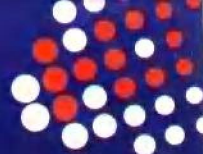


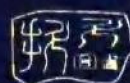
软件开发技术丛书



PowerBuilder 7

应用与开发

桂峰 陈海峰 等编著



机械工业出版社
China Machine Press

软件开发技术丛书

PowerBuilder 7

应用与开发

桂 峰 陈海峰 等编著
抖斗书屋 审



965404



机械工业出版社
China Machine Press

PowerBuilder是具有图形界面的客户/服务器模式及分布式数据库应用程序的前端开发工具,用以快速开发面向对象的数据库应用程序。PowerBuilder 7是Sybase公司推出的最新版本,它有许多崭新的特点。

本书包含两个部分。第一部分是PowerBuilder 7的操作使用指南,主要介绍该软件的使用方法和基本概念;第二部分是高级应用部分,着重讲解了在应用程序开发过程中使用到的一些高级技术,包括数据窗口、窗口控件的高级应用、拖放技术、邮件、API调用、DDE和OLE、访问注册表等等。

本书内容丰富、深入浅出,不同层次的读者或开发人员都能够从中受益。

版权所有,翻印必究。

图书在版编目(CIP)数据

PowerBuilder 7应用与开发/桂峰等编著. —北京:机械工业出版社, 1999.11
(软件开发技术丛书)

ISBN 7-111-07525-0

I. P… II. 桂… III. 数据库系统—软件工具, PowerBuilder7 IV. TP311.56

中国版本图书馆CIP数据核字(1999)第61268号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码: 100037)

责任编辑: 吴 怡

北京忠信诚胶印厂印刷·新华书店北京发行所发行

1999年11月第1版第1次印刷

787mm×1092mm 1/16·31.25印张

印数: 0 001-5 000册

定价: 49.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

前 言

数据库应用系统的解决方案往往伴随着计算机及网络技术的发展而不断地发展。在70年代出现的关系型数据库——dBase III，是最早的、成熟的桌面数据库产品，它的出现是和当时的单机以及单任务的DOS操作系统相适应的。以后，Oracle、Informix等数据库厂商经过长远发展，确立了自己在数据库市场上的地位。这一时期的产品大多是关系型数据库产品。进入90年代，出现了客户/服务器模式（Client/Server），它将一个数据库应用分为客户端与服务端部分，客户端仅向服务器发送应用请求，由服务器完成数据的存储及处理，然后把处理结果发送回客户端。

PowerBuilder是具有图形界面的客户/服务器模式及分布式数据库应用程序的前端开发工具，用以快速简洁地开发出面向对象的数据库应用程序。自它问世以来，就以其灵活强大的数据窗口、面向对象的快速开发能力和丰富的显示样式而在数据库应用程序开发领域独占鳌头，一直为很多数据库应用程序开发人员所喜爱。

PowerBuilder 7是Sybase公司推出的最新的版本。该版本在PowerBuilder 6.0的基础上做了很大的改变，尤其是在开发环境的集成、对Internet的支持、组件的开发与部署以及数据库的一些新特性等方面。

我们创作本书的目的，首先是向广大读者介绍PowerBuilder 7这一最新版本，它崭新的开发环境和适应Internet应用程序的新特点很值得开发人员关注；其次是要为开发人员提供一本可以从中得到帮助的参考书。我们以往使用PowerBuilder开发应用程序时，最大的感触是没有一本很实用的参考书，虽然目前市面上有很多关于PowerBuilder的使用和开发的图书，但是使用起来总觉得有这样或那样的不足。

本书是一本关于PowerBuilder 7的深入浅出的使用和开发指南，可满足不同层次读者的切实需要。

对于初学者，使用本书可以迅速了解PowerBuilder的开发思想和方法，并可以快速建立自己的数据库应用程序；对于有一定基础的中级水平的开发人员，本书提供了关于PowerBuilder 7的最新特点和使用方法，并提供了一些实用的范例，帮助他们能够很快地提高；对于高级开发人员，这也是一本不可多得的好书，因为它几乎包括了使用PowerBuilder开发应用程序的全部相关知识，并且还有许多融合了我们多年开发经验的技巧和建议，所以非常适合高级开发人员参考、备查。

我们编写本书时，力求与实际应用紧密结合，书中所讲解的内容以及程序代码都是用于解决在实际开发应用程序过程中常见的问题，在阅读本书、开发应用程序时，可以直接使用这些代码。

本书包含两个部分：第一部分是PowerBuilder 7的基本概念和操作使用指南，主要介绍该软件的使用方法和基本概念。第二部分是高级应用，分专题着重讲解了在应用程序开发过程中使用到的一些高级技术。

