

《电脑编程技巧与维护》杂志十周年庆典暨真情回馈读者活动
《电脑编程技巧与维护》杂志社策划

编程技巧典型案例集锦系列

《电脑编程技巧与维护》杂志社 编著

PowerBuilder 管理信息系统

编程技巧典型案例解析

- 数据移植功能的实现方法
- 随机统计与组合查询的实现
- 查询 Sybase SQL Server 的表和存储过程
- 制作中国式报表
- 发送 Notes 邮件
- 通过菜单实现 PowerBuilder 应用系统的安全性

超值1CD, 38个实例, 45000条代码,
编程高手经验汇集, 现学现用



中国电力出版社
www.infopower.com.cn

《电脑编程技巧与维护》杂志十周年庆典暨真情回馈读者活动
《电脑编程技巧与维护》杂志社策划

编程技巧典型案例集锦系列

光盘
PowerBuilder 管理信息系统
编程技巧典型案例解析

《电脑编程技巧与维护》杂志社 编著



中国电力出版社

www.infopower.com.cn

内 容 简 介

本书主要通过77个实例,从PowerBuilder数据库的编程基础与应用、查询系统、报表生成系统、网络数据库、数据库安全与维护及数据库编程技巧等6个方面对如何利用PowerBuilder实现管理信息系统的各种方法和技巧进行了详细讲解。这些实例丰富而且典型,具有较强的实用性和代表性,此外,有的实例还附加了经典的专家点评和分析,以帮助读者更好地理解 and 掌握所讲的知识点,使读者达到触类旁通、举一反三的目的。

本书语言通俗易懂、讲解精辟,是PowerBuilder程序员、数据库编程和管理人员、网络编程和维护人员及广大编程爱好者和计算机相关专业的学生的首选参考书。

图书在版编目(CIP)数据

PowerBuilder管理信息系统编程技巧典型案例解析 / 《电脑编程技巧与维护》杂志社编著. —北京:中国电力出版社,2005

(编程技巧典型案例集锦系列)

ISBN 7-5083-3276-8

I .P... II .电... III .数据库系统 - 软件工具, PowerBuilder - 程序设计 IV .TP311.56

中国版本图书馆CIP数据核字(2005)第032271号

版权声明

本书由中国电力出版社独家出版。未经出版者书面许可,任何单位和个人不得以任何形式复制或传播本书的部分或全部内容。

本书内容所提及的公司及个人名称、产品名称、优秀作品及其名称,均为所属公司或者个人所有,本书引用仅为宣传之用,绝无侵权之意,特此声明。

策 划:裴红义

姚贵胜

责任编辑:夏华香

责任校对:崔燕菊

责任印制:李志强

丛 书 名:编程技巧典型案例集锦系列

书 名:PowerBuilder管理信息系统编程技巧典型案例解析

编 著:《电脑编程技巧与维护》杂志社

出版发行:中国电力出版社

地址:北京市三里河路6号 邮政编码:100044

电话:(010)88515918 传真:(010)88518169

印 刷:利森达印务有限公司

开本尺寸:185 × 260 印 张:26.5

书 号:ISBN 7-5083-3276-8

版 次:2005年8月北京第1版

印 次:2005年8月第1次印刷

印 数:1~4000

定 价:45.00元(含1CD)

丛 书 序

在《电脑编程技巧与维护》杂志创刊 10 周年之际，为了真诚回报多年来一直关爱和支持本刊的广大读者，《电脑编程技巧与维护》杂志社和中国电力出版社共同策划出版了《编程技巧典型案例集锦系列》丛书。《电脑编程技巧与维护》杂志是为从事电脑编程、系统应用开发人员创办的专业性和实用性都很强的技术刊物，它从 1994 年创刊，十多年来始终遵循着“实用第一，智慧密集”的办刊宗旨，紧跟计算机软硬件技术发展和应用趋势，不断求变创新，针对软件开发过程中许多关键技术问题，着重提供各类解决方案。对电脑编程人员来说，程序开发能力的提高，除了对语言和算法的学习外，还要集思广益，充分借鉴参考别人的长处，深入透彻地理解其中的精髓，然后融入到自己的设计方案中去，这样无论是对于自身还是整体都有莫大的提高，这也正是我们编写这套系列丛书的初衷。

本丛书包括《Visual C++ 编程技巧典型案例解析——基础与应用篇（上）》、《Visual C++ 编程技巧典型案例解析——基础与应用篇（下）》、《Visual C++ 编程技巧典型案例解析——图形图像处理与数据库篇》、《Visual C++ 编程技巧典型案例解析——网络与通信及计算机安全与维护篇》、《Visual Basic 编程技巧典型案例解析》、《Delphi 编程技巧典型案例解析》、《C#编程技巧典型案例解析》、《Java 编程技巧典型案例解析》、《PowerBuilder 管理信息系统编程技巧典型案例解析》9 册共 545 个典型案例。每册书的编程案例，均依不同的编程应用分成若干章，条目清晰可查，使用极为方便。

