

PowerBuilder 6.0/6.5 技术丛书

刘红岩 张步达 主编

PowerBuilder 应用开发技术详解

王 蓉 等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书针对那些已经了解 PowerBuilder 开发环境、各种画板和工具的功能以及 PowerScript 面向对象语言并能够实现简单应用开发的用户,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。

全书分为五篇,共计 26 章。第一篇为用户界面技术篇,从构造 MDI 应用、管理窗口实例、实现标签等七个方面介绍了开发用户界面所涉及的一些常用技术。第二篇是数据存取技术篇,围绕着如何在 PowerBuilder 应用中实现复杂的数据处理功能而进一步介绍了事务对象、数据窗口、数据存储和数据管道等的实现方法。第三篇为程序实现技术篇,主要介绍了利用 DDE、OLE 以及其他扩展处理技术编程实现复杂的应用程序行为,尤其是应用程序之间通信、控制与交互功能的实现。第四篇为分布式应用技术篇,介绍了开发分布式应用所涉及的一些问题。第五篇为分发技术篇,讨论了分发 PowerBuilder 应用涉及的技术以及 Windows 平台和 Unix 平台上的具体分发技术。

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 应用开发技术详解/王蓉等编著. - 北京:电子工业出版社,1999.6

(PowerBuilder 6.0/6.5 技术丛书)

ISBN 7-5053-5259-8

I . P… II . 王… III . 数据库系统-软件工具,PowerBuilder - 基本知识 IV . TP311.56

中国版本图书馆 CIP 数据核字(1999)第 13103 号

丛 书 名: PowerBuilder 6.0/6.5 技术丛书

主 编: 刘红岩 张步达

书 名: PowerBuilder 应用开发技术详解

编 著 者: 王 蓉 等

策 划: 郭 立

责任编辑: 徐德霆

特约编辑: 贾 宇

排版制作: 电子工业出版社计算机排版室

印 刷 者: 北京天竺颖华印刷厂

装 订 者: 三河市金马印装有限公司

出版发行: 电子工业出版社 URL: <http://www.phei.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: 23.5 字数: 600 千字

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

书 号: ISBN 7-5053-5259-8
TP·2627

印 数: 6000 册 定价: 38.00 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 and 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 and 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

第一篇 用户界面技术

第 1 章 开发 MDI 应用程序	(3)
1.1 MDI 概述	(3)
1.1.1 什么是 MDI	(3)
1.1.2 MDI 框架窗口	(3)
1.1.3 框架	(3)
1.1.4 客户区	(4)
1.1.5 MDI Sheet	(5)
1.1.6 工具条	(5)
1.2 建立 MDI 框架窗口	(5)
1.3 使用菜单	(6)
1.4 使用 Sheet	(7)
1.4.1 打开 Sheet	(7)
1.4.2 列出所打开的 Sheet	(7)
1.4.3 排列 Sheet	(8)
1.4.4 Sheet 的最大化	(8)
1.4.5 关闭 Sheet	(8)
1.5 提供 MicroHelp	(9)
1.5.1 定义菜单项的 MicroHelp	(9)
1.5.2 定义对象的 MicroHelp	(9)
1.6 提供工具条	(11)
1.6.1 工具条	(11)
1.6.2 加入工具条的第一步	(13)
1.6.3 加入工具条的第二步	(15)
1.6.4 相关属性的设置	(17)
1.6.5 工具条的使用	(18)
1.7 调整客户区	(19)
第 2 章 管理窗口实例	(22)
2.1 什么是窗口实例	(22)
2.2 声明窗口实例变量	(22)
2.2.1 打开一个实例	(23)
2.2.2 关闭一个实例	(23)
2.3 使用窗口数组	(23)
2.4 引用继承中的实体	(25)
第 3 章 标签的实现	(27)
3.1 标签控件	(27)
3.1.1 术语	(27)
3.1.2 标签页	(28)