本书由中科辅龙计算机技术有限公司抖斗书屋策划，主要部分由桂峰、陈海峰编写，参

加本书编写工作的还有马小晶、瞿向荣、丁树军、蔡姝娟、谢慧、田军、陈建宏、赵甜、杜剑伟、高照宇、娄杰、周子社、谢忠华、李英济、张海波、白树波、桂钰、张凯、陈海樱、杨锡忠、王金星、李胜平、李宁、张磊、冯金慧、徐平、闫红梅、宋秀庆、原勇平等。全书由杨桂莲、石利文统稿，另外，郑红、郭美山、宋新波、白燕斌等做了部分审校工作。

在本书的编写过程中，抖斗书屋的全体员工付出了大量劳动，借此机会对书屋全体人员的精诚团结表示由衷的感谢！

由于时间仓促、作者水平有限，本书疏漏之处在所难免，欢迎广大读者批评指正。

对本书内容有疑问的读者可向抖斗书屋读者服务部提出咨询。

咨询电话：010－62565533-3301

作者

1999年春于中国科学院计算技术研究所

目 录

前言

第一部分 基本概念与使用

第1章 了解PowerBuilder	1
1.1 传统开发工具的不足	1
1.2 PowerBuilder的特点	1
1.2.1 专业的客户机/服务器开发工具	1
1.2.2 面向对象的开发工具	2
1.2.3 交叉平台的开发环境	2
1.2.4 强大、灵活的PowerScript	3
1.2.5 操作方便和有机结合的集成开发环境	3
1.2.6 数据窗口	4
1.2.7 对Web的完整支持	4
1.2.8 配套工具产品包	4
1.3 关于CPD	5
1.4 小结	5
第2章 PowerBuilder与DBMS	6
2.1 与主要的数据库管理系统连接	6
2.1.1 Oracle	6
2.1.2 Microsoft SQL Server	6
2.1.3 内置数据库管理系统Adaptive Server Anywhere 6.0	7
2.2 通用数据库接口ODBC	8
2.2.1 ODBC的工作原理	8
2.2.2 通过ODBCINST.INI查看数据库驱动程序	9
2.2.3 使用ODBC定义数据源	10
2.3 配置PowerBuilder中的数据库描述文件	12
2.3.1 Database Profile配置窗口	12
2.3.2 描述文件的设置	12
2.3.3 使用描述文件与数据库连接	15
2.3.4 导出与导入	15
2.3.5 PB.INI文件的使用	16

2.4 小结	17
第3章 第一个PowerBuilder应用程序	18
3.1 示例功能说明	18
3.2 准备工作	18
3.2.1 示例的数据库结构	18
3.2.2 为新的应用程序创建一个新文件夹	19
3.3 应用程序画板	19
3.3.1 启动PowerBuilder 7, 创建应用程序对象	19
3.3.2 给应用程序Lib_admin添加代码	20
3.4 数据窗口画板	21
3.4.1 创建数据窗口对象	21
3.4.2 设计数据窗口	23
3.5 窗口画板	27
3.5.1 创建窗口对象	27
3.5.2 设计窗口	28
3.5.3 为控件添加代码	30
3.6 运行应用程序	31
3.7 项目画板	32
3.8 回顾应用程序创建过程	33
3.9 小结	33
第4章 PowerBuilder 7的新特点	34
4.1 高度集成的开发环境	34
4.1.1 第一次启动PowerBuilder 7	34
4.1.2 以对象为中心的开发	35
4.1.3 非模态开发	35
4.1.4 不同画板子窗口的共同特征	36
4.1.5 子窗口的控制方法	37
4.1.6 画板的新特点	38
4.1.7 工作任务列表	42
4.1.8 定制快捷键	43
4.1.9 其他关于开发环境的新特征	44
4.2 数据库连接性	45
4.2.1 新的和增强的数据库接口	45