本丛书选编了《电脑编程技巧与维护》杂志近一两年发表的和一部分尚未发表而又极为实用、精彩的典型编程实例，特点是：其各册内容均来自编程高手的智慧，凝结了 500 余位编程高手与名家的心血，关键技术专家点评；其案例是从实际项目提炼出的开发范例，超过 800 个技术要点的经典解决方案。案例讲解部分先给出设计目标，然后介绍实现目标的基本思想和方法，最后详细给出其核心程序的源代码，对程序的关键部分进行讲解并给出程序的运行效果；其编程技巧新颖实用，构思巧妙，汇集了众多顶级程序员和业界知名专家的成功经验，告诉读者最好的创意和最实用的方法。全套书既讲究内容的深入性、专业性和权威性，同时兼顾轻松、通俗易懂、时效性强的特点，带给读者的是一份清

新、纯粹的体验感受。

本丛书是《电脑编程技巧与维护》杂志资源的二次开发，浓缩了当前主流编程语言 Visual C++、Visual Basic、Delphi、Java、C#、PowerBuilder 等程序设计的精华，其目的是力求为读者建造一个真正的知识整合，是编程思想、编程技术、技巧交流的平台，让读者从中学习到编程高手的诀窍，丰富读者的编程技巧，拓宽读者的编程思路，迅速提升读者的程序开发能力。该丛书可作为高等院校学生进行课程项目开发、毕业项目设计的参考教材，软件从业人员及编程爱好者的珍藏宝典，也可作为高等培训学校的案例教程。

实例导航学编程，自学成才成高手，思想、智慧、理念、经验、技巧无处不在……

《电脑编程技巧与维护》杂志社

2005 年 1 月

前 言

要精通编程，要成为一名优秀的程序员，这不仅需要不断地超越自我，还需要有个好的导师带你入门。超越自我需要自身的修行，“编程技巧典型案例解析”则是学习编程的好导师。对程序员来说，不断提升自己的编程功力，除了对语言和算法的深入学习和透彻理解，不断超越自我外，还需要集思广益，充分学习借鉴别人的长处，并深入透彻理解其中的精髓，然后溶入到自己的设计中去，这样无疑能较快地提高自己的编程能力和水平。

为了让更多的电脑编程人员比较集中地学习和参考 PowerBuilder 应用编程的实践经验、心得体会和技巧，在《编程技巧典型案例集锦系列》丛书中，《PowerBuilder 编程技巧典型案例解析篇》一书精选了《电脑编程技巧与维护》杂志近两年半共 30 期已发表的精彩编程实例 77 例。根据 Delphi 的不同应用对象，将精选的 77 个应用实例分为 6 章。第 1 章 PowerBuilder 数据库编程基础与应用，为初学者提供 PowerBuilder 基础应用编程入门的实例；第 2 章 PowerBuilder 数据库查询系统编程，为编程人员提供使用 PowerBuilder 设计数据库查询系统编程方法和技巧的实例；第 3 章 PowerBuilder 数据库报表生成系统编程，介绍使用 PowerBuilder 数据库报表生成系统的编程技巧的实例；第 4 章 PowerBuilder 网络数据库编程，通过一些应用实例介绍使用 PowerBuilder 实现网络与通信应用编程的方法；第 5 章 PowerBuilder 数据库安全与维护编程，通过实例，介绍使用 PowerBuilder 实现数据库安全与维护的编程方法；第 6 章 PowerBuilder 数据库编程技巧，介绍一些经验丰富的资深 PowerBuilder 编程人员使用 PowerBuilder 在编程中对所遇到的关键技术进行深入探讨的实例，这些编程实例对初学 PowerBuilder 编程和有一定基础需要进一步提高编程能力的程序员来说是很好的学习参考资料。

本书的主要特色如下：第一，每一章都是通过一个个的实例来介绍 PowerBuilder 应用编程方法和技巧，避免了枯燥、空洞的理论，并且每一个实例都具有很强的实用性和代表性。在实例的讲解上一般都是先给出设计目标，然后介绍实现该目标的基本思想和方法，最后详细给出其核心程序的源代码，并对程序的关键部分进行讲解，给出程序的运行效果。第二，所选的每一个实例都是从事 PowerBuilder 应用编程人员的经验总结，具有很强