3.2	定制标签控件	(29)
3.2.1	标签的排列方式	(30)
3.2.2	标签标识	(31)
3.3	在脚本中使用标签控件	(32)
3.3.1	引用标签页	(32)
3.3.2	引用标签页上的控件	(34)
3.3.3	打开、关闭及隐藏标签页	(34)
3.3.4	跟踪标签页	(35)
3.3.5	只创建所需标签页	(37)
3.3.6	标签控件事件	(37)
第 4 章	列表构造技术	(39)
4.1	使用列表框	(39)
4.1.1	创建列表框和 PictureBox	(39)
4.1.2	添加列表框和 PictureBox 的列表项	(39)
4.1.3	在 PictureBox 中添加图像	(40)
4.2	使用下拉式列表框	(41)
4.2.1	创建下拉式列表框和 DropDownPictureBox	(41)
4.2.2	添加下拉式列表框和 DropDownPictureBox 的列表项	(41)
4.2.3	在 DropDownPictureBox 中加入图像	(42)
4.2.4	删除 PictureBox 或 DropDownPictureBox 中的图像	(42)
4.2.5	实例	(43)
4.3	使用 ListView	(44)
4.3.1	创建 ListView	(44)
4.3.2	加入 ListView 列表项	(45)
4.3.3	向 ListView 中加入图标	(46)
4.3.4	删除 ListView 项及图标	(48)
4.3.5	使用报表视图	(48)
第 5 章	使用 TreeView 控件	(51)
5.1	TreeView 控件	(51)
5.1.1	TreeView 项的层次	(52)
5.1.2	TreeView 项的数据来源	(52)
5.1.3	TreeView 项的图标	(52)
5.1.4	TreeView 的外观	(53)
5.1.5	用户交互方式	(53)
5.2	生成 TreeView 项	(53)
5.2.1	插入项目函数	(54)
5.2.2	插入根级项目	(55)
5.2.3	插入根级以下的项目	(56)
5.3	管理 TreeView 项	(58)
5.3.1	删除项目	(59)
5.3.2	重命名项目	(59)
5.3.3	通过拖拽移动项目	(60)
5.3.4	对项目进行排序	(63)

5.4	管理 TreeView 图标	(64)
5.4.1	图标列表及属性	(64)
5.4.2	生成图标列表	(65)
5.4.3	使用覆盖图标	(66)
5.5	利用数据窗口生成 TreeView	(67)
第 6 章	拖拽技术	(70)
6.1	概述	(70)
6.2	拖拽属性、事件及函数	(71)
6.2.1	拖拽属性	(71)
6.2.2	拖拽函数	(71)
6.3	识别拖动控件	(72)
第 7 章	提供联机帮助	(73)
7.1	为开发者提供联机帮助	(73)
7.1.1	为开发者提供联机帮助	(73)
7.1.2	为用户自定义函数创建上下文有关的帮助	(74)
7.2	为最终用户提供联机帮助	(75)
7.2.1	调用 Windows 帮助	(75)
7.2.2	为 Windows 环境创建上下文帮助	(75)

第二篇 数据存取技术

第 8 章	事务对象	(79)
8.1	事务对象介绍	(79)
8.1.1	事务对象属性	(80)
8.1.2	Powersoft 数据库接口及相应的事务对象属性	(80)
8.2	使用事务对象	(83)
8.2.1	事务的基本概念	(83)
8.2.2	默认的事务对象——SQLCA	(84)
8.2.3	设置事务对象	(85)
8.2.4	从外部文件取值设置事务对象	(85)
8.2.5	与数据库连接	(86)
8.2.6	预览连接	(86)
8.2.7	关闭数据库连接	(88)
8.2.8	为多数据库连接定义事务对象	(88)
8.2.9	SQL 语句的错误处理	(91)
8.2.10	数据库事务对象共享	(91)
8.3	利用事务对象调用存储过程	(92)
8.4	DBMS 支持的存储过程特性	(99)
第 9 章	数据窗口技术	(101)
9.1	有关数据窗口对象	(101)
9.2	数据窗口对象与数据窗口控件的连接	(101)
9.2.1	使用数据窗口对象	(101)
9.2.2	数据窗口命名	(103)
9.2.3	修改数据窗口对象	(104)

