

PowerBuilder 6.0/6.5 技术丛书

刘红岩 张步达 主编

PowerBuilder 控件技术详解

邵建生 龚伟至 孟凡军 等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

PowerBuilder 控件的种类丰富,操作灵活,功能强大。本书重点介绍了标准控件中的一些功能较强使用略微复杂的控件以及对它们的高级使用方法,还介绍了如何使用 PowerBuilder 以外的应用程序所提供的控件,如 OLE、Active X 等,以丰富和增强 PowerBuilder 应用程序的功能。全书共分 8 章,3 个附录。首先介绍了 Rich-TextEdit、TreeView、ListView 等控件的特点和使用,然后叙述弹出式菜单的基本功能和特点,并重点说明创建弹出式菜单的具体方法及步骤。每章最后都有小结,并附有练习题,在附录 A 中有练习题的答案,以便参考。在附录 B 中对 PowerBuilder 6.0/6.5 所使用的对象和控件的属性、事件和函数有详细的解释,以便用户做应用开发时查阅。

本书是数据库应用开发人员首选的使用手册,适于做培训教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻版必究。

图书在版编目(CIP)数据

PowerBuilder 控件技术详解:PowerBuilder 6.0/6.5 技术丛书/邵建生等编著. —北京:电子工业出版社, 1999.6

ISBN 7-5053-5262-8

I. P… II. 邵… III. 数据库管理系统-软件工具, PowerBuilder 6.0/6.5 IV. TP311.56

中国版本图书馆 CIP 数据核字(1999)第 08118 号

丛 书 名: PowerBuilder 6.0/6.5 技术丛书

主 编: 刘红岩 张步达

书 名: PowerBuilder 控件技术详解

编 著 者: 邵建生 龚伟至 孟凡军等

策 划: 郭 立

责任编辑: 郭 晓

特约编辑: 贾 宁

排版制作: 电子工业出版社计算机排版室

印 刷 者: 北京东光印刷厂

出版发行: 电子工业出版社 URL: <http://www.phci.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 13 字数: 350 千字

版 次: 1999 年 6 月第 1 版 1999 年 6 月第 1 次印刷

书 号: ISBN 7-5053-5262-8
TP·2630

印 数: 5000 册 定价: 26.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,请向购买书店调换。

若书店售缺,请与本社发行部联系调换。电话 68279077

《PowerBuilder 6.0/6.5 技术丛书》编委会

顾 问：徐国祥 SYBASE(中国)公司副总经理 技术总监
杨孝如 SYBASE(中国)公司高级工程师

主 编：刘红岩 张步达

编 委：何 军 王 蓉 肖 斌 杨 慧 石京民
邵建生 龚伟志 孟凡军 袁 刚 黄 辉
李 雷 王怀君 刘红丽 何雅丽 戴建荣
何 杰 王 宇 陈 青

策 划：郭 立

前 言

一、PowerBuilder 6.0/6.5 的优异性能

应用软件的开发生产率与计算机硬件的巨大进步相比,已成为限制计算机应用发展的瓶颈。因而,从事应用系统开发的广大软件人员迫切需要适用、高效的工具。作为一种可视化的、面向对象的快速应用开发(RAD)工具,PowerBuilder 已被广大数据库应用开发人员所使用,并获得好评。相对于其他任何应用开发工具,PowerBuilder 可以使开发人员的工作速度更快、成本更低、质量更高、功能更强。PowerBuilder 全面支持面向对象编程;是集成强大并易于使用的第四代编程语言(4GL);内置包括数据窗口在内的多种对象类,可以方便地访问数据库。

1994 年推出的 PowerBuilder 4.0 和 1996 年推出的 PowerBuilder 5.0 版非常适用于 Client/Server 结构的应用系统,但是,随着应用系统结构的发展,对于目前的 Web 应用结构和瘦客户端的应用模式来说具有一定的局限。SYBASE 公司在 PowerBuilder 6.0 中做了很大改进,提供了强大的组件生成器,主要是提供了多种方式以支持 Web 应用。PowerBuilder 还支持 Java Proxy 组件,这大大扩展了 PowerBuilder 的适用范围,PowerBuilder 6.0 几乎支持目前通用的 Web 应用模式,但在客户端编程方面有些不足,PowerBuilder 6.5 在这方面又有了很大进步。PowerBuilder 6.5 几乎完全可在 Web 上进行所有的工作,它在 6.0 的基础上增加了 PowerSite Web 应用开发技术,成为建立 Client/Server 和多层分布式应用的一个完整的 4GL RAD 环境。PowerBuilder 6.5 使用的是业界熟悉的 Web 标准(HTML 和 Java),通过端对端的设计、调试和提交,可以控制 Web 应用的整个生命周期。

二、本套丛书的特点

1996 年 5 月,电子工业出版社出版了我们编写的《PowerBuilder 5.0 原理与应用开发指南》一书。该书较全面地向读者介绍了 PowerBuilder 5.0 的基本概念和主要功能,成为许多开发人员和读者了解和使用 PowerBuilder 的主要参考书,先后重印四次,深受读者欢迎。