的实用性，其中很多编程技巧可供借鉴。第三，每一个实例的程序源代码都经过上机调试通过，给程序开发人员移植源代码带来了方便，加快编程应用的步伐。第四，对个别版本和开发环境稍微低一些的经典实例进行点评和分析，起到触类旁通的作用。

本书是《电脑编程技巧与维护》杂志的二次开发，浓缩了 PowerBuilder 程序设计的精华，其目的是提升读者 PowerBuilder 程序开发的能力，把应用 PowerBuilder 进行编程的心得体会、经验与读者共享。该书定位于有 PowerBuilder 应用基础的编程人员和应用开发人员，对初学 PowerBuilder 编程的新手也有一定的参考价值。该书内容全面、概念清晰、层次分明、实例典型而实用，但不足甚至疏漏之处在所难免，恳请广大读者批评指正。

《电脑编程技巧与维护》杂志社

2005 年 1 月

目 录

丛书序

前 言

第 1 章 PowerBuilder 数据库编程基础与应用

实例 1 PowerBuilder 应用基础	3
实例 2 PowerBuilder 游标技术及查询方法	9
实例 3 PowerBuilder 数据窗口的应用及打印程序的设计	15
实例 4 PowerBuilder 基础类库及外部函数与外部程序的调用	22
实例 5 PowerBuilder 中调用公用控件动态链接库实现 IP 地址控件	28
实例 6 PowerBuilder 应用程序的调试和安装	33
实例 7 PowerBuilder 中数据多维分析框架的设计和实现	39
实例 8 基于 PowerBuilder 的窗口间参数传递	43
实例 9 用 PowerBuilder 7.0 简化特殊字段信息录入	47
实例 10 对数据窗口的 dwItemStatus 的研究及应用	51
实例 11 用 PowerBuilder 设计一个高效、实用的树/列表视图界面	55
实例 12 PowerBuilder 下拉式数据窗口增量式录入的实现方法	60
实例 13 PowerBuilder 中渐进式记忆输入功能的设计与实现	62
实例 14 在 PowerBuilder 中取得 DataWindow 内数据的聚合计算值的两种方法	65
实例 15 如何在 PowerBuilder 中实现界面的自动布置	70
实例 16 PowerBuilder 环境下一个高效录入界面的设计与实现	79
实例 17 PowerBuilder 环境下通用录入与编辑控件的设计与开发	83
实例 18 基于 PowerBuilder 的通用灵活打印构件的设计	88
实例 19 PowerBuilder 大文本数据的打印	93
实例 20 PowerBuilder 数据窗口的通用打印模块的设计	99
实例 21 PowerBuilder 下实现对医保病人病案的完善	105
实例 22 PowerBuilder 应用程序中的按钮设计	110
实例 23 如何在 PowerBuilder 中实现 Windows 的资源管理器	117
实例 24 在 PowerBuilder 中如何遍历树形结构数据	123
实例 25 PowerBuilder 对 INI 文件和注册表的读写实现数据库连接	128

实例 26	PowerBuilder 利用 OLE Object 对象将数据窗口导出到 Excel 表	132
实例 27	利用 PowerBuilder 数据窗口技术进行动态统计图显示	138
实例 28	如何在 PowerBuilder 中实现带帮助按钮的消息对话框	143
实例 29	PowerBuilder 如何激活数据库用户角色	147
实例 30	PowerBuilder 中事物对象的多重数据库连接	153
实例 31	如何用 PowerBuilder 实现工资软件的通用	159
实例 32	PowerBuilder 应用程序中数据移植功能的实现方法	164

第 2 章 PowerBuilder 数据库查询系统编程

实例 33	基于 PowerBuilder 综合查询自定义函数的方法和应用	175
实例 34	PowerBuilder 中一个通用查询控件的实现	180
实例 35	在 PowerBuilder 中对数据进行实时查询	184
实例 36	PowerBuilder 中实现多线程同步查询	186
实例 37	PowerBuilder PFC 编程通用中文查询模块的实现	189
实例 38	用 PowerBuilder 查询 Sybase SQL Server 的表和存储过程信息	195
实例 39	PowerBuilder 下随机统计与组合查询的实现	199
实例 40	PowerBuilder 中利用数据窗口技术实现通用查询	207
实例 41	PowerBuilder 多关系组合查询方法和技巧	212
实例 42	PowerBuilder 数据窗口中实现模糊查询	217