9.2.4	运行时改变连接的数据窗口对象	(104)
9.3	访问数据库	(104)
9.3.1	为数据窗口对象设置事务对象	(105)
9.3.2	检索和更新数据	(106)
9.4	装入外部数据	(109)
9.5	在数据窗口中操纵数据	(109)
9.5.1	数据窗口控件如何管理数据	(109)
9.5.2	访问 Edit 控件中的文本	(110)
9.5.3	操纵 Edit 控件中的文本	(111)
9.5.4	ItemChanged 事件脚本	(111)
9.5.5	ItemError 事件脚本	(112)
9.5.6	访问数据窗口的数据项	(112)
9.5.7	其他数据窗口有关函数	(113)
9.6	存取数据窗口对象属性	(113)
9.7	处理数据窗口的运行错误	(114)
9.8	数据库的更新	(115)
9.8.1	数据窗口控件如何更新数据库	(115)
9.8.2	在程序中改变行和列的状态	(116)
9.9	创建报表	(117)
9.9.1	设计数据窗口对象	(117)
9.9.2	打印报表	(118)
9.10	使用嵌套报表	(118)
9.11	使用 Crosstab	(119)
9.12	生成 HTML	(121)
9.12.1	几种生成技术	(121)
9.12.2	显示控制	(124)
9.12.3	调用 SaveAs 函数	(125)
9.12.4	以 HTML Form 的形式显示数据窗口对象	(126)
第 10 章	动态数据窗口	(131)
10.1	修改数据窗口对象	(131)
10.2	创建数据窗口对象	(132)
10.2.1	Create 函数	(132)
10.2.2	数据窗口对象源代码的定义	(132)
10.3	提供查询功能	(134)
10.3.1	查询模式的工作方式	(134)
10.3.2	查询模式的使用方法	(135)
第 11 章	数据存储	(138)
11.1	数据存储概述	(138)
11.1.1	数据存储的概念	(138)
11.1.2	数据存储的功能	(138)
11.1.3	数据存储方法	(139)
11.1.4	不支持交互信息输入	(139)
11.2	使用数据存储	(140)

11.3	使用定制的数据存储对象	(141)
11.4	用数据存储访问并操纵数据	(143)
11.5	信息共享	(145)
11.6	举例	(146)
第 12 章	图表处理	(150)
12.1	改变图表属性	(150)
12.1.1	图表的显示属性	(151)
12.1.2	图表显示属性的引用	(151)
12.2	访问数据属性	(152)
12.3	ObjectAtPointer 函数	(155)
12.4	使用图表控件	(157)
第 13 章	数据管道	(160)
13.1	创建所需的对象	(160)
13.1.1	创建管道对象	(161)
13.1.2	创建支撑用户对象	(162)
13.1.3	创建窗口	(164)
13.2	初始化操作	(165)
13.3	启动管道操作	(167)
13.3.1	启动管道	(167)
13.3.2	管道监控	(168)
13.3.3	终止管道运行	(170)
13.3.4	向数据库提交更新操作	(171)
13.4	处理行错误	(171)
13.4.1	使用管道错误数据窗口	(171)
13.4.2	修复错误行	(172)
13.4.3	放弃错误行	(173)
13.5	结束操作	(173)

第三篇 程序实现技术

第 14 章	DDE 实现技术	(177)
14.1	DDE 介绍	(177)
14.2	客户和服务端	(177)
14.3	DDE 函数和事件	(178)
14.3.1	DDE 客户函数	(178)
14.3.2	DDE 客户事件	(178)
14.3.3	DDE 服务器函数	(178)
14.3.4	DDE 服务器事件	(179)
第 15 章	OLE 实现技术	(180)
15.1	PowerBuilder 对 OLE 的支持	(180)
15.2	窗口中的 OLE 控件	(181)
15.2.1	OLE 控件包容器的特征	(181)
15.2.2	定义 OLE 控件	(182)
15.3	OLE 控件及可插入对象	(185)

15.3.1	设置 OLE 控件	(185)
15.3.2	在画板中激活对象	(186)
15.3.3	改变控件中的对象	(186)
15.3.4	用户如何与 OLE 控件交互	(187)
15.3.5	链接与嵌入	(187)
15.3.6	即席激活和离席激活	(188)
15.3.7	即席激活方式下的菜单	(189)
15.3.8	在 OLE 控件中修改对象	(190)
15.4	OLE 定制控件	(193)
15.4.1	设置 OLE 定制控件	(193)
15.4.2	OCX 编程	(194)
15.5	可编程 OLE 对象	(196)
15.5.1	OLEObject 对象类型	(197)
15.5.2	OLE 自动化方案	(199)
15.6	在脚本中使用 OLE 对象	(202)
15.6.1	OLE 自动化接口	(202)
15.6.2	自动化及 Any 数据类型	(207)
15.6.3	使用高效的 OLEObject 变量	(207)
15.6.4	错误处理	(208)
15.6.5	创建热链接	(210)
15.6.6	底层访问 OLE 对象	(211)
15.7	操纵 OLE 的高级方法	(211)
15.7.1	OLE 存储的结构	(212)
15.7.2	存储及流的对象类型	(213)
15.7.3	打开及保存 OLE 存储	(213)
15.7.4	打开流	(217)
15.7.5	使用 OLE 存储的策略	(220)
第 16 章	PowerBuilder 运行时自动化服务器	(222)
16.1	使用运行时自动化服务器	(222)
16.1.1	自动化服务器介绍	(222)
16.1.2	访问 PowerBuilder 用户对象的三种方法	(223)
16.2	用户对象作为自动化服务器	(225)
16.2.1	创建类用户对象	(225)
16.2.2	建立对象的运行库	(225)
16.2.3	注册对象	(226)
16.2.4	编写客户代码	(227)
16.3	PowerBuilder 作为自动化服务器	(228)
16.3.1	定义所需访问的对象	(228)
16.3.2	生成运行库	(229)
16.3.3	编写客户代码	(229)
16.4	创建并使用命名服务器	(231)
16.4.1	注册服务器	(231)
16.4.2	编写客户代码	(232)

