActiveX 控件	153
6.1 ActiveX 技术介绍	153
6.1.1 ActiveX 的主要特点	153
6.1.2 ActiveX 的要素及重要的概念	154
6.1.3 Delphi 5 对 ActiveX 的支持	155
6.2 使用 Delphi 5 中的 ActiveX 控件	156
6.2.1 Delphi 5 提供的 ActiveX 控件	156
6.2.2 使用 Graph 控件的示例程序	156
6.3 ActiveX 控件的制作	163
6.3.1 ActiveX 控件制作概述	163
6.3.2 ActiveX 控件制作实践	164
6.4 小结	176
第 7 章 多媒体程序开发	177
7.1 Delphi 5 提供的动画控件	177
7.2 简单的动画播放程序	183
7.3 使用 MCI 访问多媒体接口	189

7.4	自己的 CD 播放器	203
7.5	小结	212
第 8 章	高级使用技巧	213
8.1	动态控件与控件数组	213
8.1.1	动态生成一个不可视控件	213
8.1.2	动态生成可视的控件	215
8.1.3	为动态控件编写事件处理程序	216
8.1.4	使用控件数组	219
8.1	Sender	220
8.2.1	减少重复性的代码	220
8.2.2	检查接收消息的控件类型	222
8.3	使用注册表	222
8.3.1	注册表简介	223
8.3.2	TRegistry 类	224
8.3.3	使用 TRegistry 的例子	226
8.3.4	TIniFile 简介	227
8.4	多线程编程	229
8.4.1	线程	229
8.4.2	Delphi 提供的线程类 TThread	230
8.4.3	创建一个线程对象	236
8.4.4	访问共享数据	237
8.4.5	线程之间的事件触发	239
8.4.6	使用线程的示例	240
8.5	小结	245
第 9 章	网络编程	246
9.1	基本概念	246
9.1.1	网络、Internet 和 Intranet	246
9.1.2	网络的体系结构和 TCP/IP 协议	246
9.1.3	协议及常用术语	247
9.2	Delphi 5 提供的 Internet 组件	250
9.3	FTP 示例程序	256
9.4	电子邮件示例程序	269
9.5	WinSock 程序的示例：抢答器	278
9.6	Internet 高级编程技巧	291
9.6.1	为什么不使用控件	291
9.6.2	Delphi 5 对于手工实现 Internet 功能的支持	292

9.6.3 FTP 示例程序	292
9.7 小结	301
第 10 章 Delphi 5 中的 Win32 API 编程	302
10.1 API 概述	302
10.2 窗体管理	303
10.3 窗体控件	307
10.4 Windows 的外壳	308
10.4.1 任务栏	309
10.4.2 Windows 外壳的命名空间	314
10.4.3 文件的操作	317
10.5 网络支持	323
10.5.1 Window Socket API	323
10.5.2 WinInet API	330
10.6 小结	334
第 11 章 制作自己的安装程序	335
11.1 方便的 InstallShield	335
11.1.1 启动 InstallShield	335
11.1.2 新建一个工程文件	336
11.1.3 定制安装程序	338
11.1.4 编译安装程序	339
11.1.5 运行安装程序	340
11.2 InstallShield 的高级设置	341
11.2.1 Set the Visual Design 对话框设置	341
11.2.2 Select InstallShield Object for Delphi 对话框的设置	342
11.2.3 Specify Components and Files 对话框设置	344
11.2.4 Select User Interface Components 对话框	348
11.2.5 Make Registry Changes 对话框设置	349
11.2.6 Specify Folders and Icons 对话框设置	350
11.3 小结	351

第 1 章 MIDAS 与客户/服务器编程

客户/服务器模式是一种典型的两层体系结构，它将一个数据库应用系统分解为前台的客户（前端）和后台的服务器两部分，客户应用程序通过网络连接向服务器发送请求，服务器接收请求并进行处理，待处理结束后将结果返回客户应用程序。