第 3 章 PowerBuilder 数据库报表生成系统编程

实例 43	用 PowerBuilder 制作中国式报表	223
实例 44	从 PowerBuilder 数据窗口到 Excel 报表转换通用函数的实现	228
实例 45	用数据库存设计 PowerBuilder 复杂统计报表	234
实例 46	PowerBuilder 数据窗口中的多选列表框及后台数据保存的处理技术	238
实例 47	PowerBuilder 中与 Excel 间的传递数据	243
实例 48	PowerBuilder 中一个类似 Word 打印对话框的实现	248
实例 49	PowerBuilder 中实现可视对象的移动与改变大小功能	252
实例 50	在 PowerBuilder 中选择数据窗口中的某些字段生成带中文标题的 Excel 文件	257
实例 51	关于 PowerBuilder 动态报表技术的研究	264

第 4 章 PowerBuilder 网络数据库编程

实例 52	用 PowerBuilder 发送 Notes 邮件	273
实例 53	PowerBuilder 利用 mswinsck.ocx 控件进行远程通信	280
实例 54	远程网络数据库的连接及快速数据传输技术	285
实例 55	在 PowerBuilder 中利用 Delphi 的多线程机制实现对串口的实时监听	292

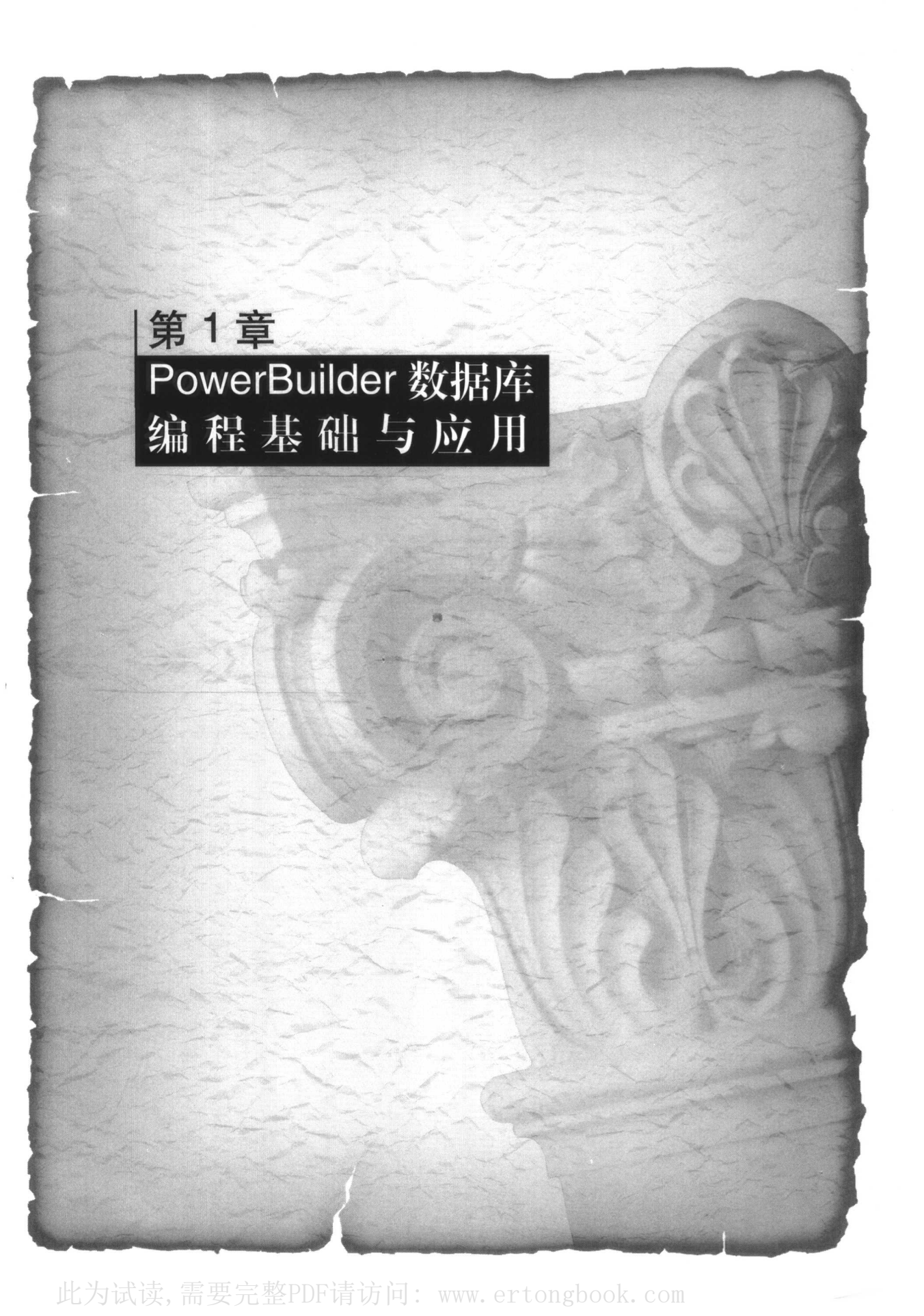
实例 56 用 PowerBuilder 编写简易的邮件收发程序	297
--	-----

