

PowerBuilder 9.0 实用教程

樊金生 张翠肖

主编

沙 金 贾玉锋

科学出版社
营销宣传

科学出版社

北 京

内 容 简 介

本书系统描述了 PowerBuilder 9.0 版的组成以及它们的语言、事件和函数,对构成 PowerBuilder 的各种对象(应用、窗口、控件、菜单、数据窗口、用户对象)以及它们的创建和使用方法等内容做了详细讲解,演示了如何用 InstallShield 制作安装盘,并介绍了数据管道、动态数据窗口、拖放技术、API 技术、分布式应用和 Web 应用等高级技术,最后一章针对 PowerBuilder 的主要技术给出了一个完整的实例。每章后面附有习题,便于学生练习。

本书可作为大专院校学生的教材和教学参考书,也可供各种培训班使用和应用开发人员参考。

图书在版编目(CIP)数据

PowerBuilder 9.0 实用教程/樊金生等主编.—北京:科学出版社,2004.6

ISBN 7-03-013418-4

I. P… II. 樊… III. 数据库系统—软件工具, PowerBuilder 9.0—高等学校—教材 IV. TP311.56

中国版本图书馆 CIP 数据核字(2004)第 045473 号

责任编辑:赵卫江/责任校对:都 岚

责任印制:吕春珉/封面设计:王 浩

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

双青印刷厂印刷

科学出版社总发行 各地新华书店经销

*

2004 年 6 月第 一 版 开本:787×1092 1/16

2006 年 7 月第三次印刷 印张:22 1/2

印数:8 001—11 000

字数:513 000

定价:30.00 元

(如有印装质量问题,我社负责调换<环伟>)

前 言

1999 年我们编写了《PowerBuilder 6.5 实用教程》，该书荣获铁道部第四届优秀教材二等奖。2003 年又出版了《PowerBuilder 8.0 实用教程》，该书已多次重印，深受读者好评。应广大读者的要求，我们又编写了《PowerBuilder 9.0 实用教程》，本书吸取了前两版的精髓，完善了不足。

在当今信息时代，数据处理技术已深入到我们工作和生活的方方面面，而数据库技术是数据处理不可缺少的手段。随着数据库技术的发展，面向对象的数据库、分布式数据库、多媒体数据库、数据库开发工具的研究方兴未艾。PowerBuilder 是一种快速构建商业应用程序的面向对象的开发工具，IDC 将之称为一种“黄金标准”，即一个用于衡量所有应用开发工具的基准。它集面向对象数据库技术、分布式应用技术、多媒体技术于一身，是目前最有代表性的数据库开发工具，它占有全球客户 / 服务器前端开发工具市场 40% 的份额，位居第一，几乎已发展成为客户 / 服务器前端开发工具的标准。由 Sybase 公司的独资子公司 PowerSoft 公司 2003 年推出的 PowerBuilder 9.0 堪称迄今最成功的数据库应用系统的辅助开发工具，它新增了对 J2EE 和 Microsoft.Net 开发环境的支持功能，其面向对象的开发方法，Internet/Intranet 和分布式应用能力，与当前流行的大型数据库和微机数据库的接口等诸多优点，使它一经推出就受到广大应用开发人员的青睐。作为一个面向对象的、手段丰富的开发环境，它能极大地提高程序员的工作效率，加快应用系统的开发速度。随着 PowerBuilder 的日益普及，希望了解和掌握它的人越来越多，在大学的很多专业也已开设有关课程介绍 PowerBuilder。本书正是应这种要求而写的，它按教材的体系、教学的特点来介绍 PowerBuilder，力求通俗易懂、由浅入深、循序渐进，并在每章后面附有习题，便于学生练习和掌握所学内容。本书除详细介绍 PowerBuilder 的基本内容，如应用、窗口、菜单、数据窗口、数据库、事件、函数、用户对象外，还介绍了数据管道、动态数据窗口、拖放技术、API 技术、分布式应用和 Web 应用等高级技术。通过本书的学习，不仅能达到入门的目的，还能掌握应用系统的实际开发能力，自己开发出实用、高效的应用系统。本书可作为大专院校学生的教材和教学参考书，也可供各种培训班使用和应用开发人员参考。

本书由樊金生、张翠肖、沙金、贾玉锋主编，邸书灵、朴春慧、刘展威、毛晚堆、张国兵、胡迎新、高志伟也参加了编写工作。其中，第一、二、三章由沙金和樊金生编写，第四、五、六章由张翠肖和张国兵编写，第七、八章由邸书灵和贾玉锋编写，第十章由沙金和朴春慧编写，第九章由樊金生和胡迎新编写，第十一章由沙金和贾玉锋编写，第十二章由贾玉锋、邸书灵、朴春慧和高志伟编写，第十三章由张翠肖和刘展威编写，第十四章由沙金和毛晚堆编写，全书由樊金生统稿。在本书编写过程中，参考了许多同行的著作，在此一并表示感谢。

由于作者水平所限，错误和不当之处在所难免，敬请读者批评指正。

作 者

2004 年 4 月 1 日

目 录

| | |
|--|----|
| 第一章 PowerBuilder 9.0 概述 | 1 |
| § 1.1 PowerBuilder 9.0 简介 | 1 |
| § 1.1.1 PowerBuilder 的优异功能和特点 | 1 |
| § 1.1.2 PowerBuilder 9.0 的新特性 | 2 |
| § 1.2 PowerBuilder 9.0 的开发环境 | 4 |
| § 1.2.1 PowerBuilder 工具栏的组成 | 5 |
| § 1.2.2 PowerBuilder 的画板 | 6 |
| § 1.2.3 PowerBuilder 菜单栏的组成 | 11 |
| § 1.3 PowerBuilder 9.0 应用系统的构成 | 12 |
| § 1.3.1 工作空间和目标 | 12 |
| § 1.3.2 窗口、控件、事件和程序 | 13 |
| § 1.3.3 函数 | 13 |
| § 1.4 本书的应用实例 | 13 |
| 习题 | 17 |
| 第二章 数据库 | 18 |
| § 2.1 数据库的建立 | 19 |
| § 2.1.1 创建本地的 Adaptive Server Anywhere 数据库 | 19 |
| § 2.1.2 创建表 | 20 |
| § 2.2 PowerBuilder 与各种数据库的连接 | 21 |
| § 2.2.1 通过 ODBC 接口连接 | 21 |
| § 2.2.2 通过专用接口连接 | 23 |
| § 2.2.3 改变连接的数据库 | 24 |
| § 2.3 操作数据库表 | 24 |
| § 2.3.1 打开、关闭和删除表 | 24 |
| § 2.3.2 修改表和列的属性 | 25 |
| § 2.3.3 修改表的定义 | 27 |
| § 2.3.4 建立索引 | 28 |
| § 2.3.5 建立主键和外部键 | 28 |
| § 2.3.6 数据库操作日志 | 30 |
| § 2.4 视图操作 | 30 |
| § 2.4.1 创建和删除视图 | 31 |
| § 2.4.2 查看视图的 SQL 语法 | 32 |
| § 2.4.3 表或视图语法的输出 | 32 |

科学出版社
营销宣传

| | |
|----------------------|----|
| § 2.5 数据操作 | 32 |
| § 2.5.1 数据检索 | 33 |
| § 2.5.2 数据修改 | 33 |
| § 2.5.3 数据排序和过滤 | 34 |
| § 2.5.4 大量数据的录入 | 35 |
| § 2.5.5 数据打印 | 35 |
| § 2.6 数据库的管理 | 35 |
| § 2.6.1 数据库存取控制 | 36 |
| § 2.6.2 执行 SQL 语句 | 37 |
| § 2.6.3 数据库的备份与恢复 | 38 |
| 习题 | 39 |
| 第三章 应用对象 | 40 |
| § 3.1 概述 | 40 |
| § 3.2 创建新的应用对象 | 41 |
| § 3.3 指定应用对象的属性 | 43 |
| § 3.4 编写应用级的程序 | 47 |
| § 3.4.1 编程环境 | 47 |
| § 3.4.2 事件处理程序 | 50 |
| 习题 | 51 |
| 第四章 窗口及窗口上的控件 | 52 |
| § 4.1 窗口的组成及类型 | 52 |
| § 4.2 创建窗口 | 53 |
| § 4.2.1 窗口工作区 | 54 |
| § 4.2.2 定义窗口的属性 | 55 |
| § 4.2.3 在窗口上放置控件 | 58 |
| § 4.2.4 窗口的保存 | 58 |
| § 4.2.5 查看定义好的窗口 | 59 |
| § 4.2.6 窗口对象的编程 | 60 |
| § 4.3 窗口上放置的控件 | 60 |
| § 4.3.1 在窗口上放置控件 | 61 |
| § 4.3.2 设置控件的属性 | 61 |
| § 4.3.3 窗口上控件的调整 | 65 |
| § 4.3.4 控件的编程 | 66 |
| § 4.4 控件的使用 | 66 |
| § 4.4.1 激活动作类型（按钮类型） | 66 |
| § 4.4.2 列出选择项类型 | 68 |
| § 4.4.3 文本显示和编辑控件 | 69 |
| § 4.4.4 列表选择 | 73 |
| § 4.4.5 图片控件 | 76 |

科学出版社
营销宣传

| | | |
|----------------------------------|---------------------|-----|
| § 4.4.6 | 超链接控件 | 76 |
| § 4.4.7 | 标签控件 | 76 |
| § 4.4.8 | 数据窗口控件 | 78 |
| § 4.4.9 | 树状浏览控件和列表浏览控件 | 79 |
| § 4.4.10 | 滚动条控件 | 82 |
| § 4.4.11 | 进度条与轨迹条控件 | 82 |
| § 4.4.12 | OLE 2.0 控件 | 84 |
| § 4.4.13 | 用户对象控件 | 84 |
| § 4.4.14 | 绘图控件 | 85 |
| § 4.5 | 窗口的继承 | 85 |
| 习题 | | 86 |
| 第五章 菜单 | | 89 |
| § 5.1 菜单的创建和设计 | | 89 |
| § 5.1.1 创建新菜单 | | 89 |
| § 5.1.2 添加菜单项 | | 91 |
| § 5.1.3 菜单的修改 | | 91 |
| § 5.1.4 菜单的一般属性和显示风格 | | 92 |
| § 5.1.5 设置加速键和快捷键 | | 93 |
| § 5.1.6 定义菜单工具栏 | | 93 |
| § 5.1.7 保存菜单 | | 94 |
| § 5.2 预览和打印菜单 | | 94 |
| § 5.3 通过继承创建菜单 | | 95 |
| § 5.4 菜单的编程 | | 95 |
| § 5.5 菜单的调用 | | 96 |
| 习题 | | 97 |
| 第六章 事件 | | 98 |
| § 6.1 应用对象的事件 | | 98 |
| § 6.2 窗口对象的事件 | | 98 |
| § 6.3 窗口上控件的事件 | | 103 |
| § 6.3.1 窗口上控件的公共事件 | | 103 |
| § 6.3.2 命令按钮、图片按钮、单选钮、复选框控件的事件 | | 106 |
| § 6.3.3 数据窗口控件的事件 | | 106 |
| § 6.3.4 图片控件的事件 | | 111 |
| § 6.3.5 成组框控件的事件 | | 111 |
| § 6.3.6 单行编辑器控件、编辑掩码控件、多行编辑控件的事件 | | 111 |
| § 6.3.7 超文本编辑控件的事件 | | 111 |
| § 6.3.8 滚动条控件的事件 | | 112 |
| § 6.3.9 列表框控件的事件 | | 113 |
| § 6.3.10 树状浏览控件的事件 | | 114 |

| | | |
|----------|-------------------------------|-----|
| § 6.3.11 | 列表浏览控件的事件 | 115 |
| § 6.3.12 | Tab 页控件的事件 | 116 |
| § 6.3.13 | OLE 控件的事件 | 116 |
| § 6.3.14 | 用户对象控件的事件 | 116 |
| § 6.3.15 | 静态文本控件和统计图控件的事件 | 116 |
| § 6.4 | 菜单事件 | 117 |
| § 6.5 | PowerBuilder 对象的用户自定义事件 | 117 |
| § 6.6 | 在程序中触发事件 | 119 |
| 习题 | | 119 |
| 第七章 | 数据窗口 | 120 |
| § 7.1 | 概述 | 120 |
| § 7.2 | 创建数据窗口 | 120 |
| § 7.3 | 使用数据窗口的步骤 | 124 |
| § 7.4 | 数据源 | 124 |
| § 7.4.1 | Quick Select 数据源 | 125 |
| § 7.4.2 | SQL Select 数据源 | 127 |
| § 7.4.3 | Query 数据源 | 134 |
| § 7.4.4 | External 数据源 | 135 |
| § 7.4.5 | Stored Procedure 数据源 | 136 |
| § 7.5 | 显示风格 | 138 |
| § 7.5.1 | Tabular 风格 | 138 |
| § 7.5.2 | Freeform 风格 | 139 |
| § 7.5.3 | Grid 风格 | 139 |
| § 7.5.4 | Label 风格 | 140 |
| § 7.5.5 | N_UP 风格 | 141 |
| § 7.5.6 | Group 风格 | 142 |
| § 7.5.7 | Composite 风格 | 143 |
| § 7.5.8 | Graph 风格 | 144 |
| § 7.5.9 | CrossTab 风格 | 145 |
| § 7.5.10 | RichText 风格 | 147 |
| § 7.5.11 | OLE 2.0 风格 | 148 |
| § 7.6 | 数据窗口画板 | 148 |
| § 7.6.1 | Design (设计) 视图 | 149 |
| § 7.6.2 | Preview (预览) 视图 | 150 |
| § 7.7 | 设置数据窗口对象的总体属性 | 151 |
| § 7.7.1 | 设置数据窗口对象的式样 | 152 |
| § 7.7.2 | 设置鼠标风格 | 152 |
| § 7.7.3 | 设置打印参数 | 152 |
| § 7.7.4 | 设置 HTML Table 属性 | 153 |

| | |
|--|-----|
| § 7.7.5 设置 HTML Generation 属性 | 153 |
| § 7.8 设置数据窗口中对象的属性 | 153 |
| § 7.9 列的编辑风格 | 154 |
| § 7.9.1 Edit (编辑框) 编辑风格 | 155 |
| § 7.9.2 DropDownListBox (下拉列表框) 编辑风格 | 156 |
| § 7.9.3 CheckBox (复选框) 编辑风格 | 156 |
| § 7.9.4 RadioButton (单选钮) 编辑风格 | 157 |
| § 7.9.5 EditMask (编辑掩码) 编辑风格 | 158 |
| § 7.9.6 DropDownWindow (下拉数据窗口) 编辑风格 | 160 |
| § 7.10 列的显示格式 | 161 |
| § 7.10.1 使用显示格式 | 161 |
| § 7.10.2 定义显示格式 | 162 |
| § 7.11 数据校验 | 165 |
| § 7.11.1 校验规则定义 | 166 |
| § 7.11.2 校验规则使用 | 167 |
| § 7.12 在数据窗口中添加对象 | 168 |
| § 7.12.1 添加静态文本 | 168 |
| § 7.12.2 添加静态图形、直线、圆形框、矩形框、圆角矩形框 | 168 |
| § 7.12.3 添加计算域 | 168 |
| § 7.12.4 添加统计图 | 169 |
| § 7.12.5 添加命令按钮对象 | 170 |
| § 7.12.6 添加成组框对象 | 171 |
| § 7.12.7 添加列 | 172 |
| § 7.13 增强数据窗口的功能 | 173 |
| § 7.13.1 数据排序 | 173 |
| § 7.13.2 数据过滤 | 174 |
| § 7.13.3 分组显示 | 174 |
| § 7.13.4 去掉重复值 | 175 |
| § 7.13.5 数据灌入 | 176 |
| § 7.13.6 数据检索 | 177 |
| § 7.13.7 数据窗口更新控制 | 177 |
| § 7.14 数据窗口的打印 | 180 |
| § 7.14.1 数据窗口打印设置 | 180 |
| § 7.14.2 数据窗口打印预览 | 180 |
| § 7.14.3 打印数据 | 181 |
| § 7.15 统计图的使用 | 181 |
| § 7.15.1 统计图的组成部分 | 182 |
| § 7.15.2 统计图的种类 | 182 |
| § 7.15.3 统计图属性定义 | 185 |

| | |
|--------------------------------------|------------|
| § 7.16 导入和导出 XML 数据 | 187 |
| § 7.16.1 导出 XML 格式数据 | 187 |
| § 7.16.2 导入 XML 格式数据 | 188 |
| 习题 | 188 |
| 第八章 PowerScript 编程语言 | 189 |
| § 8.1 语言初步 | 189 |
| § 8.2 数据类型 | 193 |
| § 8.3 变量 | 194 |
| § 8.4 常量 | 197 |
| § 8.5 数组 | 197 |
| § 8.6 运算符和表达式 | 199 |
| § 8.7 PowerScript 语句 | 201 |
| § 8.7.1 赋值语句 | 201 |
| § 8.7.2 IF...THEN 语句 | 201 |
| § 8.7.3 CHOOSE CASE 语句 | 202 |
| § 8.7.4 DO...LOOP 语句 | 203 |
| § 8.7.5 FOR...NEXT 语句 | 204 |
| § 8.7.6 CONTINUE 语句和 EXIT 语句 | 204 |
| § 8.7.7 GOTO 语句 | 204 |
| § 8.7.8 HALT 语句 | 205 |
| § 8.7.9 RETURN 语句 | 205 |
| § 8.7.10 CREATE 语句和 DESTROY 语句 | 205 |
| § 8.7.11 CALL 语句 | 205 |
| § 8.8 嵌入 SQL 语句 | 206 |
| § 8.8.1 连接语句 | 206 |
| § 8.8.2 事务处理语句 | 206 |
| § 8.8.3 数据库操作语句 | 207 |
| § 8.8.4 游标语句 | 208 |
| § 8.8.5 Blob 列操作语句 | 209 |
| § 8.8.6 存储过程语句 | 210 |
| § 8.8.7 SQICode | 210 |
| § 8.8.8 编程举例 | 210 |
| § 8.9 结构 | 212 |
| § 8.9.1 定义结构 | 212 |
| § 8.9.2 使用结构 | 214 |
| § 8.9.3 浏览与修改结构 | 214 |
| § 8.10 用户函数 | 216 |
| § 8.10.1 用户函数的定义 | 216 |
| § 8.10.2 用户函数的编译和保存 | 218 |

科学出版社
营销宣传

| | |
|-------------------------------------|-----|
| § 8.10.3 用户函数的调用 | 218 |
| 习题 | 218 |
| 第九章 函数 | 219 |
| § 9.1 数值计算函数 | 219 |
| § 9.2 日期时间函数 | 219 |
| § 9.3 字符串操作函数 | 220 |
| § 9.4 Blob (大二进制对象) 函数 | 221 |
| § 9.5 数据类型检查与转换函数 | 221 |
| § 9.6 数组函数 | 222 |
| § 9.7 文件操作函数 | 222 |
| § 9.8 打印函数 | 223 |
| § 9.9 定时操作函数 | 224 |
| § 9.10 打开与关闭窗口函数 | 224 |
| § 9.11 库管理函数 | 224 |
| § 9.12 邮件函数 | 225 |
| § 9.13 注册函数 | 225 |
| § 9.14 系统与环境函数 | 225 |
| § 9.15 其他系统函数 | 226 |
| § 9.16 对象函数 | 227 |
| § 9.16.1 公共对象函数 | 227 |
| § 9.16.2 应用对象的函数 | 228 |
| § 9.16.3 窗口对象的函数 | 229 |
| § 9.16.4 可编辑控件共有的函数 | 230 |
| § 9.16.5 单行编辑框和多行编辑框的函数 | 230 |
| § 9.16.6 编辑掩码控件的函数 | 230 |
| § 9.16.7 列表框的函数 | 231 |
| § 9.16.8 下拉列表框的函数 | 231 |
| § 9.16.9 下拉图片列表框的函数 | 231 |
| § 9.16.10 列表视图的函数 | 232 |
| § 9.16.11 树形视图的函数 | 233 |
| § 9.16.12 标签控件的函数 | 234 |
| § 9.16.13 图片框控件的函数 | 235 |
| § 9.16.14 数据窗口控件的函数 | 235 |
| § 9.16.15 数据窗口画板函数 | 240 |
| § 9.17 分布式应用函数 | 241 |
| § 9.18 Internet/Intranet 服务函数 | 242 |
| 第十章 应用的调试和发行 | 243 |
| § 10.1 应用程序的调试 | 243 |
| § 10.1.1 打开调试窗口 | 243 |

科学出版社
营销宣传

| | |
|-------------------------------------|------------|
| § 10.1.2 设置断点 | 245 |
| § 10.1.3 在调试状态下运行应用程序 | 246 |
| § 10.2 应用程序的运行 | 248 |
| § 10.3 生成可执行文件 | 249 |
| § 10.3.1 创建工程及编译 | 249 |
| § 10.3.2 应用系统 | 250 |
| § 10.4 应用程序的发行 | 250 |
| § 10.4.1 PowerBuilder 运行时库 | 251 |
| § 10.4.2 安装数据库接口 | 251 |
| § 10.4.3 配置 ODBC 数据源 | 251 |
| § 10.5 制作安装盘 | 252 |
| 习题 | 258 |
| 第十一章 用户对象 | 259 |
| § 11.1 用户对象简介 | 259 |
| § 11.2 用户对象的创建 | 261 |
| § 11.2.1 创建标准可视用户对象 | 261 |
| § 11.2.2 创建定制可视用户对象 | 263 |
| § 11.2.3 创建外部可视用户对象 | 265 |
| § 11.2.4 创建标准类用户对象 | 265 |
| § 11.2.5 创建定制类用户对象 | 266 |
| § 11.3 使用用户对象 | 267 |
| § 11.3.1 使用可视用户对象 | 267 |
| § 11.3.2 使用类用户对象 | 267 |
| § 11.4 窗口与用户对象间的通讯 | 268 |
| § 11.5 用户对象使用综合示例 | 268 |
| 习题 | 274 |
| 第十二章 PowerBuilder 高级应用 | 275 |
| § 12.1 动态数据窗口 | 275 |
| § 12.1.1 数据窗口属性 | 275 |
| § 12.1.2 使用 Modify 修改数据窗口对象 | 276 |
| § 12.1.3 动态创建数据窗口 | 278 |
| § 12.2 应用程序接口 (API) | 279 |
| § 12.2.1 外部函数定义和调用 | 279 |
| § 12.2.2 数据类型转换 | 280 |
| § 12.2.3 参数传递 | 280 |
| § 12.2.4 外部函数举例 | 281 |
| § 12.3 嵌套报表 | 282 |
| § 12.4 交叉表 | 283 |
| § 12.5 OLE 2.0 | 285 |

| | |
|--|-----|
| § 12.5.1 创建 OLE 对象 | 285 |
| § 12.5.2 操作 OLE 对象 | 286 |
| § 12.6 PowerBuilder 分布式应用 | 287 |
| § 12.6.1 Client/Server 结构与分布式应用 | 287 |
| § 12.6.2 PowerBuilder 分布式应用结构 | 288 |
| § 12.6.3 Client 与应用服务器之间通讯 | 289 |
| § 12.6.4 PowerBuilder 分布式应用实例 | 290 |
| § 12.6.5 PowerBuilder 分布式应用其他功能 | 291 |
| § 12.7 数据管道 | 291 |
| § 12.7.1 创建数据管道进行数据转换 | 292 |
| § 12.7.2 在应用程序中调用定义好的数据管道 | 293 |
| § 12.8 拖放技术 | 295 |
| § 12.8.1 拖放属性 | 296 |
| § 12.8.2 控件的拖放函数 | 297 |
| § 12.8.3 拖放事件 | 297 |
| § 12.8.4 拖放操作举例 | 297 |
| 第十三章 创建 Web 应用 | 303 |
| § 13.1 Web 应用的基本概念 | 303 |
| § 13.1.1 Web 和 Internet 之间的关系 | 303 |
| § 13.1.2 企业级 Internet 解决方案 | 303 |
| § 13.2 PowerBuilder 中主要的 Web 技术 | 304 |
| § 13.3 Web 数据窗口 | 304 |
| § 13.4 EAServer 环境下的 Web 数据窗口的实现 | 305 |
| § 13.5 IIS+MTS 环境下的 Web 数据窗口的实现 | 314 |
| 习题 | 318 |
| 第十四章 应用实例 | 319 |
| § 14.1 实例概述 | 319 |
| § 14.2 创建一个应用 | 319 |
| § 14.3 数据库设计 | 320 |
| § 14.4 创建表 | 320 |
| § 14.4.1 创建数据库 | 320 |
| § 14.4.2 创建表 | 320 |
| § 14.4.3 创建索引 | 321 |
| § 14.5 创建窗口 | 321 |
| § 14.5.1 创建主菜单窗口 | 322 |
| § 14.5.2 创建用户登录窗口 | 322 |
| § 14.5.3 创建录入和编辑窗口 | 322 |
| § 14.5.4 创建维护窗口 | 329 |
| § 14.5.5 创建查询窗口 | 333 |

| | |
|-----------------------------|-----|
| § 14.5.6 创建统计窗口 | 335 |
| § 14.5.7 创建设置窗口 | 335 |
| § 14.6 创建菜单 | 337 |
| § 14.7 将主菜单与主窗口连接 | 338 |
| § 14.8 创建数据窗口 | 339 |
| § 14.8.1 创建下拉数据窗口 | 339 |
| § 14.8.2 创建录入数据窗口 | 339 |
| § 14.8.3 创建有检索参数的数据窗口 | 340 |
| § 14.8.4 创建查询数据窗口 | 341 |
| § 14.8.5 创建表格统计数据窗口 | 342 |
| § 14.8.6 创建图形统计数据窗口 | 343 |
| § 14.8.7 创建编码设置数据窗口 | 345 |
| 主要参考文献 | 346 |

科学出版社
营销宣传

第一章 PowerBuilder 9.0 概述

§1.1 PowerBuilder 9.0 简介

PowerBuilder 9.0 版是 Sybase 公司 2003 年推出的新一代 4GL Plus 数据库开发工具，它不仅继承了以往版本的各种先进技术，而且适应目前开放式软件的发展潮流，在快速集成开发环境中增加了构建 Web Services、JSP Web、EJB 的功能，满足了在 4GL RAD 环境中创建完整应用和应用组件的需求，并使企业应用可以平滑过渡到基于浏览器和 Web 的应用模式下。同时，PowerBuilder 9.0 卓越的应用开发效率及强大的 Web 功能为新版本增色不少。

§1.1.1 PowerBuilder 的优异功能和特点

(1) 对多种数据库的支持

PowerBuilder 同时支持多种关系型数据库管理系统，如 Sybase、Oracle、Informix、MS SQL Server 等，提供了多种数据库的专用接口和 ODBC 接口。

(2) 数据窗口 (DataWindow) 对象

PowerBuilder 拥有数据窗口这个具有专利技术的智能对象，利用该对象可以操作数据库的数据而无需编写 SQL 语句。它具有丰富的数据窗口数据源和多种样式的数据显示格式，可以查询、修改、插入、删除、浏览、打印数据，打开和存储数据，也可在数据窗口中直接定义功能按钮实现预定义的系统功能，它还支持数据库的事务管理和并发控制等机制。

(3) 面向对象的程序设计方法

PowerBuilder 采用面向对象的开发方法，这可以使系统开发人员无需精通专用语言就可以迅速转向面向对象的开发方式。PowerBuilder 应用由一系列对象组成，包括窗口、菜单、函数、数据窗口和各种控制对象，它支持对象的继承、封装和多态性。

(4) 集成的开发环境

PowerBuilder 是一个集成的开发环境。它提供了应用的各种成分，包括窗口、数据窗口、菜单、事件处理程序、数据库、图形、报表的建立和维护，数据库管理、调试、编译等工作，都能在 PowerBuilder 集成开发环境中完成，从而可以大大加快开发的速度。

(5) 多种平台的开发环境

PowerBuilder 是一个基于多平台的开发工具，它提供了良好的跨平台性。它有 Windows、Macintosh、Unix 等多个版本。如果要把一个平台上开发的代码移植到另外一个平台上，只要重新编译就可以了。

(6) 分布式应用

PowerBuilder 使用户能够在网络上分布 PowerBuilder 对象, 灵活而方便地创建多级分割。从而允许开发人员根据特定的性能、维护或安全方面的需求, 决定将代码的某一部分驻留于应用服务器上或是客户机上。

应用在分布配置后, 客户端应用可以访问应用服务器中定义的方法, 共享应用服务器中的对象, 应用服务器对客户应用消息回送, 构成多层应用结构, 即 N 层结构。

(7) 支持 Internet/Intranet 下的 Web 应用开发

由于 PowerBuilder 对分布式应用体系结构的支持, 可采用该工具建立基于 Web 的应用服务。在 Web 服务器上的 HTML 页面中可以嵌入 PowerBuilder 的客户应用程序, 也就是 PowerBuilder 提供的 Web Framework。当用户用浏览器访问页面, 调用 Web Framework 时, Web Framework 访问用 PowerBuilder 编写的服务器应用, 构成动态发布数据库信息的 Web 应用。

(8) PowerBuilder 开发工具包和 Powersoft 企业系列产品

PowerBuilder 开发工具包是配合 PowerBuilder 基本开发环境所使用的一些工具, 可生成 PowerBuilder 应用对象的交叉引用文档, 查看数据窗口的列规则与数据库表的扩展属性中的列规则是否一致, 指出应用从一种数据库(如 Oracle)移植到另一种数据库(如 Sybase)时需要改动的 SQL 语句, 把 C++ 程序编译成 PowerBuilder 可调用的 DLL 文件, 同步制作 HTML 页面等。

Powersoft 还有一个辅助产品 Informaker, 它是数据库查询和报表制作的工具, 可使最终用户访问本地和整个企业的数据库, 并能以多种报表的形式显示。

Powersoft 还开发了用于数据流和数据库设计的工具 PowerDesigner。该工具可以进行企业的数据流分析, 完成数据库的概念和物理设计, 把数据库的物理设计结果直接送给 PowerBuilder, 自动生成应用的原型系统, 从而使 Powersoft 的产品线拓展到了应用的整个生命周期。

Powersoft 提供的复杂报表制作工具 Visual Component 可以把电子表格贴在窗口上, 与数据窗口相连, 生成复杂表头的中国式报表, 从而真正解决了在应用开发中的报表编制难题。

§1.1.2 PowerBuilder 9.0 的新特性

与 PowerBuilder 8.0 版本相比, PowerBuilder 9.0 扩展并增加了多种功能。它集设计、建模、开发、部署、管理等各项功能为一体, 还包含一个优秀的代码基础、许多新的增强特性以及与 Web 和 N 层应用开发的进一步整合。PowerBuilder 9.0 的新增功能包括了对 .Net、PBDOM、XML DataWindow、RAD JavaServer Pages 与 PBNI 等的支持。同时, 它还提供了与 Sybase 应用服务器 Enterprise Application Server (EAServer) 更为紧密的集成, 能够在 EAServer 中调用 Enterprise JavaBean (EJB), 并能够将现有的组件连接到 EAServer。PowerBuilder 9.0 的出现将会帮助开发者在更为开放、高度集成的开发环境中利用 PowerBuilder 9.0 的强大功能开发出理想的系统。

(1) 开放式快速应用程序开发第四代语言基础

PowerBuilder 9.0 基于原来第四代语言的快速应用程序开发能力, 使其继续构筑新的

基础来创造 Client / Server、Web 和 N-Tier 系统架构的应用程序。PowerBuilder 9.0 使这个过程更加轻松, 更加快速, 更加有效率, 成本也更加低廉。PowerBuilder 9.0 作为一种领先的第四代开发语言, 从强大的针对 XML 的 PowerBuilder 文件对象模型到其他 J2EE 应用程序服务器整合, 给开发人员提供了一流的程序撰写开发环境。

(2) XML DataWindow

PowerBuilder 获有 DataWindow 专利技术, 可使用户在开发项目中充分运用 XML 的优势。DataWindow 画板, 可用图形接口建置 XML 样板, 并作为 DataWindow 对象的一部分。这些样板可在各种自订结构中汇入或汇出 XML 资料。DataWindow 的表示可以存成 XSL-FO, XSL-FO 为 XML 语言, 用以描述分页报告、打印、XML-to-PDF 应用程序。

(3) PowerBuilder 文件对象模型 (PBDOM)

PowerBuilder 的 Document Object Model (PBDOM) 是一套完整的 API, 可使用标准的 PowerScript, 建立、读取、写入及操作 XML 文件。PBDOM 以众多的非可视化类别为基础, 并采用 PowerBuilder Native Interface 技术建构而成。

(4) RAD Java Server Page

PowerBuilder 9.0 提供了一个容易使用的开发环境来建立 JSP Web 应用程序。它可以自动完成许多开发任务, 允许部署与 JSP 1.2 服务器兼容的 Web 应用程序。JSP Web Wizard 自动地建立 JSP Web Target-J2EE Web 应用程序, 甚至还包括了部署参数。JSP Web Target 可以直接从 PowerBuilder 部署到 EAServer 或者是 Tomcat 中, 其中的命令行选项使它也可以配置到其他流行的 JSP 容器 (Container) 中。

(5) 高速开放式整合开发环境

PowerBuilder 9.0 快速应用程序开发 (RAD) 取得了巨大突破。第四代整合开发环境提供了建立不同应用的机制以配置不同的系统架构, 如 Microsoft R.NET 框架和 J2EE。PowerBuilder 9.0 可以开发一流的 Client / Server 应用程序, 也可以开发 Web Services、RAD JavaServer Pages, 同时给其他应用程序服务器提供支持, 这样就使原来的应用程序能平稳移植到分布式架构和 Web Base 架构。

(6) Web Services

PowerBuilder IDE 中的 Web Services Wizard, 可建立代理器对象或定制标签库, 内含可从 PowerBuilder 应用程序或 JSP 页面呼叫 Web Services 的必要信息。PowerBuilder 可以作为客户端在 Intranet 或 Internet 上提供 Web Services。通过使用 Soap 和 Wsdl, PowerBuilder 可以将一系列函数作为一个实体远程发布, 并且成为 PowerBuilder 应用程序的一部分。通过使用 EAServer ToolBox, PowerBuilder 制作的组件可以很容易地部署为 Web Services。PowerBuilder 整合开发环境中的 Web Services Wizard 建立一个 Proxy 或定制标签库, 它们包含了必要的信息, 以便 PowerBuilder 应用程序或 JSP 页面调用 Web Services。

(7) 客户端 EJB 服务器支持

使用 PowerBuilder 可以调用 Java 服务器端商业逻辑, 这样不需要常年累月地使用 Java 建立应用程序, PowerBuilder 就能够快速建立一个合适的应用程序来呼叫运行在与 J2EE 兼容的应用程序服务器上的 EJB, 如 Sybase EAServer、IBM WebSphere、BEA WebLogic 应用程序服务器以及其他应用程序服务器。