面对 PowerBuilder 6.0/6.5 这样一个内容博大精深的应用开发产品,广大应用开发人员迫切需要一套更加系统而全面地介绍 PowerBuilder 的丛书。应广大用户的热情要求,我们编写了这套《PowerBuilder 6.0/6.5 技术丛书》。

本丛书在内容的编写和结构的安排上做了较为精心的设计。如果您不熟悉 PowerBuilder,利用本套丛书能很快入门并循序渐进;如果您对 PowerBuilder 有所了解,使用本套丛书能尽快提高应用水平;如果您长期使用 PowerBuilder,也能从本套丛书中获得很多应用开发的高级技巧。

本丛书包括以下分册:

- 1.《PowerBuilder 原理与应用指南》是利用 PowerBuilder 开发应用程序或学习如何使用 PowerBuilder 时所必读的入门书,该书全面介绍了 PowerBuilder 开发应用的步骤、各种工具和使用方法。

2.《PowerScript 语言、事件与函数》是利用 PowerBuilder 开发应用程序,或遇到诸如 PowerScript 语句、事件或函数的使用而不清楚时,所必备的参考书。

3.《PowerBuilder 应用开发技术详解》针对那些已经掌握用 PowerBuilder 进行应用开发的基本方法的用户,围绕着一组常用的应用实现技术,指导读者进一步掌握开发、分发与维护 PowerBuilder 应用程序的方法。

4.《PowerBuilder 数据窗口技术详解》全面详尽地介绍了开发人员常常涉及到的数据窗口的有关技术,更重要的是,通过对数据窗口技术内部机制的深入剖析和对大量程序片段的解释,使得开发人员能够快速而全面地掌握数据窗口的各种技术。

5.《PowerBuilder 控件技术详解》主要介绍一些功能强大而略微复杂的控件以及其高级使用方法,并介绍如何使用 PowerBuilder 以外的应用程序所提供的控件,如 OLE、Active X 等,来丰富和增强 PowerBuilder 应用程序的功能。

6.《PowerBuilder Internet 技术详解》由浅入深、详尽地介绍了用 PowerBuilder 6.0/6.5 开发 Internet 应用的方法。此外,还花费了相当的篇幅讨论 Web 站点及其应用开发的规划问题,这对打算建立 Web 站点的用户会有较大的帮助。

7.《PowerBuilder 高级开发技术》全面而详细地介绍了 PowerBuilder 基础类库(PFC)、C++ 类构造器以及版本控制系统 ObjectCycle、版本控制接口 PVCS 和 PowerBuilder 中的各种高级分析和开发工具的使用,对于熟练的 PowerBuilder 开发人员来说,使用这些工具能使您的应用程序锦上添花。

三、编写人员简介

这套丛书由刘红岩博士(清华大学经管学院)、张步达先生(北方计算中心)担任主编,何军副教授(中国人民大学数据库所)在本丛书的组织及编写工作中起了重要作用。

本套丛书是在征求了 SYBASE 公司培训和技术支持专家的意见并进行了多次讨论后,由本书主编与电子工业出版社的人士共同策划和组织的。参加本书编写的多数人员是国内应用 PowerBuilder 历史最久的数据库专家,他们在研究、使用和开发 PowerBuilder 中积累了较为丰富的经验。

四、致谢及其他

本套丛书在编写过程中得到了许多 PowerBuilder 用户的热情支持,他们是我们编写这套丛书的动力;SYBASE(中国)公司给我们提供了大量的技术资料和宝贵建议,他们的帮助是高质量编写本套丛书的保证;电子工业出版社的领导和编辑也对本丛书的编写给予了极大的关心,并付出了艰辛的劳动,在此一并表示感谢。

由于时间仓促,书中难免会有不足和错误之处,敬请广大读者提出宝贵意见,以便我们在下一版中修正。

刘红岩 张步达
1999 年 3 月于北京

目 录

第 1 章 概述	(1)
1.1 系统对象、属性、事件和函数	(2)
1.1.1 PowerBuilder 系统对象	(2)
1.1.2 对象的属性	(3)
1.1.3 对象的事件	(3)
1.1.4 内置式函数	(4)
1.2 系统对象的继承层次	(5)
1.3 查看系统对象	(7)
1.4 正确选择控件	(10)
1.4.1 在窗口中放置控件	(10)
1.4.2 选择控件	(12)
1.4.3 调整控件的属性	(12)
1.4.4 各种控件的使用	(17)
1.5 练习题	(20)
1.6 小结	(20)
第 2 章 使用 RichTextEdit 控件	(21)
2.1 RichText 的基本功能和特点	(21)
2.1.1 什么是 RichText	(21)
2.1.2 RichText 的特点	(22)
2.2 RichText 类型的数据窗口	(22)
2.2.1 RichText 数据窗口的特点	(22)
2.2.2 创建 RichText 数据窗口	(23)
2.2.3 设置 RichText 数据窗口属性	(26)
2.2.4 增加数据窗口的列	(26)
2.2.5 设置输入字段属性	(28)
2.2.6 操作 RichText 数据窗口	(29)
2.2.7 使用 RichText 数据窗口	(30)
2.3 RichTextEdit 控件	(31)
2.3.1 什么是 RichTextEdit 控件	(31)
2.3.2 向 RichTextEdit 控件载入文本	(32)
2.3.3 创建一个 RichTextEdit 控件	(33)
2.3.4 RichTextEdit 数据共享	(34)
2.3.5 用 RichTextEdit 控件来保存 RichText 文本	(35)
2.4 预览 RichTextEdit 报表	(35)
2.4.1 RichText 数据窗口的预览	(36)