16.5	通过自动化访问远程对象	(233)
16.6	用户对象及注册	(235)
16.6.1	GUID、CLSID 和 ProgID	(235)
16.6.2	对象信息存储地点	(235)
16.6.3	创建注册信息	(237)
16.7	分发自动化服务器应用	(240)
16.8	PowerBuilder.Application 服务器对象	(240)
第 17 章	在数据窗口中使用 OLE	(242)
17.1	数据窗口对 OLE 的支持	(242)
17.2	OLE 对象及 OLE 显示风格	(243)
17.2.1	在数据窗口中加入 OLE 对象	(243)
17.2.2	使用 OLE 显示风格	(244)
17.2.3	预览数据窗口	(245)
17.2.4	定义 OLE 对象数据	(245)
17.2.5	属性窗口中的附加设置	(246)
17.2.6	在事件程序中操纵 OLE 对象	(247)
17.3	在数据窗口中使用 OLE 列	(249)
17.3.1	创建一个 OLE 列	(249)
17.3.2	预览数据窗口列	(252)
17.3.3	在应用程序中使用 OLE 列	(253)
第 18 章	增加扩展处理功能	(255)
18.1	外部函数的使用	(255)
18.1.1	外部函数的声明	(255)
18.1.2	外部函数声明示例	(256)
18.1.3	传递参数	(257)
18.2	在 Unix 上调用外部函数及程序	(259)
18.2.1	调用外部函数	(259)
18.2.2	运行其他程序	(260)
18.3	发送窗口消息	(260)
18.3.1	Post 和 Send 函数	(261)
18.3.2	触发 PowerBuilder 事件	(261)
18.4	消息对象	(261)
18.5	上下文信息	(262)
18.5.1	上下文信息服务	(263)
18.5.2	关键字服务	(266)
18.5.3	Internet 服务	(267)
18.6	使用实用函数管理信息	(269)

第四篇 分布式应用技术

第 19 章	PowerBuilder 环境下的分布计算	(273)
19.1	为何采用分布式计算	(273)
19.2	分布式应用的体系结构	(274)
19.2.1	服务器应用	(275)

19.2.2	客户应用	(275)
19.2.3	通信驱动程序	(277)
19.3	设计分布式应用的指导性原则	(277)
19.4	分布式应用的开发与测试	(279)
19.5	分布式计算及 World Wide Web	(279)
第 20 章	客户应用的开发	(280)
20.1	创建用户界面	(280)
20.2	与服务器连接	(280)
20.3	激活远程对象函数	(281)
20.4	生成代理对象	(284)
20.5	处理通信错误	(285)
20.5.1	编写连接对象的 Error 事件脚本	(286)
20.5.2	编写应用对象的 SystemError 事件脚本	(286)
20.6	管理程序变量空间	(287)
20.7	使用 ConnectionInfo 对象	(287)
20.8	分发客户应用	(288)
第 21 章	服务器应用的开发	(289)
21.1	创建用户界面	(289)
21.2	管理客户连接	(289)
21.2.1	监听客户连接	(289)
21.2.2	处理客户请求	(291)
21.3	建立远程对象	(291)
21.3.1	编写用户对象方法	(292)
21.3.2	设置远程对象	(293)
21.4	数据库存取	(294)
21.4.1	执行检索操作	(294)
21.4.2	数据库更新	(296)
21.4.3	数据库事务共享	(302)
21.5	使用共享对象	(303)
21.5.1	共享对象的工作原理	(303)
21.5.2	执行过程	(304)
21.6	处理异步请求	(307)
21.7	向客户推送消息	(307)
21.8	避免死锁	(308)
21.8.1	死锁产生的原因	(308)
21.8.2	如何避免死锁	(309)
21.9	分发服务器应用	(310)
第 22 章	分布式应用的故障测试	(311)
22.1	关于测试分布式应用	(311)
22.2	用 Local 驱动程序测试	(311)
22.3	在本地使用通信驱动程序测试	(312)
22.4	使用跟踪功能	(313)
第 23 章	开发多平台应用程序	(315)