（8）PowerBuilder 本地接口（Native Interface）

PowerBuilder 本地接口（PBNI）是 PowerBuilder 一项强大的功能，它允许开发人员将他们自己的函数加入到 PowerBuilder 中。这是一个 API，它通过两种方式使 PowerBuilder 与外界联系。第一种方式，通过建立 DLLs 在 PowerBuilder 中作为 NonVisual Object Class 来提供一个或多个 Class。这允许开发人员撰写扩展函数库，然后在 PowerBuilder 中建立，并像可视化或非可视化对象一样使用它们。第二种方式，PowerBuilder 虚拟机（PowerBuilder Virtual Machine）可以嵌入 C++、Java 应用程序中，或者其他能够调用由 PowerScript 撰写的商业逻辑应用程序中。

（9）源代码控制

为了显著提高功能源代码控制（SCC, Source Code Control）接口的功能，PowerBuilder 9.0 引入了初始化工作空间的执行线程，这样就消除了大型应用程序的启动延迟。同时，修订数的引入提供了巨大的改进，可以决定同步状态，支持新的集成扩展。此外，OrcaScript 功能可以用来直接从 SCC 仓库创建应用 PBL。

（10）OrcaScript

OrcaScript 允许撰写批处理语法在 PowerBuilder 开发环境之外处理 PowerBuilder 应用程序和文件。OrcaScript 能够从源代码控制中获取最新的 PowerScript 对象、建立 PBL 对象、部署 EAServer 组件、编译 PowerBuilder 可执行文件，所有这些都不需要操作人员干涉。

（11）PDF 支持

PowerBuilder 提供了两种方法将数据窗口对象保存为 PDF 格式的报表。默认的是提取的方法，它能够保存数据窗口对象的所有类型。同时，数据窗口的数据和表示也可以保存为文档或字符串，使用 XSL 格式对象（XSL-FO）或 PDF 格式。

科学出版社
营销发行

§1.2 PowerBuilder 9.0 的开发环境

PowerBuilder 是面向对象的开发工具，与传统的开发工具有着质的区别。传统的开发思路是“流控制”，给程序预先定义运行路径，需要开发人员编写大量的程序去控制程序的流程。程序运行时，用户不能偏离这一路径。而 PowerBuilder 的思想方法是事件驱动，在编程过程中首先创建各种对象，然后为对象赋予属性，为对象编写事件驱动程序。

PowerBuilder 9.0 为应用程序的开发提供了一个集成的开发环境，如图 1-1 所示。它包括顶部的菜单栏和工具栏、左边的系统树（System Tree）和文件夹（Clip）窗口、底部的输出窗口。当你开始操作对象时，灰色区域将显示你要打开的编辑器。

这是一个典型的多文档界面（MDI），在这个环境下，用户可以通过菜单、工具栏以及一些功能键等方便快速地创建对象，编写事件处理程序。了解 PowerBuilder 的开发环境是应用开发的第一步，本节将介绍 PowerBuilder 9.0 开发环境的定制（调整适合自己的开发环境）、环境中各部分的功能和简单操作。

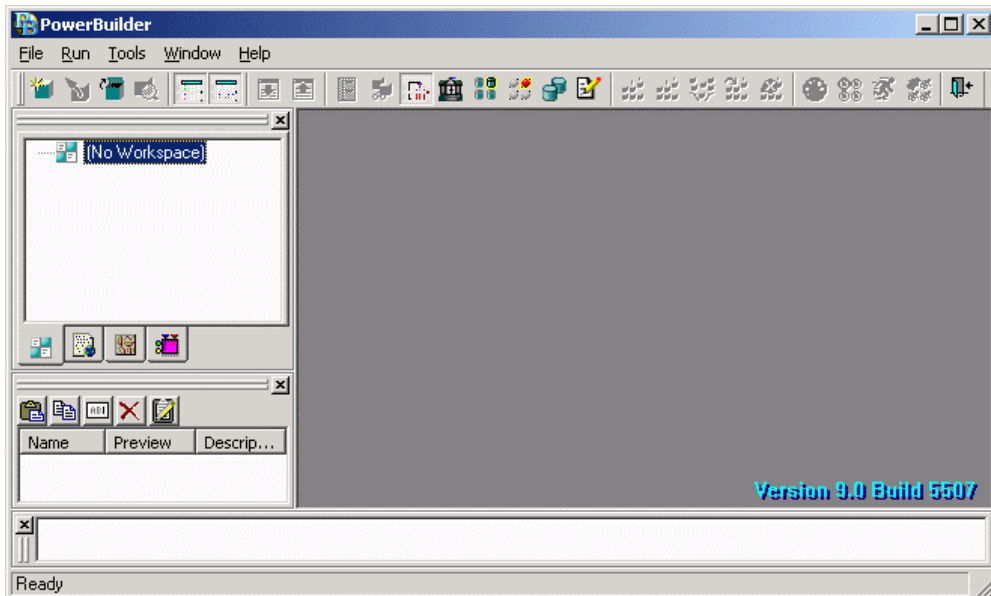


图 1-1 集成开发环境

§1.2.1 PowerBuilder 工具栏的组成

PowerBuilder 的工具栏（Toolbars）由以下三部分组成：

- **PowerBar**: 由 PowerBuilder 的画板组成，用来打开 PowerBuilder 的各种画板。
- **PainterBar**: 由当前画板的常用操作的图标按钮组成，用于执行当前画板的操作。
- **StyleBar**: 用于改变文本的字体、字号等属性。

其中 PainterBar 只有在打开某个画板后才出现，StyleBar 只有在对文本进行操作时才出现。为了说明方便，我们打开一个窗口对象，屏幕如图 1-2 所示。



图 1-2 工具栏

由于打开了窗口画板，所以与窗口画板对应的 PainterBar 出现在当前屏幕上，因为窗口对象上可以放文本，所以出现了 StyleBar。PainterBar 是对当前的画板组件操作的，因此当打开不同的画板时，PainterBar 是不一样的。

在 PowerBar 上放置着 PowerBuilder 的各种画板，这些画板相互配合，构成了 PowerBuilder 强大的集成开发环境。画板由基本画板和扩展画板组成，基本画板是安装 PowerBuilder 后就出现在 PowerBar 中的画板；扩展画板最初并不出现在 PowerBar 中，用户可以根据自己的需要将扩展画板加到 PowerBar 中，以方便使用。将扩展画板加到 PowerBar 中的方法是：

- ① 把鼠标移到 PowerBar 上，单击鼠标右键，弹出一个弹出式菜单，如图 1-3 所示。

② 选择 Customize..., 打开 Customize 对话框, 如图 1-4 所示。

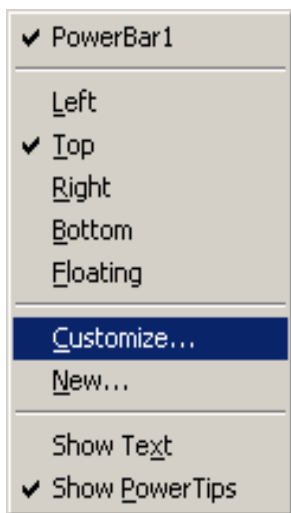


图 1-3 设置工具栏菜单

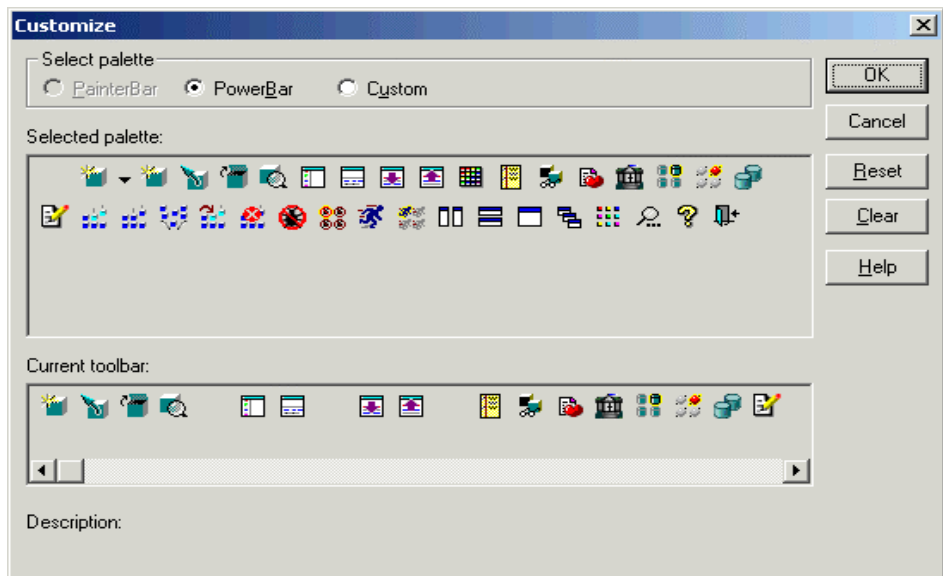












图 1-4 定制工具栏

该对话框包括上下两个部分：上部分为 PowerBuilder 所能提供的所有画板（含基本画板和扩展画板），下部分为目前已经在 PowerBar 中的画板。可以通过拖拽的方式把扩展画板追加到 PowerBar 中。

图 1-3 中其他菜单项的功能是：Left、Top、Right、Bottom、Floating 用于设置 PowerBar 工具栏在 PowerBuilder 工作区的位置，选择 Left 放在工作区的左侧，选择 Top 放在顶端、选择 Right 放在右侧、选择 Bottom 放在底端，选择 Floating 则悬浮在工作区中。选择 Show Text 时，工具栏的图标按钮上显示此按钮的英文说明；选择 Show PowerTips 时，当鼠标移到某一个按钮上时，鼠标指针后面将出现该按钮的英文说明。

§1.2.2 PowerBuilder 的画板

PowerBuilder 的基本画板如下：

- | | |
|---|--------------------------------|
|  | 新建对象画板 (New) |
|  | 继承对象画板 (Inherit) |
|  | 打开对象画板 (Open) |
|  | 运行/预览画板 (Run/Privew Object) |
|  | 系统树画板 (System Tree) |
|  | 输出画板 (Output) |
|  | 下一个画板 (Next Error/Message) |
|  | 上一个画板 (Previous Error/Message) |
|  | 处理列表画板 (To-Do List) |
|  | 对象浏览器画板 (Browser) |



下面简述 PowerBuilder 基本画板的功能。

新建画板 (New Painter): 创建各种对象，单击它后弹出如图 1-5 所示的对话框。

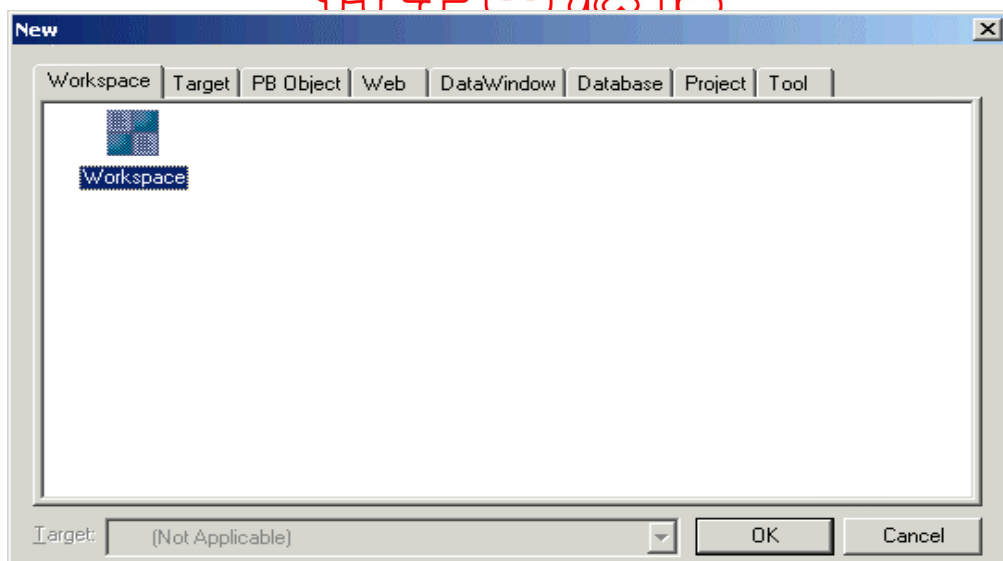


图 1-5 新建对象

该对话框是由多个标签页组成的，包括了 PowerBuilder 的所有对象（分别见图 1-6 至图 1-12），关于每个对象的具体创建过程在以后的章节中会逐一介绍。如果没有打开 Workspace（工作空间），那么除了 Workspace 标签页和 Tool 标签页的内容可用外，其他的都是不可用的。

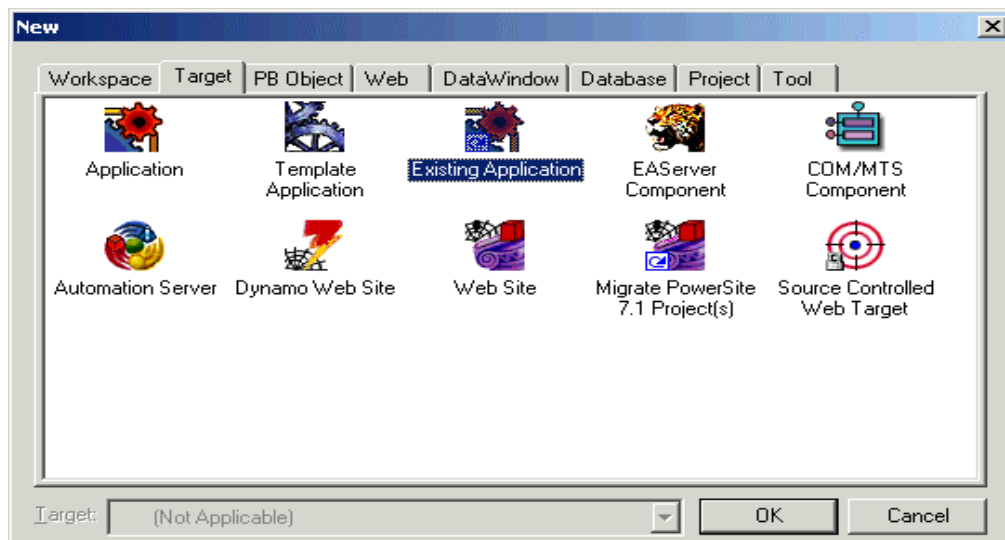


图 1-6 新建对象

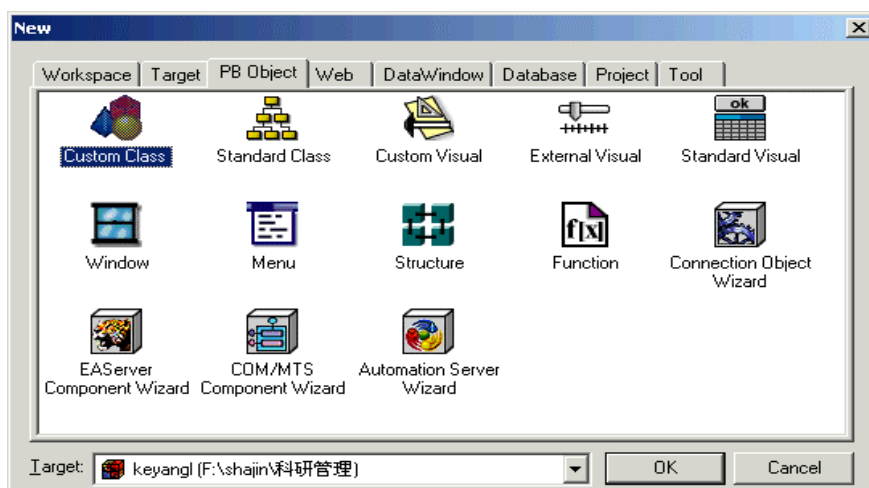


图 1-7 新建对象

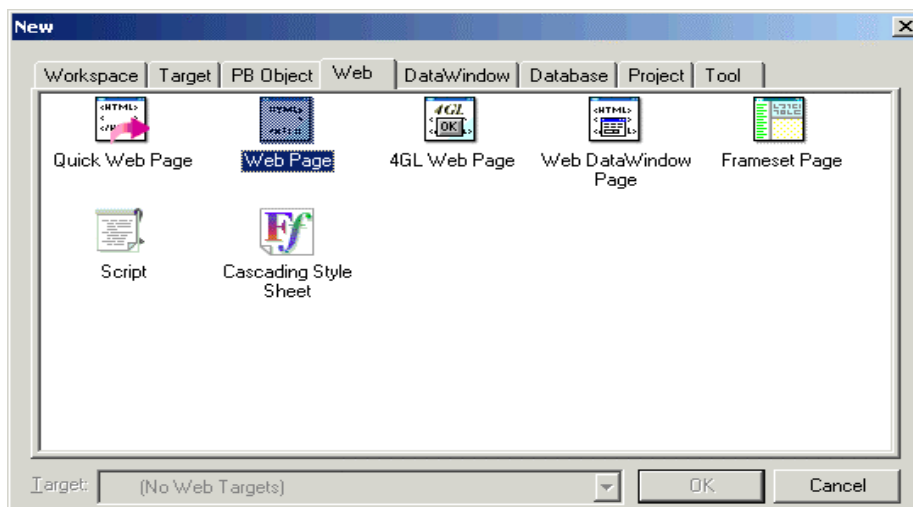


图 1-8 新建对象

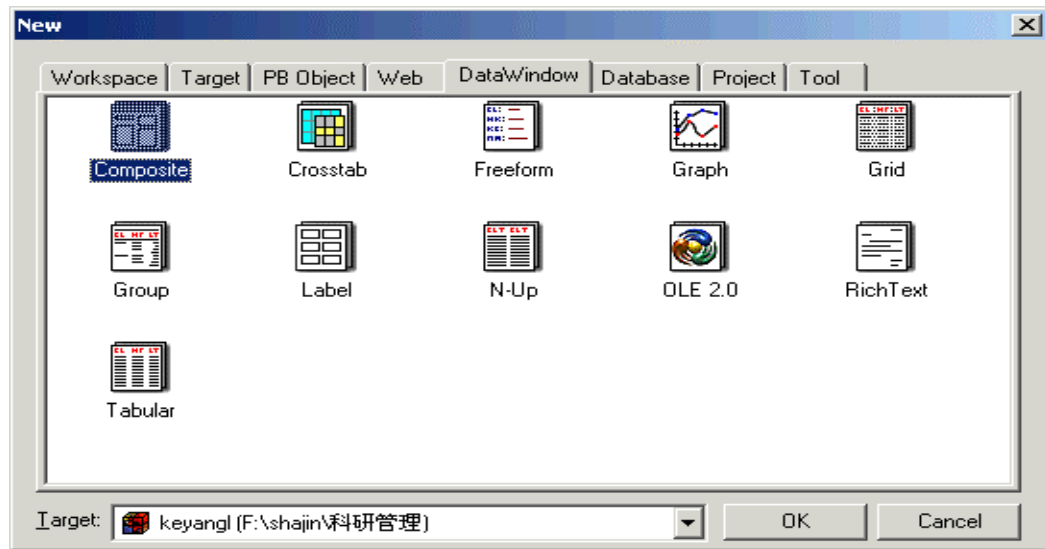


图 1-9 新建对象

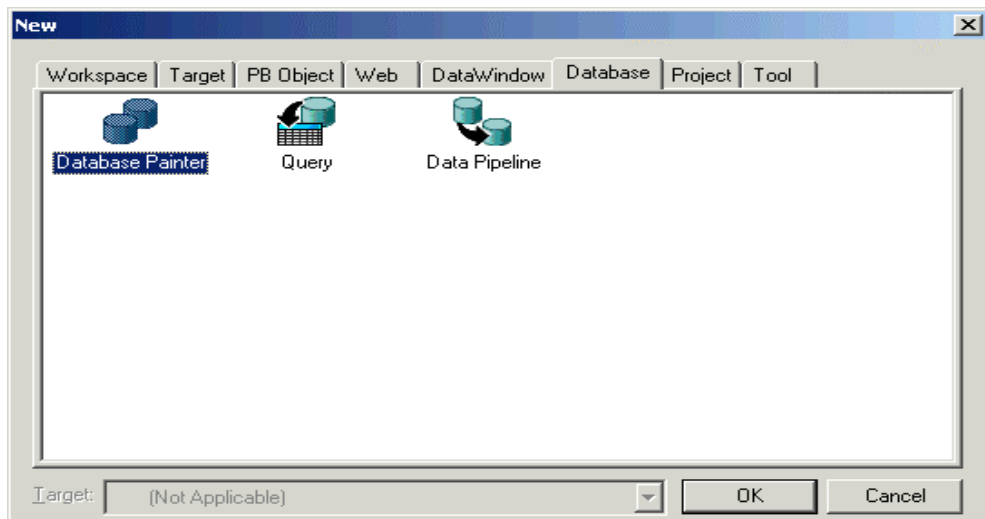


图 1-10 新建对象

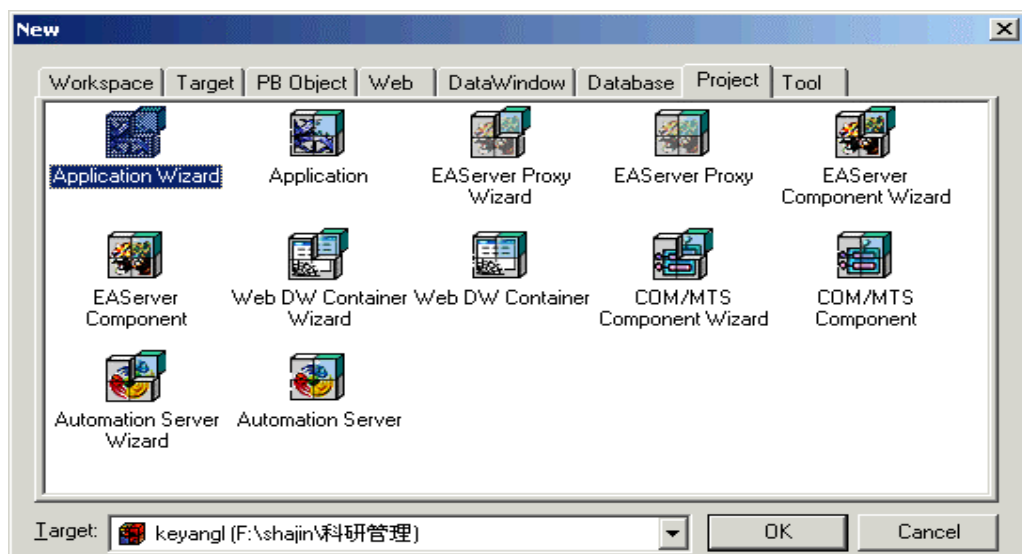


图 1-11 新建对象

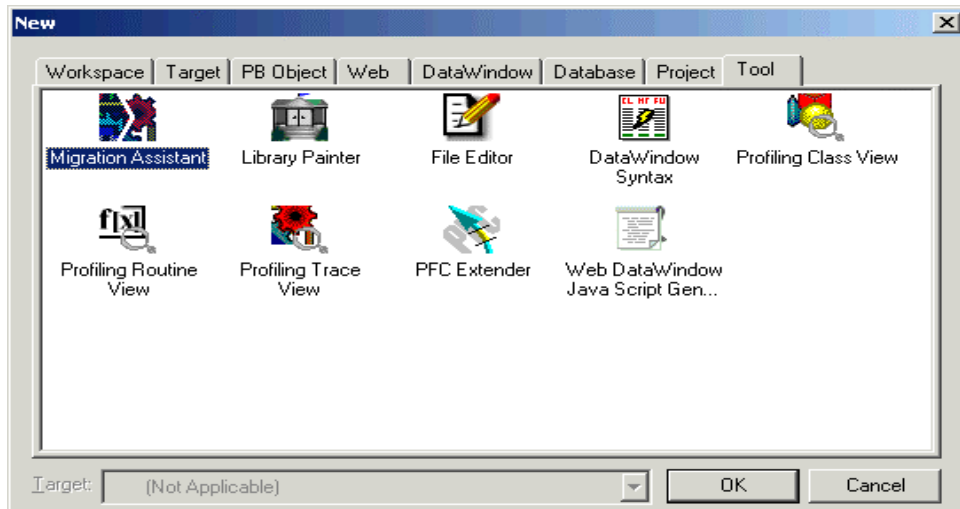


图 1-12 新建对象

继承对象画板 (Inherit Painter): 同 New 画板一样, 也是用来新建一个对象。它们的区别是: 以 New 方式创建对象时, 主要继承 PowerBuilder 的标准类; 而 Inherit 可以继承开发者自定义的对象。如果还没有创建任何对象, 这个菜单项是不可用的。如果已经创建了对象, 单击它后出现的对话框如图 1-13 所示。



图 1-13 继承对象

可以从 Target 下拉式列表框中选择当前工作空间中的目标; 从 Libraries 列表框中选择当前目标中的库文件; 从 Object of Type 列表框中选择要继承对象的类型; 在 Object Name 文本框中输入对象的名称, 也可从上面的详细列表中选择对象的名称。

打开画板 (Open Painter): 用于打开一个已经存在的对象, 以便对它进行编辑。它的对话框和 Inherit 对话框相似, 只是标题不同。

运行、预览对象画板 (Run/Preview Painter): 单独地运行或预览一个独立的对象而非整个应用, 从而方便地独立调试应用的各个组成部分, 提高开发效率。

系统树画板 (System Tree Painter): 打开系统树窗口。系统树窗口 (System Tree Window) 是资源浏览窗口, 该窗口以树形列表形式提供当前开发过程的 Target 对象的所

有编程资源系统视图信息。开发人员不仅可以通过它得到信息，也可以将树形列表中的对象直接拖放到画板视图使用中。

在该窗口的下部有四个标签，它们依次是：Workspace 标签、Page 标签、Language 标签、Components 标签。其中 Page 标签在 Web 对象列出 IE 对象模型和对象之间的集成关系；Language 标签列出 Web 对象应用过程相关的语言、语法元素；最后一个标签用于列出在程序中使用的诸如 ActiveX 控件、EAServer CTS 控件等控件名。

输出窗口画板 (Output Painter)：Output 窗口主要用于显示建立、调试、执行、保存和搜索等操作的结果。

处理列表画板 (To Do List)：打开处理列表窗口，选择目标，进行处理操作。

对象浏览器画板 (Browser Painter)：可方便地浏览应用中的各个对象及其组成，对浏览的内容还可生成 RTF 格式的文档。

文件夹窗口画板 (Clip Painter)：用于打开文件夹窗口。Clip 窗口可以方便地在应用程序中对频繁使用的代码进行拷贝。对于经常使用的代码段，开发人员可以将代码复制到文件夹窗口进行保存，当需要使用时，可以拖拽或粘贴到所需的画板视图中。在 Clip 窗口中存储了用户编程中所用 Clip 的名称及其内容预览信息列表。

库管理画板 (Library Painter)：提供了管理应用的各个组成部分的手段。PowerBuilder 把开发出的所有对象放在开发人员指定的库中，用库管理画板不仅可方便地管理这些对象，而且还可以使开发人员共享这些对象，加强对公共标准和开发环境的控制。

数据库配置画板 (DB Profile Painter)：配置 PowerBuilder 到数据库的连接参数。

EAServer 配置画板 (EAServer Profile Painter)：配置 PowerBuilder 到 EAServer 的连接参数。

数据库画板 (Database Painter)：以交互方式生成和维护数据库表、控制用户对数据库的访问、操作数据库的数据、定义列的扩展属性（包括列标题、显示格式、编辑格式、校验规则等）。

文本编辑画板 (Edit Painter)：文本编辑器。

调试画板 (Debug Painter)：使应用在调试状态下运行（单步或多步运行），找出对象及事件处理程序中的错误。

PowerBuilder 提供的主要扩展画板和功能如下：

- **系统设置画板 (System Option)：**系统设置工作。由于许多系统设置工作已在应用的属性对话框中定义，这里的系统设置工作比较简单。
- **纵向排列窗口画板 (Vertical Tile)：**将打开的多个窗口纵向排列。
- **横向排列窗口画板 (Horizontal Tile)：**将打开的多个窗口横向排列。
- **平铺窗口画板 (Layer)：**将打开的多个窗口平铺。
- **层叠窗口画板 (Cascade)：**将打开的多个窗口层叠。
- **排列图标画板 (Arrange)：**排列工作空间中缩成图标的窗口。
- **查找画板 (Search)：**打开 PowerBuilder 的搜索窗口，查找信息。

§1.2.3 PowerBuilder 菜单栏的组成

菜单的应用和画板的应用是相似的，这里只介绍几个画板中不存在的菜单项。

- File→Open Workspace...菜单项用于打开一个已有的工作空间。单击它后,会出现一个对话框,它的作用是可以选择打开工作空间的路径和名称。
- File→Recent Objects、Recent Workspaces 和 Recent Connections 三个菜单项分别用于打开最近打开过的对象、最近打开过的工作空间和最近连接的数据库。
- Tool→Toolbars...菜单项用来对 PowerBuilder 的工具栏进行修改。单击该菜单项后,出现的对话框如图 1-14 所示。

其中, Select Toolbar 列表框列出了当前界面上所有的工具栏,用户可以从中选择作为设置的对象。Move 分组框列出了工具栏可以移到的位置。Show Text 复选框用来决定工具栏的图标上是否要显示文字。Show PowerTips 复选框用来决定当前用户将鼠标在工具栏的图标上停留片刻后是否会显示提示性文字。Font Name 和 Font Size 下拉式列表框分别用来选择工具栏上字体的名称和大小。右边的 Customize 按钮用来对当前工具栏进行定制。New 按钮用来创建一个新的工具栏。

- Tools→Keyboard Shortcuts...菜单项用来修改和设置菜单项命令的快捷键,如图 1-15 所示。
- Tools→System Options...菜单项用来修改和设置系统参数,在下次启动系统的时候这些参数才会生效。

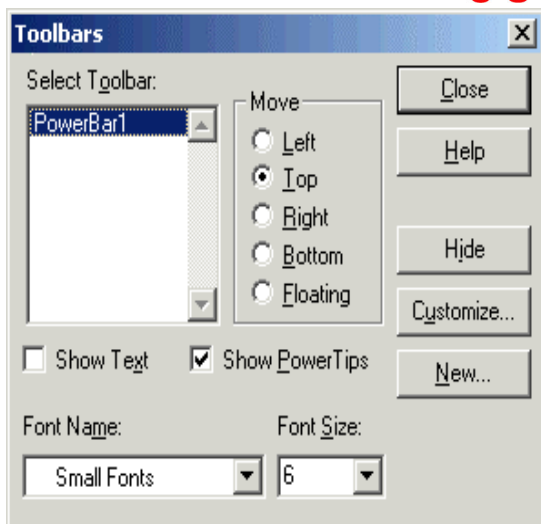


图 1-14 设置工具栏

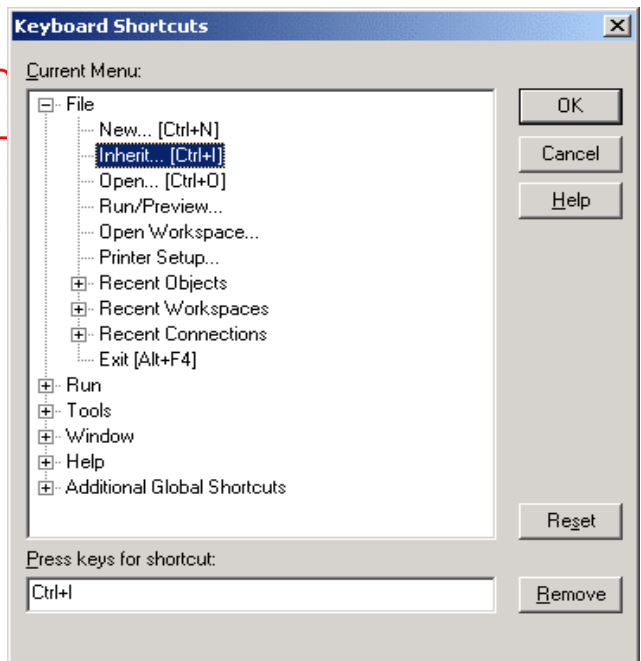


图 1-15 设置快捷键

§1.3 PowerBuilder 9.0 应用系统的构成

完整的应用系统由工作空间、目标、应用对象、窗口对象、窗口上的菜单对象、数据窗口对象和控件对象,以及这些对象的事件处理程序构成。

§1.3.1 工作空间和目标

在 PowerBuilder 8.0 以前的版本中,程序员一次只能开发一个应用程序,这样使得

应用程序无法取得多个可用的组件信息，因此在构建复杂应用时会大大增加难度。

PowerBuilder 8.0 之后提供了 Workspace 功能，这让开发人员同时多个应用或组件上进行开发，同时 PowerBuilder 用 Target 来描述加到 Workspace 中的应用或组件。PowerBuilder 中存在的三类 Target 是：Window 客户、EAServer 组件和 Web 站点。在 PowerBuilder 中，只有打开 Workspace 才能更有效地工作，开发者可以在 Workspace 中定义一个新的 Target 或者加入一个已经存在的 Target。因此，Workspace 已经变成了 PowerBuilder 中所有活动的核心。

§1.3.2 窗口、控件、事件和程序

在 PowerBuilder 中，窗口（Window）是主要的用户界面，软件开发人员往往把各种控制功能（如录入、查询、打印、统计等）做成按钮或其他控件放在窗口上由用户选择。应用对象、窗口对象、菜单对象、窗口上的数据窗口控件和其他控件对象都对应着若干个可能发生的事件，如应用对象和窗口对象上有 Open（打开）、Close（关闭）等事件，菜单项上有 Clicked（鼠标左键单击或按回车键）等事件，窗口上的控件有 Clicked、DoubleClick（鼠标左键双击）、Getfocus（得到焦点）、EditChanged（数据被修改）等事件。PowerBuilder 的程序是嵌套在事件中的，也就是说是由事件驱动的，一旦用户选定了控件也就同时触发了某个事件。软件开发人员只需为各个控件的事件编写事件处理程序，指定该事件被触发时要进行的处理，应用系统就这样被构造出来了。

事件处理程序用 PowerBuilder 的专用语言——PowerScript 语言编写。PowerScript 语言是一种高水准、面向对象的语言，提供了丰富的内置函数供软件开发人员使用。事件处理程序由 PowerScript 语句和函数组成，来响应被触发的事件并做相应的处理。

例如，菜单项上有 Clicked 事件，此事件的处理程序可能是打开另一个窗口，用户用鼠标左键单击此菜单项时，Clicked 事件被触发，指定的窗口被打开。又如窗口上有一个数据窗口和一些按钮，按钮上的文字为“录入”、“删除”、“存盘”等，这些按钮的功能分别为在数据窗口中插入一行、删除一行、将修改结果存盘，我们只需为这些按钮的 Clicked 事件编写相应的程序完成这些功能。