2.4.2 RichTextEdit 控件预览	(36)
2.5 练习题	(36)
2.6 小结	(37)
第3章 使用 TreeView 控件	(38)
3.1 TreeView 的基本功能和特点	(38)
3.1.1 TreeView 控件的基本功能	(38)
3.1.2 TreeView 控件的特点	(39)
3.1.3 TreeView 控件的结构	(40)
3.1.4 TreeViewItem 的图标	(42)
3.2 创建 TreeView 控件	(46)
3.2.1 新建 TreeView 控件	(46)
3.2.2 设置 TreeView 控件的属性	(47)
3.2.3 TreeView 控件的属性列表	(52)
3.3 扩展 TreeView 控件	(54)
3.3.1 扩充 TreeView 控件的方式	(54)
3.3.2 TreeView 控件的事件	(55)
3.3.3 如何扩展 TreeView 控件	(56)
3.3.4 TreeView 控件的函数	(58)
3.4 使用 TreeView 控件项	(58)
3.4.1 指定一个 TreeViewItem	(59)
3.4.2 修改 TreeView 图片	(61)
3.4.3 编辑 TreeViewItem	(61)
3.5 练习题	(64)
3.6 小结	(64)
第4章 使用 ListView 控件	(65)
4.1 ListView 控件的基本功能和特点	(65)
4.1.1 ListView 控件的特点	(65)
4.1.2 ListView 控件的结构	(66)
4.2 创建 ListView 控件	(67)
4.2.1 创建 ListView 控件	(67)
4.2.2 设置 ListView 控件的属性	(67)
4.3 扩展 ListView 控件	(71)
4.3.1 加入 ListView 项	(71)
4.3.2 向 ListView 中加入图像	(73)
4.3.3 删除 ListView 项及图像	(74)
4.4 使用 ListView 控件	(75)
4.4.1 存取 ListView 项的序号	(76)
4.4.2 增加 ListView 项	(77)
4.4.3 改变控件中数据的显示方式	(77)
4.4.4 使用报表视图	(78)

4.5 使用 ListView 控件的事件	(79)
4.6 练习题	(81)
4.7 小结	(81)
第 5 章 使用弹出式菜单	(82)
5.1 弹出式菜单的基本功能和特点	(82)
5.1.1 关于菜单	(82)
5.1.2 弹出式菜单	(83)
5.1.3 弹出式菜单的优缺点	(84)
5.2 创建弹出式菜单	(85)
5.3 显示弹出式菜单	(87)
5.3.1 显示弹出式菜单的步骤	(87)
5.3.2 利用键盘显示菜单	(88)
5.4 练习题	(89)
5.5 小结	(89)
第 6 章 使用 Tab 控件	(90)
6.1 标签控件的基本功能和特点	(90)
6.1.1 标签控件的基本概念	(90)
6.1.2 标签控件的功能和特点	(91)
6.2 创建 Tab 对话框	(92)
6.2.1 创建标签控件及标签页	(92)
6.2.2 访问标签对话框对象	(95)
6.2.3 标签控件的属性	(98)
6.2.4 标签控件的属性列表	(102)
6.3 创建一个祖先标签页	(102)
6.3.1 创建祖先标签页的步骤	(102)
6.3.2 动态地创建标签页	(103)
6.3.3 动态访问标签属性	(105)
6.3.4 标签控件的事件和函数	(105)
6.4 练习题	(106)
6.5 小结	(106)
第 7 章 使用 OLE 容器	(107)
7.1 部件的基本概念	(107)
7.1.1 部件的功能	(108)
7.1.2 关于 ActiveX 控件	(108)
7.1.3 注册 ActiveX 控件	(109)
7.1.4 OLE 容器	(110)
7.2 选择 ActiveX 控件	(111)
7.3 使用 ActiveX 控件的属性、方法和事件	(113)
7.3.1 浏览和操纵一个 ActiveX 控件	(113)
7.3.2 设置容器的属性	(115)

7.3.3 OLE 控件的属性列表	(119)
7.3.4 设置 ActiveX 控件的属性	(120)
7.3.5 使用 ActiveX 控件的属性	(120)
7.3.6 事件的冲突	(121)
7.4 OLE 的自动化	(122)
7.4.1 什么是 OLE 的自动化	(122)
7.4.2 自动化脚本	(123)
7.4.3 调用部件函数	(124)
7.4.4 OLE 异常处理自动化	(126)
7.5 分发含有部件的应用	(129)
7.6 练习题	(130)
7.7 小结	(130)
第 8 章 实现 OLE 类型的数据窗口	(131)
8.1 OLE 2.0 数据窗口表现风格	(131)
8.1.1 数据窗口的 OLE 对象	(131)
8.1.2 OLE 2.0 数据窗口表现风格	(131)
8.1.3 OLE 对象与 OLE 表现风格	(132)
8.2 统一数据传输器	(132)
8.2.1 什么是统一数据传输器	(132)
8.2.2 OLE 2.0 数据窗口与 UDT	(134)
8.3 创建一个 OLE 2.0 数据窗口	(134)
8.4 数据窗口中的 OLE 2.0 容器	(138)
8.4.1 在数据窗口中放置一个 OLE 容器	(139)
8.4.2 操作服务器应用	(140)
8.4.3 在数据窗口对象中使用 OLE 列	(141)
8.5 OLE 2.0 自动化和数据窗口	(146)
8.6 练习题	(146)
8.7 小结	(146)
附录 A 练习题答案	(148)
附录 B 对象和控件的属性、事件和函数	(154)
附录 C 中英文词汇对照表	(193)