MIDAS，英文全称为 Multi-tiered Distributed Application Services，是 Delphi 中最重要的技术之一。它打破了传统的客户/服务器模式数据管理，提出了革命性的“多层体系”概念，大大提高了网络数据处理的效率。

本章主要介绍 C/S（客户/服务器）应用原理、多层体系的基本原理以及 MIDAS 页中各个组件的用法，最后结合例子分析一下如何建立多层体系应用程序。

1.1 客户/服务器结构简介

下面介绍一些在客户机服务器编程中所涉及的基本概念。如果你对这部分比较熟悉，可以跳过本节。

1.1.1 客户/服务器体系结构

客户/服务器结构包括三部分，客户机和服务器之间的网络连接，如图 1-1 所示。客户机一般选用微机。而服务器多为大型机或小型机，但当对数据处理要求不是很高的情况下，可以用微机来代替。在 C/S 环境下，可以利用称作中间件(Middleware)的程序设计接口单元来达到客户机与服务器通信的目的。这些接口为前端应用程序和后台数据库之间提供了通路。

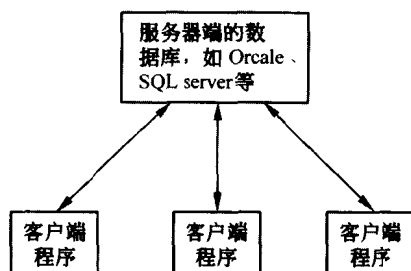


图 1-1 客户/服务器模式示意

下面简单列出服务器和客户机各自的功能。

数据库服务器的功能：

- 为多用户管理一个独立的数据库；
- 管理和处理接收到的数据访问请求，包括管理请求队列、管理缓存、响应服务并进行处理和通知服务完成等；
- 管理用户账号、控制数据库访问权限和其他安全性；
- 维护数据库，包括数据库的备份和恢复；
- 保证数据库的完整性。

客户机的功能：

- 建立与服务器的连接和断开与服务器的连接；
- 提交数据访问请求；
- 等待服务通告、接收请求结果或错误；
- 处理数据库访问结果或错误，包括重发请求或终止请求；
- 提供应用程序的友好界面；
- 数据输入输出及验证。

1.1.2 客户机、服务器以及中间件概述

客户机运行那些使用户能阐明其服务请求的程序，并将这些请求传送到服务器。由客户机执行的计算称为前端处理。前端处理具有所有与提供、操作和显示数据相关的功能。

客户机软件由网络接口软件、支持用户需求的应用程序以及实现网络能力的实用程序组成。网络接口软件提供各种数据传输服务。应用程序软件执行具体的任务，如字处理、电子表格和数据库查询生成。实用程序软件通常执行几乎所有网络用户都要求的标准任务。

在服务器上执行的计算称为后端处理。后端硬件是一台管理数据资源并执行数据库引擎功能(如存储、操作和保护数据)的计算机。

服务器软件既包括遵循于 OSI 或其他网络结构的网络软件，又包括由该服务器提供给网络上客户机的应用程序或服务软件。

中间件是一个软件层，它保护应用程序开发人员避免受到各种通信协议、操作系统以及数据库管理系统的影响。它为建立可与以前沿袭下来的应用程序并存的新应用程序打下了基础。

中间件的类型包括应用程序设计接口(API)、远程过程调用(RPC)、网络通信、数据库访问以及计算机辅助软件工程(CASE)工具。

由于客户/服务器系统需要集成各种不同结构的机器和技术，因而应用程序设计相当复杂。选择适当的中间件可以消除程序设计人员为每个单独协议和操作系统编写代码的麻烦。

1.1.3 客户/服务器型数据接口

客户/服务器的前端应用程序实际上不直接与数据库引擎打交道。每个客户服务器提供一个数据库通信接口, 该数据库通信接口运行于前端。这些接口也称为数据库通信 API 等。数据库通信接口的工作流程如下:

- (1) 前端应用程序发送命令给数据库通信接口;
- (2) 接口通过网络把命令传给数据库引擎;
- (3) 数据库引擎在服务时做查询或更新操作之类的工作, 通过网络文件系统访问物理数据;
- (4) 数据库引擎将结果返送给工作站上的通信接口;
- (5) 前端从接口上接到结果后, 显示或按用户要求做其他处理。

客户/服务器型比本地型 DBMS 更接近 ODBC 的原理。因为由前端向数据库发送命令和由数据库向前端返回结果都是透明的, 无须知道具体传送的方式, 各系统存在差别的地方是: 客户/服务器系统在管理工作站和服务端间通信的方式不同, 彼此会不兼容。而这些问题在 IDAPI 中得到了有效解决。

1.1.4 Delphi 5 的解决方案

IDAPI(集成数据库应用程序设计接口)是 Delphi 5 解决客户/服务器连通性的方案。它是通过 BDE(Borland Database Engine)和 SQL Links 来解决本地型数据库和客户/服务器型数据库接口问题的。

Delphi 的数据库特性使程序员能很容易构造数据库应用程序。这些应用程序能访问 Visual dBASE、Paradox、Local InterBase Server for Windows 等本地数据库和 Oracle、Sybase、Informix、SQL Server 和 Remote InterBase Server 等客户/服务器数据库。

BDE 是支持 Delphi 5 Client/Server Suite 的核心数据库引擎和互连软件。BDE 提供了丰富和强壮的特性支持客户/服务器应用的开发。

提供支持多种数据库如 dBASE、Paradox、Text、InterBase、Oracle、Sybase 和 Microsoft SQL Server 以及任何 ODBC 数据源的统一和一致的应用程序编程接口(API)。开发者不用修改数据库应用程序就能访问不同的数据库站点和数据库格式:

- BDE 是用于开发客户/服务器数据库应用程序的理想工具, 它使程序既可访问本地数据库又可访问远程数据库;
- 允许数据库直接或间接地访问数据源;
- BDE 对于 Paradox 和 dBASE 文件格式来说是高性能的数据库引擎;
- 支持使用 ISAM(Indexed Sequential Access Method)SQL 和 QBE 访问数据;
- BDE 是数据集成化引擎, 以提高不同数据库的共享服务。支持不同数据库格式的相互转化, 如 dBASE 和 Oracle 表、从 InterBase 到 Paradox 拷贝数据, 甚至建立 InterBase 和 Oracle 表之间的一对多关系;

- BDE 查询引擎为 SQL、QBE 和面向集合访问提供一致性的查询语言。支持用户定义和访问基本 SQL 的服务器和基于文件数据库的能力；
- BDE 支持全 32 位功能，如多线程、抢占式多进程、长文件名和 UNC，用户可在后台执行多个查询，多个数据库应用可访问同一个数据库文件 Paradox、dBASE 和文本数据库 BDE 包含的数据库驱动程序，支持对标准数据源的一致性访问。用户可以增加 ODBC 驱动和 Borland SQL Links 产品以支持对 SQL 服务器的数据访问，如 InterBase Oracle 和 Sybase 等。此外 BDE 给予 Windows 95 和 Windows NT 应用开发者以直接、独立共享的对多种数据源的高级访问。

BDE 在设计上是面向对象的。在运行时，数据库应用通过建立各种类型的 BDE 对象与 BDE 交互，这些运行的对象用于操作数据库实体如数据库表、查询。BDE 扩展的 API 支持 C、C++、Delphi 等对数据库引擎的访问。

在 Delphi 应用程序中访问数据库是通过调用 BDE 的 API 函数。Delphi 在库单元 BDE 中提供了大约 30 多个 API 函数和各种 BDE 消息和结构。由于 Delphi 应用程序的开发是基于部件的，有关 BDE API 的调用都嵌入了 Delphi 可视部件类库，因此，建立数据库应用时可以不必管 BDE API 的细节。只要正确安装 IDAPI 的 Drivers，并进行正确的配置，就能使你的数据库应用程序与服务器连接并访问数据库。当然为了提高应用程序的数据库访问性能，可以在程序中直接调用 BDE API 函数。