§1.3.3 函数

函数是 PowerScript 语言的重要组成部分，PowerBuilder 为软件开发人员提供了近 800 个内置函数，主要用于完成以下工作：访问和操纵数据，操纵对象、数字或文本，进行计算，字符串处理，数据类型转换，处理日期和时间，文本文件的输入和输出，生成商务图形，打印数据，与其他窗口的数据通讯，处理邮件，Web 应用，分布式应用等。第九章给出了 PowerBuilder 的常用函数。

此外，软件开发人员还可以通过建立用户自定义函数来满足某个应用系统的特殊要求。需要时还可调用 Windows 的 SDK 函数和外部 DLL 函数。

§1.4 本书的应用实例

本书内容的讲解和习题围绕着由下面的表组成的数据库进行：

(1) 科研人员简历表 (employees)

```

CREATE TABLE "dba"."employees"

    ("em_code" char (6) NOT NULL,           //人员编码
    "name" char (8) NOT NULL,               //姓名
    "sex" char (2) NOT NULL,                //性别
    "birthdate" date,                       //出生日期
    "tech_title" char (8) ,                 //职称
    "address" char (20) ,                   //住址
    "telephone" char (20) ,                 //电话
    "d_code" char (3) ,                     //所在部门编码
    "picture" long binary,                  //照片
    "record" long varchar                    //简历
, PRIMARY KEY ("em_code")) ;               //主键为 em_code

```

(2) 科研项目情况表 (items)

```

CREATE TABLE "dba"."items"

    ("it_code" char (4) NOT NULL,           //项目编码
    "itemname" char (30) NOT NULL,          //项目名称
    "first_department" char (20),           //甲方单位
    "second_department" char (20),          //负责单位
    "level" char (4),                       //项目级别
    "leader" char (8),                      //课题负责人
    "fund" numeric (7,0) ,                  //总经费
    "contract_date" date,                   //合同签订日期
    "finish_date" date,                     //完成日期
    "status" char (4) ,                     //状态
    "contents" long varchar,                //合同内容
    "remarks" long varchar                  //备注
, PRIMARY KEY ("it_code")) ;               //主键为 it_code

```

(3) 项目参加人员表

```

CREATE TABLE "dba"."member"

    ("it_code" char (4) NOT NULL,           //项目编码
    "em_code" char (6) NOT NULL,           //人员编码
    "order" integer,                       //名次
    "duty" char (28)                        //承担内容
, PRIMARY KEY ("it_code", "em_code")) ;    //主键为 it_code+em_code

```

(4) 获奖情况表 (prize)

```

CREATE TABLE "dba"."prize"

    ("prize_code" char (4) NOT NULL,       //获奖编码

```

```

        "prize_date" date,           //获奖日期
                                     //颁奖部委
        "conferer" char (8) ,       //获奖等级
        "prize_level" char (4) ,    //项目编码
        "it_code" char (4)          //主键为 prize_code
    , PRIMARY KEY ("prize_code"));

```

(5) 辅助编码表

```

CREATE TABLE "dba"."code"
    ("code" char (3) NOT NULL,       //编码
     "c_name" char (16) NOT NULL     //名称
    , PRIMARY KEY ("code"));        //主键为 code

```

还有以下的表在习题中要用到（表名和各列名、类型、宽度由自己定义）。

(6) 学术论文登记表

论文编码,论文题目,刊物编码,第一作者,发表年月,卷号,期号,内容简介。

(7) 学术刊物表

刊物编码,刊物名称,主办单位,电话,邮编,刊物级别。

(8) 著作登记表

著作编码,书名,第一作者,出版社,出版年月,字数,单价,获奖情况。

(9) 论文参加人员表

论文编码,人员编码,名次。

(10) 著作参加人员表

著作编码,人员编码,名次,完成字数,有关说明。

employees（科研人员简历）表的初始记录如下：

| em_code | name | sex | birthdate | tech_title | address telephone | d_code |
|---------|------|-----|------------|------------|-------------------|--------|
| 100001 | 刘俊海 | 男 | 1948-9-12 | B01 | 39314588 | A04 |
| 100002 | 王沛奇 | 男 | 1963-11-24 | B02 | 38784866 | A04 |
| 100003 | 丁宏伟 | 男 | 1970-4-27 | B03 | 37626346 | A04 |
| 100004 | 郝丽萍 | 女 | 1968-7-5 | B03 | | A04 |
| 100005 | 张志文 | 男 | 1958-8-21 | B02 | 38868733 | A03 |
| 100006 | 严华武 | 男 | 1949-5-16 | B02 | 37754685 | A03 |
| 100007 | 杨文彬 | 男 | 1965-6-18 | B05 | 38848573 | A04 |
| 100008 | 赵 磊 | 男 | 1975-1-30 | B04 | | A04 |
| 100009 | 陈伟锋 | 男 | 1959-5-24 | B02 | 36937222 | A04 |
| 100010 | 潘浩林 | 男 | 1968-3-18 | B03 | 34456832 | A03 |
| 100011 | 徐艳丽 | 女 | 1969-6-28 | B06 | 36590562 | A03 |
| 100012 | 刘 刚 | 男 | 1971-7-14 | B03 | 34851908 | A04 |

items（科研项目情况）表的初始记录如下：

| it_code | itemname | first_depart | second_depart | level |
|---------|---------------|--------------|---------------|-------|
| 2001 | 电子售票订票系统统计子系统 | 铁道部 | A04 | 部级 |

| | | | | |
|------|-----------------|-----|-----|----|
| 2002 | 饲料配料自动化系统 | | A04 | |
| 2003 | 制药厂管理信息系统 | 河北省 | A04 | 省级 |
| 2004 | 电厂自动控制系统 | 电力部 | A03 | 部级 |
| 2005 | 电子售票系统 2.0 版的优化 | 铁道部 | A04 | 部级 |

| leader | fund | contract_date | finish_date | status | contents | remarks |
|--------|------|---------------|-------------|--------|----------|---------|
| 100009 | 10 | 1996-12-2 | | 在研 | | |
| 100007 | 5 | 1990-3-15 | 1992-12-30 | 完成 | | |
| 100001 | 10 | 1987-5-12 | 1989-12-22 | 完成 | | |
| 100005 | 15 | 1995-2-18 | 1996-12-20 | 完成 | | |
| 100002 | 10 | 1998-1-18 | | 在研 | | |

member（项目参加人员）表的初始记录如下：

| it_code | em_code | order | duty |
|---------|---------|-------|------|
| 2001 | 100009 | 1 | 负责人 |
| 2001 | 100002 | 2 | 系统设计 |
| 2001 | 100007 | 3 | 软件设计 |
| 2002 | 100001 | 1 | 负责人 |
| 2002 | 100002 | 2 | 硬件设计 |
| 2002 | 100010 | 3 | 软件设计 |
| 2002 | 100012 | 4 | 系统测试 |
| 2003 | 100009 | 1 | 总体设计 |
| 2003 | 100003 | 2 | 网络设计 |
| 2003 | 100004 | 3 | 软件设计 |
| 2003 | 100011 | 4 | 系统维护 |
| 2004 | 100005 | 1 | 总体设计 |
| 2004 | 100006 | 2 | |
| 2004 | 100010 | 3 | |
| 2005 | 100002 | 1 | 总体设计 |
| 2005 | 100009 | 2 | |
| 2005 | 100008 | 3 | |

prize（获奖情况表）表的初始记录如下：

| prize_code | prize_date | conferer | prize_level | it_code |
|------------|------------|----------|-------------|---------|
| 3001 | 1996-3-5 | 河北省 | C13 | 2002 |
| 3002 | 1997-12-16 | 学院 | C31 | 2004 |
| 3003 | 1997-12-28 | 电力部 | C12 | 2003 |

code（辅助编码）表的初始记录如下：

| code | c_name |
|------|--------|
| A01 | 化工系 |
| A02 | 机械系 |

| | |
|-----|-------|
| A03 | 电子系 |
| A04 | 计算机系 |
| A05 | 经济管理系 |
| B01 | 教授 |
| B02 | 副教授 |
| B03 | 讲师 |
| B04 | 助教 |
| B05 | 高级工程师 |
| B06 | 工程师 |
| C01 | 国家一等奖 |
| C02 | 国家二等奖 |
| C03 | 国家三等奖 |
| C11 | 省部一等奖 |
| C12 | 省部二等奖 |
| C13 | 省部三等奖 |
| C31 | 院一等奖 |
| C32 | 院二等奖 |
| C33 | 院三等奖 |



科学出版社 营销宣传

1. PowerBuilder 9.0 具有哪些功能和特点？
2. PowerBuilder 9.0 由哪几部分组成？各起什么作用？
3. 怎样定制 PowerBar（将部分扩展画板加到 PowerBar 中）？
4. PowerBuilder 的各个基本画板的功能是什么？
5. PowerBuilder 的应用系统由哪些部分构成？
6. 你了解数据库的基本概念吗？能看懂§1.4 的内容吗？如果看不懂，请自学数据库概论的有关书籍，了解关系数据库的基本概念和结构查询语言 SQL。
7. 你熟悉某种程序设计语言吗？如果不熟悉，请自学第八章 PowerScript 语言部分。
8. 请自学第九章的前 5 节，对 PowerBuilder 的简单函数有一个基本的了解。

第二章 数 据 库

PowerBuilder 应用程序的核心是数据库。在关系数据库中，一个数据库是一系列表的集合，表存储某个实体或实体间联系的有关信息，列表示属性，行表示记录。数据库管理系统（DBMS）提供了定义数据、使用数据及与他人分享数据的全面控制。PowerBuilder 包括了 DBMS 的全部功能，可以对本地和网络服务器上的数据库进行操作，它包括：

- 创建本地数据库，连接本地和网络服务器上已存在的数据库。
- 创建、修改、删除数据库中的表和视图。
- 定义和修改表中各列的属性和规则。
- 创建和删除表的索引。
- 定义表间的关系，即主键和外部键的关系。
- 对表中的数据进行插入、删除、修改和检索。
- 数据库管理：定义表的访问权限，创建、修改和删除用户，创建和删除组，编写和执行 SQL 文件。

在 PowerBar 中与数据库操作有关的画板有数据库画板、数据库参数配置画板、EAServer 参数配置画板。

数据库操作的步骤：

① 创建数据库（如果已创建，此步骤可省）：

如果要创建的是本地数据库 ASA（Adaptive Server Anywhere），单击数据库画板，再选择 ODB ODBC→Utilities→Create ASA Database 进行操作（见后述）。

如果要创建的是 Sybase、Oracle、FoxPro 等数据库，则用 Sybase、Oracle、FoxPro 等系统创建数据库。

② 连接数据库：

创建 Adaptive Server Anywhere 数据库时，PowerBuilder 会自动连接到该数据库上。

如果已经为某个数据库建立了 DB Profile 文件，则选择菜单 File→Connect 连接此数据库即可。

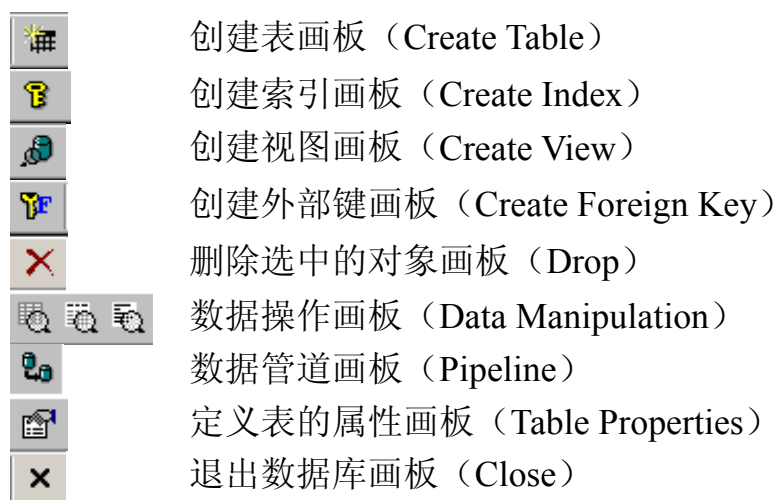
如果要通过 ODBC（Open DataBase Connectivity）连接到一个数据库上，则使用 ODBC 配置画板进行 ODBC 数据源的配置，再使用 DB Profile 画板配置 PowerBuilder 数据库的连接参数并连接。

如果要通过专用接口连接到数据库服务器上，使用 DB Profile 画板配置 PowerBuilder 到数据库的连接参数并连接。

③ 创建和修改表，创建视图，创建和修改表的索引、主键、外部键，数据输入、修改、浏览与输出，数据库管理等。

在数据库的操作中，使用最多的是 Database 画板。在 PowerBar 工具条上单击 Database

画板即可进入 Database 画板工作区，并出现此画板的 PainterBar，下面对此 PainterBar 上的画板做一简要介绍，以方便后续各节的使用。



§ 2.1 数据库的建立

§ 2.1.1 创建本地的 Adaptive Server Anywhere 数据库

Adaptive Server Anywhere 数据库可以与 PowerBuilder 运行在同一台机器上，作为 PowerBuilder 的数据库服务器，与 PowerBuilder 构成客户/服务器体系结构。由于 Adaptive Server Anywhere 在功能上已具备了大型数据库的特点，因此使用 Adaptive Server Anywhere 有如下好处：

由于 Adaptive Server Anywhere 本身已经是一个完整的数据库管理系统，所以在项目的开发阶段，单机即可开发大型数据库应用，而不必在网络状态下与大型数据库管理系统连接起来开发。

开发完成后，几乎不用更改，把开发出的应用系统拿到网络下即可运行在其他数据库上。

创建 Adaptive Server Anywhere 数据库的步骤是：

① 在 PowerBar 上单击 DB Profile 画板 ，弹出 Database Profile 对话框，如图 2-1 所示。在对话框中选择 ODB ODBC→Utilities→Create ASA Database，双击此项，弹出 Create Adaptive Server Anywhere Database 对话框，如图 2-2 所示。

② 在 ASA Version 中可以选择 ASA 7.0 或 ASA 8.0，在 Database Name 编辑框中给出数据库文件名（比如 kygl），在 User ID 框中给出用户名（缺省为 dba），在 Password 框中键入口令（缺省为 sql），然后单击 OK 按钮。PowerBuilder 开始创建指定的数据库。创建完成后，系统自动连接到该数据库上。

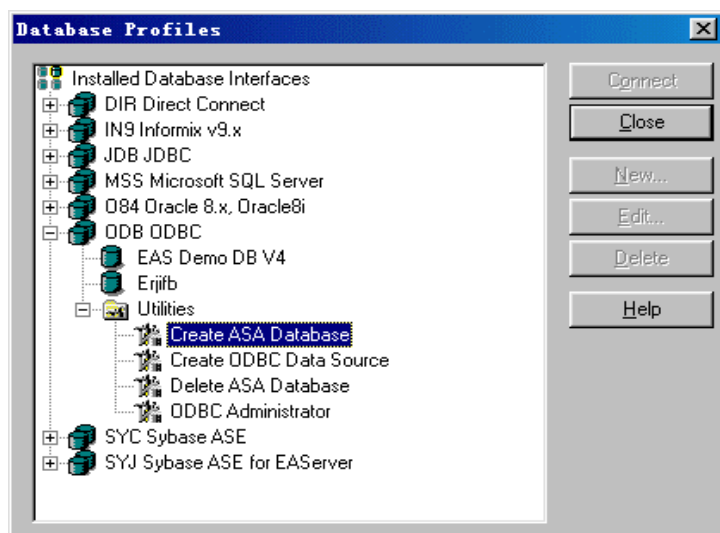


图 2-1 创建数据库

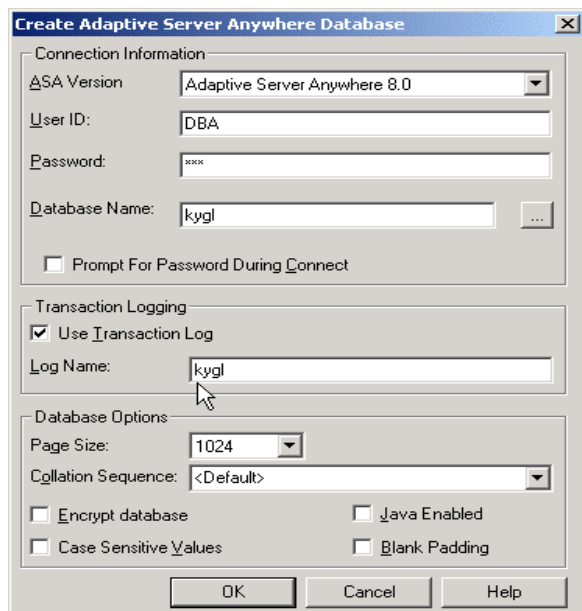


图 2-2 数据库信息

§ 2.1.2 创建表

创建并连接好数据库后就可在此数据库上创建表。创建表可使用 PowerBar 中的 Create Table 画板，或者选中 Tables 项单击鼠标右键 选择弹出菜单的 New Table...。具体步骤如下：

- ① 单击 Create Table 画板，出现 Columns 工作区，如图 2-3 所示。
- ② 输入列名、数据类型、长度和是否为空等列的基本信息。
- ③ 重复第②步，直到所有列输入完毕。
- ④ 单击 Save Change 画板，弹出 Create New Table 对话框。给定表名后单击 OK 按钮，指定的表被创建，同时在 Object Layout 工作区显示该表。

| Column Name | Data Type | Width | Dec | Null | Default |
|-------------|-----------|-------|-----|------|---------|
| em_code | char | 10 | | No | (None) |
| name | char | 10 | | No | (None) |
| sex | char | 2 | | No | (None) |

图 2-3 表列信息

- ⑤ 在输入或修改当前列时，还可输入列的扩展属性：
 - Display: 列中数据的显示格式。
 - Edit Style: 输入数据时的编辑风格。当为该列输入数据时，显示该列的编辑格式，以帮助用户按正确格式输入数据。
 - Validation: 数据的有效性规则。当为该列输入数据时，使用有效性规则检验输入数据的正确性。

- Headers: 列标题, 显示在列上面的提示。如果不输入, 系统会用列名作为提示。
- Label: 标签, 显示在列旁边的提示。如果不输入, 系统会用列名作为提示。
- Comments: 该列的注释, 以便于理解该列的含义。
- Justify: 指定列的对齐方式。
- Display ers: 指定列的高度。
- Display Width: 指定列的宽度。
- Initial Value: 指定列的初值。

§ 2.2 PowerBuilder 与各种数据库的连接

上一节讲的是创建本地数据库, 然后在此数据库上创建新表。如果要使用已存在的数据库, 则应先通过数据库接口软件与此数据库建立连接。

PowerBuilder 是专为各种数据库设计的客户端的应用开发工具, 它可以和其他数据库服务器(如 Sybase、Oracle、MS SQL、Informix 等)构成客户/服务器体系结构, 也可以连到如 dBase、FoxPro、Excel 等文件类型的数据源上, 对这些数据文件进行操作。

对于不同类型的数据源(数据库), PowerBuilder 通过采用特定接口软件的方式进行连接。PowerBuilder 与各种数据库的连接方式有两种:

- 通过 ODBC (Open DataBase Connectivity) 通用接口软件。
- 通过 PowerBuilder 为 Sybase、Oracle、MS SQL、Informix 提供的专用接口软件。

ODBC 通用接口是微软公司推出的开放式数据库互连标准接口。ODBC 接口以 SQL 作为标准的查询语言来存取连接的数据源, 它允许单个应用同时访问多个不同的 DBMS, 用户可以通过加载连接到不同数据库的驱动程序来建立与各种数据库的连接。但由于它调用层次多, 通过这种接口操作数据库的效率较低。

专用接口的特点是调用直接, 速度快, 并充分发掘了特定数据库的特色, 因此, 如果要与 Sybase、Oracle、MS SQL 等数据库连接, 建议使用专用接口。

§ 2.2.1 通过 ODBC 接口连接

具体操作步骤如下:

- ① 如果 PowerBuilder 要连接到网络上的数据库服务器, 先安装并调通网络。
- ② 安装数据库接口软件和 ODBC 软件。

③ 在 PowerBuilder 中配置 ODBC, 单击 PowerBar 上的 DB Profile 画板, 出现 Database Profiles 设置窗口, 如图 2-4 所示。

④ 选择 ODB ODBC→Utilities→Create ODBC Data Source 或 ODBC Administrator, 双击该选项, 出现 ODBC Data Source Administrator 设置窗口, 单击“添加”按钮, 出现 Create New Data Source 设置窗口, 选择要建立的数据源的驱动, 例如我们选择 ASA 8.0, 如图 2-5 所示。单击“完成”按钮后又出现 ODBC Configuration for Adaptive Server Anywhere 设置窗口, 在 ODBC 页中的 Data source name 项输入数据源的名字, 例如输入 kygl, 如图 2-6 所示。在 Login 页中选择 Use integrated login 或选择 Supply user ID and password 项, 并填写用户名和密码, 如图 2-7 所示。在 Database 页中的 Database name

和 Database file 项中填写已经存在的数据库名和数据库文件，如图 2-8 所示，完成后单击“确定”按钮。

⑤ 选定 Database Profile 窗口的 ODB ODBC 项，单击 New 按钮出现 Database Profile Setup-ODBC 设置窗口，在 Connction 页中的 Profile Name 项中填写配置文件的名字，在 Data Source 项中选择已经建立好的数据源，填写用户名 dba 和密码 sql，如图 2-9 所示。完成后单击 OK 按钮，PowerBuilder 自动连接此配置文件。



图 2-4 ODBC 管理



图 2-5 选择数据源驱动

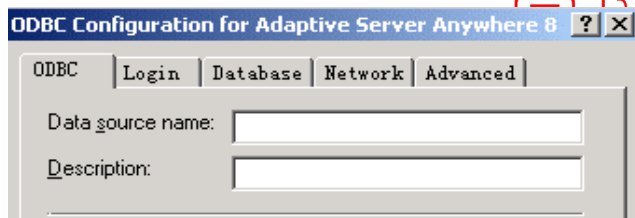


图 2-6 配置 ODBC

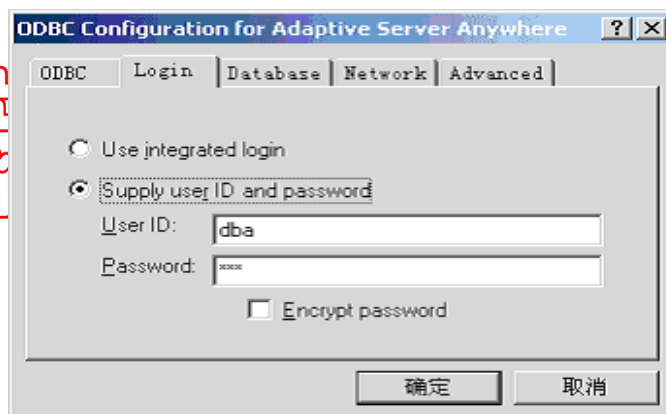


图 2-7 设置登陆参数

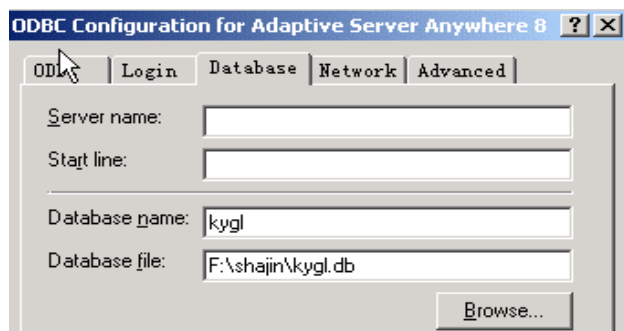


图 2-8 设置数据库

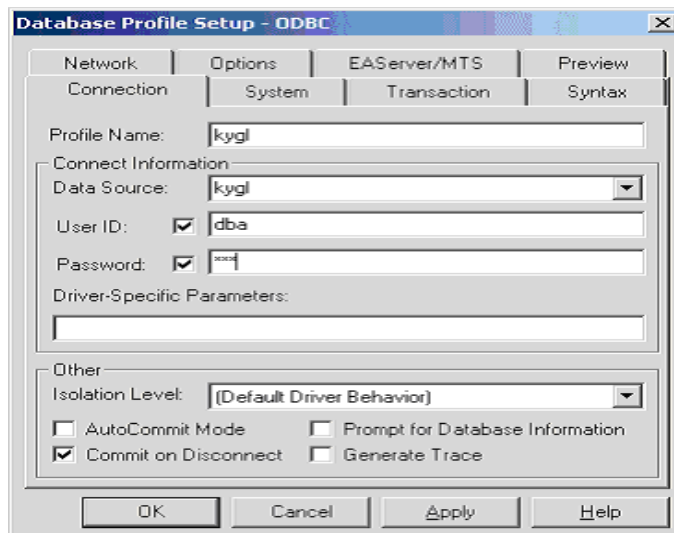


图 2-9 设置连接参数

§ 2.2.2 通过专用接口连接

下面以 Sybase11 为例讲解如何通过专用接口连接, 其他数据库类似, 不同的地方可以参考相应的技术资料。

- ① 安装并调通网络。
- ② 安装数据库软件和专用接口软件。
- ③ 配置到数据库的连接参数。

在 PowerBar 上单击 DB Profiles 画板时, 打开参数配置窗口 Database Profiles, 如图 2-10 所示。选择 SYC Sybase ASE→Open Client 11→Directory Services Editor, 双击出现 InterfacesDriver 设置窗口, 如图 2-11 所示。在 Server 项中添加服务器名, 在 Server Address 项中添加网络协议和地址, 用 ping 按钮测试以确定配置正确。

在 Database Profiles 设置窗口中选定 SYC Sybase ASE 项, 单击 New 按钮, 出现 Database Profile Setup-Adaptive Server Enterprise 设置窗口, 如图 2-12 所示。

在此窗口中填入要连接的数据库参数, 其中:

- Profile Name: 连接参数的名称 (任取)。
- Server: Sybase 服务器的名称。此名称是在客户端配置工具如 DSEEDIT 中指定的。
- Login ID: 登录用户的名称。
- Password: 用户的口令。
- Database: 要连接的数据库名称。

设置好连接参数后, 单击 OK 按钮, 保存创建的 Profile 并关闭设置窗口。这个 Profile 就显示在 Database Profiles 窗口中了。

④ 在 Database Profiles 设置窗口中选定 SYC Sybase ASE→Utilities→Open Client 11→SQL Advantage 项, 双击鼠标, 出现 SQL Advantage 窗口。在工具栏上单击 Open 图标, 打开 PowerBuilder 9.0 安装盘上的 Server 目录下的四个 SQL 文件, 如图 2-13 所示。分别选中它们, 再单击 SQL Advantage 窗口工具栏上的执行图标, 运行这些 SQL 语句。

⑤ 单击 Database Profiles 设置窗口中的 Connect 按钮可以连接到选中的数据源。此后我们在 PowerBuilder 中做的所有操作都是对此数据源进行的。



图 2-10 数据库接口

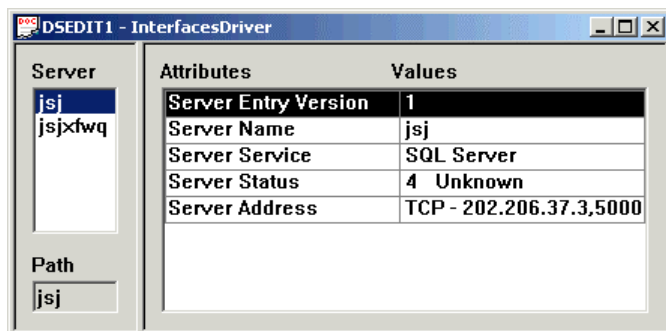


图 2-11 设置参数

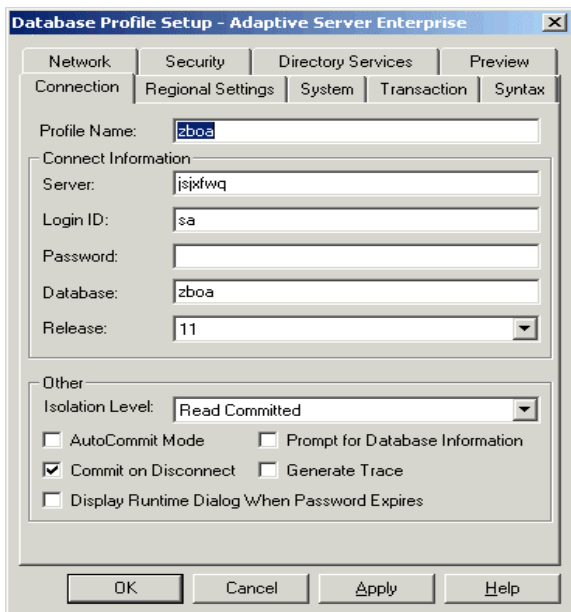


图 2-12 设置连接参数

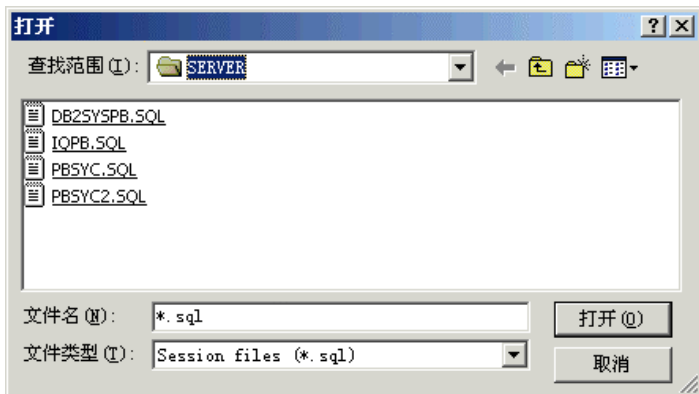


图 2-13 选择 SQL 文件

§ 2.2.3 改变连接的数据库

在进入数据库画板工作区后，PowerBuilder 就连接到一个指定的数据库上。在这之后所做的操作都是针对这个数据库的。如果想要操作其他的数据库，需要改变连接的数据库。有两种方法可以改变当前连接的数据库。

方法一：在 PowerBar 上单击 DB Profile 画板，在 Database Profiles 对话框中选择要连接的数据库，然后单击 Connect 按钮，即可连接到该数据库上。

方法二：在数据库工作区中，选择要连接的数据库源，单击鼠标的右键，在弹出式菜单中选 Connect 项，即可连接到该数据库上。

§ 2.3 操作数据库表

§ 2.3.1 打开、关闭和删除表

(1) 打开表

在 PowerBar 上单击 Database 画板，连接数据库，单击 Tables 项，出现所有的表，选定要打开的表，单击鼠标的右键，在弹出式菜单中选 Add to Layout 项，或者用鼠标将此表拖放到 Object Layout 工作区，此表就出现在 Object Layout 工作区中。

(2) 关闭表

如果想要关闭打开的表，可将鼠标指针移到该表名上，单击鼠标右键，在弹出的菜单上选取 Close 菜单项，即可关闭该表。

(3) 删除表

在数据库画板工作区中选择要删除的表，然后在 PainterBar 上单击 Drop Object 画板；或者将鼠标指到要删除的表名上，单击鼠标右键，在弹出的菜单上选择 Drop Table；或者从菜单条中选择 Object→Delete。这时弹出一个对话框，提示用户确认删除，单击“是

(Y)”按钮删除该表，单击“否 (N)”按钮则不删除该表。

§ 2.3.2 修改表和列的属性

对于创建好的表，可以修改表属性和表中各列的扩展属性。

1. 修改表属性

在数据库画板工作区中，选中要修改的表，然后在 PainterBar 上单击 Table Properties 画板，弹出表属性对话框，如图 2-14 所示。

该对话框共有四个 Tab 页，在 General 页中可为表加注释，在 Data Font 页中可修改表中数据的显示字体，在 Heading Font 页中可修改列标题的显示字体，在 Label Font 页中可修改列标签的显示字体。

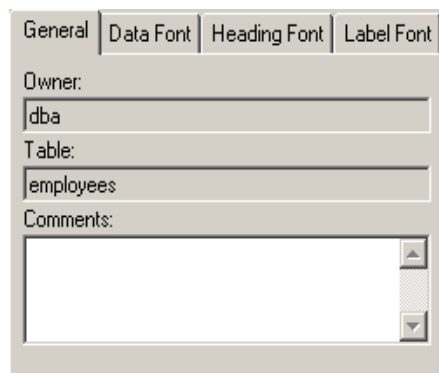


图 2-14 设置表属性

2. 修改列属性

在数据库画板工作区中，将鼠标指针移到要修改的列上，单击鼠标右键，弹出一个菜单，从中选择 Properties 菜单项，弹出列属性修改画面，如图 2-15 所示。

该对话框共有五个 Tab 页，在 General 页中可编辑列的注释，在 Headers 页中可修改列标题和列标签，在 Display 页中可创建、编辑列的显示格式，在 Validation 页中可创建、编辑列的有效性规则，在 Edit Style 页中可创建、编辑列的编辑风格。

(1) Display (显示风格) 页

显示风格是指该列中的数据在显示时的风格。在 Display 页 (图 2-16) 的 Display Format 框中显示已有的显示风格，单击其中的某项可以选择某个显示风格。

修改和创建显示风格的操作基本相同，下面介绍如何创建新的显示风格。

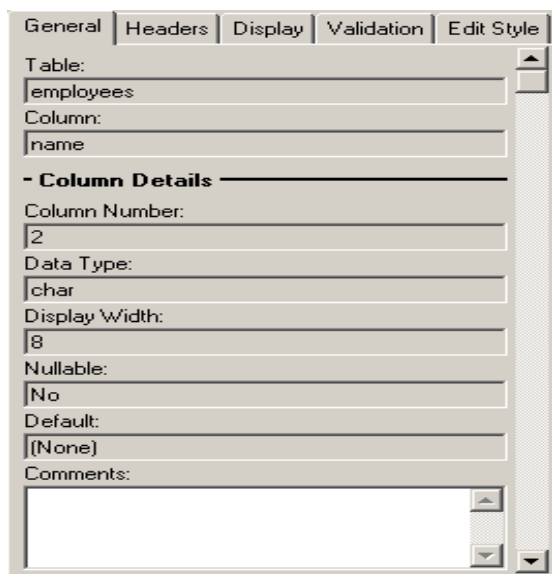


图 2-15 列的一般属性

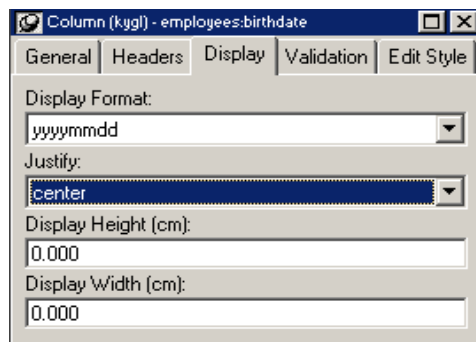


图 2-16 列的显示格式

① 在 View 菜单中选择 Extended Attributes，出现 Extended Attributes 工作区，如图

2-17 所示。单击鼠标的右键，在弹出式菜单中选择 New，出现的窗口如图 2-18 所示。

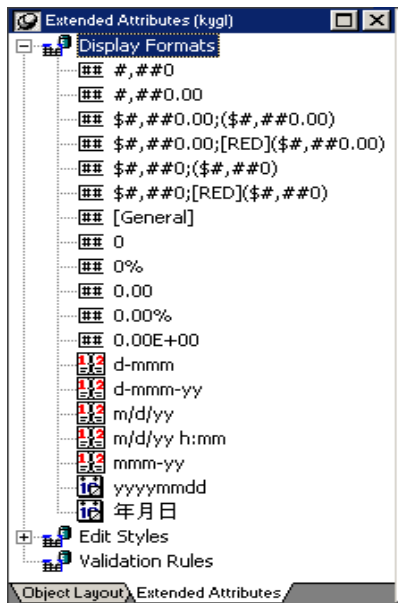


图 2-17 列的扩展属性



图 2-18 定义数据格式

② 在 Style Name 框中输入要创建的显示风格的名字，在 Data Type 框中选择数据类型，在 Format 框中输入或选择显示格式。

③ 保存定义的显示风格。定义显示风格可以使数据在显示或打印输出时更加符合用户的要求和习惯。例如，对于大多数用户，在打印日期型数据时希望以“yyyy 年 mm 月 dd 日”的样式输出。为了满足这一要求，为该列定义显示风格为“yyyy 年 mm 月 dd 日”即可。