第 1 章 概 述

PowerBuilder 作为一种典型的面向对象的、可视化的、专业的客户/服务器开发工具,不仅具备了支持客户机/服务器结构、多媒体、Internet 开发等功能,而且还具有同时支持多种关系数据库管理系统的强大功能。

PowerBuilder 6.0 是 1997 年推出的最新版本,每一次新版本的推出都增加了很多新的功能,使得该产品更加完善。其中,PowerBuilder 对于控件技术的支持也是相当出色的,控件的种类更丰富,操作更灵活,功能也更强大。

PowerBuilder 提供了大量的 Window 控件,以满足开发人员的不同需要。既有标准的控件,又有用户根据需要在标准控件的基础上重新组合定义生成的自定义控件,以及通过定义为用户对象使用的 PowerBuilder 的外部对象或者 Active X 等控件。

PowerBuilder 中的标准控件按照功能划分包括:

- 显示并(或)操纵值,如静态文本(StaticText)、单行编辑器(SingleLineEdit)等;
- 做出选择,如单选钮(Radio Button)、复选框(CheckBox)、列表框(ListBox)等;
- 实施动作,如命令按钮、图形命令按钮;
- 操纵管理数据,如 PowerBuilder 中最富特色的数据窗口(Datawindow)控件;
- 复杂列表组织数据,如列表浏览(ListView)和树状浏览(TreeView);
- 图形化显示信息,如图形(Graph)、水平滚动条(HscrollBar)等;
- 装饰美化窗口,如直线(Line)、图像(Picture)等;
- 组织其他控件,如分组框(GroupBox)、标签页(Tab)。

本书的重点就是介绍一些功能更强而使用略微复杂的标准控件以及它们的高级使用方法,还要介绍如何使用 PowerBuilder 以外的应用程序所提供的控件(如 OLE、Active X 等)来丰富和增强 PowerBuilder 应用程序的功能。

PowerBuilder 6.0 的工作界面如图 1.1 所示。

用户在使用 PowerBuilder 开发应用程序时,使用得最多的将会是各种各样的系统对象。可以说整个应用程序是由两部分组成的,一部分是系统对象,另一部分是进行数据处理和流程控制的 PowerScript 语言。本章将对 PowerBuilder 系统对象的有关概念做简单的介绍,以便用户在使用该产品的各种控件之前,能够对系统对象有一个整体的印象和较全面的了解。

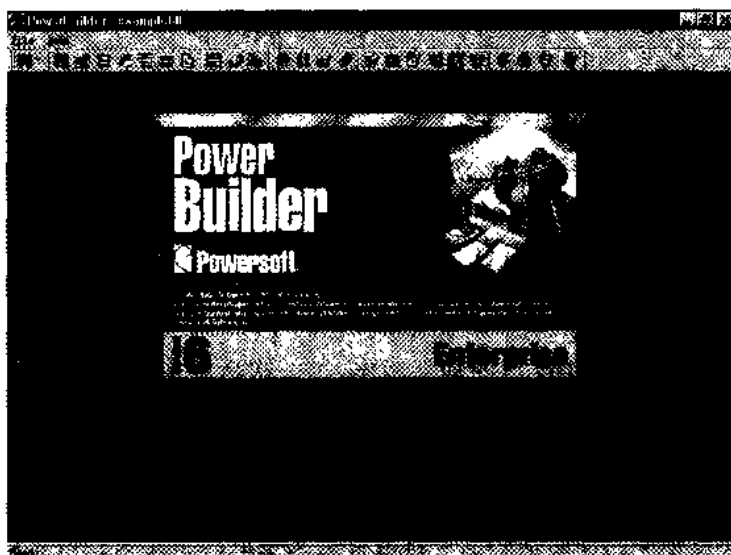


图 1.1 PowerBuilder 6.0 窗口

1.1 系统对象、属性、事件和函数

用户应用程序通过选择各种系统,设置它们的属性,在对象的事件中写入相应的处理程序,调用各种嵌入式函数,来完成各种功能。在使用它们之前,先对这些概念做一介绍。

1.1.1 PowerBuilder 系统对象

PowerBuilder 系统对象指的是用户在开发应用程序时,使用的嵌入式对象。所谓嵌入式对象,也称为内嵌对象,指的是由 PowerBuilder 提供的一系列封装好的、可供用户直接使用的对象,这些对象具有不同的属性、事件和函数。用户可以像盖房子一样,将它们嵌入到应用程序中,不需要考虑它们底层的具体结构。这些对象能够很好地和用户的应用程序结合在一起,使之成为应用程序的一部分,而丝毫没有嵌入的痕迹。PowerBuilder 系统对象包括图像控件、窗口、菜单等各种控件,当然也包括用户在应用程序中预定义的实体,比如消息和错误对象。