第 5 章 PowerBuilder 数据库安全与维护编程

实例 57 PowerBuilder 用户权限分配的实现策略	303
实例 58 PowerBuilder 中任意设置操作员对菜单项的操作权限	308
实例 59 信息系统应用软件自动升级的方法与工具	313
实例 60 通过菜单实现 PowerBuilder 应用系统的安全性	321
实例 61 使用 PowerBuilder 设置 Windows 98 应用程序的使用权限	329
实例 62 利用 PowerBuilder 生成软件的注册号	332
实例 63 在 PowerBuilder 中利用动态链接库实现数据加密	336
实例 64 在 PowerBuilder 中实现对 Oracle 数据库的系统维护	339
实例 65 利用 PowerBuilder 数据窗口和树型控件生成多层结构数据库	348

第 6 章 PowerBuilder 数据库编程技巧

实例 66 用 PowerBuilder 实现通用的右键菜单程序	359
实例 67 使用 PowerBuilder 开发应用程序过程中遇到的一些问题及解决办法	362
实例 68 基于 PowerBuilder 下的结构实现通用窗口动态连接数据窗口对象及 改变窗口属性	374
实例 69 PowerBuilder 8.0 中数据窗口行选择的应用技巧	377
实例 70 在 PowerBuilder 中处理图片字段	379
实例 71 数据项的动态管制方法与实现	381
实例 72 PowerBuilder 中大图片的浏览	389
实例 73 在 PowerBuilder 中利用用户对象封装 API 函数的方法	392
实例 74 PowerBuilder 子数据窗口的设计技巧	396
实例 75 PowerBuilder 8.0 中通用打印控制界面的实现技巧	400
实例 76 让 PowerBuilder 中的文字动起来——PowerBuilder 动态 About 窗口实现	404
实例 77 PowerBuilder 编程中的若干经验和技巧	408



第 1 章

PowerBuilder 数据库 编程基础与应用

■实例 1

PowerBuilder 应用基础

PowerBuilder 是一个用来进行客户机 / 服务器开发的完全可视化开发环境, 采用面向对象技术、图形化应用开发环境, 是数据库的前端开发工具, 可以用一种可视的、直观的方式来创建应用程序的用户界面和数据库接口。所开发的应用系统广泛地应用于金融、证券、邮电、商业等各个部门和各个领域, 并发挥了巨大的作用。据有关机构统计, PowerBuilder 目前在中国数据库前端开发市场占有率达到 40% 左右。

PowerBuilder 采用的是目前流行的图形化的界面和可视化的编程方法, 通过引入独具特色的数据窗口对象, 使得程序员可以可视化地完成对数据库的操作。PowerBuilder 提供了对目前流行的绝大多数的大型数据库和桌面数据库的支持, 同时它自身也附带了一个小的数据库管理系统 Sybase SQL Anywhere。PowerBuilder 支持多种软、硬件平台, 它不仅跨平台共享程序中的各种对象, 还支持应用程序的跨平台开发和发布, 极大地减轻了程序员在不同平台上移植程序的工作量。

本实例以 PowerBuilder 8.0 为蓝本, 由浅入深地向大家介绍 PowerBuilder 应用与开发技巧。我将整个实例分成 PowerScript 语言及 PowerBuilder 对象、鼠标技术及查询方法、数据窗口及打印程序的实现、基础类库及外部函数程序的调用、调试和安装五部分, 将自己在开发大型 MIS (某监狱管理局管理信息系统) 中的经验和体会渗入其中, 希望大家能从中得到启示, 学有所得。

一、PowerScript 语言

PowerScript 是 PowerBuilder 使用的程序设计语言, 它不是简单的第四代语言, 而是一种高级的、结构化的编程语言, 我们使用 PowerScript 编写应用程序中使用的函数和事件处理程序。如果大家以前使用过其他高级语言的话, 就会发现 PowerScript 十分简单、易学, 有点类似于 Visual Basic 与 C 的混合物。

PowerScript 与众不同之处在于它提供了一套完整的嵌入式 SQL 语句, 开发人员能够像使用其他语言成分 (比如赋值语句) 那样使用它们, 而且需要的话也可以使用特定数据库管理系统专用的 SQL 语句, 这就大大增强了程序操纵和访问数据库的能力。需要嵌入式 SQL 语句时, 无需逐字键入, 可利用 PowerBuilder 提供的可视化工具生成嵌入式 SQL 语句, 然后把它粘贴到程序中。