各种显示风格的例子请见§7.5。

在 Display 页上，除了可以选择显示风格外，还可以设置一些有关显示的其他属性。例如：Justify（对齐方式），Display Height（显示高度）和 Display Width（显示宽度）。如果该列为字符型数据，则还有 Case（大小写）和 Picture（使用一个字符列显示图片，图片文件的类型有 BMP 和 WMF 两种）两项属性。

（2）Validation 页

校验规则用来校验用户输入的数据。如果输入的数据违反了校验规则，系统将拒绝接收数据，并且给出提示信息。在 Validation 页的 Validation Rule 框中列出了已有的校验规则，单击其中的某项可以选择某个校验规则。

修改和创建校验规则的操作基本相同，下面介绍如何创建新的校验规则。

① 在 Extended Attributes 工作区中，选定 Validation Rules 项，单击鼠标的右键，在弹出式菜单中选择 New...，弹出 Validation Rules 设置工作区，如图 2-19 和图 2-20 所示。

② 在 Validation Name 框中输入要定义的校验规则的名字，在 Data Type 框中选择数据类型。

③ 在 Message 框中输入出错提示信息。这个提示信息应是一个字符串，缺省的提示信息是 'Item ~'+@col+'~' does not pass validation test.'，你也可以定义自己的提示信息。

④ 在 Validation Rule 框中输入校验表达式，这个表达式必须是有关的逻辑表达式。在表达式输入过程中，可以利用粘贴工具来加快表达式的录入速度。

定义检验规则可以满足用户对数据的一些特殊要求。例如，某表中有一列为 Age（年龄），它的数据类型为整型，长度为 2 位，用于记录在职员工的年龄。如果不做任何限制，它可以接收 0~99 之间的任意数字，这显然是不合适的。因此，需要为该列定义一个校验规则，其校验表达式为"@col>17 And @col<61"，这样就可以保证输入的员工年龄在 18~60 岁之间。

（3）Edit Style 页（图 2-21）

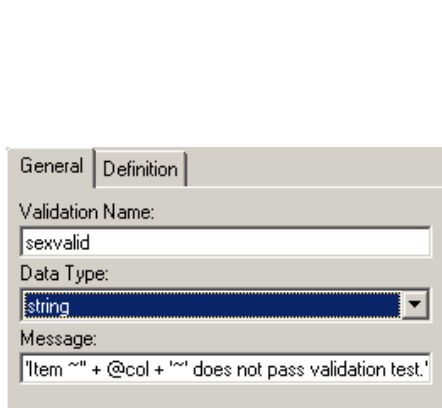


图 2-19 设置合法规则

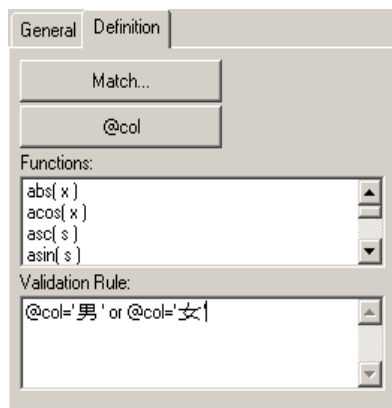


图 2-20 定义合法规则

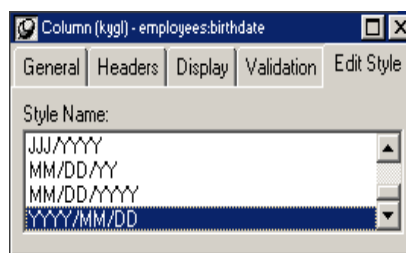


图 2-21 设置编辑格式

编辑风格是指用户输入数据的格式。定义恰当的编辑风格不仅可以提高数据的输入速度，而且可以减少由于操作失误而录入错误的数据。例如，对于日期类型的字段，如果不定义编辑风格，在输入数据时就需要自己输入分隔符“/”或“-”；如果对此字段定义了日期型的编辑风格，则只需输入数字即可，还可避免输入无效的日期。

在 Edit Style 页的 Style Names 框中列出了已有的编辑风格，单击其中的某项可以选择某个编辑风格。各种编辑风格的例子请见§7.9。

§ 2.3.3 修改表的定义

当一个表被创建后，可以对它做如下的修改：

- 增加列。但增加的列必须允许为空。
- 在一些数据库管理系统中，可以修改一个字符列的长度或将它改为可空，但不能将一个已经定义为可空的列改为非空。
- 修改表和列的所有可修改信息。

修改时，选定要修改的表，单击鼠标的右键，选择 Alter Table...，其后的操作与创建表时类似。

在创建和修改表时，PowerBuilder 会把相应的操作转化成一组 SQL 语句。在保存修改结果时，系统自动执行这些 SQL 语句完成对表的修改。在保存修改结果前，可以查看这些 SQL 语句，方法是单击弹出式菜单的 Export Syntax 项。

在查看 SQL 语句时，可拷贝 SQL 语句到剪切板，或将 SQL 语句存成一个扩展名为.sql 的文件，或打印 SQL 语句。

§ 2.3.4 建立索引

对于一个基本表，可以根据需要建立若干个索引，以提供多种存取路径。索引的建立和删除通常由 DBA（数据库管理员）或表的属主（即建立的用户）负责。存取路径的选择由系统自动进行，用户不能在存取数据时选择索引。

根据索引所包含的列数分为单值索引（包含一列）和多值索引（包含多列）；根据索引列的值是否允许重复分为惟一索引（不允许重复值）和可重复索引。

创建索引时，选定要创建索引的表，再单击鼠标的右键，在弹出式菜单中选择 New → Index 项。出现创建索引工作区，如图 2-22 所示。

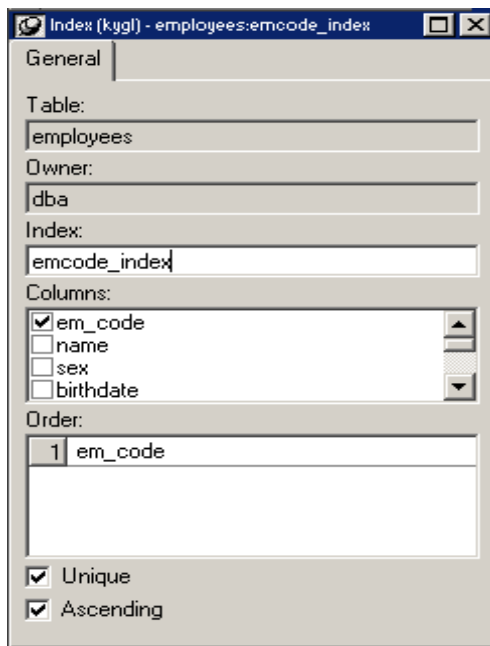


图 2-22 定义索引

在 Index 框中输入索引名，单击要做索引的列（可以选择多列），选中的列显示在 Columns 框中被打上对勾。再选择惟一索引（Unique）或可重复索引（Duplicate），升序（Ascending）或降序（Descending）。

修改索引与创建索引类似，在 Indexes 页中选中要修改的索引，单击 Edit 按钮。在 Create Index 工作区中根据需要选择或取消列。

删除索引时，在 Indices 项中选中要删除的索引，单击 Drop Object 画板。

§ 2.3.5 建立主键和外部键

在关系数据库系统中，有三类完整性约束，即实体完整性、参照完整性和用户自定义的完整性。其中实体完整性和参照完整性是关系系统必须满足的完整性约束，它们分别由主键和外部键来实现。如果使用的 DBMS 支持主键和外部键，那么就可以在 PowerBuilder 中使用它们。

1. 创建和修改主键

① 选定要创建主键的表，再单击鼠标的右键，在弹出式菜单中选择 New →

PrimaryKey 项，出现创建主键工作区，如图 2-23 所示。

② 在 Columns 框中选择一列或多列作为主键，但是要注意，只有非空列才能作为主键列，否则在存盘时会得到一个错误信息。

③ 根据不同的 DBMS 指定一些其他信息。这些信息是因 DBMS 而异的。例如，Sybase 要求给出主键名，指出主键约束是簇类（Clustered）还是非簇类（Nonclustered）。

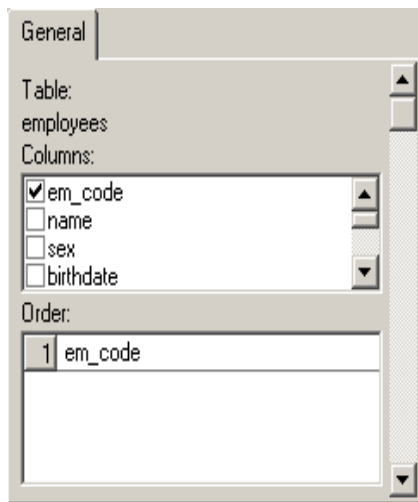


图 2-23 定义主键

2. 删除主键

要删除主键，只需在图 2-23 中的 Columns 框中取消所有作为主键的列，操作过程同上。

3. 创建和修改外部键

① 选定要创建外键的表，再单击鼠标的右键，在弹出式菜单中选择 New→Foreign Key 项，出现创建外键工作区，如图 2-24 所示。在 General 页中的 Column 框中选中要建立外键的列。

② 在 Primary Key 页上选择关联的表，如图 2-25 所示。

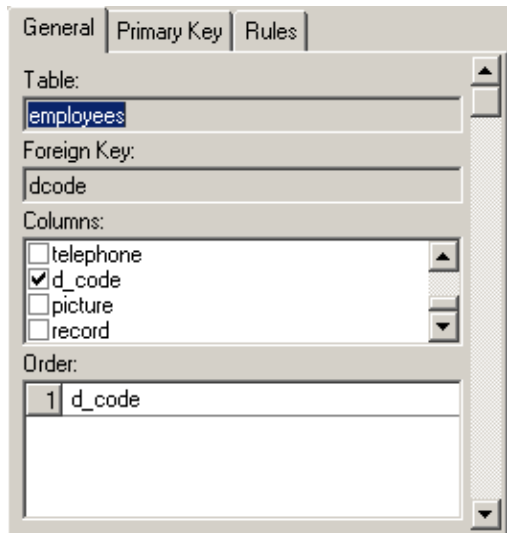


图 2-24 定义外键

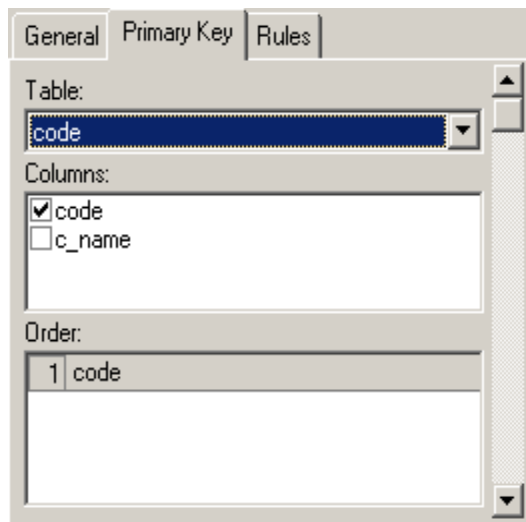


图 2-25 定义关联

③ 在 Rules 页上的 On Delete of Primary Table Row 框中有三个选项，它们决定了当删除主键表中的一行时，外部键表中的相应行如何处理：

- Disallow if Dependent Rows Exist: 如果外部键表中存在参照行，则不允许删除主键表中的行。
- Delete any Dependent Rows: 删除主键表中的行时，把外部键表中相应的行也删除。
- Set Dependent Columns to NULL: 删除主键表中的行后，把外部键表中的参照列的值置为空。

④ 在 Primary Key 页的 Table 框中选择外部键要参照的表，该表的主键显示在 Columns 框中。注意：作为外部键的列必须与被参照表的主键列相匹配。

⑤ 根据不同的 DBMS 指定一些其他信息。

4. 删除外部键

要删除外部键，只需选中某个外部键后单击 Drop Object 画板即可。

§ 2.3.6 数据库操作日志

在 PowerBuilder 中使用各种画板对数据库进行的各种操作，其实质是向 DBMS 发出一系列 SQL 指令。例如用 Create Table 画板创建一个表，相当于一条 Create Table 指令。利用 PowerBuilder 的数据库操作日志，可以记录这些指令。如果需要，还可以把它们以文本文件的形式保存起来。具体方法如下：

① 在数据库操作中选取 View→Activity Log 菜单，打开 Activity Log 工作区，在此工作区中单击鼠标的右键，选择 Start Log 启动 PowerBuilder 操作日志。以后所做的操作将出现在 Activity Log 工作区中。

② 选取 File→Save As... 菜单可以把日志保存为扩展名为 .sql 的文本文件。

③ 选取 Stop Log 菜单停止记录 PowerBuilder 的操作日志。

§ 2.4 视图操作

视图是从一个或几个基本表（或视图）导出的表。但视图和基本表不同，视图是一个虚表，即视图所对应的数据并不实际存储在数据库中，数据库中只存储视图的定义（存在数据字典中）。视图一经定义就可以和基本表一样被查询、更新，也可以在视图的基础上定义新的视图。通常由于以下原因可能要用到视图：

- 需要多次执行一些 SELECT 语句。
- 限制对一个表某些字段的访问。
- 例如，对职工情况表，可建立一个视图包括除工资和住址以外的其他字段，这样就可以让某些用户只能查询职工情况中除工资和住址以外的信息。
- 为了更方便地同时访问多个表的信息。

§ 2.4.1 创建和删除视图

1. 创建视图

- ① 在数据库画板中，单击 **Create New View** 画板，打开 **Select Tables** 对话框。
- ② 在 **Select Tables** 对话框中选择要基于的表和视图，然后单击 **Open** 按钮。这时选中的表和视图显示在视图工作区中（图 2-26）。
- ③ 选择需要的列，而且还可以加入计算列。
- ④ 如果选择了多个表，那么应该连接这些表。初始情况下 **PowerBuilder** 按如下规则连接表：
 - 如果表之间有主键/外部键关系，自动按主键/外部键连接它们。
 - 如果没有主键/外部键关系，**PowerBuilder** 试图用列名和类型相同的列连接表。

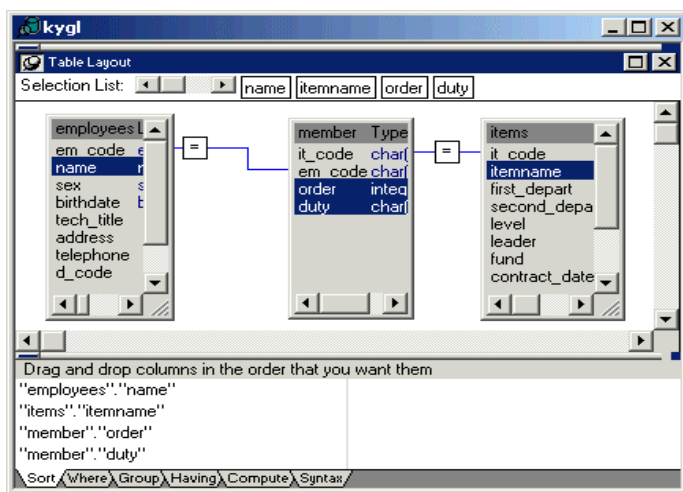


图 2-26 选择表和视图

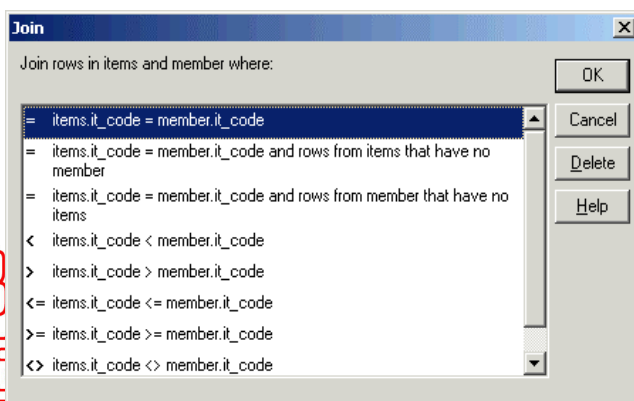


图 2-27 Jion 对话框

如果要创建新的连接关系，可在 **PainterBar** 上单击 **Join** 画板，然后单击用于连接的列即可。如果要建立非等值连接，那么在画板工作区中单击连接表达式，这时 **Jion** 对话框弹出（图 2-27）。

从连接对话框中选择需要的表达式，单击 **OK** 按钮返回画板工作区。

⑤ 在 **Where** 页中指定限制条件；在 **Group** 页中指定分组条件；在 **Having** 页中指定对组的限制。

⑥ 当视图定义完成后，单击 **Return** 画板，弹出 **Save View Definition** 对话框，在 **Name** 框中输入视图名。

⑦ 单击 **Create** 按钮，创建视图并返回到数据库画板工作区。

2. 删除视图

在数据库画板工作区选择要删除的视图，在 **PainterBar** 上单击 **Drop** 画板，**PowerBuilder** 提示是否要删除该视图，确认后即可删除。

§ 2.4.2 查看视图的 SQL 语法

1. 查看正在创建的视图的 SQL 语法

在创建新视图的过程中,可以通过选取 Syntax 页来查看视图的 SQL 语法,如图 2-28 所示。

2. 查看已存在的视图的 SQL 语法

在数据库画板工作区中打开一个视图,双击该视图,出现 View 属性工作区,视图的语法显示在 Definition 框中。

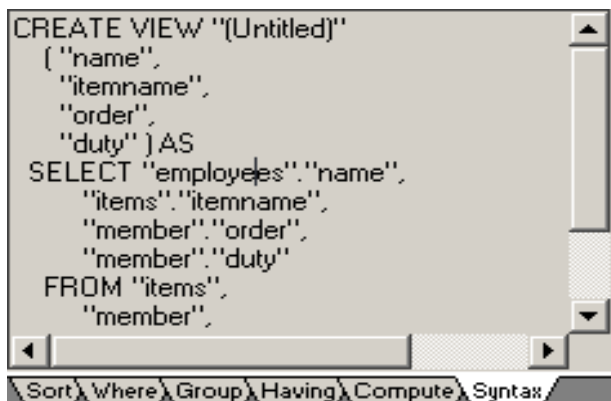



图 2-28 查看视图的 SQL 语法

§ 2.4.3 表或视图语法的输出

PowerBuilder 可将表或视图的语法导出到日志中。当你想在修改表或视图之前为之建立一个备份,或者想在其他数据库上创建相同的表时,这个功能就非常有用。具体操作步骤如下:

- ① 在数据库画板工作区中选取要导出语法的表或视图。
- ② 单击鼠标右键,从弹出式菜单中选取 Export Syntax 菜单项。
- ③ 如果你选的是视图,那么 PowerBuilder 立即将语法导出到日志中;如果你选的是一个表,并且安装了多个 DBMS,则出现 DBMS 对话框,选择正确的 DBMS。如果选了 ODBC,再出现数据源对话框让你选择数据源,单击 OK 按钮导出语法。

§ 2.5 数 据 操 作

当你在数据库画板中工作时,可能经常需要查看已有数据或输入一些新的数据,以便测试显示风格、校验规则和编辑风格等。为此,PowerBuilder 提供了三个 Data Manipulation (数据操作)画板 ,分别用来完成:

- 检索和操作数据。
- 把检索到的内容存成多种形式(如 Excel、dBase、Text 和 HTML Table 等格式)。

§ 2.5.1 数据检索

① 在数据库画板中选取要对其进行操作的表或视图。

② 单击 **Grid**（网格格式）、**Tabular**（表格格式）、**Freeform**（自由格式）三个画板之一。数据操作画板打开，并且开始检索数据，检索完毕后屏幕显示出所检索的数据。

数据操作画板实际上是一个数据窗口，它的编排风格和数据窗口对象是一致的。如果想要重新检索数据，可在 **PainterBar** 上单击 **Retrieve** 画板。对于检索到的数据，还可以文件的形式保存起来。具体方式是：从菜单中选取 **File**→**Save Rows As...**，弹出“另存为”对话框（图 2-29）。

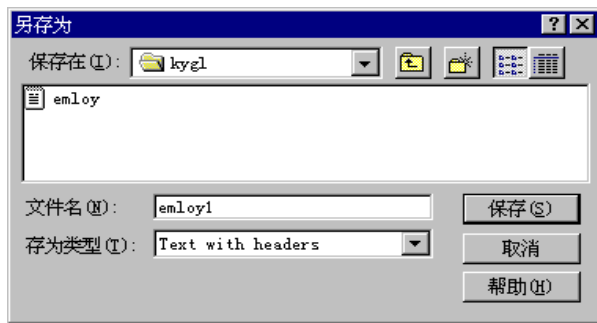


图 2-29 “另存为”对话框

在“存为类型”框中选择要保存的文件格式，在“文件名”框中输入文件名，再单击“保存”按钮，保存检索到的数据为指定格式的文件。

§ 2.5.2 数据修改

在数据操作画板中，可以增加、修改和删除表中的数据。当完成这些操作后，可以把这些改变保存到数据库中。但如果是对视图操作，是否可对数据进行修改取决于所使用的 **DBMS** 及视图的类型。

- 修改已有的数据：将光标移到要修改的数据上进行修改。
- 增加一行：单击 **Insert Row** 画板，一个空行被插入，然后为每一个字段输入数据。
- 删除一行：单击 **Delete Row** 画板，当前光标所在行被删除。
- 查看当前修改的情况：从菜单中选取 **Rows**→**Describe** 菜单项，弹出 **Describe Rows** 对话框（图 2-30）。该对话框显示了被删除的行数（未提交）、显示的总行数、过滤掉的行数、被修改的行数（未提交）。

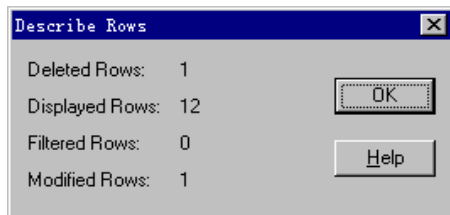


图 2-30 Describe Rows 对话框

- 保存修改结果（提交）：单击 **Save Changes** 画板，保存操作结果。

在操作过程中，最好是增加、修改、删除一行，就提交一次，并重新检索一下。

§ 2.5.3 数据排序和过滤

在数据操作画板中，可以为行定义排序标准和过滤器。但在数据操作画板中定义的排序标准和过滤器不会影响表和数据窗口。

1. 数据排序

① 在数据操作画板中，从菜单中选择 Rows→Sort 菜单项。弹出 Specify Sort Columns 对话框（图 2-31）。

② 把要排序的列从 Source Data 框拖到 Columns 框中。在缺省情况下，数据是按升序排列的。如果要按降序排列，单击右边的校验框。

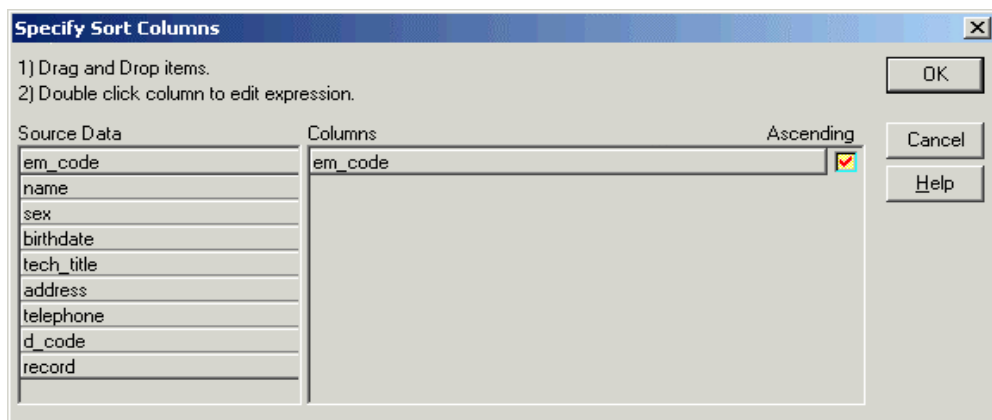


图 2-31 Specify Sort Columns 对话框

③ 如果要按多个字段排序，则依次拖动字段。如果要编辑表达式则双击 Columns 中的字段，弹出 Modify Expression 对话框（图 2-32）。在对话框中输入表达式，单击 OK 按钮，返回 Specify Sort Columns 对话框。

④ 单击 OK 按钮返回到画板工作区，这时数据已按要求排好序。

2. 数据过滤

① 在数据操作画板中，从菜单中选择 Rows→Filter 菜单项。弹出 Specify Filter 对话框（图 2-33）。

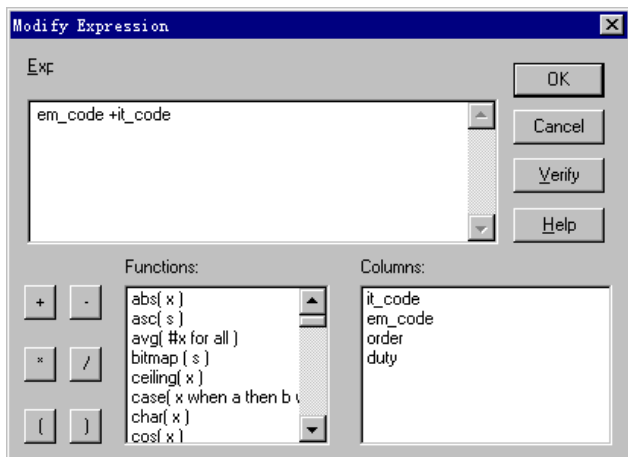


图 2-32 Modify Expression 对话框

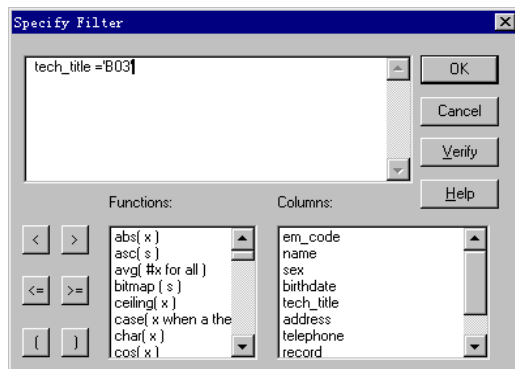


图 2-33 Specify Filter 对话框

② 输入一个逻辑表达式。表达式中可包含表中的列、操作符和 PowerScript 函数。

③ 单击 OK 按钮。系统过滤表中的数据，将符合条件的行显示出来。

如果要删除过滤器，重复步骤①，然后删除逻辑表达式，单击 OK 按钮。

§ 2.5.4 大量数据的录入

在数据操作画板中也可以从一个外部文件中导入数据到数据库的某个表中。具体步骤如下：

① 在数据库画板工作区中选取要导入数据的表，然后打开数据操作画板，从菜单中选取 Rows→Import，弹出 Select Import File 对话框。

② 选择要导入的数据文件。

可导入的文件类型有两种：文本文件 (*.txt) 和 dBase 文件 (*.dbf)。对于文本文件，要求它的每一行对应表中的一条记录，而且每个数据项之间由一个制表符 (Tab) 分隔。另外，不管哪种类型的文件，导入的数据一定要与表中对应列的数据类型相匹配。

§ 2.5.5 数据打印

在数据操作过程中，可以把显示的数据打印出来，打印前可以以不同尺寸进行打印预览。预览和打印数据的步骤如下：

① 打开数据操作画板，从菜单中选取 File→Print Preview 菜单项。

② 改变预览尺寸：从菜单中选取 File→Print Preview Zoom 菜单项，从 Zoom 对话框中选取合适的预览比例。

③ 再次选取 File→Print Preview 菜单项，结束预览。

④ 在 PainterBar 上单击 Print 画板（或选择 File→Print 菜单），弹出 Print 对话框（与 Windows 98 中的打印对话框相同）。

⑤ 填写打印对话框后单击 OK 按钮，打印数据。

§ 2.6 数据库的管理

在客户/服务器体系结构中，PowerBuilder 属于客户端的开发工具，但是通过 ISQL Session 工作区，PowerBuilder 也可以完成一定的数据库管理工作。PowerBuilder 的 ISQL Session 工作区提供了以下功能：

- 控制对当前数据库的访问。
- 创建 SQL 语句并执行。

在数据库画板中，选择 View→Interactive SQL 菜单项，即可打开 ISQL Session 工作区，如图 2-34 所示。编写 SQL 语句，选择菜单项 Design→Execute ISQL...就可执行。

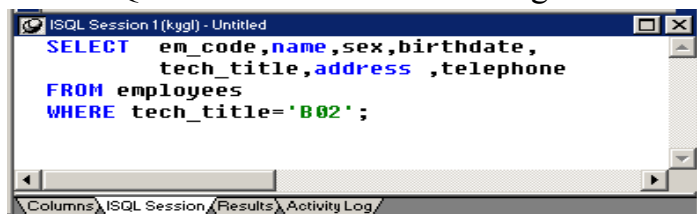


图 2-34 SQL 语句

§ 2.6.1 数据库存取控制

PowerBuilder 可以连接到不同的数据库上, 如 Adaptive Server Anywhere、Oracle、Sybase 等。由于这些数据库的生产厂商不同, 在对数据库进行存取控制时存在着一些差别。以下的操作都是针对 Adaptive Server Anywhere 数据库的。

1. 创建、修改和删除用户

(1) 创建用户

① 在 Object 工作区中, 选中 Users 项, 单击鼠标右键, 在弹出式菜单中选择 New User...项, 出现 User 工作区 (图 2-35)。

② 在 User 框中输入用户名, 在 New Password 和 Confirm New Password 框中输入口令, 然后单击 Save 按钮, 即可创建指定的用户。

(2) 修改用户口令

双击 Users 选项, 选中一个用户名, 双击鼠标, 出现 User Properties 工作区。在此工作区中可修改用户的口令。

(3) 删除用户

双击 Users 选项, 选中一个用户名, 单击鼠标右键, 在弹出式菜单中选择 Delete User 菜单项, 即可删除该用户。

2. 创建和删除用户组

(1) 创建用户组

① 在 Object 工作区中, 选中 Groups 项, 单击鼠标右键, 在弹出式菜单中选择 New Group...项, 出现 Group 工作区 (图 2-36)。

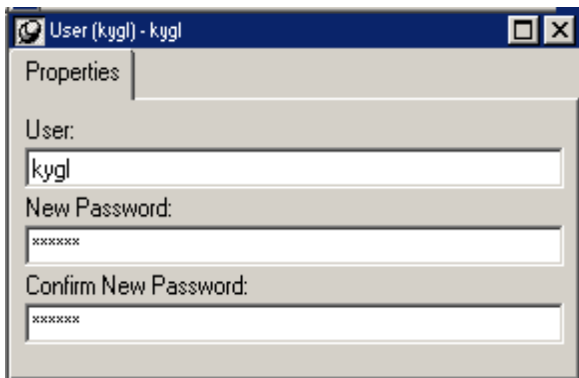


图 2-35 User 工作区

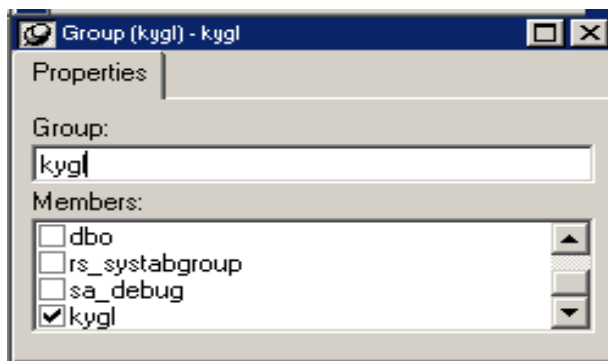


图 2-36 Group 工作区

② 在 Group 编辑框中输入组名, 在 Members 框中选择用户, 然后单击 Save 画板, 即可创建指定的组。另外, 还可以选中组名, 双击鼠标修改组的成员。

(2) 删除组

双击 Groups 选项, 选中一个用户名, 单击鼠标右键, 在弹出式菜单中选择 Delete Group 菜单项, 即可删除该组名。

3. 定义表的访问权限

① 进入 Database 画板工作区, 选择 Design→Table Security 菜单, 弹出 Table Security 对话框 (图 2-37)。

② 在 Groups, Users 框中选择用户或组。

③ 在 Tables 框中选择要修改权限的表。

④ 在 Privileges 组框中选择所需权限。

注意: 通过复选框授予的权限都是表级的, 如果要改变某列的更新 (Update) 权限, 可单击图中右下角的 Update 按钮。

⑤ 单击 Apply 按钮确认修改。

⑥ 单击 Done 按钮, 返回 Database 画板工作区。

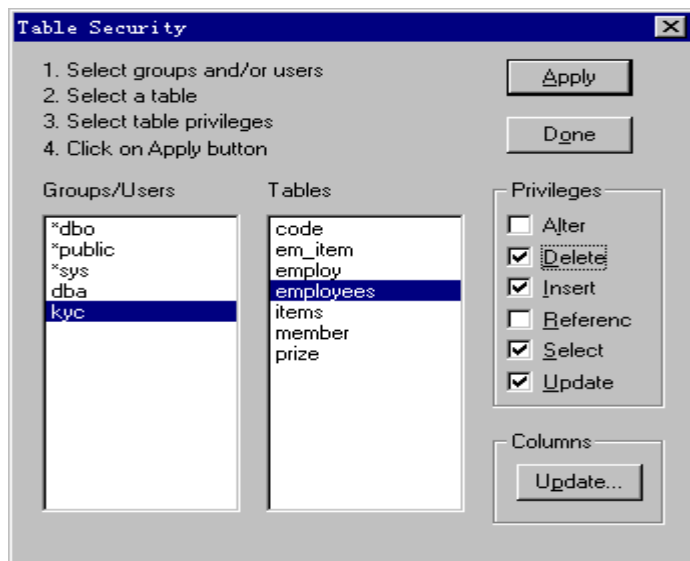


图 2-37 设置表的权限

§ 2.6.2 执行 SQL 语句

在 PowerBuilder 中 SQL 语句必须以一个特定符号结束, 缺省的 SQL 语句结束符是分号 “;”。可用 Design→Option 菜单项定义自己的 SQL 语句结束符。