PowerBuilder 控件是系统对象的子集,用户可以放置在窗口或是其他对象中,用来实现和丰富应用程序的各种功能。比如在设计窗口的时候,放置一些图形对象,使得窗口更加美观。图 1.2 就是 PowerBuilder 窗口对象和常用的控件。

PowerBuilder 6.0 为用户提供的控件分类请参见附录。

PowerBuilder 系统结构是指包含了相互间有联系的系统对象的集合,它也是系统对象的子集。例如环境对象就是一个包含了 PowerBuilder 应用程序运行的计算机信息的结构。

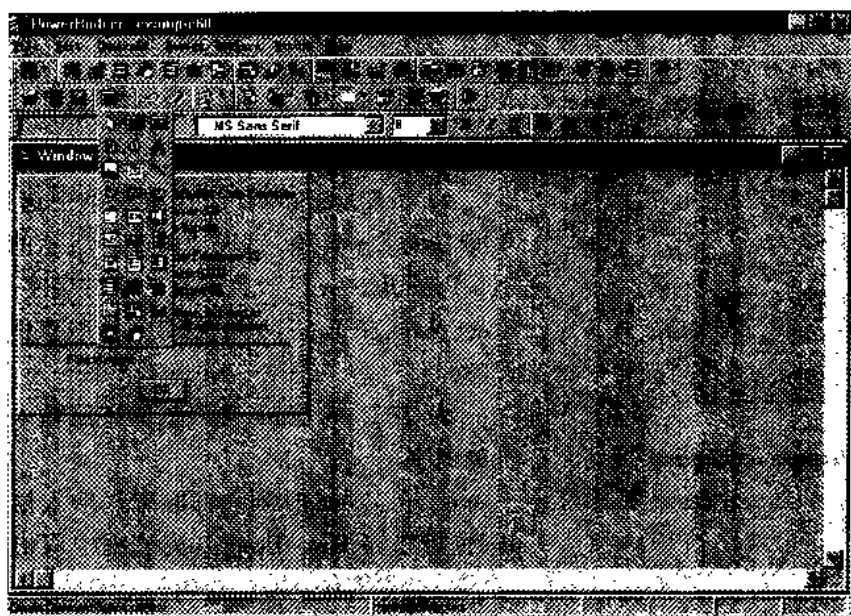


图 1.2 PowerBuilder 窗口对象和常用控件

1.1.2 对象的属性

属性指的是反映对象本身性质的各种特性。每一个系统对象都具有很多种属性,通过对各种属性的规定,来定义它们的特性。例如 CheckBox 控件利用 Height 和 Width 的属性来控制它的大小,利用 Background Color 属性来控制它的背景颜色。用户能够利用对象的属性标签页或者是具体的程序脚本来设置对象的属性值。

1.1.3 对象的事件

事件指的是作用在对象上的动作。PowerBuilder 的应用程序是事件驱动的,即用户通过事件的触发和相应的程序来控制应用程序的运行。比如说,当用户用鼠标点中一个按钮,从菜单中选定一项,或是在编辑框中输入数据时,一个事件就被触发了。用户可以编写相应的 PowerScript(PowerBuilder 使用的编程语言)脚本,来指定当事件被触发时要进行的处理动作。PowerBuilder 能够传递参数给事件,帮助用户的应用程序判断在事件触发时要去执行哪一部分程序。另外,对大部分事件来说,还可以通过设定返回值来决定下一步的流程,或者是去触发另一个事件。

PowerBuilder 的大部分控件(GroupBox 和 Drawing(直线、椭圆、长方形和正方形)控件除外)都有很多相关事件。但也有一些系统对象,比如系统结构,就没有任何与之相关事件。

虽然 PowerBuilder 系统本身提供了很多的事件,这些预先定义的事件几乎能够满足通常情况下的需求,但是,有时候还希望窗口、用户对象或控件声明自己的事件。这些用户自己声明的事件称为用户事件。在解决窗口与用户对象之间的通信问题、解决按键的处理问题或是解

决多种方法完成一个功能的问题时,可能就会用到用户事件。

给窗口、用户对象或者控件定义用户事件的操作步骤如下:

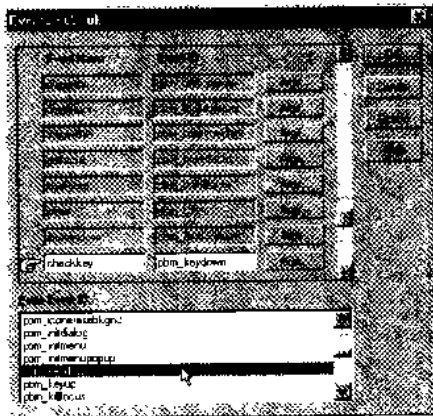


图 1.3 Events 对话框

一个 PowerBuilder 定制的事件(pbm_custom01 到 pbm_custom75)。

f. 单击 OK 按钮,结束操作。

a. 打开窗口画板或用户对象画板。

b. 选定要定义事件的窗口、用户对象或控件。

c. 从 PowerBuilder 工作菜单条中选择 Declare 菜单的 User Events 项,出现如图 1.3 所示的 Events 对话框。PowerBuilder 已经预定义的事件名字和 ID 显示在 Event Name 和 EventID 列表框中,这些事件都是不可修改的。

d. 在 Event Name 列表框的末尾输入要声明的事件的名字。

e. 在 Event ID 列表框的末尾输入要声明事件的 ID,也可以在 Paste EventID 列表框中双击所需要的事件 ID。如果事件不对应任何一个 Window 消息,选择一个