Borland 的 IDAPI 包含对 ODBC 的支持，因此通过 BDE，编写的应用程序能访问一切与 ODBC 兼容的数据库如 Access 等。

Local InterBase Server 提供了一个单用户多实例的 SQL 服务器平台，特别是在将数据库应用程序转移到对 Oracle、Sybase 和 Informix 等远程数据库的访问之前，可以在 Local InterBase Server 平台建立和测试数据库应用程序。

对客户/服务器型的数据库应用程序来说，SQL Links 在 Delphi 数据库体系结构中起着至关重要的作用。

Borland SQL Links 支持访问局部和远程 SQL 数据库的 BDE 应用程序。为了访问特定的远程 SQL 服务器，必须在客户端安装相应的 SQL Links 驱动程序。安装了 SQL Links 驱动程序后，SQL 表达式才能被传送给相应的服务器执行。

一旦安装了 SQL Links 的驱动程序，并建立了 SQL 驱动程序的 Alias，就能采用下列方法使数据库应用程序像访问本地 Paradox 和 dBASE 数据库一样访问远程数据库：

- 通过应用程序用户接口(界面)；
- 通过包含嵌入 SQL 表达式的应用程序；
- 直接将 SQL 表达式传送给服务器。

安装了 SQL Links 驱动程序，就能用 SQL 访问数据。驱动程序负责 SQL 服务器的链接，将查询语句转换成兼容的 SQL 语句，并将它们传送给 SQL 数据库。等处理完成后，SQL 数据库把结果以应用程序平台能显示的格式返回给客户。其工作流程如下：

(1) BDE 客户查询 SQL 数据库；

(2) SQL Links 驱动程序建立客户工作站与 SQL Server 的链接，并将查询发送给 SQL 服务器；

(3) SQL 服务器进行 SQL 表达式的错误和语法检查, 处理查询并将结果返回给 BDE 客户;

(4) SQL Links 驱动程序将结果通过 SQL 转换成客户端能识别的形式, 客户端将其进行格式化并将数据显示给用户。

在应用程序中使用 SQL Links 驱动程序对使用 SQL 的数据库具有如下好处:

- 直接查询 SQL 服务器的能力;
- 支持 SQL 网络的传输;
- 增强了记录的快速存取;
- 数据锁定;
- 在 SQL 结果集和数据库表的双向变换;
- 使用索引给数据排序;
- 为数据库当前 Session 建立书签, 并可在以后重用书签;
- 通过动态访问数据源来动态处理 SQL 数据。

1.2 Delphi 客户/服务器应用开发环境的构造

下面将向读者介绍如何正确配置数据库开发环境。

选择菜单项 Borland Delphi 5 中的 BDE Administrator 项, 单击该图标, 会出现如图 1-2 所示的窗口, 该窗口包括了两个面板。

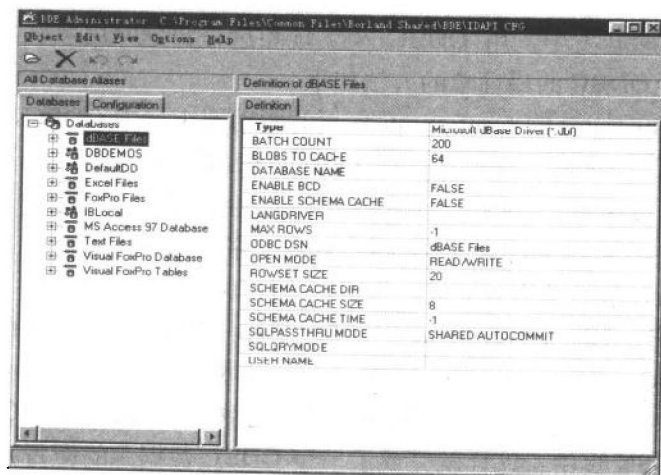


图 1-2 BDE 管理工具

左部的面板以层次树的形式显示数据库对象, 在 Databases 页中, 列出了所有的数据库别名; 在 Configuration 页中, 列出了数据库驱动和 BDE 结构设置信息。右部的多页面显示了左部面板被选中项的具体信息。