1. 输入 SQL 语句

输入 SQL 语句有三种方法。

(1) 粘贴

① 单击 Paste SQL 画板, 在 SQL Statement Type 对话框中选择语句类型 (SELECT、INSERT、UPDATE 和 DELETE 之一)。

② 在 Select Tables (选择表) 对话框中选择 SQL 语句要引用的表, 单击 Open 按钮, 在打开的表中选择 SQL 语句要引用的列。

③ 完成后, 单击 Return 画板, 返回画板工作区。

(2) 键盘录入

使用键盘可以输入任何 SQL 语句, 包括一些不能用粘贴画板建立的 SQL 语句, 例

如创建存储过程和创建触发器。

(3) 导入文本文件

从菜单中选择 Edit→Paste Special→From File..., 在 File Import 对话框中选取要导入的文件, 单击 OK 按钮, 导入文件内容。

2. 执行 SQL 语句

在 PainterBar 上单击 Execute 画板, 可以执行 ISQL Session 工作区中的 SQL 语句, 其实质是 PowerBuilder 把工作区中的 SQL 语句向 DBMS 提交。在 ISQL Session 工作区中可以执行如下四类 SQL 语句:

- 数据查询语句 Select。
- 数据定义语句, 用于创建和删除表、视图、存储过程和触发器等数据库对象。
- 数据操纵语句, 用于对表中的数据进行 Insert (插入)、Delete (删除) 和 Update (修改) 操作。
- 数据控制语句, 用于控制用户对数据的存取权力, 如授权语句 (Grant) 和回收语句 (Revoke)。

§ 2.6.3 数据库的备份与恢复

数据的安全性越来越受到人们的重视, 保证数据安全性的措施之一就是要定期地对数据库进行备份。当操作系统瘫痪、数据库文件被破坏或硬盘被损坏时, 我们可以从备份的文件恢复数据。

不同的数据库有不同的数据备份与恢复措施, 下面以 ASA 8.0 为例说明数据库的备份与恢复方法。有关其他数据库的数据备份与恢复, 请参考相应的资料。

(1) Sybase Central 工具

启动应用程序, 然后连接要备份的数据库。选中数据库, 单击鼠标右键, 在弹出式菜单中选择 Backup Database..., 出现备份数据库向导。按照向导中的说明操作即可完成数据库的备份。

如果要恢复数据库, 那么在右边的面板中打开 Utilities 文件夹, 双击 Restore Database, 按照向导中的说明操作即可完成数据库的恢复。

(2) DOS 命令

ASA 8.0 自带一个数据库备份命令文件 dbbackup.exe, 该命令可以完成数据库的备份功能。例如在 DOS 命令模式下, 执行下面的命令能把数据库 (例如 kygl.db) 和日志文件备份到 c:\backup 目录下:

```
dbbackup -c "uid=sa;pwd=sql;dbf=kygl.db" c:\backup
```

如果要恢复数据库, 那么把备份的文件拷贝到初始目录下即可。

(3) SQL 语句

用 SQL 语句也可达到数据库备份的目的。例如我们要把数据库和日志文件备份到 c:\backup 目录下, 在程序中只需编写如下的代码:

```
string ls_backupSql
```

```
ls_backupSql="backup database directory ' c:\backup' "
```

```
execute immediate :ls_backupSql;  
if sqlca.sqlcode <> 0 then  
    messagebox ("提示信息","数据库备份失败") //执行失败  
else  
    messagebox ("提示信息","数据库备份成功") //执行成功  
end if
```

恢复方法和上面相同。

(4) 脱机备份

以上三种数据库备份方法都是在数据库运行状态下执行的，也可以在脱机状态下把数据库拷贝到指定的位置，恢复方法和上面相同。

习 题

1. 如果要通过 ODBC 接口连接到一个数据库，有哪些步骤？

如果要通过专用接口连接到数据库服务器，有哪些步骤？

2. 请创建一个 Adaptive Server Anywhere 数据库 science.db，并用本章所介绍的方法创建§1.4 给出的表，并输入前 5 个表的数据：

(1) 科研人员简历表 (employees)，主键为 em_code，按 em_code 索引。

(2) 科研项目情况表 (items)，主键为 it_code，按 it_code 索引。

(3) 项目参加人员表 (member)，主键为 it_code+em_code，按 it_code+em_code 索引，it_code、em_code 为外部键。

(4) 获奖情况表 (prize)，主键为 prize_code，按 prize_code 索引。

(5) 辅助编码表 (code)，主键为 code，按 code 索引。

(6) 学术论文登记表。

(7) 学术刊物表。

(8) 著作登记表。

(9) 论文参加人员表。

(10) 著作参加人员表。

3. 数据库表的列有哪些扩展属性？各个扩展属性的作用是什么？如何定义和修改这些扩展属性？

4. 根据表 employees, items, member 创建视图 menber2, 取 employees 中的字段 name, sex, position, items 中的 itemname, level, member 中的 order, duty, 连接条件为按各表的主键做自然连接。

5. 用数据库管理画板练习创建用户、修改用户口令和删除用户，定义表的访问权限。

第三章 应用对象

§ 3.1 概 述

应用也称应用程序，一般来讲，一个应用可完成一组相关的功能。在 PowerBuilder 中，应用是由一系列 PowerBuilder 对象组成。这些对象包括：PowerBuilder 的应用对象（Application Object），窗口对象（Window Object），数据窗口对象（DataWindow Object），菜单对象（Menu Object），函数对象（Function Object）以及用户对象（User Object）等。

应用库是 PowerBuilder 提供的一种用于存储 PowerBuilder 应用的文件。这种文件以 .pbl 作为后缀（其含义是 PowerBuilder Library）。这种文件指示应用存放的位置，存储开发的内容（如应用对象、窗口、数据窗口、菜单、函数、用户对象等）。当我们完成了应用开发后，所有应用开发的内容就存放在这样的库文件中了。一个完整的应用既可存放在一个库文件中，也可存放在多个库文件中。

应用对象是用来标识应用、进入应用的入口点。它是 PowerBuilder 应用中包含的众多对象之一，同窗口对象、数据窗口对象、菜单对象、函数对象、用户对象一样存储在应用库中。

应用对象定义了应用的应用级行为。在建立应用时，要定义存储应用的应用库，应用中文本的字体，以及应用开始、终止及出现系统错误时应执行的操作。

PowerBuilder 为应用对象定义了六个可在特定情况下触发的事件，来控制应用在启动、终止及出现系统错误时应执行的操作。当用户启动应用程序时，应用对象中的 Open 事件被触发，可以在这个事件中打开应用的第一个窗口并对系统进行一定的设置；当应用终止时，应用对象的 Close 事件被触发，以完成系统必须的一些清理工作，如关闭数据库或重写初始化文件等；如果应用在执行期间发生系统错误，应用对象的 SystemError 事件被触发，向用户显示出错信息并按应用的要求跳转到特定的位置。

由此可知，应用对象在应用系统中起主程序的作用，用于处理系统级的事件，是应用系统的起点和终点，在运行期间执行状态检测等任务。

所以，应用即为应用系统，应用对象指应用中的一个对象，应用库是应用的物理存储方式。

PowerBuilder 中有以下几种应用程序：

- 执行应用程序：这类应用程序是包括窗口、数据窗口和菜单等内容的集合，一般用于传统的 C/S 模式。应用程序是分布式系统中的客户/服务器程序。
- 服务器应用程序：包含多个用户自定义对象的组件集合，这些对象提供了分布式系统（比如 Jaguar CTS）需要的企业级信息。
- 网络应用程序：在 IE 中运行的一些窗体。PowerBuilder 为我们提供了创建网络应用程序的很多扩展信息，读者可以在以后的工作中灵活调用。

§ 3.2 创建新的应用对象

因为应用对象是进入应用的入口点，所以创建应用的第一步是创建应用对象，同时创建存储应用对象的应用库。创建应用对象之前，首先要创建一个工作空间(Workspace)，它可以方便地组织整个程序结构，具体步骤如下。

1. 创建工作空间

工作空间是创建、修改、调试和运行 PowerBuilder 目标的地方。你可以在一个工作空间中建立多个目标(Target)。

下面是我们建立工作空间的步骤：

- ① 选择 File→New 菜单项，或单击 PowerBar 上的 New 图标。
- ② 在出现的对话框中选择 Workspace 标签页中的 Workspace 项，并单击 OK 按钮。
- ③ 在出现的对话框中输入要保存的工作空间名。

2. 创建新的应用对象

创建新的应用对象的具体步骤如下：

- ① 选择 File→New 菜单项，或单击 PowerBar 上的 New 图标。
- ② 选择 Target 标签页中的 Application 项，并单击 OK 按钮，如图 3-1 所示。



图 3-1 创建应用对象对话框

③ 在出现的对话框中输入应用对象的名称、应用库的名称和目标对象的名称，单击 Finish 按钮，如图 3-2 所示。

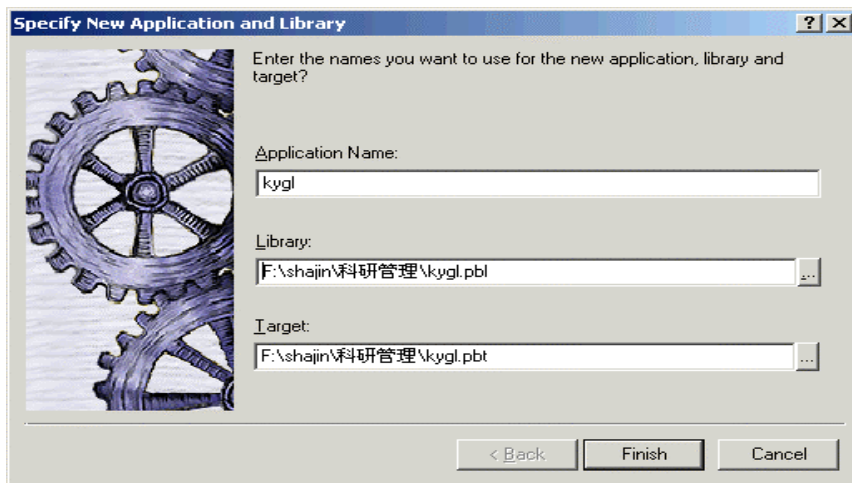


图 3-2 指定应用、库和目标文件

3. 打开应用对象

如何在 PowerBuilder 中打开或者链接别的应用程序, 并进行移植是一个重要的问题。

① 打开新创建的工作空间 kygl。

② 选择 File→New 菜单项, 或单击 PowerBar 上的 New 图标, 选择 Target 标签页中的 Existing Application 项, 并单击 OK 按钮, 如图 3-3 所示。

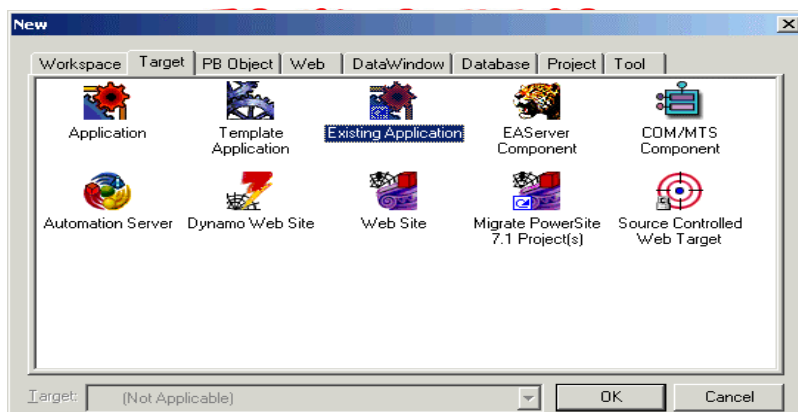


图 3-3 新建目标

③ 在出现的对话框中选择应用对象库的应用对象, 单击 Next 按钮, 如图 3-4 所示。

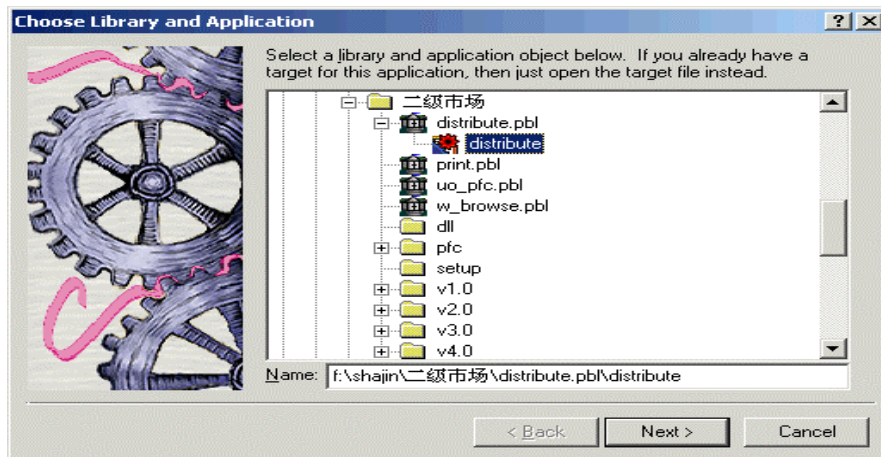


图 3-4 指定应用

- ④ 在出现的对话框中设置应用程序的搜索路径，单击 Next 按钮，如图 3-5 所示。
- ⑤ 在出现的对话框中，单击 Finish 按钮。
- ⑥ 如果打开以前版本的应用，则出现升级移植应用程序对话框，单击 OK 按钮完成移植，这样就可以在 PowerBuilder 9.0 中修改原有的应用程序，如图 3-6 所示。

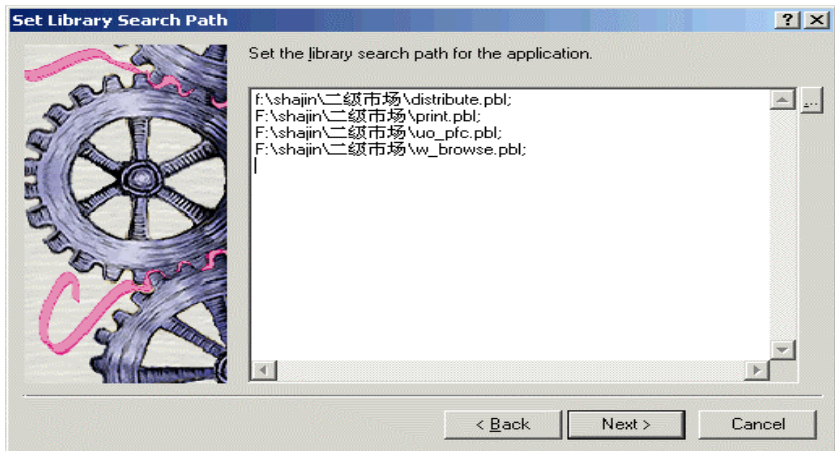


图 3-5 设置搜索路径

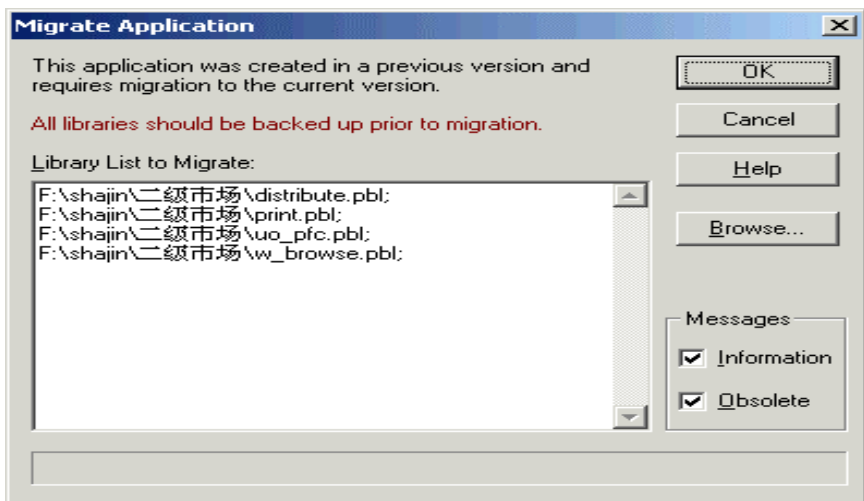


图 3-6 移植应用程序

§ 3.3 指定应用对象的属性

PowerBuilder 通过应用对象的属性设置工作区来指定应用对象的属性，可指定的属性有：

- 默认的文本属性（字体、风格、大小和颜色）。
- 应用的图标。
- 缺省的全局对象。

指定应用属性的具体操作步骤如下：

- ① 在 System Tree 工作区中双击应用对象，在工具栏上单击 Properties 图标，出现属性工作区，如图 3-7 所示。在此工作区中可以设置应用对象的一般属性，各属性的功

能如表 3-1 所示。

② 单击 Additional Properties 按钮，出现附加属性对话框，如图 3-8 所示，各属性页的功能如表 3-2 所示。

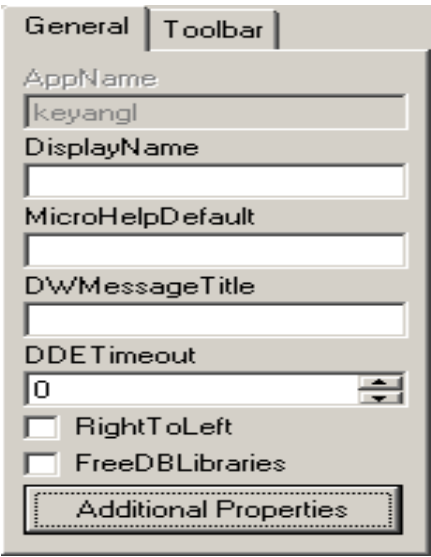


图 3-7 应用对象一般属性

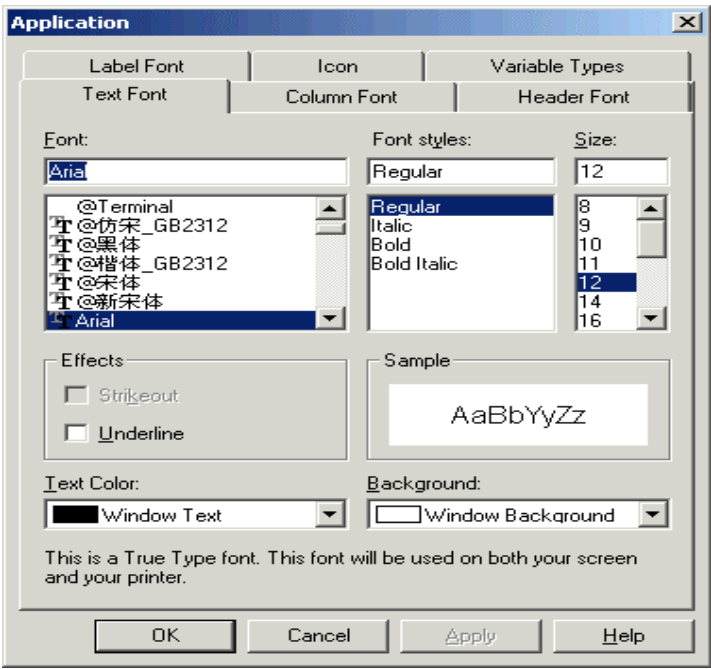


图 3-8 应用对象附加属性

科学出版社
表 3-1 应用对象的一般属性
营销宣传

| 属 性 | 数据类型 | 功 能 |
|----------------------|---------|---|
| AppName | String | 指定应用对象的名称，该属性不能使用代码在运行时修改，它是个只读属性 |
| DWMessageTitle | String | 指定程序运行时 DataApplication 消息对话框的标题 |
| MicroHelpDefault | String | 指定 MicroHelp 对象的默认文本，默认值为 Ready |
| ToolbarPopupMenuText | String | 应用程序工具条菜单位置选项的文本，默认值为 "Left,Top, Right, Bottom, Floating"，你可以将其修改为 "左,上,右,下,浮动"，各菜单项之间使用西文逗号分隔，如图 3-10 所示 |
| ToolbarText | Boolean | 指定工具条上是否显示各工具按钮的文本。有效值是：True: 工具条上显示文本；False: 工具条上不显示文本 |
| ToolbarTips | Boolean | 当工具条按钮上未显示文本时，该属性指定是否显示跟随提示。有效值为：True: 显示跟随提示，即当鼠标指针移动到工具条按钮上，并稍微停留时，显示指示按钮功能的简短提示；False: 不显示跟随提示 |
| ToolbarUserControl | Boolean | 指定用户是否能够使用工具条弹出菜单来隐藏/显示工具条、移动工具条以及显示工具条文本。有效值为：True: 用户可以使用该弹出菜单；False: 用户不能使用该弹出菜单 |

表 3-2 应用对象的属性页

| 属性页 | 功 能 |
|----------------|----------------------------|
| Text Font | 指定显示在窗口对象、用户对象上的静态文本的默认字体 |
| Column Font | 指定数据窗口上录入或显示的数据的默认字体 |
| Header Font | 指定表状格式和网状格式的数据窗口的标题栏中的默认字体 |
| Label Font | 指定自由格式的数据窗口列标签的默认字体 |
| Icon | 指定应用的图标 |
| Variable Types | 指定应用的全局变量 |

1. 指定默认的字体属性

应用对象是一个非可视对象，它的四个字体属性页用于定义应用程序中其他对象使用的缺省字体。把应用对象的字体属性设置成应用程序中使用最频繁的字体可避免为每个对象设置所需字体，从而加快应用程序的开发速度。

下面以指定 Text Font 属性页中的属性为例，说明默认文本属性的指定方法。操作步骤如下：

① 择属性设置对话框中的 Text Font 属性页，显示如图 3-8 所示。该属性页中各个属性的功能如表 3-3 所示。

科学出版社
表 3-3 应用对象的字体属性

| 属 性 | 功 能 |
|-------------|--|
| Font | 指定文本字体，该方框下面的列表框是系统预定的字体，从中选择其一，则该应用的窗口对象、用户对象和数据窗口对象上的静态文本的默认字体为指定的字体 |
| Font Styles | 指定字体的风格，从该方框下面的列表框中选择其一：Regular（标准）、Italic（斜体）、Bold（粗体）、Bold Italic（粗斜体），不指定此项时默认为 Regular |
| Size | 指定字体的大小，从该方框下面的列表框中选择其一，文本按照指定的大小显示 |
| Text Color | 指定文本颜色，从下拉列表框中选择文本的颜色，不指定此项时默认为黑色 |
| Background | 指定背景颜色，从下拉列表框中选择文本的背景色，不指定此项时默认为白色 |

② 根据需要选择文本的字体、风格、大小和颜色。定义完毕后，单击框中的 OK 按钮。

按照同样的步骤，从应用对象属性对话框中选择 Column Font 属性页、Header Font 属性页、Label Font 属性页，定义相应文本的字体、风格、大小和颜色。

2. 指定应用的图标

如果为应用指定了图标，则当用户在执行应用的过程中，Windows 98/NT 状态行或建立快捷方式时会出现这个指定的图标。指定应用的图标的具体操作步骤如下：

① 选择应用对象属性设置对话框中的 Icon 属性页，如图 3-9 所示。

② 在 Icon Name 框中键入图标的文件名（.ico 文件）或使用 Browse 按钮选择 .ico 文件，被指定的图标文件名显示在 Icon Name 框中。



图 3-9 设置应用程序的图标



图 3-10 工具条的弹出式菜单

③ 修改完毕后，单击 OK 按钮，返回到应用对象工作区。

注意：该图标文件指向了具体路径，这将有碍于应用编译后的安装环境（因为安装环境的路径不同），因此，建议去掉路径，将图标放入当前目录下。

3. 指定应用库的查找路径

在各种画板中建立的对象都存储在 PowerBuilder 库中，扩展名为.pbl，PowerBuilder 应用存放在一个或多个库文件中。一个应用的所有库文件都应定义在库的查找路径中。

PowerBuilder 在运行期间使用查找路径来查找引用的对象。当引用一个新的对象时，PowerBuilder 按照库查找路径中指定的顺序查找所有的库，直到找到该对象为止。

定义库查找路径的具体步骤如下：

① 在 System Tree 工作区中选择目标对象，单击鼠标右键，选择 Properties... 菜单项，弹出目标属性设置对话框，如图 3-11 所示。在 Library Search Path 列表框中键入库名，或使用 Browse 按钮将其他路径下的库包含到此路径下。

② 如果要从 Library Search Path 列表框中删除库，只要从 Library Search Path 列表框中选中想要删除的库并按下键盘上的 Delete 键即可。

③ 单击 OK 按钮，应用库的查找路径定义完成。

4. 缺省的全局对象

全局对象是指作用域为整个应用的对象，也就是在应用的任何地方都可使用的对象。PowerBuilder 为所有的应用预定义了五个全局对象，如表 3-4 所示。

表 3-4 全局对象

| 全局对象 | 用 途 |
|---------|---|
| SQLCA | Transaction 对象，用来与数据库通讯 |
| SQLDA | DynamicDescriptionArea 对象，用于动态 SQL |
| SQLSA | DynamicStagingArea 对象，用于动态 SQL |
| Error | Error 对象，用于在执行过程中报告错误信息 |
| Message | Message 对象，用于处理非 PowerBuilder 定义的事件的信息，及对象间传递参数 |

这五个全局对象都有缺省的类型，通常可不修改。需要时开发者也可对其进行修改以方便应用的开发，比如，我们可以定制 Error 对象，使其能够提供中文的出错信息。

修改全局对象的类型的步骤为：

① 使用用户对象（User Object）画板创建标准类用户对象，该类必须继承相应的标准类（比如，想修改 Error 对象的类，必须定义一个继承原 Error 对象标准类的类），具体方法参见第十一章。

② 在应用画板的应用对象属性对话框中选择 Variable Types 属性页，如图 3-12 所示，在相应的编辑框中键入新的类型，单击 OK 按钮，Error 对象定制完毕。

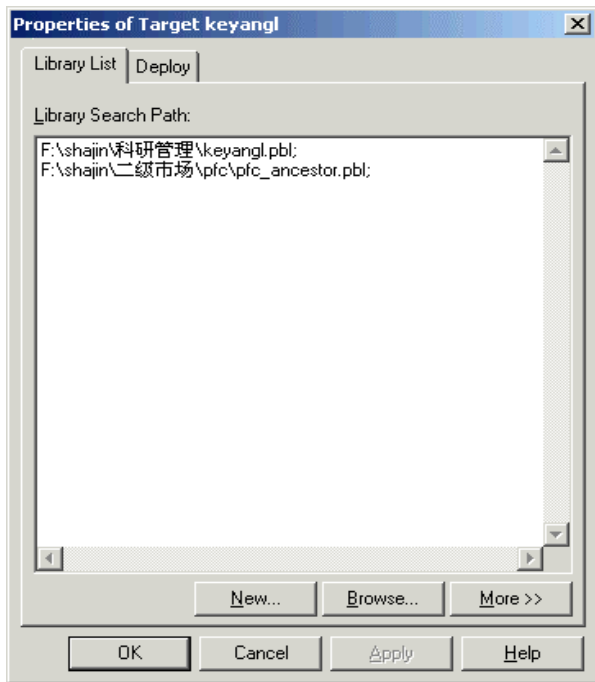


图 3-11 设置搜索路径

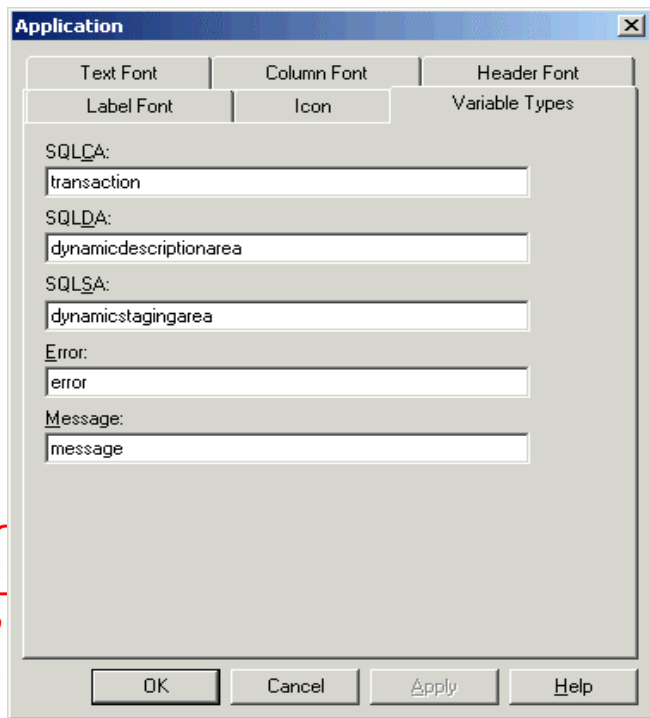


图 3-12 全局对象

§ 3.4 编写应用级的程序

§ 3.4.1 编程环境

应用对象的编程环境包括菜单、工具栏、定义函数、事件、结构、声明变量和外部函数。

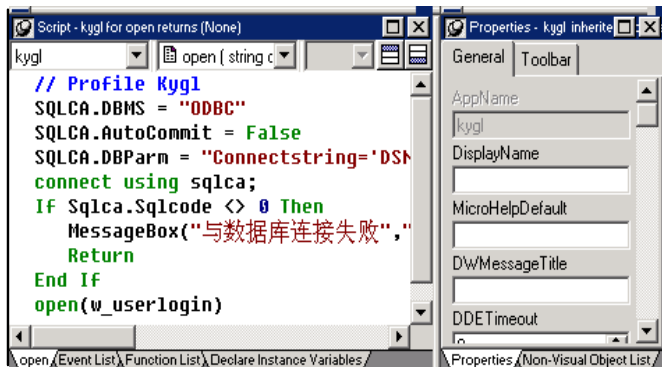


图 3-13 默认布局图

1. 关于两个菜单

从菜单栏的 View 菜单可以打开几种视图：非可视化对象列表、属性视图、事件列表、函数列表、脚本视图、变量视图和结构列表，选择菜单项 View→Layouts→Default 可以设置默认的布局图，如图 3-13 所示。

从菜单栏的 Insert 菜单可以插入对象（如 Connection、DataStore 等）、事件、函数和结构。

2. 关于脚本视图

可以用脚本视图为函数和事件编写代码，自定义函数和事件，声明变量和外部函数。在应用、窗口、用户对象、菜单和函数画板中，脚本视图是默认布局图的一部分。在应用、窗口、用户对象画板中，初始的布局图中有一个脚本视图，它显示此对象事件的脚本，还有一个脚本视图用于声明实例变量。还可以打开多个脚本视图，并在某一个脚本视图中完成编码任务。

（1）标题栏

脚本视图的标题栏显示名称和当前事件或函数的返回类型，也显示当前控件的名称、事件名称和函数列表。如果脚本视图用来声明变量或函数，那么标题栏显示声明类型。

（2）下拉列表

在脚本视图的顶部有三个下拉列表，如图 3-14 所示。

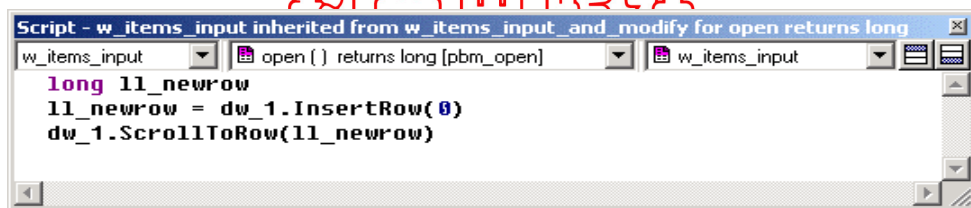


图 3-14 脚本视图的三个下拉列表

在第一个下拉列表中，为了编写对象、控件或菜单项的脚本，可以选择对象、控件或菜单项；为了编写函数脚本，可以选择函数；还能声明变量和外部函数。

在第二个下拉列表中，为了编辑函数或事件，可以选择函数或事件；还能选择要声明的变量的类型。与事件名称相邻的脚本图标指示这个事件有脚本，并且脚本图标的外观能告诉你关于脚本的更多信息，如表 3-5 所示。相同的脚本图标同样显示在事件列表视图中。

表 3-5 脚本图标

| 假如有脚本 | 脚本图标显示 |
|---------------|----------|
| 在当前的对象或控件中 | 白色 |
| 仅在祖先对象或控件中 | 彩色 |
| 在祖先和当前的对象或控件中 | 一半白色一半彩色 |

在派生对象中，第三个下拉列表是可利用的，它列出了当前的对象及其他的所有祖先对象，以便浏览祖先对象的脚本。

(3) 显示/隐藏原型和错误窗口的按钮

当定义一个函数或事件时，原型窗口显示在脚本视图的顶部。当有编译错误时，错误窗口显示在脚本视图的底部。通过列表右边的按钮，能锁定显示这两个窗口。

3. 编辑脚本

从菜单栏选择菜单项 **View→Script**，打开一个新的脚本视图，为函数或事件编写脚本。双击事件列表或函数列表视图中的项，或者从一个脚本视图的第二个下拉列表中选择事件或函数，我们就可以在当前的脚本视图中为事件或函数编写脚本。

从菜单栏的 **Edit→Paste Special** 菜单项或者从弹出式菜单或者从工具栏，可以粘贴函数、SQL 语句、全局变量、共享变量、实例变量、窗口、对象和参数到脚本中。还能用 **AutoScript** 工具，加速编程速度。从菜单栏的 **Design→Option...** 菜单项可以设置关于 **AutoScript** 的一些选项，如图 3-15 所示。还可以为此工具定义快捷键，选择菜单栏的 **Tools→Keyboard→Shotcuts...** 菜单项，在设置快捷键对话框中，找到 **Edit→Activate AutoScript**，假如设置快捷键为 **F8**，那么在脚本视图中，按 **F8** 键就可以激活自动脚本工具。

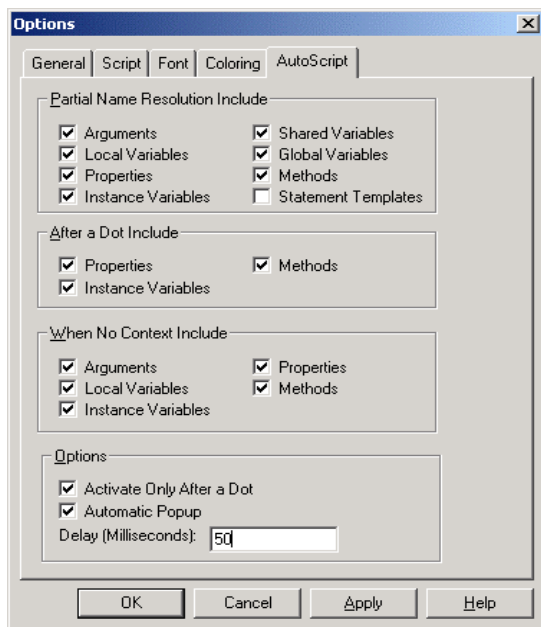


图 3-15 设置 AutoScript

除了通过帮助菜单和 **F1** 键查看帮助外，还可以在脚本视图中用上下文相关的帮助为保留字和内置的函数显示帮助窗口。把光标放置于保留字或内置函数上，按 **Shift+F1** 键，则显示关于保留字和内置函数的帮助信息。可以把帮助文本从帮助窗口拷贝到脚本视图中。