1.1.4 内置式函数

PowerScript 提供了丰富的内置式函数类,每个函数可以完成某种特定的功能,使用户可以在应用程序中随心所欲地使用。所谓内置式函数,也称为内嵌函数,是 PowerBuilder 提供的,用户在编程时可以直接调用的函数。这些函数在程序脚本中被调用,用以完成大量的应用处理工作。每一个系统对象,都提供了一套内置式函数供用户使用操纵。此外,用户还可以通过自己创建的函数,来完成对应用程序的特殊处理。

函数可以分为以下三种类型:

- 系统函数(System function by category);
- 控件函数(Function for controls);
- 系统对象函数(Function for system objects)。

系统函数按照功能可以分为支持动态 SQL 的函数、动态数据交换函数、日期函数、时间函数、字符串处理函数、数字处理函数、Blob 类型函数、数据类型转换函数、打印函数、文件支持函数、邮件函数、外部函数、支持函数、PowerBuilder 库管理函数、特殊函数以及在数据窗口中使用的统计函数等类型。

对每一种控件来说,系统提供的控件函数都不相同,这些函数只能用于相应的控件,对其他的控件来说,是不能被调用的。

系统对象函数,用来调用对象的事件处理程序、操作对象,并返回有关对象的信息。每一个系统对象,都提供了不同的函数,这些函数和控件函数一样,也只能用于各自的系统对象。

附录 B 为用户提供了绝大部分对象和控件的属性、事件和函数的列表。用户如果希望更为详细地了解对象的属性、事件和函数,可以查看 PowerBuilder 提供的在线帮助。单击 Power-

Builder 窗口中菜单条上的 Help,或直接按键盘上的[F1]键,就能进入在线帮助对话框,如图1.4所示。

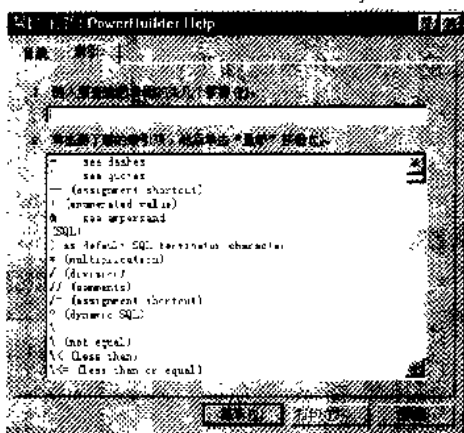


图 1.4 PowerBuilder 在线帮助对话框

用户可以在“索引”页面中的检索栏中输入要查找对象名称的头几个字符或是整个单词,比如 Checkbox,光标就会根据用户输入的单词,自动去定位匹配,然后单击“显示”按钮,就能了解关于该对象所有的详细信息,不仅包括对事件、属性和函数的详细讲解,还有部分具体的示例。

1.2 系统对象的继承层次

用户在开发应用程序时,可能会遇到这样的情况:需要创建一批界面、功能都很相似的窗口对象。如果一个个地创建,然后再对每个窗口分别编程,将会是一件极其繁琐的事情,而且其中包含大量的重复劳动,这会导致效率低下。但这些工作又是必不可少的,如何解决这个矛盾呢? PowerBuilder 为用户提供了一种继承的功能,利用它可以大大简化这种情况下的工作量,使用户方便快捷地完成任务。

PowerBuilder 最显著的特色之一就是继承。用户在创建窗口、菜单等各种对象时,可以利用已经存在的对象派生出新的对象来,这个过程就叫做继承,而被派生出来的对象和原来的对象具有完全相同的性质。当用户用一个对象继承创建另一个新的对象时,就产生了一个层次(或者称为树型结构),这两个新旧对象分别称为后代(子)对象和祖先(父)对象。

在整个层次最上端的对象为基类对象,而所有其他的对象都是这个对象的后代对象,每一个后代都从它的祖先身上继承信息。

使用继承不仅能够使应用中同类型的对象风格一致,而且不仅当祖先对象发生变化时,后代对象也会跟着变化,使用户的应用程序更容易维护。后代对象继承了原对象的所有内容,而且可以修改其属性,增加新的控件,增加事件处理程序,定义新的函数、结构、变量和对象。因此,根据不同的需要,每一个后代能通过进行适当的修改和调整,达到各自的目的。需要注意的一点是,不能删除后代对象的控件。

PowerBuilder 系统对象构成了这样一个层次,在最顶端的基类对象被称为 PowerObject,而所有其他的对象和控件都称为它的后代。这个层次还包含了一些其他类的基类对象,它们对整个应用程序的开发可能并不重要,但是对整个逻辑继承结构来说,却是必不可少的一部分。