数据类型、嵌套语句、操作符、系统函数等语言基础, 这里就不一一赘述了, 主要谈谈 PowerScript 语言与其他高级语言的不同之处。其中, PowerScript 语言有一种标准数据类型 BLOB (Binary Large Objects, 大型二进制对象) 是其他语言所没有的, 非常具有代表性。

当遇到下面 4 种情况时, 我们要使用到 Blob 数据类型。①当遇到要将 OLE 对象 (如图形、声音等) 存入数据库时; ②当要将大型的二进制对象存入数据库时; ③当要操纵的文本对象过大以致一般的字符串函数无法对其操作时; ④当所使用数据库的数据类型 PowerBuilder 不能支持时, 只能

使用 Blob 函数对其进行操纵。在我们编写的程序中，记录罪犯信息的照片就是由 Blob 类型存储的。

Blob 型变量声明的语法为：

Blob {{大小}}变量名 //其中第一层大括号表示可以省略其中的成分。

声明 Blob 型变量时，可用大括号括起一个整数，该整数指明 Blob 变量以字节为单位的长度，默认该项时，默认长度为 0，且系统会根据数据自动调整其长度。

Blob {100} lb_full_photo //声明一个长度为 100 的 Blob 型实例变量。

Blob lb_photo, lb_blob //声明一个默认长度为 0 的 Blob 型变量。

需要注意的是，Blob 型变量不能设置初值，而且一般用 UPDATEBLOB 和 SELECTBLOB 两个 SQL 语句来实现数据的写入。

```

* * * * *
Blob lb_full_photo, lb_photo, lb_blob
IF GetFileName ( "选择图片", ls_FullFileName, ls_FileName, "bmp", "BMP FILES, *. bmp" )
= 0 THEN
    IF NOT IsNull ( lb_Photo ) THEN
        p_photo. Visible = TRUE
        SetPicture ( p_photo, lb_Photo )
    .....
//获取当前选定的照片文件大小
FOR i = 1 to loops
    //将照片读取到共享 Blob 变量 sb_photo 中
    FileRead ( li_file_open_return_value, lb_blob )
    //将所有的 Blob 变量相加
    lb_full_photo = lb_full_photo + lb_blob
NEXT
//关闭照片文件
FileClose ( li_file_open_return_value )
//将 Blob 照片变量更新到表中
UPDATEBLOB CLT_JTCLDA SET zhaopiao = :
lb_full_photo where CLID = :ls_person_id;
* * * * *

```

丰富的数据类型和众多的函数进一步提高了 PowerBuilder 的适应性和灵活性，熟练地掌握 PowerScript 语言是灵活高效地开发高质量 PowerBuilder 应用程序的基本要求。

二、对象的属性、事件和函数

丰富的对象是 PowerBuilder 的特色之一，它们为应用程序的开发提供了极大的方便，熟悉和掌握各种对象的功能、特点，有利于快速开发出高效、实用的应用程序。

1. 属性

属性也就是数据，包括系统属性和用户定义属性两类，描述该对象的各种特性。如在窗口对象中的系统属性，包括标题、高度、宽度等，用户定义的属性可以是实例变量或共享变量，对这种属性的访问同系统属性相同，只是在封装性上有所不同。

2. 事件

事件是 PowerBuilder 应用系统开发平台中一个非常重要的概念。在 PowerBuilder 环境中，应用系统的运行是由事件来驱动的，例如单击一个命令按钮，这个按钮的某个事件中的 PowerScript 脚本就会被执行，PowerBuilder 编程就是编写事件脚本。PowerBuilder 中的事件与 Windows 的事件存在着

映射关系。当用户的操作或系统本身产生了 Windows 的标准事件，就转化成 PowerBuilder 中的事件。例如命令按钮有一个叫 Clicked 事件，当用户按按钮时触发，操作系统首先检测到鼠标被单击，并把 Clicked 这一消息转给该按钮，PowerBuilder 运行时的工作引擎翻译了这一事件，并执行适当的 PowerScript 语句。PowerBuilder 中大部分的事件都是可以映射成由用户激发的 Windows 事件。另一类事件是用户定义事件，这种事件是用户声明并可以在任何时刻由用户触发的，PowerBuilder 中保留了 75 个用户事件可由程序员使用。

3. 函数

用户通过函数来使对象完成某些操作。建议用户通过调用对象事件和函数来修改对象而不要直接修改对象属性，以满足对象封装的要求，例如采用 Window.hide() 而不要用 Window.visible = false 来使窗口不可见。