在左部面板中, 对象旁边的一个“+”标志表示该对象还包含可扩展的信息。只要

单击“”，就可以看到这些扩展信息了。如果对象已不能再扩展，“”就变成了“”。如果要隐藏这些扩展信息，可以单击“”，这样就回到初始的对象信息。

要想查看左部面板中对象的详细信息，可以双击该对象。这时，右部面板就会以一页或多页的形式显示对应对象的信息。例如，当在左部面板中选中的一个数据库别名时，右部面板就会显示 **Definition** 页，其中包含了 **Type**、**DATABASE NAME** 等参数。这些参数中以粗体显示的表示不可修改，其他参数则是可以编辑的。

下面介绍用 **BDE Administrator** 来管理 BDE 的别名，这些操作包括查看、创建和修改别名。这几种操作的步骤简介如下：

查看别名 首先，单击数据库对象旁边的“”标记，可以看到一组别名对象；然后，在左部面板中选择要查看的别名，该别名的定义会自动出现在右部的面板中。

创建别名 首先在左部面板中选择一个数据库对象；右击调出上下文菜单；选择 **New** 项；在随后出现的 **New Database Alias** 对话框中选择一个驱动器并选择 **OK**；在左部面板中输入此别名；在右部面板的 **Definition** 页的 **PATH** 项或 **SERVER NAME** 项输入别名对应的路径，也可以在 **Default Driver** 参数项指定驱动器名；右击左部面板中的数据库对象，在弹出的菜单中选择 **Apply** 项或单击确认创建该数据库。

修改别名 在左部面板中选择要修改的别名，右部面板中会出现 **Definition** 页；根据需要编辑 **PATH** 参数项和 **Default Driver** 项；在左部面板中的数据库对象处右击，在弹出式菜单中选择 **Apply** 项或单击以更新数据库。

删除别名 在左部面板中选择要删除的别名；右击调出上下文菜单；选择 **Delete** 项删除别名；在左部面板中右击数据库对象，在弹出式菜单中选择 **Apply** 项或单击以更新数据库。

在分布式数据库开发中，用 **ODBC** 来实现和远程数据库服务器的连接是一个较为普遍的做法。配置 **ODBC** 的最简单的方法如下：

在 **BDE Administrator** 的 **Configuration** 页的驱动器中选中 **ODBC**，或在 **Database** 页中选中一个 **ODBC** 数据库；选择菜单项 **Object|ODBC Administrator**，在随后出现的对话框的用户 **DSN** 页中单击“添加”按钮；这时，会出现一个“创建新数据源”的对话框，可从中选取待添加的 **ODBC** 数据源的驱动程序。为举例方便，我们从这些驱动程序选择 **SQL Server**，然后单击“完成”按钮，如图 1-3 所示。

随后出现 **ODBC SQL Server Setup** 对话框，其中一般需要设定 **Data Source Name** 项、**Server** 项和 **Network Address** 项，如果需要进行详细的设置，还可以单击 **Options>>** 按钮，进行细致的设定；然后再按“确定”按钮。可以看到，设置的 **ODBC** 数据源出现在最初对话框的用户 **DSN** 页中，这样就完成了一个 **ODBC** 数据源的配置。

上面介绍的配置方法主要是由系统来修改配置文件，属于自动配置。如有必要，也可以尝试手工配置 **ODBC**。其一般过程为：先配置 **ODBC** 结构文件，然后配置 **BDE** 结构文件以支持 **ODBC**。

有两个文件包含了 **ODBC** 的配置信息，一个是 **Windows** 的注册表文件 **ODBCINST.INI**，这个文件列出了 **ODBC** 驱动器；另一个文件是 **ODBC.INI**，它列出了 **ODBC** 数据源。图 1-4 所示是 **Windows** 系统注册表中关于 **ODBC** 的键值。