23.1	交叉平台设计策略	(315)
23.2	在多台环境开发和分发	(315)
23.2.1	在不同平台之间移动和共享文件	(315)
23.2.2	PowerBuilder 库文件	(316)
23.2.3	图像文件	(316)
23.2.4	文本文件	(317)
23.2.5	INI 文件	(318)
23.2.6	可执行文件	(320)
23.2.7	机器码动态库	(320)
23.2.8	如何定位外部文件	(320)
23.3	根据当前平台执行程序	(321)
23.4	应用程序的外观设计	(322)
23.4.1	如何在窗口中放置控件	(322)
23.4.2	三维显示控件	(322)
23.4.3	字体的使用	(323)
23.5	窗口行为	(324)
23.6	菜单设计	(326)
23.7	其他方面	(327)
第五篇 分发技术		
第 24 章	封装应用	(331)
24.1	生成应用的可执行版本	(331)
24.1.1	编译基础	(331)
24.1.2	封装内容	(332)
24.1.3	选择封装模式	(334)
24.1.4	实现封装模式	(336)
24.1.5	测试可执行应用程序	(336)
24.2	交付给最终用户	(337)
第 25 章	Windows 上的分发技术	(339)
25.1	分发过程的自动化	(339)
25.2	安装分发软件包	(339)
25.3	实现分发过程	(340)
25.4	分发后的应用	(343)
第 26 章	Unix 上的分发技术	(347)
26.1	分发应用程序	(347)
26.2	安装分发软件包	(348)
26.2.1	安装前的预备知识	(348)
26.2.2	安装软件包	(348)
26.2.3	最终用户设置	(351)
26.3	其他交付方法	(353)
26.4	分发后的应用	(354)
附录	名词术语英中文对照表	(356)

第一篇

用户界面技术

用户通过用户界面与应用程序进行交互。如果用户界面设计、开发得不理想,会严重影响用户使用应用程序。本篇将介绍用户界面的一些实现技术。熟练掌握本篇介绍的用户界面技术,可以开发出用户界面更友好、更灵活的应用程序。

本篇包括:

- 第 1 章 开发 MDI 应用程序
- 第 2 章 管理窗口实例
- 第 3 章 标签的实现
- 第 4 章 列表构造技术
- 第 5 章 使用 TreeView 控件
- 第 6 章 拖拽技术
- 第 7 章 提供联机帮助

第 1 章 开发 MDI 应用程序

本章介绍如何利用 PowerBuilder 开发多文档界面(MDI)应用程序。

1.1 MDI 概述

1.1.1 什么是 MDI

MDI 是 Multiple Document Interface 的英文缩写,它的中文含义是多文档界面。MDI 代表着一种应用窗口的风格。在 MDI 应用中,有一个称为 MDI 框架(MDI Frame)的主窗口,在这个主窗口中可以打开多个子窗口(在 PowerBuilder 中称为 Sheet——工作表),并且可以在打开的子窗口之间进行切换。在主窗口中打开的子窗口只能在主窗口的范围内移动。

MDI 是一种很流行的应用风格,Windows 上的许多软件都采用这种风格。例如大家熟悉的 Microsoft Word 和 Microsoft Excel,实际上,PowerBuilder 开发环境的主窗口就是一个 MDI 窗口。若想建立一个多窗口的应用程序,并且希望在打开的多个窗口之间能方便地进行切换,则最好采用 MDI 界面风格。建立 MDI 应用,首先需要定义一个类型为 MDI Frame 或 MDI Frame with MicroHelp 的窗口,然后再根据需要创建一些只能在 MDI 框架范围中打开的子窗口(Sheet)。下面让我们来看看 MDI 应用的各个组成部分。

1.1.2 MDI 框架窗口

可以把 MDI 框架窗口看成是 MDI 应用的主窗口。一个 MDI 框架窗口由菜单条(Menu bar)、框架(Frame)、客户区(Client Area)、Sheet(工作表)以及可显示 MicroHelp(微帮助)的状态行五部分组成。MicroHelp 是对当前菜单项或当前动作的简短描述。图 1.1 标识出了 MDI 框架窗口的各个组成部分。

1.1.3 框架

MDI 框架(Frame)包容客户区的 MDI 窗口边框。PowerBuilder 提供了两种类型的 MDI 框架:标准的(Standard)MDI 框架和定制的(Custom)MDI 框架。

标准的 MDI 框架窗口带有一个菜单条,通常还有一个可显示 MicroHelp 的状态行。当没打开任何 Sheet 时,客户区是空的。每个 Sheet 既可以有自己的菜单,也可以继承 MDI 框架窗口