在应用默认的布局图中包含一个声明变量的脚本视图。在第二个下拉列表中选择变量类型或函数类型，在脚本视图中编写声明变量或外部函数。

4. 定义函数、事件和结构

PowerBuilder 有许多内置的函数和事件，根据程序的需要，也可以编写自己的函数和事件，具体的定义过程请参考以后的章节。在刚才介绍的脚本视图中可以编写自定义

函数和事件的脚本。在应用对象中定义的结构属于对象级的结构，在以后的章节中也有详细的介绍。

§ 3.4.2 事件处理程序

在应用执行的过程中，当应用对象上的事件发生时，便执行该事件的事件处理程序。

表 3-6 列出了应用对象的事件、触发时机及其作用。

表 3-6 应用对象的事件、触发时机及其作用

| | |
|-------|--|
| 事件: | Open |
| 触发时机: | 当应用被打开时 |
| 作用: | 此事件的事件处理程序一般打开应用的第一个窗口，设置系统的一些参数（例如，此应用使用哪个数据库管理系统，数据库名，用户名和口令等） |
| 事件: | Close |
| 触发时机: | 当用户关闭应用时 |
| 作用: | 此事件的事件处理程序关闭所有对象（如关闭数据库连接及写预定义文件） |
| 事件: | Idle |
| 触发时机: | 当 Idle 函数被调用且经过此函数规定的秒数后鼠标和键盘仍未动作 |
| 作用: | 在此事件上通常写一些屏幕保护程序或存盘程序 |
| 事件: | SystemError |
| 触发时机: | 当应用执行过程中出现系统错误时 |
| 作用: | 如果没有为这个事件编写事件处理程序，PowerBuilder 显示一个带有错误号和错误信息的信息框；如果有事件处理程序，则 PowerBuilder 执行其上的错误处理程序 |
| 事件: | ConnectionBegin |
| 触发时机: | 用于分布式应用的服务器端。当客户程序试图连接到服务器程序时，此事件在服务器程序应用对象上触发 |
| 作用: | 在此事件上通常写一些提示信息，通知用户连接服务器成功或失败 |
| 事件: | ConnectionEnd |
| 触发时机: | 用于分布式应用的服务器端。当客户程序试图断开与服务器程序的连接时，此事件在服务器应用对象上触发 |
| 作用: | 在此事件上通常写一些提示信息，通知用户断开服务器成功或失败 |

在上面的六个事件中，Open 事件是应用对象中最重要的事件，必须为 Open 事件编写事件处理程序。如果 Open 事件中没有事件处理程序，则系统不知道要打开哪一个窗口，从而无法运行应用。这是因为：

- PowerBuilder 应用是一个基于窗口环境的应用系统，应用在运行时要首先打开该应用的主窗口。
- PowerBuilder 是客户端的开发工具，仅仅开发客户端的软件，该软件运行时还需要与服务器连接，需要知道服务器的一些参数。
- 配置 PowerBuilder 自身运行需要的参数。例如：事务提交模式（是否为 Auto Commit），数据窗口错误信息风格等。

下面举一个 Open 事件的处理程序的例子。该程序的作用是：当应用启动时，设置要连接的数据库的信息，打开应用的主窗口。具体操作步骤如下：

① 打开当前的应用，在 Event List 工作区中，选择 Open 事件，双击鼠标，出现 Open 事件脚本编辑区。在 PowerBar 上单击 DB Profile 图标，选中要连接的数据库配置文件，单击鼠标右键，在出现的对话框中选择 Preview 标签页，如图 3-16 所示，单击 Copy 按钮，将其粘贴到应用的 Open 事件中，如图 3-17 所示。

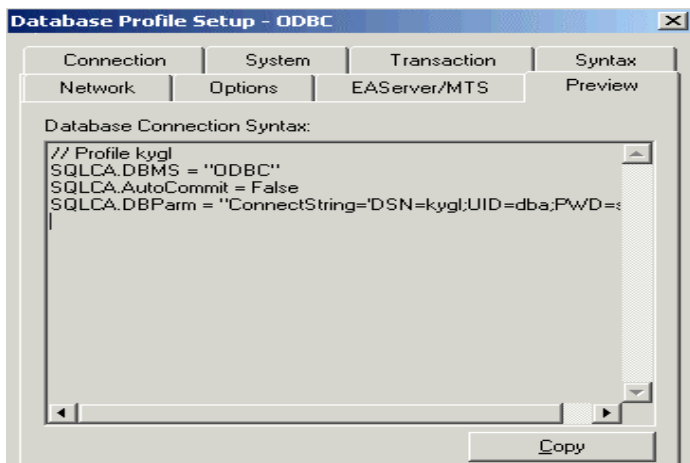


图 3-16 数据库连接语法

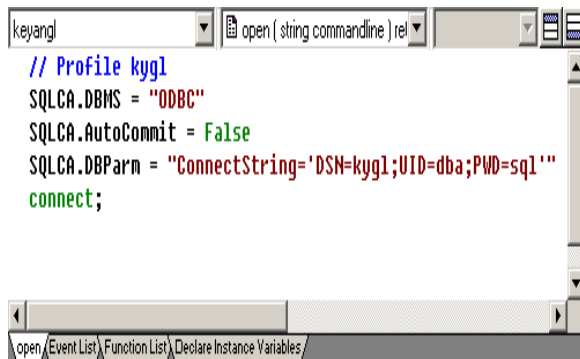


图 3-17 连接数据库代码

② 在这段程序的后面加上下面这条语句：

```
//Open(w_main)
```

该语句表示要打开应用的主窗口 w_main，即应用的第一个窗口。如果此时还未定义 w_main 窗口，则应给此语句加上注释标记（//），否则因为这个窗口不存在，编译无法通过（有关 Open 函数的用法和 Script 语言的语法，请见后面的有关章节）。

在开发实际应用时，SystemError 事件的处理程序也是必不可少的。因为，在应用运行期间，由于各种各样的原因，可能会导致运行错误“Execution-time error”的发生。一旦发生这种错误，将会导致整个应用终止运行。为此，应编写一个程序来管理应用执行期间会遇到的运行错误，告诉错误发生的原因和正确的操作方法，然后继续运行应用。

完成上述功能的具体方法是：在应用级的事件 SystemError（系统错误）下编写处理“Execution-time error”（运行错误）的事件处理程序。在程序中通常采用多分支语句 Choose Case 逐个判断错误的类型号，再根据不同的错误号显示不同的提示信息，以帮助用户改正或处理错误。

如果不为事件 SystemError 编写事件处理程序，当应用遇到运行错误时，会弹出一个 PowerBuilder 本身提供的错误信息显示框，然后终止程序的运行。

习 题

1. 应用对象有哪些属性？怎样设置这些属性？
2. 创建一个应用程序，指定应用的图标、默认的字体等属性，并编写 Open 事件的处理程序。
3. 在编写事件处理程序时，怎样选择事件？怎样粘贴参数、对象、变量？怎样粘贴函数、SQL 语句、分支语句和循环语句？试上机练习。

第四章 窗口及窗口上的控件

窗口（Window）是 Windows 应用程序中最常见，也是最重要的界面元素之一，是 PowerBuilder 中主要的组成部分。用户通过窗口向应用程序提供必要的的数据、指示所需完成的工作；应用程序通过窗口显示运行状态和执行结果。

§ 4.1 窗口的组成及类型

1. 窗口的组成

窗口对象是组成 PowerBuilder 应用程序的主要成分，是应用程序的操作界面，其他对象（菜单、数据窗口及用户对象等）只有加到窗口上，才能显示出来。

用 PowerBuilder 建立的窗口与其他 GUI 环境所使用的窗口是类似的，由以下三部分组成：

- 属性：定义窗口的外观和行为（例如窗口的标题栏、最小化按钮等）。
- 事件：由用户动作触发而进行的动作。
- 控件：放在窗口上，用于请求、输入或显示信息。

创建窗口时，可以定义窗口的属性，也可以编写脚本在程序运行时改变窗口的属性；还可以为窗口的各种事件编写事件处理程序（也称为脚本）以完成需要的操作；通过在窗口中放置各种控件，可以接收用户输入信息，并把信息显示给用户。把控件放到窗口上后，还可以定义控件的属性，为控件的事件编写脚本以完成需要的操作。

2. 窗口的类型

PowerBuilder 提供下列六种类型的窗口。

（1）主窗口（Main Window）

主窗口是一个普通的窗口，可以覆盖其他的窗口，也可以被其他窗口覆盖。任何时候主窗口都可以显示在任何地方，它可以有菜单，能够被最大化、最小化以及改变大小。对功能单一的应用，通常选用主窗口作界面基础。

（2）子窗口（Child Window）

子窗口总是从属于其父窗口，父窗口可以是主窗口或弹出式窗口。

子窗口只能在父窗口中移动，不能把它移到父窗口之外。当把子窗口的一部分移出父窗口时，PowerBuilder 将裁剪掉移出的部分。当父窗口移动时，子窗口随之移动并保持相对位置。关闭父窗口时，子窗口随之关闭。子窗口没有菜单，不能被当作活动窗口。子窗口可以被最大化和最小化，最大化时，它充满整个父窗口而不是整个屏幕，最小化时，它缩小成图标显示在父窗口的底部。

（3）弹出式窗口（Popup Window）

弹出式窗口一般由其父窗口打开，显示在父窗口的上面（即不能被父窗口覆盖），它既能移出父窗口，也能覆盖父窗口。

弹出式窗口可以最大化和最小化，最大化可以充满整个屏幕，最小化图标显示在屏幕的下方。但是，当父窗口最小化时，它也被最小化，当父窗口恢复时，它也恢复成原来的大小。弹出式窗口可以有菜单和标题栏。

（4）响应窗口（Response Window）

响应窗口一般用于显示信息，由其父窗口打开，但是，用户必须关闭该窗口后才能对应用中的其他窗口进行操作。响应窗口不能被最大化和最小化。

应用程序中经常使用的 About 窗口、MessageBox 对话框就是响应窗口应用的具体实例。

（5）MDI 框架窗口（MDI Frame）和带微帮助的 MDI 框架窗口（MDI Frame With Microhelp）

MDI 框架窗口及其子窗口构成了应用程序的多文档界面风格，它有两种形式：普通的 MDI 框架窗口和带微帮助的 MDI 框架窗口，两者的区别在于，后一种框架窗口在底部多了一个状态栏，用于显示应用程序简短的当前信息，我们将其称为微帮助（Microhelp）。

典型的 MDI 框架窗口由标题栏、菜单栏、工具栏、工作区和状态栏组成，如图 4-1 所示。PowerBuilder 9.0 开发环境使用的就是带微帮助的 MDI 框架窗口。

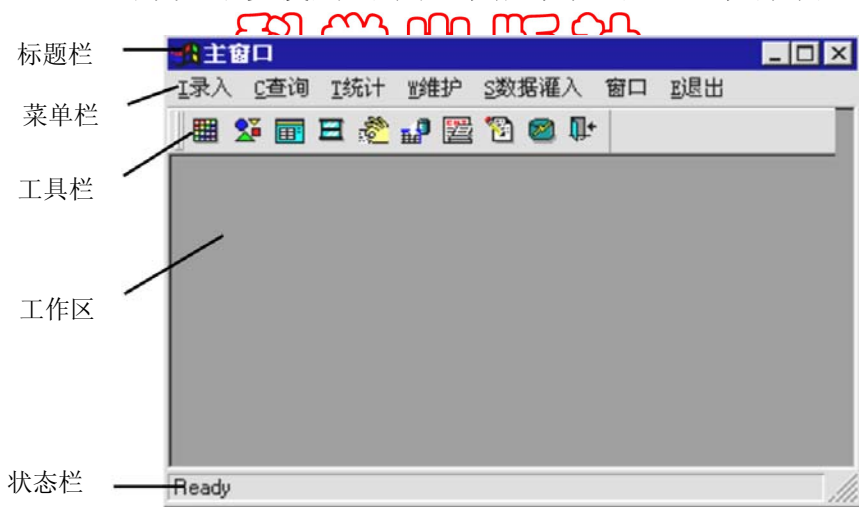



图 4-1 带微帮助的 MDI 框架窗口

§ 4.2 创建窗口

创建窗口有两种方法：创建新窗口和从已有的窗口中继承。本节我们介绍创建新窗口，其操作步骤如下：

- ① 从 File 菜单中选择 New 或者单击 New 工具图标 ，打开 New 对话框。
- ② 在 PB Object 页，选择 Window，如图 4-2 所示。

③ 单击 OK 按钮，出现如图 4-3 所示的 Window 画板工作区。

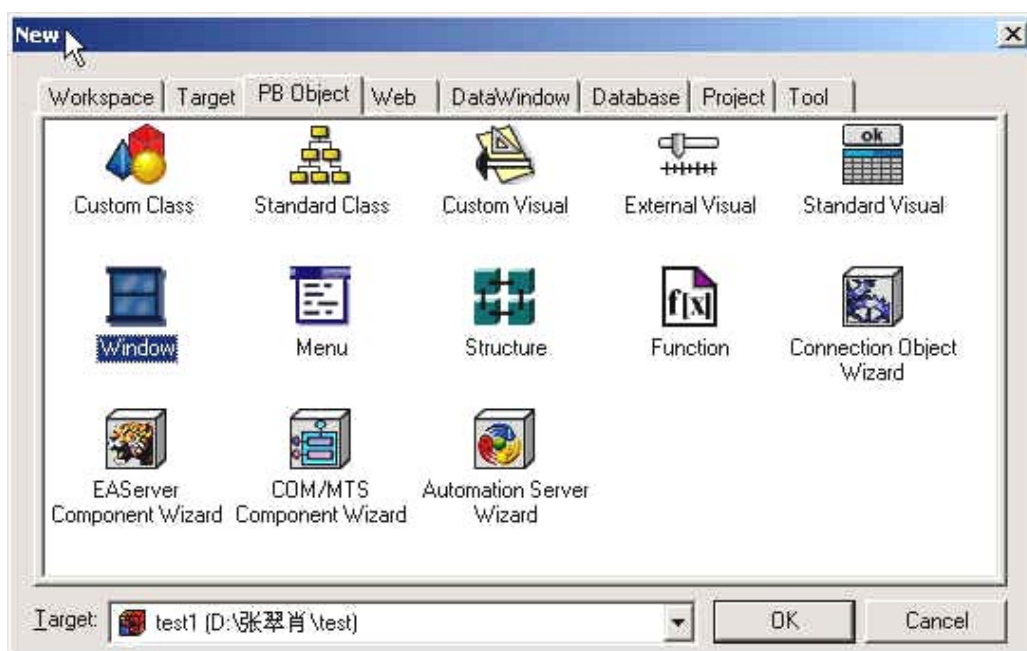


图 4-2 New 对话框

§ 4.2.1 窗口工作区

窗口工作区由菜单栏、PainterBar、StyleBar 以及创建窗口的工作区组成，如图 4-3 所示。

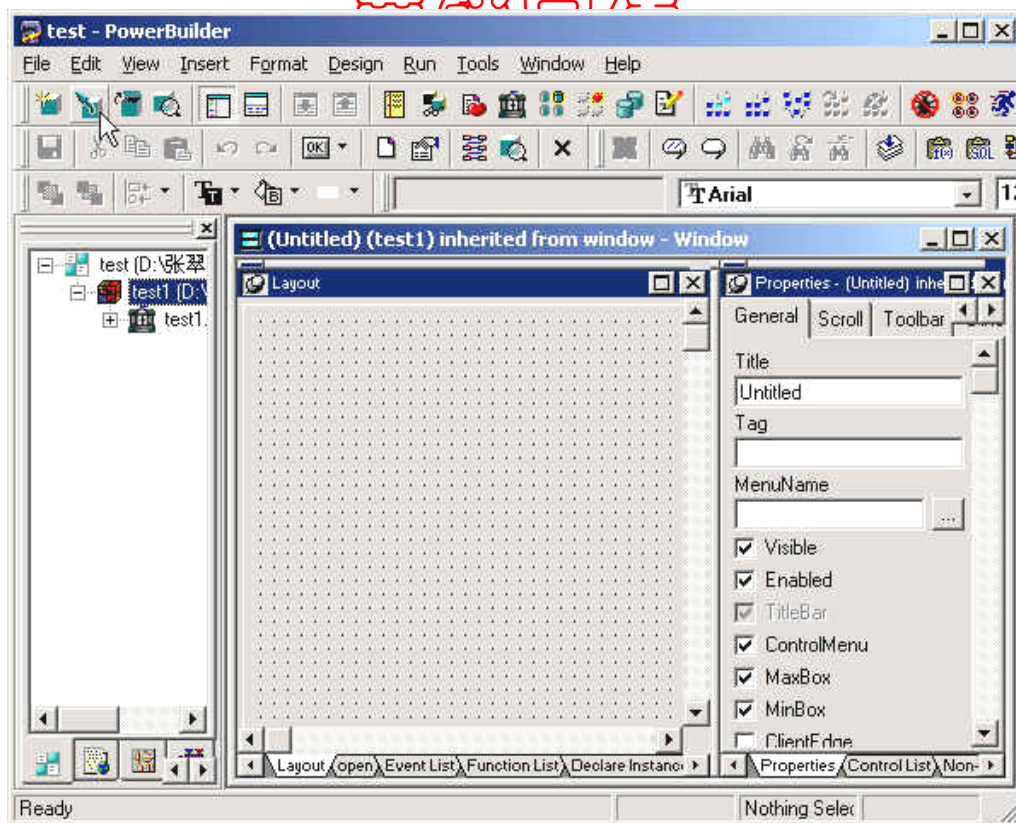



图 4-3 Windows 画板工作区

§ 4.2.2 定义窗口的属性

窗口的属性用于指定窗口的表现形式。窗口的属性可以在设计窗口时通过打开窗口的属性框来定义或修改，也可以在程序运行时动态地修改。

1. 在设计窗口时定义窗口的属性

在设计窗口时定义窗口的属性，是通过打开窗口的属性框为各属性页中的属性赋值实现的。在窗口工作区打开窗口的属性框有以下几种方法：

- 在窗口的空白区域（无控件的区域）中单击鼠标右键，显示弹出式菜单，从中选择 Properties（属性）项。
- 单击工具栏属性图标.

用以上任一方法，都可打开窗口的属性框，如图 4-4 所示。

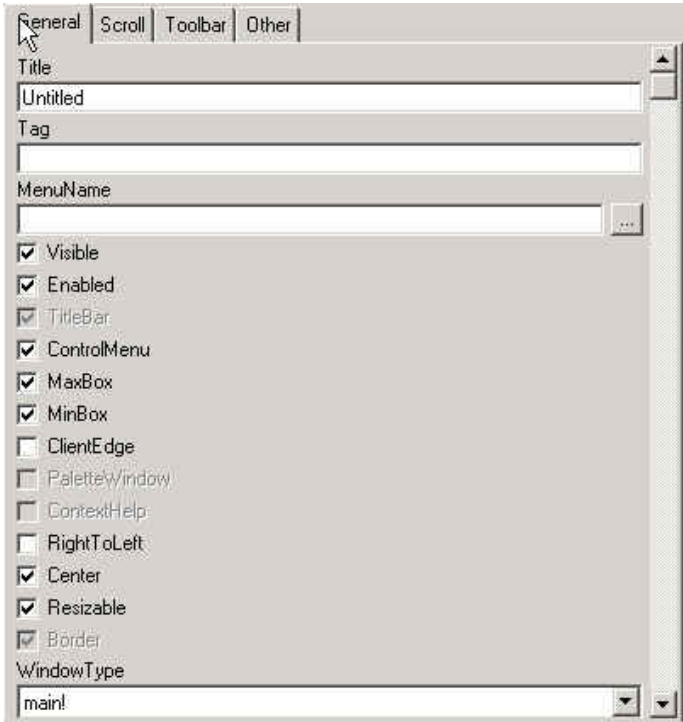


图 4-4 窗口的属性框

图 4-4 中列出了四个属性页，每个属性页中包含一类属性值，详细情况见表 4-1。

表 4-1 窗口的属性页

| 属 性 页 | 属 性 |
|---------|----------------------|
| General | 窗口标题、类型、颜色、风格以及相关的菜单 |
| Scroll | 水平和垂直滚动条位置 |
| Toolbar | 工具栏位置 |
| Other | 窗口的大小、位置和鼠标指针 |

下面详细介绍窗口的属性，以后若窗口中的控件属性与此类似不再介绍。

（1）General 页

General 属性页如图 4-4 所示，表 4-2 是该属性页中各属性的功能。

表 4-2 窗口 General 属性页的属性

| 属 性 | 功 能 |
|---------------------------------|---|
| Title | 编辑定义窗口的标题 |
| MenuName | 用于输入连接到该窗口的菜单名，也可以用 Browse 查找 |
| Tag | 窗口的一些说明信息 |
| Visible | 定义该窗口是否可见 |
| Enabled | 定义该窗口是否可用 |
| ControlMenu MaxBox MinBox | 指定该窗口是否有控制菜单、最大化和最小化按钮。选中表示相应按钮出现，否则不出现 |
| Resizable | 指定该窗口大小是否可变 |
| Border | 指定窗口是否要边框 |
| WindowType | 定义窗口类型（main, child, popup, response, mdi） |
| Back Color | 定义窗口的颜色 |
| MDIClientColor | 定义 MDI 框架窗口中客户区域的颜色 |
| Icon | 定义窗口图标 |

设置一般属性时应注意以下几点：

- 主窗口必须有标题栏。
- 子窗口不能有菜单。
- 响应式窗口不能有菜单、最大化框和最小化框。

（2）Scroll 页

该属性指定本窗口的滚动属性，如图 4-5 所示，各个属性的功能如表 4-3 所示。



图 4-5 窗口属性的 Scroll 页

表 4-3 窗口 Scroll 属性页的属性

| 属 性 | 功 能 |
|--------------|--|
| HScrollBar | 设置横向滚动条 |
| VScrollBar | 设置纵向滚动条 |
| UnitsPerLine | 设置用户单击纵向滚动条中的上箭头或下箭头时，向上或向下滚动的距离。若为 0，则滚动窗口的 1/100 |

续表

| 属 性 | 功 能 |
|----------------|--|
| UnitsPerColumn | 设置用户单击横向滚动条中的左箭头或右箭头时，向左或向右滚动的距离。若为 0，则滚动窗口的 1/100 |
| ColumnsPerPage | 当用户单击横向滚动条时窗口滚动的列数。若为 0，则滚动 10 列 |
| LinesPerPage | 当用户单击纵向滚动条时窗口滚动的行数。若为 0，则滚动 10 行 |

(3) Toolbar 页

在 Toolbar 属性页中设置窗口工具栏的初始位置，该属性页如图 4-6 所示，各个属性的功能如表 4-4 所示。

表 4-4 窗口 Toolbar 属性页的属性

| 属 性 | 功 能 |
|-----------------------------|------------------------|
| ToolbarVisible | 指定工具条是否可见 |
| ToolbarAlignment | 指定工具栏显示位置 |
| ToolbarX, ToolbarY | 指定工具条的位置：X 为横坐标，Y 为纵坐标 |
| ToolbarWidth, ToolbarHeight | 指定工具条的宽度和高度 |

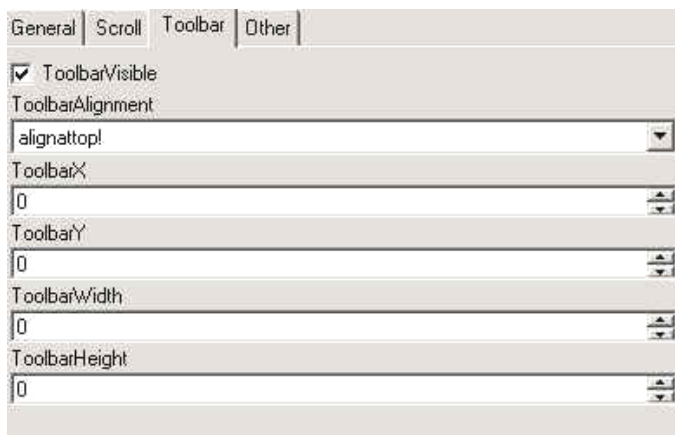


图 4-6 窗口属性的 Toolbar 页

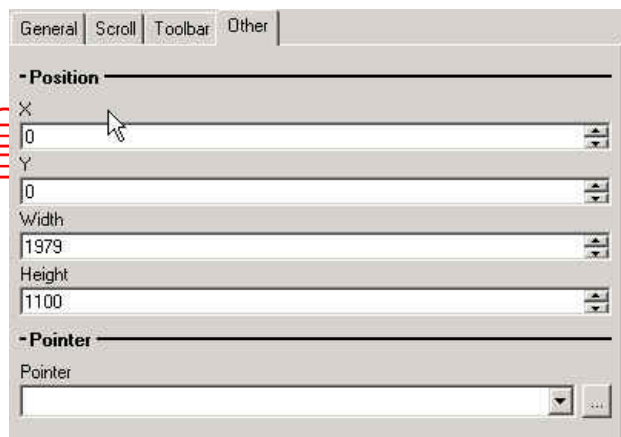


图 4-7 窗口属的 Other 页

(4) Other 页

Other 页主要用来设置窗口的位置、大小和鼠标指针，该属性页如图 4-7 所示，各个属性的功能如表 4-5 所示。

表 4-5 窗口 Other 属性页的属性

| 属 性 | 功 能 |
|---------------|--------------|
| X, Y | 指定窗口左上角的位置 |
| Width, Height | 分别指定窗口的宽度和高度 |
| Pointer | 指定窗口鼠标指针的形状 |

2. 在程序中修改窗口的属性

窗口的属性可以在程序中动态地修改，其方法是在事件处理程序中用语句给窗口的属性赋值。语句为：窗口名.属性=属性值。

例如：在图 4-8 所示的窗口 w_1 中，窗口的标题为“标题一”，有一个命令按钮控件 cb_1，此命令按钮的显示文本为“改变窗口的标题为：标题二”。

cb_1 的 Clicked 事件处理程序为：w_1.title=" 标题二"，功能是：在程序运行时，用户用鼠标单击命令按钮 cb_1，就会改变窗口 w_1 的标题为“标题二”。如图 4-9 所示。



图 4-8 初始的 W_1 窗口




图 4-9 按下按钮后的 W_1 窗口

§ 4.2.3 在窗口上放置控件

创建新窗口时，除了要设置窗口的属性外，还要在窗口上放置控件，只有这样窗口才能和用户进行交互，即接收用户的输入信息，并把信息呈现给用户。把控件放置到窗口上后，还可以定义控件的属性及对控件编程。详见 § 4.3。

§ 4.2.4 窗口的保存

应随时保存正在创建或修改的窗口。保存窗口的方法如下：

① 从菜单栏中选择 File→Save，也可单击画板 PainterBar 工具栏上的 Save 图标 ，如果以前保存过窗口，则 PowerBuilder 会在同一个库中保存此新版本并返回到 Window 工作区。如果以前没有保存过，PowerBuilder 会显示如图 4-10 所示的 Save Window 对话框。

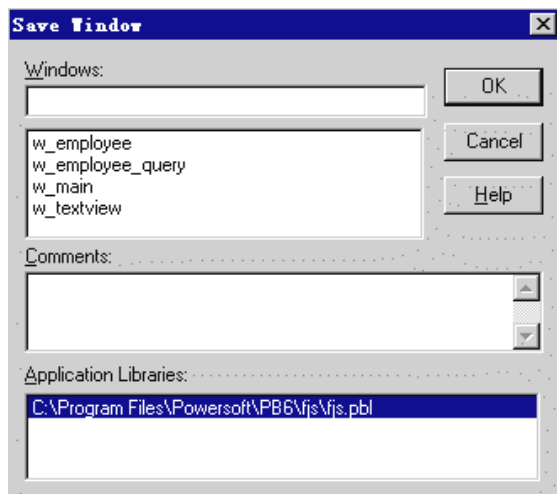


图 4-10 Save Window 对话框


- ② 从列出的库中选择一个要保存该窗口的库。
- ③ 在 Windows 文本框中为该窗口命名。
- ④ 在 Comments 文本框中输入注释文本进行说明。
- ⑤ 单击 OK 按钮。

§ 4.2.5 查看定义好的窗口

1. 预览窗口

创建窗口的过程中，可以从 Window 画板中预览它的外观，使用户对其有一个直观的印象。但是在预览时，不能修改窗口的属性，不能触发事件，也不能连接数据库。

要预览窗口，可以选择菜单栏上的 Design→Preview，或单击窗口工具图标 ，该窗口就以其定义的属性出现在屏幕上。

要退出预览状态，可再一次选择菜单栏上的 Design→Preview，或单击 。

2. 打印窗口

选择菜单栏的 File→Print 项，可以将一个窗口的描述、控件以及它们的属性和事件处理程序打印出来。

3. 运行窗口

在开发过程中，可以测试单个窗口而不必运行整个应用程序，其具体操作步骤如下：

- ① 单击 PowerBar 上的工具图标 ，弹出 Run 对话框，如图 4-11 所示。

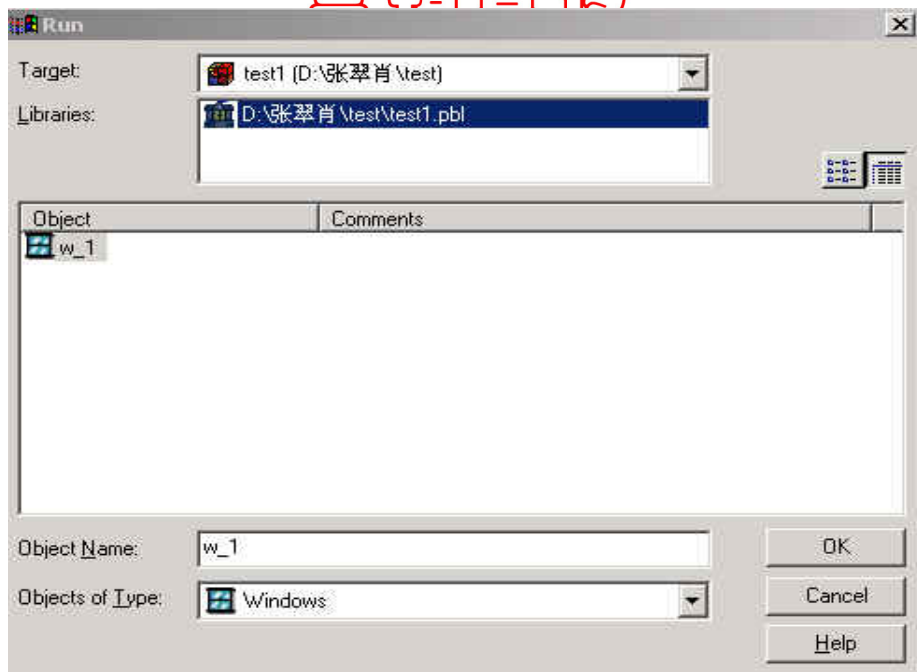


图 4-11 Run 对话框

- ② 从列表框中选择将要运行的窗口，然后单击对话框中的 OK 按钮，开始运行该窗口。

与预览窗口不同，在运行窗口时，可以触发事件、打开其他的窗口、连接数据库等，


但是此时窗口不能访问用户自定义的全局变量。另外，若窗口运行出错也不会触发 `SystemError` 事件，因为 `SystemError` 事件是应用对象的事件。因此用此方法运行窗口有一定的局限性，不能与应用的运行一致。

§ 4.2.6 窗口对象的编程

窗口对象本身有一些触发事件，包括窗口被打开时触发的 `Open` 事件和窗口关闭时触发的 `Close` 事件。

同样，窗口上的控件也有一些事件，所以，编程时要注意打开正确的编程窗口，明确是在为哪个对象的哪个事件编程。

下面介绍窗口本身的编程方法：

① 单击窗口的空白区（确保没有选中窗口上的控件），然后单击 **Script**（编程）图标 ，此时，显示窗口的编程窗口，如图 4-12 所示。

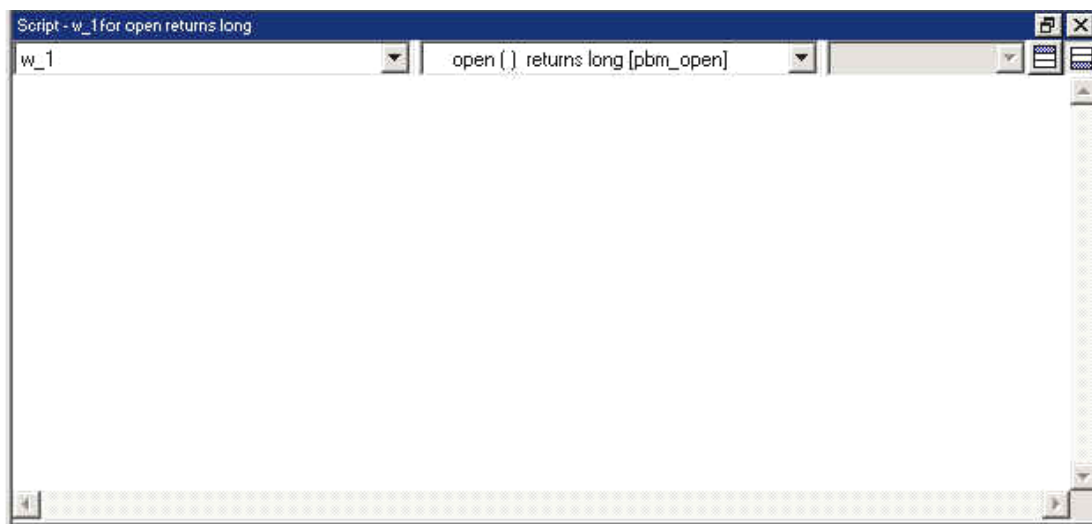


图 4-12 窗口的编程脚本视图

② 从 **Select Event** 下拉列表框中选择相应的事件。

③ 在编程工作区为该事件编写事件处理程序。

在编写事件处理程序中，当引用窗口的属性时，用点号来指示窗口对象及其属性。例如使用下列语句将一个名为 `w_subject` 窗口的标题改为“维护职称表窗口”：

```
w_subject.Title="维护职称表窗口"
```