【例 1.1】通过继承创建窗口的操作步骤如下:

a. 单击 PowerBuilder 工具条中的窗口(Window)画笔图标,打开 Select Window 对话框,如图 1.5 所示。

b. 单击对话框中的 Inherit...按钮,打开 Inherit From Window 对话框,如图 1.6 所示。

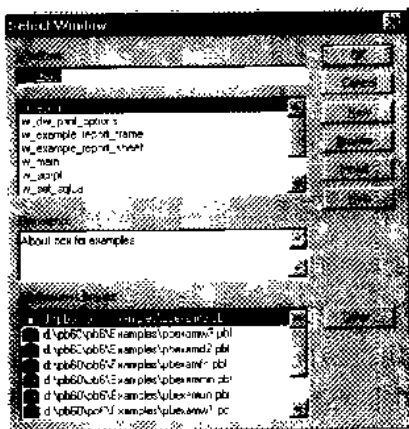


图 1.5 Select Window 对话框

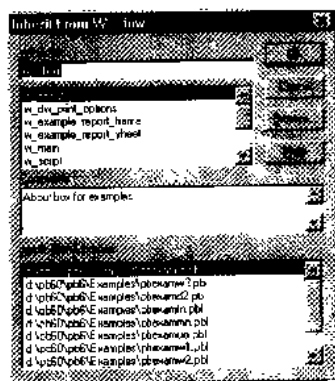


图 1.6 Inherit From Window 对话框

c. 在对话框列出的所有可以继承的窗口中选择被继承的窗口,即祖先窗口,再选择第一个窗口 W_about,然后单击 OK 按钮,就通过继承生成了新的窗口,如图 1.7 所示。

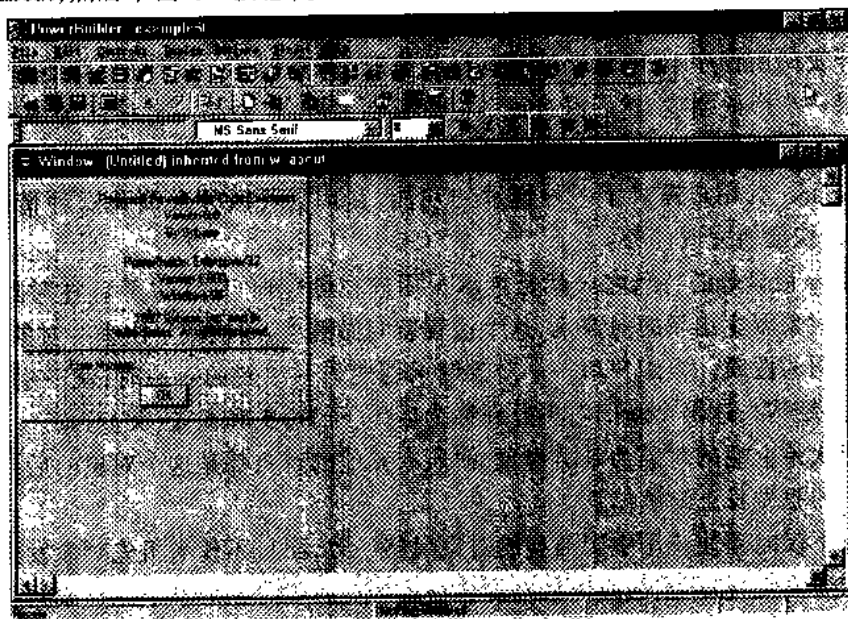


图 1.7 继承 W_about 生成的新窗口

d. 单击 PowerBuilder 主菜单 File 中的 Save, 或按键盘上的 [Ctrl] + [S] 键, 将新的继承窗口命名, 并保存, 如图 1.8 所示。余下的工作就是修改继承的后代窗口了。

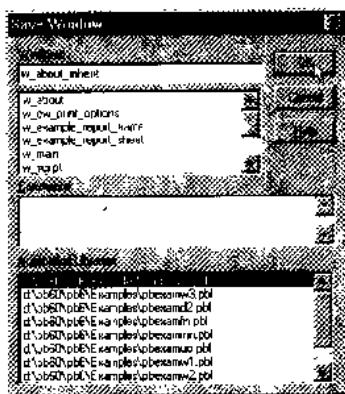


图 1.8 保存后代窗口为“w_about_inherit”

1.3 查看系统对象

用户使用 PowerBuilder 开发较大的应用程序时, 会创建许多的窗口、菜单、函数或是数据窗口, 定义各种类型的变量, 使用各种事件进行编程, 也许还需要好几个应用程序分工合作来完成功能。随着用户应用程序的不断增大, 各种控件和对象的不断增多, 用户自己可能都不是非常清楚究竟定义了多少变量, 创建过多少窗口, 这样有时可能会造成一些混乱。尤其是几个用户共同开发一个项目的时候, 清楚地了解各自应用程序所包含的内容, 就更显重要。因此, 用户非常需要有一个“管家”, 能够清晰地列出应用程序中包含的详细内容, 不仅是所有的对象、控件, 还应该有各种属性、事件, 以及各种数据类型等方面。