函数和事件可以统称为方法，调用的语法为：

```
{ objectname. } { type } { calltype } { when } functionname ( { argumentlist } )
```

type：所调用方法的类型，可以是 EVENT 或 FUNCTION，其中 FUNCTION 为默认。

calltype：所调用的类型，可以是动态（关键词为 DYNAMIC）或静态（关键词为 STATIC），其中 STATIC 为默认。

when：调用的时间，POST 或 TRIGGER，其中 TRIGGER 为默认。

4. 消息对象和事务对象

PowerBuilder 提供了 70 多个系统预定义对象，有连接（Connection）对象、数据存储（DataStore）对象、子数据窗口（DataWindowChild）对象等等，应用程序还可以通过继承对象的方法定制出符合自己特殊需要的用户对象。在这里我们通过对典型的语言消息（message）对象和事务（Transaction）对象的描述，使用户加深对 PowerBuilder 对象的属性、事件和函数的理解。

（1）消息（Message）对象

消息对象有很多用途，它既可以用于处理非 PowerBuilder 预定义事件（这类事件指没有定义为 PowerBuilder 事件的 Windows 消息），也可以在打开或关闭窗口时在窗口之间传递消息，还能保存执行 TriggerEvent() 或 PostEvent() 函数时传递的事件参数。

PowerBuilder 为每一个应用程序都预先定义了一个全局消息变量 Message。可以使用用户对象画笔来定义继承了消息对象的标准类用户对象，并通过在应用对象的属性对话框中把消息对象的类型更改为开发人员定义的标准类用户对象，从而定制出自己所需版本的消息对象。

消息对象的 Handle、Number、WordParm、LongParm 4 个属性对应于 Microsoft Windows 消息结构的前 4 个属性，而下面 6 个属性是 PowerBuilder 独有的见表 1-1。

PowerBuilder 中函数 OpenWithParm()、CloseWithReturn()、OpenSheetWithParm() 等都有一个在进行窗口操作（打开或者关闭）时存入 Message 消息对象的参数 parameter 或 returnvalue。从 Message 消息对象的不同属性中提取该函数并传送给被打开窗口的值。消息对象 Message 有 3 个属性用于存储这些函数传递给窗口的数据（依赖于窗口操作时参数赋予的类型）。根据 parameter / returnvalue 参数数据类型不同，该参数的值保存在 Message 对象的不同属性中。

在 PowerBuilder 中如果是传递一个参数，是很容易实现的。在调用脚本中，可以使用如下代码：

```
openwithparm(w_prisoner, parmtotrans)
接着在打开窗口 w_prisoner 的 open 事件中输入：
string ls_getparm
ls_getparm = message.stringparm
...
```

表 1-1 PowerBuilder 属性表

属 性	数据类型	意 义
ClassDefinition	PowerObject	指明该对象的类定义信息
DoubleParm	Double	数值参数值或数值型变量
StringParm	String	字符串参数值或字符串型变量
PowerObjectParm	PowerObject	用于传递任何 PowerBuilder 对象类型,包括结构在内
Processed	Boolean	
ReturnValue	Long	在用户定义事件或 Other 事件的事件处理程序中设置的值,用于指明该事件的进一步处理方式
		当 Message.Processed 属性的值为 True 时,该参数指定要返回给 Windows 的值;当值为 False 时,该参数被忽略

如果需要传递多个参数,可以使用如下方法:

定义结构变量: `lstr_declaredstr`, 其中包含想传递的各种参数:

变量名	变量类型
Prisoner_ID	Unsignedlong
Prisoner_Name	Character
Prisoner_address	Character
Prisoner_old	Number

...

在调用脚本中, 给结构变量赋值:

```
lstr_declaredstr lstr_parmtotrans
lstr_parmtotrans.Prisoner_ID = 0102451
lstr_parmtotrans.Prisoner_Name = "陈超"
lstr_parmtotrans.Prisoner_address = "湖南长沙"
lstr_parmtotrans.Prisoner_old = "23"
...
```

```
openwithparm(w_prisoner, parmtrans)
```

接下来在打开窗口 `w_prisoner` 的 `open` 事件中, 获取结构信息:

```
lstr_declaredstr lstr_getparm
integer li_getid
string ls_getname
string ls_geteaddress
string ls_getold
lstr_getparm = message.powerobjectparm
li_getid = lstr_parmtotrans.Prisoner_ID
ls_getname = lstr_parmtotrans.Prisoner_Name
ls_geteaddress = lstr_parmtotrans.Prisoner_address
ls_getold = lstr_parmtotrans.Prisoner_old
```