在编写事件处理程序过程中，可以定义窗口对象的自定义事件、函数、结构、变量等，具体内容参见有关编程方法的章节。

④ 编写完后存盘即可。

§ 4.3 窗口上放置的控件

在窗口上可以放置图 4-13 所示的控件，不同的控件有不同的功能和用途，关于每个控件的详细介绍见 § 4.4。



图 4-13 窗口上的控件

§ 4.3.1 在窗口上放置控件

在窗口上放置控件的方法如下：

① 单击窗口工具条中的控制按钮下拉箭头 ▼，弹出的下拉列表框中显示了所有控件的图标（图 4-13）。


② 从中选择相应的控件图标，返回窗口工作区，在窗口上要放置控件的位置上单击鼠标左键，控件就显示在该位置。

注意：可以使用 Ctrl+T 键来创建多个同样的控件。方法是用鼠标选中已创建好的控件，按 Ctrl+T 键，即在此控件下面创建好一个同样的控件。

§ 4.3.2 设置控件的属性

与设置窗口的属性相同，设置控件的属性也是通过设置其属性对话框中的属性来实现的。

打开控件的属性对话框有以下几种方法：

- 选中放置在窗口上的控件，单击鼠标右键，显示弹出式菜单，从中选择 Properties（属性）项。
- 选取窗口上的控件（控件的四角显示黑点），再选取画板工具条上的属性图标 , 以 CommandButton（命令按钮）控件为例，其属性对话框如图 4-14 所示。

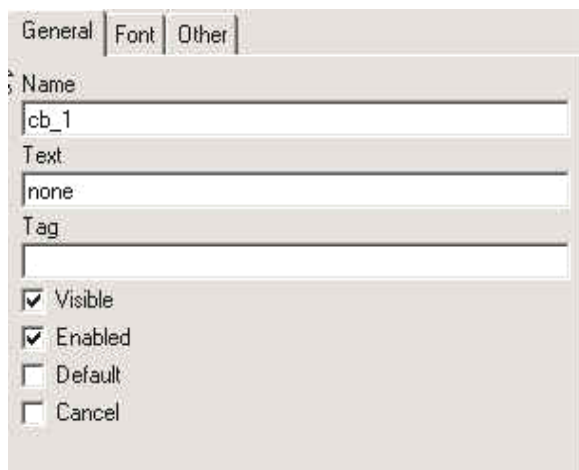


图 4-14 命令按钮的属性框

下面介绍大多数控件的通用属性。

1. 设置控件名字

控件的名字用于标识该控件并通过名字对其进行引用。在 PowerBuilder 中，对各控件都有一个缺省的名字，如图 4-14 中的命令按钮控件的名字为 cb_1。在我们的应用中最好也用系统规定的名称前缀。如静态文本都以 st_ 开头，命令按钮都以 cb_ 开头。表 4-6 给出了所有控件的缺省前缀。

表 4-6 各控件的缺省前缀

| 控 件 | 名 字 |
|----------------------------------|--------|
| 命令按钮 (CommandButton) | cb_ |
| 图片按钮 (PictureButton) | pb_ |
| 复选框 (CheckBox) | Cbx_ |
| 单选钮 (RadioButton) | rb_ |
| 静态文本 (StaticText) | st_ |
| 静态超链接 (StaticHyperLink) | Shl_ |
| 图片框 (Picture) | p_ |
| 图片超链接 (PictureHyperLink) | Phl_ |
| 组框 (GroupBox) | gb_ |
| 线段 (Line) | ln_ |
| 椭圆 (Oval) | oval_ |
| 矩形控件 (Rectangle) | r_ |
| 圆角矩形 (RoundRectangle) | rr_ |
| 单行编辑框 (SingleLineEdit) | sle_ |
| 编辑掩码控件 (EditMask) | em_ |
| 多行编辑框 (MultiLineEdit) | mle_ |
| 超文本编辑框 (RichTextEdit) | rte_ |
| 水平滚动条 (HscrollBar) | hsb_ |
| 垂直滚动条 (VscrollBar) | vsb_ |
| 水平进度条 (HprogressBar) | hpb_ |
| 垂直进度条 (VprogressBar) | Vpb_ |
| 水平轨迹条 (HtrackBar) | htb_ |
| 垂直轨迹条 (VtrackBar) | vtb_ |
| 下拉列表框 (DropDownListBox) | ddlb_ |
| 下拉图片列表框 (DropDownPictureListBox) | ddplb_ |
| 列表框 (ListBox) | lb_ |
| 图片列表框 (PictureListBox) | plb_ |
| 列表浏览控件 (ListView) | lv_ |
| 树形控件 (TreeView) | tv_ |
| Tab 控件 (Tab) | tab_ |
| 数据窗口控件 (DataWindow) | dw_ |

续表

| 控 件 | 名 字 |
|----------------------|------|
| 统计图 (Graph) | gr_ |
| OLE 控件 (OLE) | ole_ |
| 用户对象控件 (User Object) | uo_ |

2. 设置控件标题

控件标题就是显示在控件上的文本。大多数控件都有显示文本，用户可以修改标题文本、字体字型及对齐方式等，其方法如下：

① 选中控件。

② 在窗口画笔风格栏 (Style) 左边的“标题编辑框”中输入标题文字，在“字体”、“字号”列中选择相应的字体字号，需要加黑、采用斜体和加下划线的可以选择相应按钮，再选择相应的对齐方式。如图 4-15 所示。

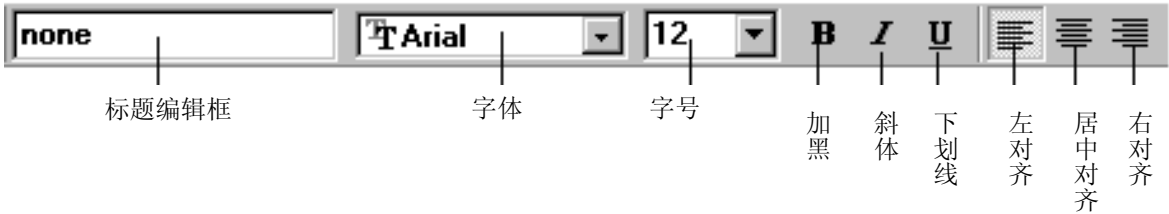


图 4-15 设置控件标题的工具栏

③ 在窗口的工具栏上单击背景图标和前景图标可以选择背景色和文本颜色，如图 4-16 所示。



图 4-16 设置背景色和文本颜色

也可以在控件属性页的 General 页和 Font 页中进行设置。

3. 控件的位置、大小

控件的位置和大小可以通过其属性框的 Other 页来具体设置其横坐标、纵坐标、高度和宽度（与窗口的设置方法相同），也可以使用鼠标和键盘来移动控件的位置和改变控件的大小。下面介绍后一种方法。

(1) 用鼠标

位置：用鼠标拖动并将其移动到理想位置。

大小：用鼠标抓住控件的边界并拖动来改变大小。

(2) 用键盘

位置：通过选择控件并按相应方向的箭头键来移动其位置。

大小：选择控件，并按住 Shift+相应箭头键来改变其大小。各箭头键的作用如下：

→：宽， ←：窄， ↓：低， ↑：高

4. 定义加速键

可以为控件定义加速键从而允许用户通过按 Alt+加速键切换到相应的控件。如何定

义加速键取决于该类控件是否有与之关联的文本。下面分别讲述。

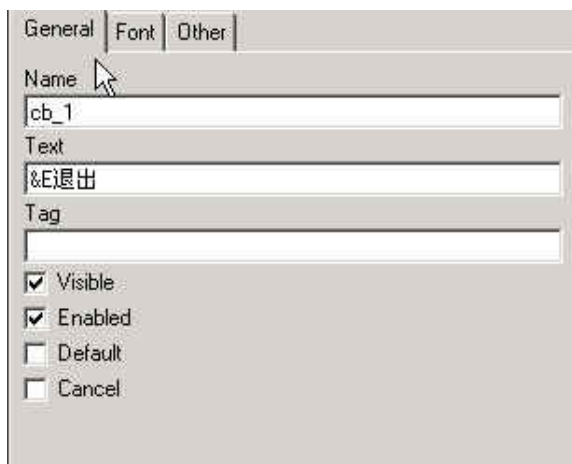


图 4-17 命令按钮属性的 General 页

(1) 为与之有关联的文本控件（如 CommandButton）定义加速键。

以命令按钮（CommandButton）为例，定义 Alt+E 为加速键。具体步骤如下：

① 选择命令按钮控件属性的 General 页，显示如图 4-17 所示。

② 在显示文本中，在定义加速键的字符前加上“&”，PowerBuilder 显示下划线以指示该加速键。

(2) 为与之没有关联的文本的控件指定加速键。

以 SingleLineEdit（单行编辑器）为例，具体步骤如下：

① 选择单行编辑器控件属性的 General 页，显示如图 4-18 所示。

② 在 Accelerator 框中键入加速键（只需键入 Alt 后的字母，例如要定义 Alt+i 为加速键，只需键入字母 i）。

③ 单击 OK 按钮返回到窗口工作区，加速键已经定义，但是用户并不知道，还需要为之做标签。

④ 在控件旁边放一个静态文本控件，用来标识该加速键。

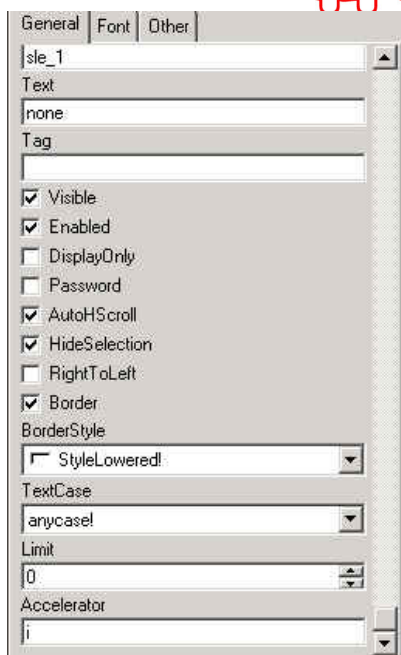


图 4-18 单行编辑器控件属性的 General 页

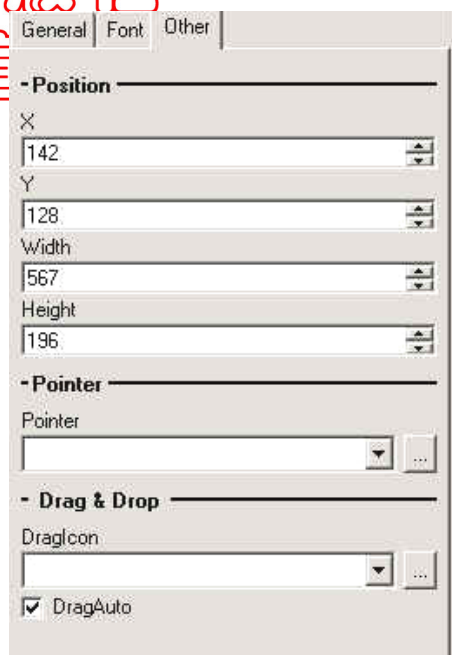


图 4-19 单行编辑器控件属性的 Other 页

5. 定义控件的鼠标指针



控件的鼠标指针是指在运行时，当鼠标放在控件上时鼠标的指针形状。

通过设置控件的 Pointer 属性可定义控件的鼠标指针，其界面和操作方法与窗口的鼠标指针相同。

6. 指定控件的拖拉图标

通过设置控件的 Drag and Drop 属性进行指定, 如图 4-19 所示。

在指定控件的拖拉图标时, 必须先选中 DragAuto 复选框。因为只有选中该复选框时, 此控件才是可拖动的, 为其定义的拖动图标才有效。

可键入图标文件名, 也可以单击后面的按钮  进行查找, 或从 DragIcon 下面的下拉箭头  中选择 PowerBuilder 提供的图标。

§ 4.3.3 窗口上控件的调整

1. 选中窗口上的控件

(1) 选中一个控件

单击该控件, 控件的四周出现黑点, 表示该控件被选中。

(2) 选中多个控件

单击要选中的第一个控件, 按下 Ctrl 键不放, 再单击其他控件, 直到所需的控件都被选中。

若这些控件相邻, 可以先在此控件区域的一角单击, 然后拖动鼠标, PowerBuilder 显示一个约束框, 当所有的控件都在此框中时, 释放鼠标, 则选中此区域的所有控件。


2. 调整窗口上的控件

注意: 所有的调整都有一个标准, 如: 对齐都以第一个控件为标准, 调整距离都以前两个控件的距离为标准。

具体方法如下:

① 首先选中作为标准的控件。


② 按住 Ctrl 键不放, 再单击其余需要调整的控件。

③ 单击窗口工具栏中的对齐图标的下拉箭头 , 显示图 4-20 所示的下拉列表, 从中选择所需的对齐方式的图标即可。

3. 定义光标在控件上的跳转顺序

光标的跳转顺序是指在窗口的执行过程中, 当用户按 Tab 键时, 焦点从一个控件移到另一个控件的顺序。可定义控件的 Tab 值, 当用户按 Tab 键时, 光标按照 Tab 值从小到大的顺序从一个控件跳转到另一个控件循环。

定义光标的跳转顺序的具体步骤如下:

① 单击窗口的 PainterBar 工具栏 Tab Order (跳转顺序) 图标 , 或从菜单栏选择 Format→Tab Order, 如图 4-21 所示, 显示当前的跳转顺序。

② 使用鼠标将插入点移到要修改 Tab 值的控件上。

③ 输入新的 Tab 值 (0~9999); 0 代表从 Tab 顺序中除去该控件 (跳过该控件)。

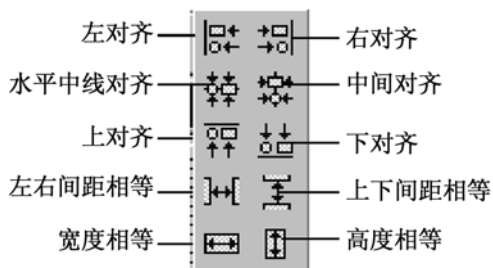


图 4-20 控件布局下拉工具栏

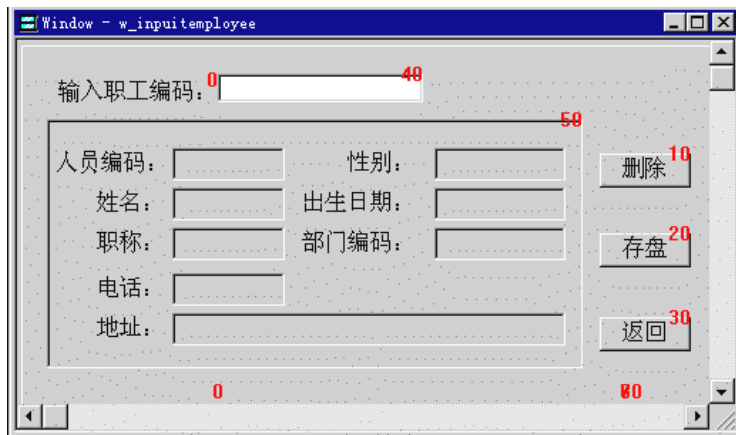


图 4-21 控件的 Tab 顺序

④ 重复以上步骤，直到得到所有的 Tab 顺序。

§ 4.3.4 控件的编程

窗口上的每个控件都有相应的事件，用户可根据需要编写事件处理程序。方法如下：

① 选择要编程的控件。

② 单击鼠标右键，从弹出菜单中选择 Script 项，或单击图标 ，进入控件的编程工作区。

③ 选择相应的事件，并编写事件处理程序。

科学出版社
营销宣传

§ 4.4 控件的使用

在窗口上可以放置的控件种类很多，根据其功能可分成四种基本类型：

- 激活动作类型（按钮类型）：这些控件包括 CommandButton 和 PictureButton。
- 列出选择项类型：这些控件包括 RadioButton 和 CheckBox。可以用 GroupBox 控件对这些控件进行分组。
- 显示数据类型：这些控件包括 ListBox、PictureListBox、DataWindow、DropDownListBox、DropDownPictureListBox、StaticText、StaticHyperLink、Graph、ListView、TreeView、RichTextEdit、Picture、PictureHyperLink、SingleLineEdit、MultiLineEdit、EditMask 以及 OLE 控件。

以上三类是带事件的控件，通过为其事件编写事件处理程序来决定当控件的事件被触发时所应执行的处理。

- 修饰类型：这些控件是绘图对象，包括 Line、Rectangle、RoundRectangle、Oval。这些控件不带事件，不需要编写事件处理程序。

下面对这四类控件的使用逐一进行介绍。

§ 4.4.1 激活动作类型（按钮类型）

这种类型包括命令按钮（CommandButton）和图片按钮（PictureButton），都是用来

激活动作的控件。将执行某一动作的事件处理程序写在该控件的事件中，用户通过触发该事件来执行这一动作。

1. 命令按钮

命令按钮控件除了显示文本、是否可见、是否可用等一般属性外，还有下列两种属性：选中 **Default** 复选框，表示该按钮是缺省按钮，程序运行时，即使焦点处于其他控件上，用户按下回车键，也会触发该按钮的 **Clicked** 事件；选中 **Cancel** 复选框，表示该按钮成为当前窗口的取消按钮，用户按下 **Esc** 键将触发此按钮的 **Clicked** 事件。

在使用命令按钮时，应按其所执行的功能，设置此按钮的属性，并为此按钮的事件编写事件处理程序。例如，在职工详细情况录入窗口 `w_inputemployee` 中，设置了三个按钮（如图 4-22 所示），分别实现删除、存盘和退出功能。



图 4-22 职工详细情况录入窗口 `w_inputemployee`

这三个按钮的程序如表 4-7 所示。

表 4-7 删除、存盘、返回按钮的程序代码

| 对 象 | 事 件 | 程 序 代 码 |
|---------------|---------|---|
| cb_delete（删除） | Clicked | <pre>dw_1.deleterow(0) if update(dw_1)=1 then commit; else rollback; end if</pre> |
| cb_save（存盘） | Clicked | <pre>if update(dw_1)=1 then commit; else rollback; end if</pre> |
| cb_exit（返回） | Clicked | <pre>close(parent)</pre> |

2. 图片按钮

图片按钮控件的功能与命令按钮类似，区别在于图片按钮可以在该按钮上显示 **BMP**、**RLE** 或 **WMF** 格式的图片，而且能以不同的图片表示按钮处于允许和不允许两种状态。

注意：在同一个窗口中，不要把多个按钮指定为缺省按钮或取消按钮。因为设置多个后，只有最后一个起作用。指定多个只能使自己困惑不解。

§ 4.4.2 列出选择项类型

这种类型包括单选钮、复选框和成组框。单选钮和复选框是用来表示某选项的状态，而不用于激活动作。单选钮经常和成组框联合使用。

1. 成组框（GroupBox）

成组框是个简单的控件。它本身不做任何事情，只是把其他控件按功能分成组，使用户界面更加清晰，也易于理解。

2. 单选钮（RadioButton）

单选钮控件是表示彼此排斥的圆按钮，它们一般存在于一个成组框中，每个组框中只能有一个被选中，当选中一个单选钮后，原来选中的就变成不再选中了。

单选钮控件的属性如图 4-23 所示。



图 4-23 单选钮控件的属性

3. 复选框（CheckBox）

与单选钮控件相似，复选框控件通常也只用来表示某选项的状态，而不用于激活动作。它们也常放在成组框中。但是与单选钮不同，无论它们是放在成组框中还是其他位置，它们彼此之间是相互独立的，选中一个，不会对其他的产生任何影响。

复选框的属性与单选钮的属性类似，在此不再赘述。

通常在应用中，当用于互相排斥的选项时，使用单选钮控件；而用于同时选择多个，各项之间关系独立时，使用复选框控件。

例如：在图 4-24 所示的窗口中，要对某个单位的人员进行查询，各单位之间是彼此排斥的，所以采用了三个单选按钮控件和一个成组框控件。而查询人员的信息（姓名、性别、出生年月、联系电话）之间是相互独立的，没有排斥关系，所以采用了四个复选框和一个成组框。图 4-24 是查询电子系人员的姓名和性别的情况，选中了一个单选钮和两个复选框。

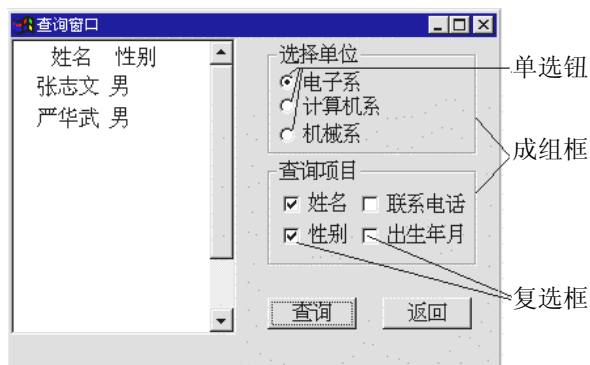


图 4-24 单位人员查询窗口

§ 4.4.3 文本显示和编辑控件

文本一直是最常用的输入输出形式。尽管现在可以使用图形界面，但文本在用户界面中仍然充当着重要的角色。下列的控件可以进行文本的输入输出。

1. 静态文本 (StaticText) 控件

静态文本控件主要用来向用户显示信息，或用来说明某些控件的功能。例如，在单行编辑框前面加一个静态文本来说明应该向单行编辑框中输入什么样的数据。它的属性如图 4-25 所示。



图 4-25 静态文本控件的属性

静态文本控件也有事件，但很少对它编程。

2. 单行编辑器（SingleLineEdit）控件

单行编辑器是用户用来输入一行文本的方框，它也常常被称为编辑框。单行编辑器一般用于数据的输入输出。其属性可以进行设置，如图 4-26 所示。

静态文本和单行编辑器都可以用来向用户显示数据，不同点在于在执行过程中，静态文本的内容不可修改，而单行编辑器的内容可以修改，可以在单行编辑器上输入数据和修改数据。

例如：图 4-22 中的“输入职工编码”就是一个静态文本控件。其后的方框是一个单行编辑器控件。

3. 多行编辑器（MultiLineEdit）控件

多行编辑器控件的功能基本上与单行编辑器相同，不同点在于单行编辑器只能输入一行，而多行编辑器可以输入多行。其属性与单行编辑器也基本相同，在此不再赘述。



图 4-26 单行编辑器控件的属性

例如：可以在多行编辑器中显示职工的个人简历，如图 4-27 所示。

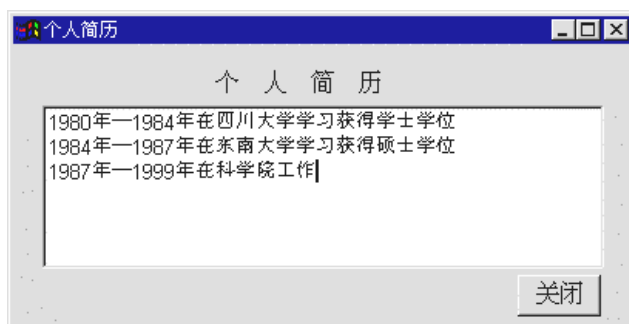


图 4-27 多行编辑器控件

4. 编辑掩码 (EditMask) 控件

编辑掩码控件是一种灵活的、多功能的单行编辑器，可以用它创建一个屏蔽，便于用户输入一些有效的值，并且能够自动对输入进行格式化。它还可以在编辑框旁边加上一个微调控制，以便在一些数字或一系列选项中滚动。

有时用户需输入有固定格式的数据，例如，要求输入四位数字的年代，则可以使用一个指定格式的编辑掩码控件，便于用户输入。

编辑掩码控件的属性除了常用的属性外（如显示文本、字体等），主要是设置掩码格式和设置渐变控制。

(1) 设置掩码格式

双击该控件得到其属性定义对话框，选择 Mask 页，如图 4-28 所示。

在图 4-28 中的 MaskDataType 框中选择要输入的数据或显示数据的数据类型，在 Mask 框中选择已有的掩码格式或输入希望的掩码格式。图 4-28 是设置输入四位年代的掩码格式的例子。

(2) 设置渐变控件

可以将编辑掩码控件设置为一个带渐变控制的控件，它是一个如 1991 年 所示的编辑框，用户可以单击向上箭头或向下箭头按渐变增量增减或在固定值中循环。

如图 4-28 所示，选择 Spin 微调控制，就将其设置为一个渐变控件。有两种类型的渐变方式：面向列表的和面向值的，通过 UseCodeTable 复选框来决定是哪一种，选中则是面向列表的，否则是面向值的。

可以为面向值的渐变控件设置一个最大值、一个最小值和一个渐变增量。运行时，每单击一次向上箭头就把增量加到编辑框的值上，直到它达到最大值，单击向下箭头则相反。



图 4-28 编辑掩码控件的属性的 Mask 页

而面向列表的渐变控件可以在一系列值中滚动，在设计时，可以在图 4-28 的 DisplayValue 列中输入显示值，在 DataValue 列中输入数据值。并且可以用 Add、Delete 和 Insert 按钮进行调整。运行时，单击向上箭头或向下箭头就在这些列表值中循环滚动。

5. 超文本编辑（RichTextEdit）控件

超文本编辑控件是一个功能十分强大的文本编辑器，它可以使用 Windows 系统支持的所有字体、字型及颜色，它有自己的工具栏和标尺。利用该控件，用户能够输入、编辑、格式化、保存与打印文本。

此控件的一般属性比较简单，除滚动条、设置编辑框的背景颜色和边框风格以外，主要是 Document 属性页，如图 4-29 所示。

各选项的意义如下：

- 选中 TabsVisible，系统以字符“→”来显示用户输入到超文本编辑框中的 Tab 字符。
- 选中 ReturnsVisible，系统以字符“¶”来显示用户输入到超文本编辑框中的换行字符。
- 选中 RulerBar、TabBar、ToolBar，分别显示标尺、Tab 条和工具栏。
- HeaderFooter 用于指定是否加上题头 / 脚注节。
- 选中 PopMenu，用户可以右击鼠标打开编辑的弹出式菜单，弹出式菜单用于进行文本的复制、剪切、粘贴或清除。
- LeftMargin、RightMargin、TopMargin、BottomMargin 分别指定打印页的左空白、右空白、上空白和下空白的大小。

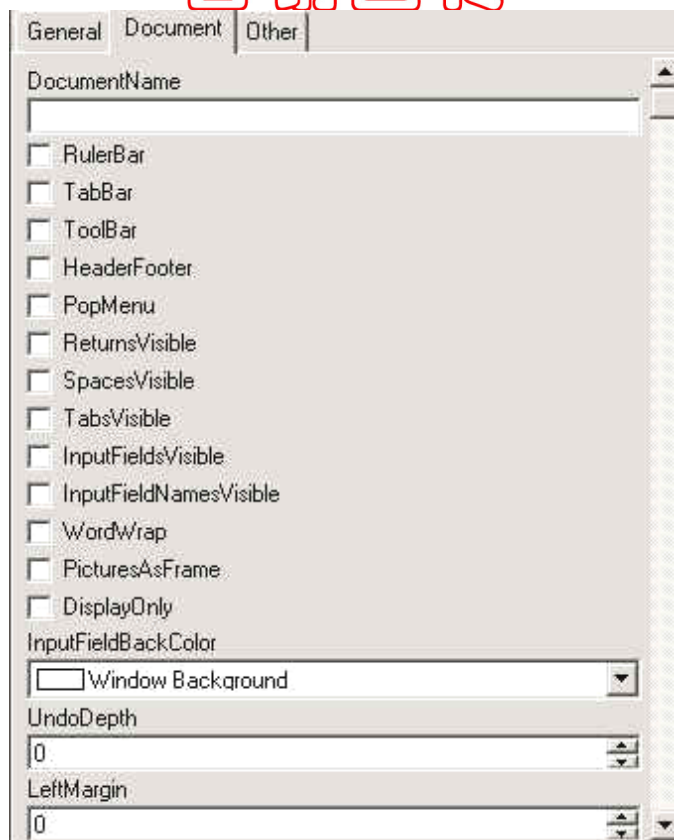


图 4-29 超文本编辑控件属性的 Document 页

§ 4.4.4 列表选择

当应用程序只用到少数几个选项时，使用单选按钮和复选框既简单又方便。但是当选项较多且运行时可能变化，单选按钮和复选框就显得力不从心了。此时，我们可以利用列表方式为用户提供列表选择。**PowerBuilder** 提供了以下四个控件完成列表选择：列表框、图片列表框、下拉列表框和下拉图片列表框。下面分别进行介绍。

1. 列表框

列表框是一种以列表方式显示数据的控件。通常情况下，列表框用于列出一些可选选项供用户选择，这些选项可以在其属性中指定，也可以在程序中动态修改。列表框的一般属性比较简单，主要是设置滚动条、设置是否允许选择多项及排序等，在此不再说明。下面详细说明其属性的 **Items** 页，主要用于输入显示列表框中的数据选项，如图 4-30 所示。

输入列表框的各选项后，按 **OK** 按钮即可。图 4-30 中各个选项前的数字表示此选项的序号（**index**）。

在实际应用中，当用户选择列表框中的一项后，程序应作出反应。下面我们用一个简单的例子来说明如何对列表框进行编程，该程序的功能是当选中列表框中的一项时，复制此项文本到单行编辑器中。

在图 4-31 的窗口上放一个列表框 **lb_1** 和一个单行编辑器 **sle_1**。

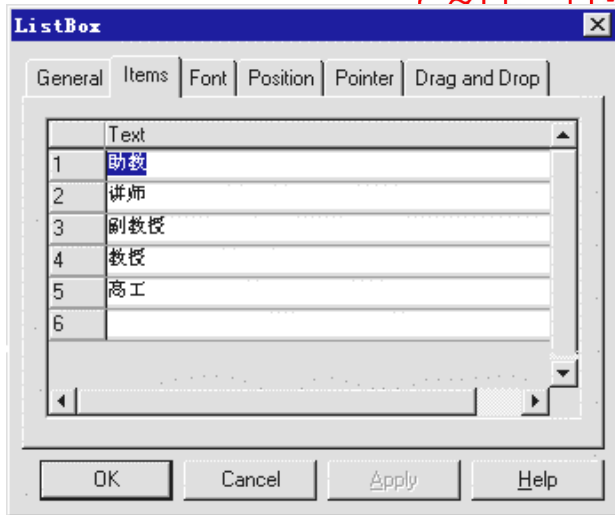


图 4-30 列表框控件属性的 **Items** 页

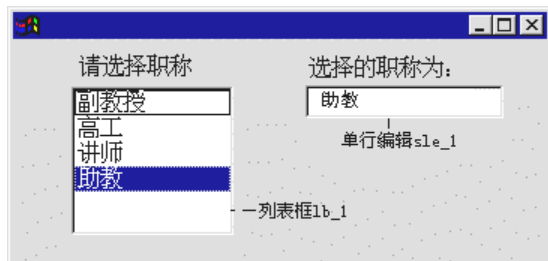


图 4-31 列表框的应用

列表框的 **SelectionChanged** 事件处理程序为：

```
sle_1.text=this.text(index)
```

index 是 **SelectionChanged** 事件的整型参数，用来标识当前选中的是列表框的第几项。运行的结果如图 4-31 所示（说明：**SelectionChanged** 事件在列表框的当前选项改变时触发）。

2. 图片列表框（**PictureListBox**）控件

图片列表框与列表框控件的功能基本相同，不同的是图片列表框既可显示文本又可

显示图片，如图 4-32 所示。

图片列表框的使用方法如下：

- ① 在窗口上放置一个图片列表框控件。
- ② 双击该控件得到其属性，选择 Pictures 页，如图 4-33 所示。

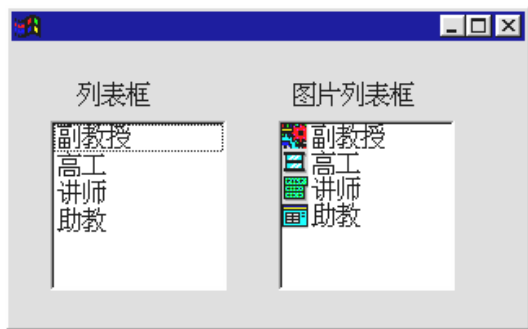


图 4-32 图片列表框

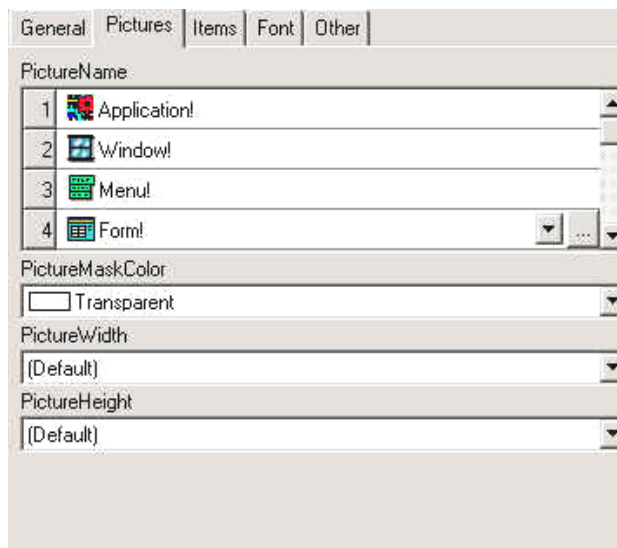


图 4-33 图片列表框属性的 Pictures 页

在 PictureName 下输入图片文件名或用后面的按钮查找，或按向下箭头从中选择 PowerBuilder 提供的图片。各图片从 1 开始编号，此编号在 Items 页中要用到。

③ 选择 Items 页，如图 4-34 所示。在 Text 下输入各选项文本并在 Picture Index 下输入每项所对应的图片编号。

至此就得到了图 4-32 所示的图片列表框。

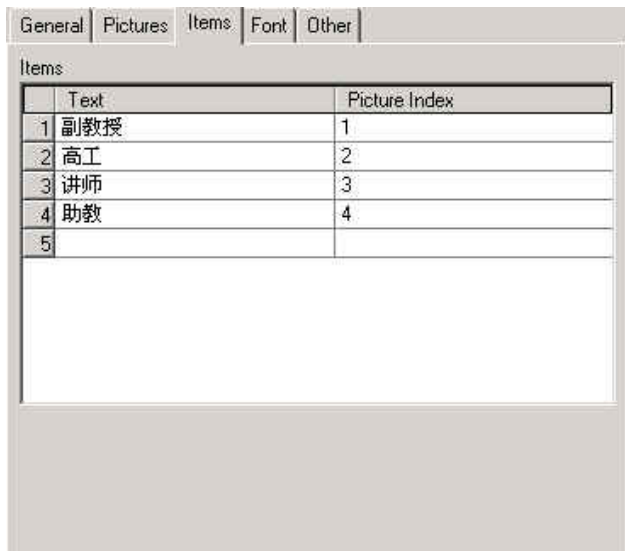


图 4-34 图片列表框属性的 Items 页

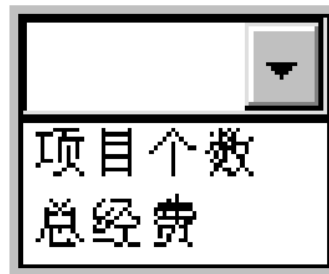


图 4-35 下拉列表框

3. 下拉列表框 (DropDownListBox) 控件

下拉列表框组合了单行编辑框和列表框的特点，即该控件将所有可选项放置在列表中，当选择列表中的一项时，该项显示在图 4-35 所示的方框中，然后隐藏列表框。

下拉列表框有两种类型：一种是不可编辑型，另一种是可编辑型。若用户仅从一个固定的选项集合中做出选择，可设置其为不可编辑型。如果希望给用户输入其他值的机会，可设置其为可编辑型。