PowerBuilder 为用户提供的“管家”叫做浏览器 (PowerBuilder Browser), 用户可以使用它查看所有系统对象的列表, 以及它们的属性、事件、函数和各种用户定义的各种局部、全局变量。

在 PowerBuilder 系统工具条上, 有一个称为“浏览器” (Browser) 的画笔图标, 如图 1.9 所示, 用鼠标单击该图标, 就可以打开浏览器, 显示系统对象。

图 1.10 显示的就是打开 Browser 之后的系统对象对话框, 缺省显示的是按照字母顺序列出的当前应用的标签页。

可以看到, 该浏览器有很多标签页, 将用户的应用程序的内容, 按照不同的范畴分成了十二类。它们分别是: 应用 (Application)、数据窗口 (Data Window)、窗口 (Window)、菜单 (Menu)、用户对象 (User Object)、函数 (Function)、代理 (Proxy)、系统 (System)、枚举 (Enumerated)、结构 (Structure)、数据类型 (Data Type) 和对象连接嵌入 (OLE)。

整个浏览器的主体分为左右两个区域 (OLE 标签页除外)。左边的区域根据上述的 12 种分类方法, 列出了应用包含的全部内容, 比如: “窗口”标签页, 就列出了应用中的所有创建的窗口, “菜单”标签页, 列出的就是应用中的所有创建的菜单。右边的区域, 列出的是每一类包含的所有属性 (Properties)、事件 (Events)、函数 (Functions)、外部函数 (External Functions)、实例变量

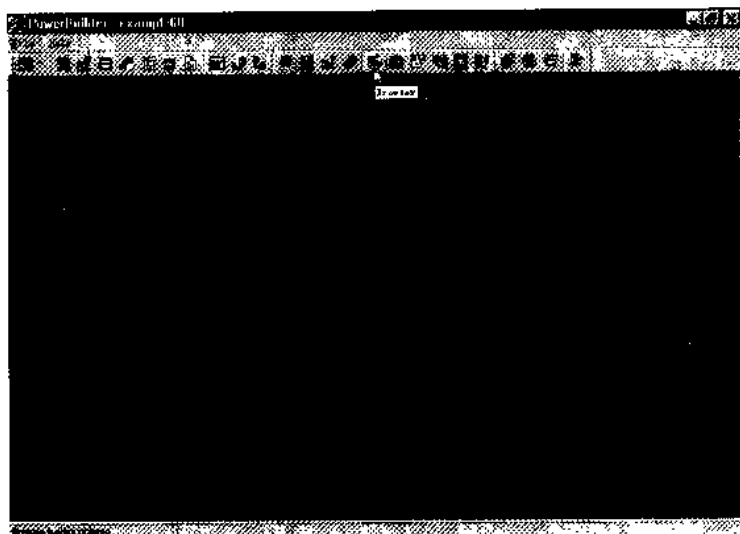


图 1.9 系统工具条上的 Browser 画笔图标

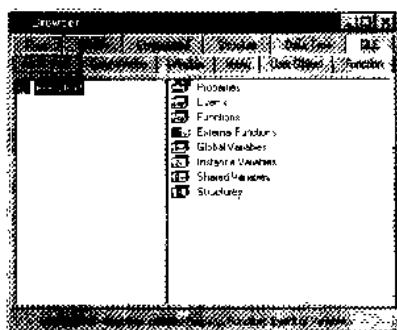


图 1.10 浏览系统对象对话框

(Instance Variables)、共享变量 (Shared Variables) 和结构 (Structures)。

要显示一个指定对象的属性、事件或者函数, 首先选中对话框左边区域中的某个对象, 然后双击右边窗口中该对象的属性、事件或函数项中的某一项, 即可进行查看。

【例 1.2】查看窗口 `w_about` 的属性和事件的操作步骤如下:

a. 单击 PowerBuilder 窗口工具条上的“浏览器”(Browser)画笔图标, 打开浏览器, 显示应用程序 `example60`, 如图 1.11 所示。

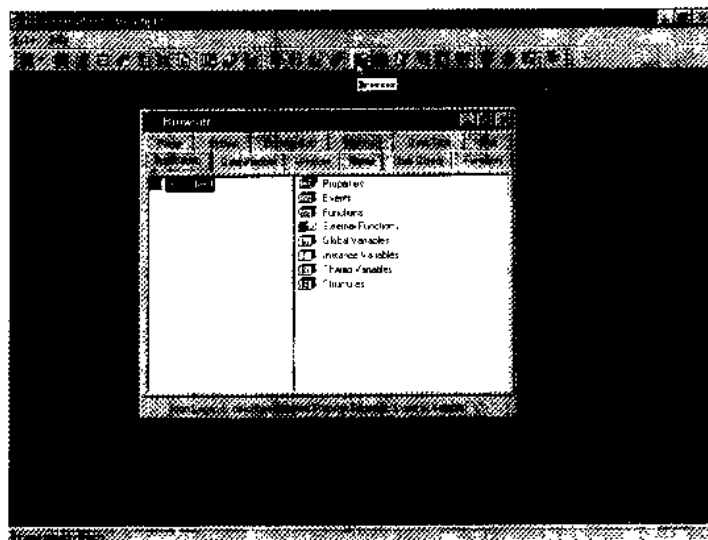


图 1.11 用浏览器打开 example60 应用