另外, 用 `Message` 传递参数必须注意以下几点:

● 最好执行完窗口操作 (打开或关闭) 后马上将 `Message` 消息对象的值传给变量, 这是因为 `Message` 是全局使用的对象, 每一次事件被触发或传送时都会被使用, 其他 `Script` 也有可能用到 `Message` 消息对象, 因此它有被覆盖的危险。

● 避免传送 NULL 对象，以免程序出错，极容易出现在 `Openwithparm()` 函数要打开的结构中，有变量没有被赋值，传入 Null，致使窗口无法打开。

(2) 事务 (Transaction) 对象

事务对象定义 PowerBuilder 应用程序用以连接到数据库的连接参数，PowerBuilder 为每一个应用程序都定义了一个全局的事务对象实例变量 `SQLCA`，大家可以在应用程序中直接使用该实例变量，也可以声明和创建自己所需要的事务变量，还可以使用用户对象画笔通过继承 `Transaction` 对象定义自己的标准类用户对象，然后声明和创建该标准类用户对象的实例，从而定制出自定义版本的事务对象。笔者认为 PowerBuilder 中事务对象的最大功效就是保证数据库操作的完整性，正确的管理事务可以保证数据的完整性，所做的工作全部完成和得到确认之前，没有任何数据物理地写进数据库。

提示：根据笔者编程经验所得，如果没有特别需要，用户最好使用 PowerBuilder 提供的 `SQLCA` 变量，使用 `SQLCA` 作为事务对象变量，那么在编程过程中即使偶尔忘记连接事务对象，PowerBuilder 也会按照默认的 `SQLCA` 去运行。

对事务的操作是这样进行的：先定义开始一个事务，然后对数据做修改、添加、删除等操作，这时如果提交 (`COMMIT`)，这些操作的结果就永久地保存下来了，如果回滚 (`ROLLBACK`)，数据库管理系统将放弃所做的所有操作而回到开始事务时的状态。此外，有些数据库支持事务的“存储点” (`SAVEPOINT`) 这一概念：即在一个事务进程中任意一点都可以进行当前状态的存储，回滚时只是回到所设定的存储点，而无需退回全部的事务。如果该事务可以分成几组对数据库的操作，那就可以设置多个存储点，根据需要回退到任意一个存储点，而避免所有事务的操作数据全部丢失。

常用的事务管理的语句有 4 个：`COMMIT`、`ROLLBACK`、`CONNECT`、`DISCONNECT`。

当应用程序需要与数据库建立连接时，使用 `CONNECT` 操作命令，取消连接时执行 `DISCONNECT`，这两个命令一般分别用在应用的开始和结束，也就是 `Application` 的 `Open` 和 `Close` 事件中。

当一个事务的数据库操作都成功地完成后，将操作提交给数据库，`COMMIT` 语句是一个旧事务结束和一个新事务开始的界线。在操作被提交前，数据库的数据并没有被真正地修改，这些修改被保留在某个工作区，只有作修改的用户才能看到这些被修改后的值，提交之后，则所有的用户就都可以看到新值了。

在事务的进程中发生某些错误，或者在操作中出于种种原因打算中止事务，需用 `ROLLBACK` 命令回滚事务，如果已作的操作不用 `ROLLBACK` 命令取消，这些操作必将错误地作为下一个事务的一部分而导致数据库的混乱。

事务对象有两个属性比较重要：`AutoCommit` 和 `SQLCode`。`AutoCommit` 的属性可以使开发者简化对事务管理的操作，这一布尔型的属性可以用 `TRUE` 或 `FALSE` 来对其赋值。其值为真时，PowerBuilder 不通过其他额外的交互就将 SQL 语句传输给后台数据库，而且执行完毕自动提交。当 `AutoCommit` 属性为假时，需使用 `COMMIT` 或 `ROLLBACK` 提交或回退事务。在大多数应用中，一部分的数据库操作是要成组提交的，而另一些则不用。因此我们可以利用 `AutoCommit` 的特性来确定事务的起点，当我们把 `AutoCommit` 的属性设为 `False` 时，系统设定此时为事务的起点。当 `AutoCommit` 设为真时，系统自动取消这一事务。因此可以先把 `AutoCommit` 设为真，当需要开始一个事务时，将其置为 `false`，此刻即为事务起始点。

提示：`AutoCommit` 属性的默认值为 `False`，笔者认为如果没有成组提交操作的要求，提倡大家在应用开始时，将事务对象的 `AutoCommit` 属性设为 `True`，以保证操作的自动提交。

如果大家对 SQL 语言比较熟悉或有过编写数据库程序经历的话，会知道 `SQLCode` 在程序语句