对于可编辑型的下拉列表框，可以选择是让该列表一直显示还是隐藏该列表。对于不可编辑型的下拉列表框，不能选择一直显示该列表，用户可以通过单击向下箭头来显示该列表。两种类型的确定是通过对其属性对话框中 `AllowEdit` 复选框的选取来设置的。其他属性比较简单，在此不再赘述。

列表框和下拉列表框都可用来从多项中选择内容，但下拉列表框既可节省空间，又可自己输入列表中没有的值。

例如，在图 4-36 所示的窗口中采用下拉列表框来选择统计内容。

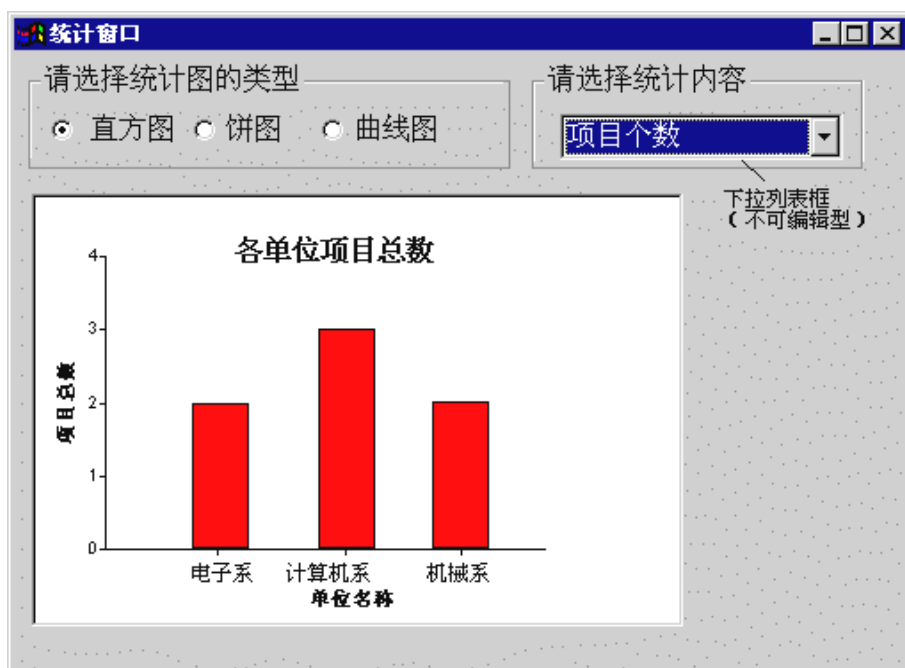


图 4-36 下拉列表框的应用

下拉列表框中的选项可以在设计时指定，也可以在程序中动态构造。而后一种方法在系统开发中尤其常用（例如：其选项可能是从数据库中取的，而数据库中的内容是动态改变的，此时，下拉列表框的选项必须动态构造）。下面我们通过一个例子来说明其构造方法。

在图 4-37 所示的窗口 `w_1` 中有一个下拉列表框控件 `ddlb_1`，我们在打开窗口 `w_1` 时构造列表选项。

窗口 `w_1` 的 `Open` 事件处理程序为：

```
ddlb_1.reset()           //下拉列表清空  
ddlb_1.additem("教授")  
ddlb_1.additem("副教授")  
ddlb_1.additem("高工")  
ddlb_1.additem("讲师")
```

运行程序，打开此窗口，结果如图 4-37 所示。

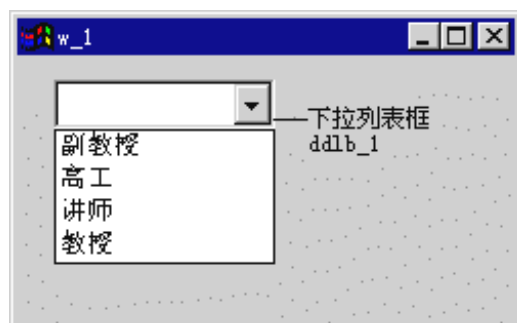


图 4-37 窗口 `w_1`

4. 下拉图片列表框 (DropDownPictureListBox) 控件

下拉图片列表框的功能与下拉列表框基本相同, 不同点在于下拉图片列表框不仅能显示文本, 还能显示图片。其图片和选项的设置方法与图片列表框相同, 在此不再赘述。

§ 4.4.5 图片控件

图片 (Picture) 控件是 PowerBuilder 中的特殊控件, 它用于显示一个位图 BMP 文件、RLE 文件或者 Aldus 风格的 WMF 文件。

下面介绍图片控件的一般属性, 如图 4-38 所示。

图片控件与静态文本的作用类似, 都是起修饰辅助说明的作用, 它本身也有事件, 通常并不使用。

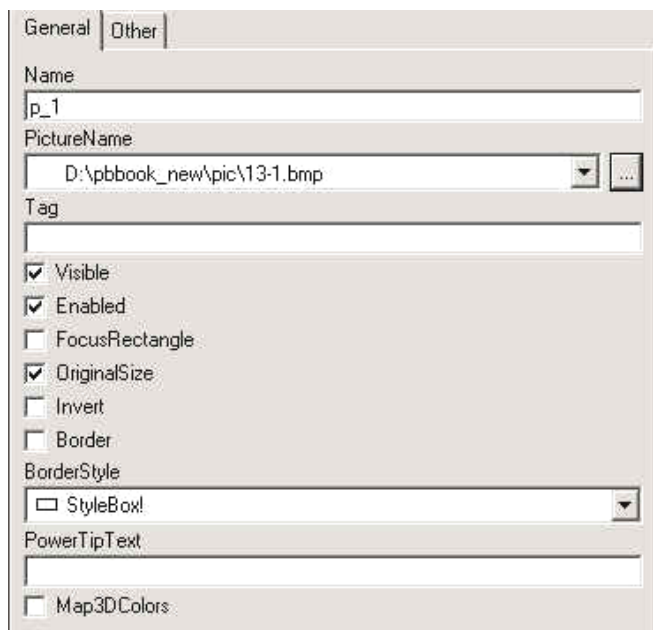


图 4-38 图片控件的一般属性

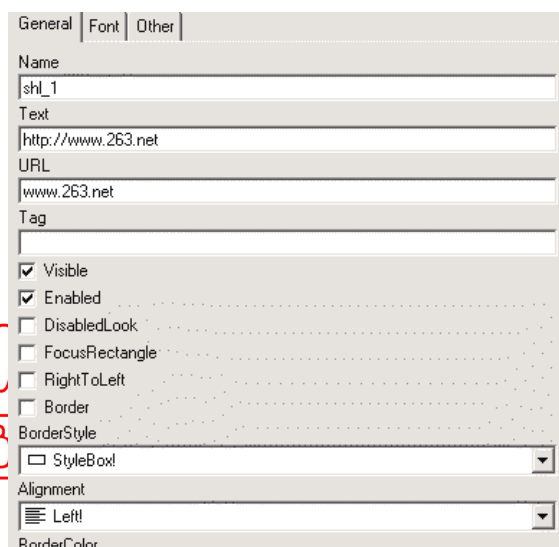


图 4-39 静态文本超链接控件的一般属性

§ 4.4.6 超链接控件

超链接控件主要用来链接到特定的 Web 页上, 包括静态文本超链接控件 (StaticHyperLink) 和图片超链接控件 (PictureHyperLink), 下面分别介绍。

1. 静态文本超链接控件

下面介绍其一般属性, 如图 4-39 所示。其中, Name 为控件名字, Text 为显示文本, URL 为链接到的网址。

通过设置其 URL 属性, 使运行时单击此控件就可以打开浏览器浏览相应的网页。

2. 图片超链接控件

与静态文本超链接相似, 只是需要设置图片文件名。

§ 4.4.7 标签控件

标签 (Tab) 控件是 PowerBuilder 开发环境中使用最多的控件之一。当设置一个对

象的属性时，打开的属性对话框使用的就是标签控件。一个标签控件由多个 Tab 页（标签页）组成，用户能看到的总是当前的 Tab 页。

（1）为标签控件增加标签页

① 在窗口中放置一个标签控件，新放上的标签控件只有一个 Tab 页，还需添加新的 Tab 页到该控件中。

② 在标签控件的标签上（而不是 Tab 页上）单击鼠标右键，显示弹出式菜单，单击弹出菜单中的 InsertTabPage 项，则添加一个新的 Tab 页到标签控件中。

③ 重复步骤②，直到添加完所需的 Tab 页。

（2）修改标签控件中 Tab 页的属性

① 双击选中的 Tab 页（确保鼠标不在该 Tab 页的标签位置），得到该 Tab 页的属性，如图 4-40 所示。



图 4-40 标签控件中 Tab 页的属性

② 选择 Picture 项，可以为该 Tab 页的标签设置一个图片。



图 4-41 标签控件属性的 General 页

(3) 修改标签控件的属性

① 双击标签控件（把鼠标放在标签位置上），得到该标签控件的属性，选择 **General** 属性页，如图 4-41 所示。

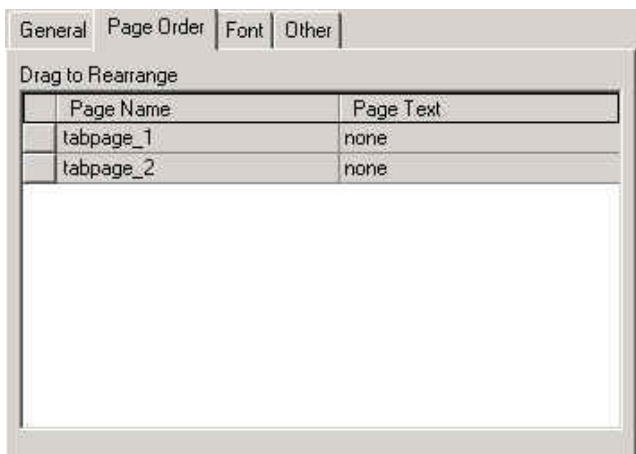


图 4-42 标签控件属性的 PageOrder 页

② 选择 **PageOrder** 属性页，如图 4-42 所示，根据需要修改该控件中 Tab 页的顺序，方法是用鼠标拖动一行到指定位置，然后释放鼠标。

标签是一个控件，上面可以放多个标签页，标签控件的属性如图 4-41 和图 4-42 所示。而每一个标签页也有属性，如图 4-40 所示。在每个标签页中，可以像在窗口上一样放置控件。还可以对放置的控件进行编程。

§ 4.4.8 数据窗口控件

这里的数据窗口（DataWindow）控件只不过是窗口上的一个空白的控件，与第七章中讲的数据窗口对象不是一回事。当这个空白的数据窗口控件建立好后，再将它与某个数据窗口对象建立联系，在系统运行时，用户通过窗口上的数据窗口控件来操纵数据窗口对象，完成对数据库的操作（录入、修改、删除等）。

数据窗口控件的属性如图 4-43 所示。



图 4-43 数据窗口控件的属性

数据窗口是 PowerBuilder 中的一大特色, 灵活使用数据窗口是开发应用的关键所在。关于数据窗口的详细介绍请见第七章。

§ 4.4.9 树状浏览控件和列表浏览控件

树状浏览 (TreeView) 和列表浏览 (ListView) 是表示数据项的两种非常直观的形式, 下面分别进行介绍。

1. 树状浏览控件

Windows 98 资源管理器你一定不会陌生, 在它左边的列表中以层次结构列出了驱动器、目录及子目录, 用户通过单击层次结构中的加号 (+)、减号 (-) 来展开或折叠目录, 同时, 每个目录前面都有一个代表其类别的图标。这种表示方法使用户一目了然。

在 PowerBuilder 中也可以用这种方法来表示数据, 这就要使用树状浏览控件。

(1) 添加 TreeView 项及其图片

具体步骤如下:

① 在窗口中放一个树状浏览控件, 并得到其属性对话框, 选择 Pictures 属性页, 如图 4-44 所示。

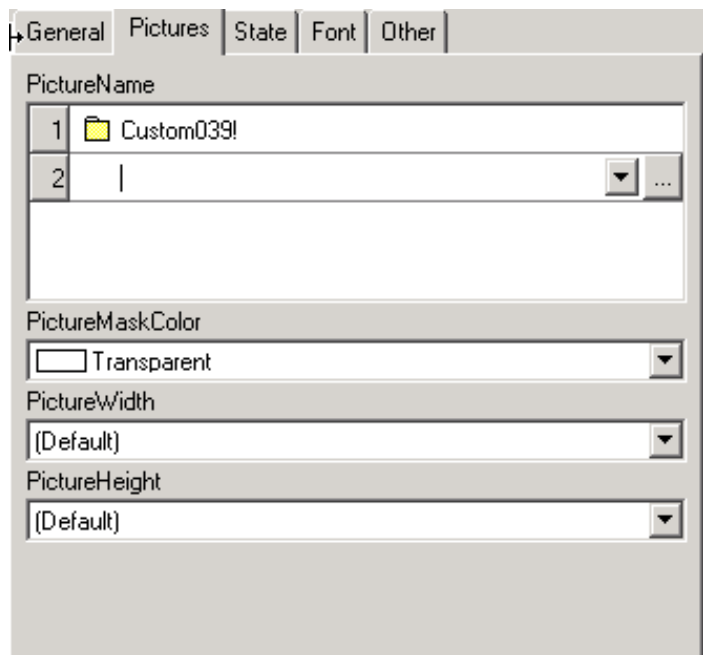


图 4-44 树状浏览控件属性的 Pictures 页

② 在 PictureName 中输入图片名或单击后面的按钮进行查找或从 Stock Pictures 框中选择 PowerBuilder 提供的图片。

③ 重复直到键入所需的所有图片, 所有的图片从 1 开始编号作为其索引值。

④ 单击 OK 按钮, 返回到窗口工作区。

⑤ 编写事件处理程序为树状浏览控件添加项。在程序中注意把图片的索引值和该项联系起来。

(2) 添加状态图片

状态图片用于表示该项所处的状态。可通过在不同的情况下激活不同的状态图片来

指示该项所处的状态。具体操作步骤为：在树状浏览控件的属性中选择 **State** 属性页，其界面和操作方法与添加浏览项图片相同，在此不再赘述。

（3）设置树状浏览的其他属性

选择其属性的 **General** 页，如图 4-45 所示。

以上讲述了如何设置树状浏览控件的属性。但是，要使它按要求显示数据，还得为它编写程序。下面举例进行说明。

用树状浏览显示职工和其参加的科研项目：按检索按钮开始检索，按展开按钮展开所有项，按退出按钮关闭窗口，如图 4-46 所示。

下面是本例的具体做法（首先建立两个数据窗口对象 `dw_employee` 和 `dw_item`。关于数据窗口的详细情况参见第七章）：



图 4-45 树状浏览控件属性的 General 页

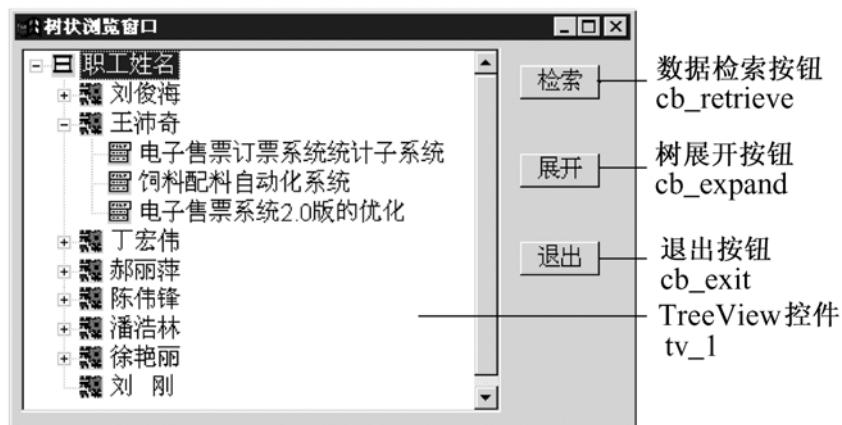


图 4-46 树状浏览窗口

① 创建一个窗口，窗口上放一个 `TreeView` 控件：`tv_1`，三个按钮：检索按钮 `cb_retrieve`、展开按钮 `cb_expand`、退出按钮 `cb_exit`。

② 定义窗口实例变量如下：

```
datastore ids1, ids2
long il_rootitem
```

③ 编写处理程序。

在窗口的 `Open` 事件中编写如下程序：

```
ids1=create datastore
ids2=create datastore
ids1.dataobject="dw_employee"
ids2.dataobject="dw_item"
```

在窗口的 `Close` 事件中编写如下程序：

```
destroy ids1
destroy ids2
```

在检索按钮 `Clicked` 事件中编写如下程序。这段程序的功能是检索职工及其参加项目信息，并把检索结果加到 `TreeView` 控件中去，但只显示树根。

```
long newitem
integer row1, row2, i, j
treeviewitem item
string name, code
tv_1.setredraw (false)
tv_1.deleteitem (il_rootitem)
item.label="职工姓名"
item.pictureindex=1
item.selectedpictureindex=1
il_rootitem=tv_1.insertitemlast (0, item)
row1=ids1.retrieve ()
row2=ids2.retrieve ()
for i=1 to row1
    code=ids1.getitemstring (i, "em_code")
    name=ids1.getitemstring (i, "name")
    item.label=name
    item.pictureindex=2
    item.selectedpictureindex=2
    newitem=tv_1.insertitemlast (il_rootitem, item)
    for j=1 to row2
        if code=ids2.getitemstring (j, "member_em_code") then
            item.label=ids2.getitemstring (j, "items_itemname")
            item.pictureindex=3
```

科学出版社
营销宣传


```

item.selectedpictureindex=3
tv_1.insertitemlast (newitem, item)
end if
next
next
tv_1.setredraw (true)

```

在展开按钮的 Clicked 事件中编写如下程序。功能是展开树状结构，显示检索结果。

```
tv_1.expandall (il_rootitem)
```

在退出按钮的 Clicked 事件中编写如下程序，用于关闭该窗口。

```
close (parent)
```

运行结果如图 4-46 所示。

2. 列表浏览控件



图 4-47 列表浏览控件的表现形式

列表浏览控件以图标的形式显示数据项及其标签值，如图 4-47 所示。

此控件的属性与树状浏览控件类似，不同点在于列表浏览控件有大图标和小图标两种图标。

在应用中，一般也要对其编程来显示需要的数据，其方法与树状浏览控件类似，在此不再赘述。

§ 4.4.10 滚动条控件

PowerBuilder 提供了两种滚动条控件，即：横向滚动条（HScrollBar）控件和纵向滚动条（VScrollBar）控件。这两种控件会在窗口上产生一个滚动条，它们的属性也比较简单，此控件很少使用。

§ 4.4.11 进度条与轨迹条控件

1. 进度条控件

进度条控件分为 HProgressBar（水平进度条）和 VProgressBar（垂直进度条）两种，前者在窗口上水平放置，后者在窗口上垂直放置。

进度条控件是用来显示某个过程的执行进度的。例如，一些大型软件的安装程序一般都有进度条提示安装进度。

其属性比较简单，如图 4-48 所示。下面以水平进度条为例进行说明。

- **MinPosition:** 给出控件最左端的起始值，默认为 0。
- **MaxPosition:** 给出控件最右端的最大值，默认为 100。
- **Position:** 指出进度的当前位置。
- **SmoothScroll:** 确定进度条的显示形式是间断离散的还是平滑连续的。

- **SetStep**: 指出进度条增量的步长, 默认为 10。

下面通过一个简单的例子来说明进度条的使用。

假设某个软件安装需要 4 分钟 (240 秒), 在安装程序中使用 **HProgressBar** 控件, 功能如下:

- 通过窗口的 **Timer** 事件和 **Timer** 函数每 2.4 秒使进度条前进 1%。
- 若安装在 4 分钟之内完成, 则直接使进度条前进到 100%。
- 若安装在 4 分钟之内完成不了, 当进度条前进到 98%时停止触发 **Timer** 事件, 当安装完成时直接使进度条前进到 100%。
- 在进度条控件下面放置一个静态文本控件来显示安装进度。

为完成以上功能, 所需语句如下:

```
long I
```

在窗口的 **Open** 事件中初始化 **HProgressBar** 控件:

```
hpb_1.minposition=0
```

```
hpb_1.maxposition=100
```

```
hpb_1.position=0
```

```
hpb_1.smoothscroll=true
```

程序安装开始时触发 **Timer** 事件:

```
timer (1.0)
```

在窗口的 **Timer** 事件中有如下语句:

```
I++
```

```
hpb_1.position=I
```

```
st_1.text=string (i) + " %"
```

```
if I=98 then timer (0)
```

在程序安装完成时执行语句:

```
hpb_1.position=100
```

```
st_1.text="100%"
```

```
messagebox ("提示", "安装完成。")
```

```
close (parent)
```

2. 轨迹条控件

轨迹条控件分为 **HTrackBar** (水平轨迹条) 和 **VTrackBar** (垂直轨迹条) 两种, 前者在窗口上水平放置, 后者在窗口上垂直放置。

轨迹条控件可以控制其他控件的变化 (如缩小、放大等)。它的外观包括标尺、标尺

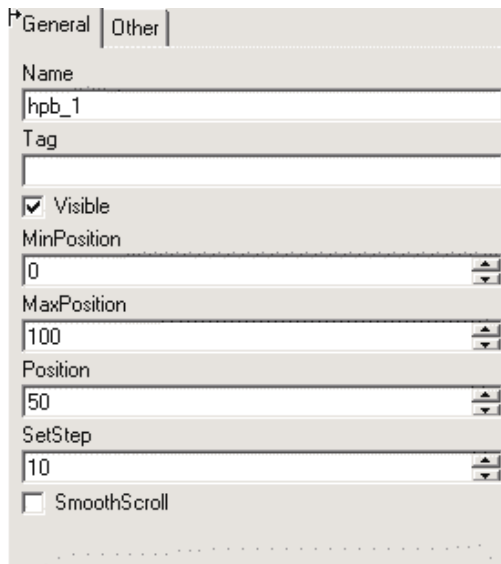


图 4-48 水平进度条控件的属性

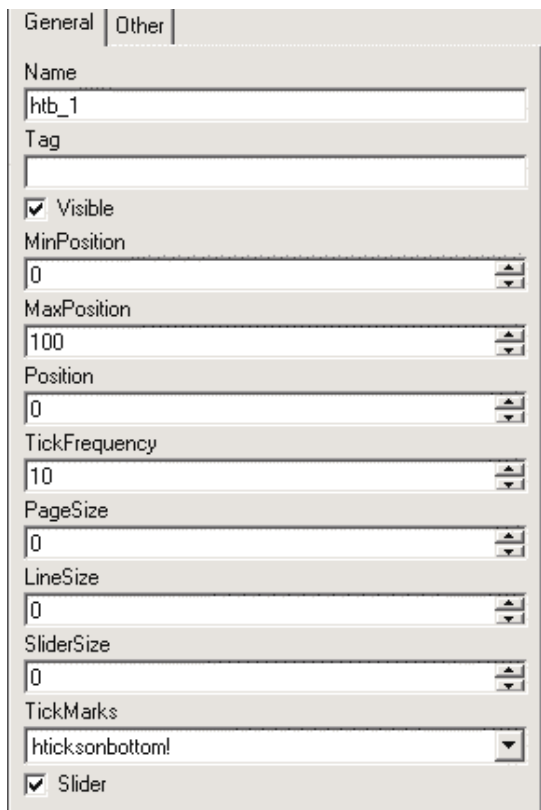


图 4-49 水平轨迹条控件的属性

刻度和滑动游标。标尺说明了控制范围，刻度说明变化的递进，滑动游标控制变化的方向。

两种控件的功能完全一样。下面以水平轨迹条为例说明其属性，如图 4-49 所示。

- **MinPosition:** 指出标尺左起始点的最小值。
- **MaxPosition:** 指出标尺右终止点的最大值。
- **Position:** 确定游标在控件中的当前位置。
- **TickFrequency:** 指出标尺刻度的增量。
- **PageSize:** 指出鼠标在控件上按 PageUp 键和 PageDown 键时游标向左或向右移动的刻度值。
- **LineSize:** 指出鼠标在控件上按左箭头和右箭头时游标向左或向右移动的刻度值。

- **SliderSize:** 指出滑动游标自身的宽度。
- **TickMarks:** 指出滑动游标和刻度的显示方式。
- **Slider:** 指出是否显示滑动游标。

科学出版社
营销宣传

§ 4.4.12 OLE 2.0 控件

在窗口上放置 OLE 2.0 控件，用于将 Windows 支持的 OLE 对象链接到 OLE 2.0 控件中，实现在 PowerBuilder 应用中利用 OLE 服务器的命令和函数完成对 OLE 对象的操作功能。例如，可以在窗口上定义 OLE 控件，定义该控件链接到微软公司的 Excel，利用该应用在 PowerBuilder 应用中输出电子表格。

关于 OLE 控件的详细情况参见第十二章。

§ 4.4.13 用户对象控件

用户对象（UserObjects）控件用来将可视用户对象放置在窗口中。用户对象是在用户对象画板中创建的。关于其详细内容，请参阅第十一章。

用户对象的放置方法为：

- ① 在窗口工作区的下拉列表控件中选择用户对象图标，显示 Select User Object 对话框。
- ② 选择已经定义好的用户对象，然后单击 OK 按钮。

§ 4.4.14 绘图控件

PowerBuilder 提供了四种绘图控件：线（Line）、矩形（Rectangle）、圆角矩形（RoundRectangle）、椭圆（Oval）。这些控件主要起修饰作用，没有事件。它们的使用比较简单，在窗口工作区的控件下拉框中选择相应的绘图控件，放置在窗口上，并设置其一般属性即可。

§ 4.5 窗口的继承

继承是 PowerBuilder 强有力的特点之一，使用继承能够方便地从已有的窗口中派生出新的窗口。使用继承创建窗口有如下优点：

- 当改变祖先窗口时，所有的后代窗口也都随之改变，无需手工修改，从而可以节省时间，使应用程序更易于维护。
- 后代窗口继承祖先窗口的事件处理程序，从而避免了重复工作。

1. 通过继承创建窗口

具体步骤如下：

- ① 在 File 下拉菜单中选择 Inherit...
- ② 在弹出的 Inherit from Object 对话框中选择要继承的窗口，如图 4-50 所示。
- ③ 单击 OK 按钮进入窗口画板工作区，在此对话框中选择一个窗口作为父窗口，然后单击 OK 按钮，则窗口画板显示通过继承创建的新窗口。

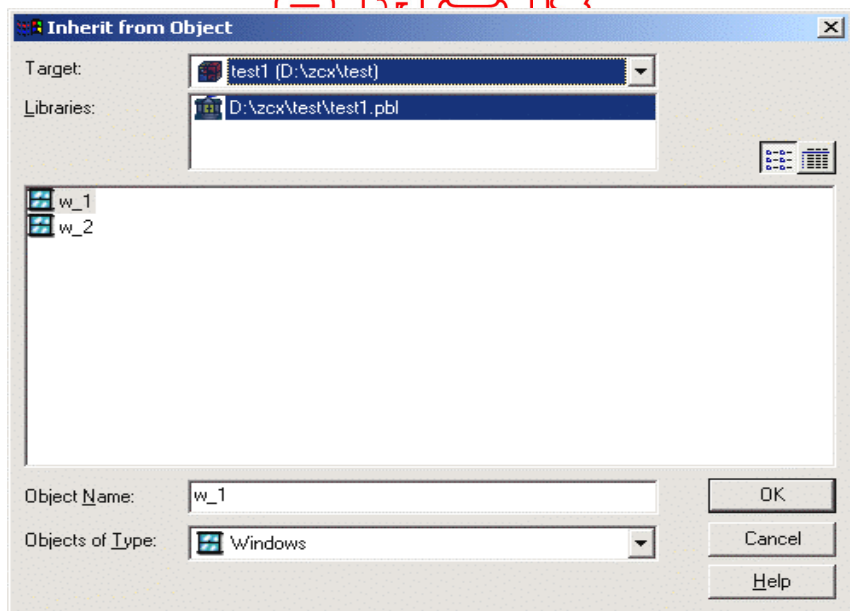


图 4-50 Inherit from Object 对话框

2. 继承窗口的修改

对于继承的窗口，可根据需要对其进行修改，但有些修改是不允许的。下面介绍在新窗口中可做的工作和不可做的工作。

(1) 可做的工作

- 修改窗口的属性。
- 添加新控件。
- 编写事件处理程序。
- 引用祖先窗口的函数、事件和结构。
- 定义窗口的函数、结构和变量。
- 定义窗口及窗口上控件的用户自定义事件。

(2) 不可做的工作

删除已有的控件。

3. 继承窗口的事件处理程序

(1) 查看继承的事件处理程序

在后代窗口的编程工作区，不能直接看到祖先窗口的事件处理程序，要查看祖先窗口的事件处理程序可以从菜单栏选择 **Design→Display Ancestor Script** 项。

(2) 对后代窗口的编程

在由祖先窗口和后代窗口构成的继承层次结构中，每个后代窗口都继承了其父窗口的事件处理程序。在后代窗口运行时，PowerBuilder 提供了以下两种执行事件处理程序的方式。

- 重载祖先程序：执行后代窗口的程序而不执行祖先窗口的程序。

重载祖先窗口程序的步骤如下：

- ① 选择相应的窗口。
- ② 进入编程工作区。
- ③ 选择事件。
- ④ 从菜单中选择 **Design→Override Ancestor Script**。
- ⑤ 在编程工作区编写事件处理程序。
- ⑥ 编程完毕后返回到窗口工作区。

- 扩展祖先程序：先执行祖先窗口的程序，然后执行后代窗口的程序。

扩展祖先窗口程序的步骤如下：

- ① 重复上述步骤①、②、③。
- ② 单击菜单栏中的 **Design→Extend Ancestor Script**。
- ③ 在编程工作区编写程序。
- ④ 编程完毕后返回窗口工作区。

习 题

1. PowerBuilder 的窗口有哪些类型？它们之间有何区别？
2. 请创建如图 4-51 所示的窗口 w_xt1，并运行此窗口。
其中，w_xt1 的 Open 事件处理程序为：

```
sle_1.setfocus()
```


sle_1 的 Modify 事件处理程序为：

```
sle_2.setfocus ()
```

sle_2 的 Modify 事件处理程序为：

```
cb_1.setfocus ()
```

cb_1 的 Clicked 事件处理程序为：

```
sle_3.text=string (integer  
(sle_1.text)+integer  
(sle_2.text))
```

cb_2 的 Clicked 事件处理程序为：

```
sle_3.text=string (integer  
(sle_1.text)-integer  
(sle_2.text))
```

3. 请创建图 4-22 所示的窗口，并
存为 w_inputemployee。

4. 请说明单选钮控件和复选框控件的区别，并创建图 4-24 所示的窗口。其中的数据窗口控件对应的数据窗口对象可以为空。

5. 请创建图 4-37 所示的窗口。

6. 请说明 MessageBox 对话框的功能和使用方法。创建图 4-52 所示的窗口 w_xt2，
然后请运行此窗口。

该窗口的功能是：用户用鼠标选择日期，回车后，在 sle_1 中显示该日期。当单击“退出”按钮时，弹出如图 4-53 的退出窗口（MessageBox 对话框），在退出窗口中，选择“是”按钮则关闭窗口 w_xt2，否则继续。

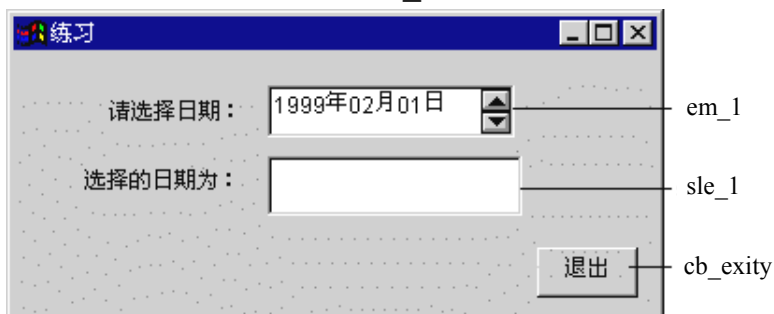


图 4-52 窗口 w_xt2

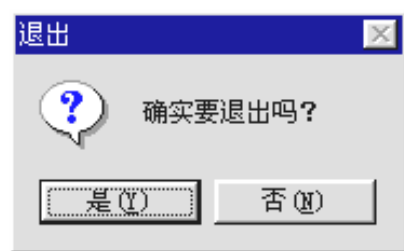


图 4-53 退出窗口

下面给出各事件脚本。

em_1 的 modify 事件处理程序为：

```
date d
```

```
em_1.GetData (d)
```

```
sle_1.Text = String (d, "yyyy-mm-dd")
```

cb_exit 的 clicked 事件处理程序为：

```
int i
```

```
i=messagebox ("退出", "确实要退出吗?", question!, yesno!)
```

```
if i=1 then  
    close (parent)  
end if
```

7. 请创建图 4-54 所示的统计窗口 w_tj。

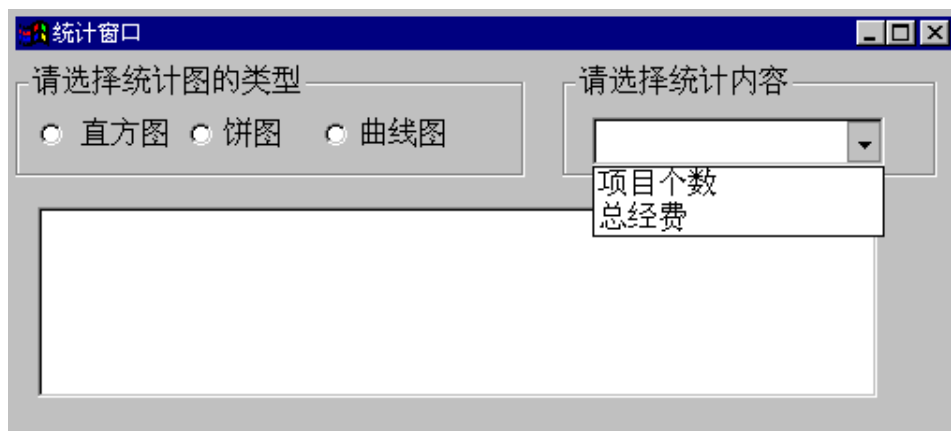


图 4-54 统计窗口 w_tj

科学出版社
营销宣传