4.2.2 不再支持的数据库接口	46	5.4 用户表间关系	75
4.2.3 数据库描述文件的导出和导入	46	5.5 添加、更新和删除数据	76
4.2.4 快速访问数据库实用工具	46	5.5.1 SQL语句的执行环境	76
4.3 组件的开发和部署	47	5.5.2 向表中插入行	77
4.3.1 关于Jaguar CTS	47	5.5.3 更新表中的数据	79
4.3.2 创建Jaguar组件	49	5.5.4 从表中删除数据	81
4.3.3 创建Jaguar客户	49	5.6 通过交互方式操纵数据	82
4.3.4 创建COM/MTS组件	50	5.7 数据操纵的安全性	82
4.3.5 创建COM/MTS客户	50	5.8 SQL的数据查询功能	83
4.3.6 远程调试	50	5.8.1 SQL查询初步	84
4.3.7 创建自动服务器	51	5.8.2 无数据源检索	84
4.4 其他新特点	51	5.8.3 指定检索的列	84
4.4.1 ActiveX和OLE 特点	51	5.8.4 检索表中所有的列	85
4.4.2 Internet图像格式的支持	51	5.8.5 指定检索的行: WHERE子句	85
4.4.3 新的窗口控件	52	5.8.6 处理查询结果	89
4.4.4 使用存储过程更新数据库	52	5.9 高级SQL查询	93
4.4.5 数据窗口Web控件	52	5.9.1 多表连接	94
4.5 小结	53	5.9.2 使用子查询	94
第5章 SQL和PowerBuilder	54	5.10 视图	96
5.1 SQL的数据定义语言和PowerBuilder 的交互界面	55	5.10.1 视图的好处	96
5.1.1 创建和删除数据库	55	5.10.2 建立视图	96
5.1.2 创建和删除表	56	5.10.3 删除视图	98
5.2 数据的完整性	65	5.11 存储过程和触发器	98
5.2.1 数据完整性的类型	66	5.11.1 存储过程	98
5.2.2 用来实现数据完整性的特性	66	5.11.2 创建和执行存储过程	99
5.2.3 约束	67	5.11.3 触发器	99
5.2.4 使用IDENTITY性质实施数据 完整性	71	5.12 小结	101
5.3 索引	72	第6章 应用程序画板	102
5.3.1 使用索引的好处	72	6.1 应用程序对象	102
5.3.2 使用索引的坏处	73	6.1.1 新建应用程序对象	103
5.3.3 选择建立索引的列	73	6.1.2 选择当前应用程序对象	105
5.3.4 选择性	73	6.2 应用程序画板介绍	106
5.3.5 建立索引的准则	73	6.3 应用程序对象属性	106
5.3.6 不使用索引的准则	73	6.3.1 General属性	106
5.3.7 索引的特性	73	6.3.2 Toolbar属性	107
5.3.8 唯一索引	74	6.3.3 附加属性	108
5.3.9 复合索引	74	6.4 应用程序对象事件	109
		6.4.1 Open事件	109
		6.4.2 Close事件	110

6.4.3 Idle事件	111	7.7 语句	134
6.4.4 SystemError事件	111	7.8 PowerBuilder的长度单位	135
6.4.5 ConnectionBegin和ConnectionEnd 事件	111	7.9 系统对象	136
6.5 应用程序对象函数	112	7.9.1 Error	136
6.6 库搜索路径	112	7.9.2 Message	138
6.7 小结	113	7.10 小结	140
第7章 PowerScript编程	114	第8章 PowerScript编程环境	141
7.1 PowerScript基础	114	8.1 Script子窗口	141
7.1.1 大小写	114	8.2 定制编程环境	142
7.1.2 注释	114	8.2.1 Font选项页	142
7.1.3 断行、分隔符和继续符	115	8.2.2 Coloring选项页	142
7.1.4 标识符	115	8.2.3 AutoScript选项页	143
7.1.5 标签	115	8.2.4 Script选项页	143
7.1.6 保留字	116	8.2.5 Prefixes 1和Prefixes 2选项页	143
7.1.7 对象、属性和方法	116	8.3 编程工具和编译	144
7.1.8 操作符	117	8.3.1 注释	144
7.1.9 操作符的优先级	118	8.3.2 粘贴	144
7.1.10 ASCII字符	118	8.3.3 查找和替换	145
7.2 PowerScript数据类型	119	8.3.4 编译	146
7.2.1 标准数据类型	119	8.4 使用对象浏览器	146
7.2.2 增强数据类型	120	8.5 函数画板的使用	147
7.2.3 数据类型转换	123	8.5.1 新建全局函数	147
7.2.4 字符和字符串	123	8.5.2 修改全局函数	148
7.2.5 数组	124	8.6 结构画板的使用	148
7.3 变量	127	8.7 小结	149
7.3.1 变量的访问控制	127	第9章 窗口画板	150
7.3.2 变量的作用域	128	9.1 窗口对象类型	150
7.4 常量	129	9.1.1 主窗口	150
7.5 函数	129	9.1.2 子窗口	150
7.5.1 函数搜索链	129	9.1.3 多文档窗口	151
7.5.2 参数传递方式	130	9.1.4 弹出窗口	151
7.5.3 特殊类型的参数传递	130	9.1.5 响应窗口	152
7.5.4 函数调用方式	130	9.2 应用程序风格	152
7.5.5 使用Windows系统对话框	132	9.3 窗口画板	152
7.6 代词	132	9.3.1 窗口画板介绍	152
7.6.1 This	133	9.3.2 定制窗口画板	153
7.6.2 Parent	133	9.3.3 创建窗口对象	154
7.6.3 Super	134	9.3.4 打开窗口对象	154
		9.3.5 在窗口上放置控件	154

9.3.6 控件的操纵	155	9.10 控件的Tab Order	183
9.4 窗口对象属性	157	9.11 预览和运行窗口	184
9.4.1 常规属性	157	9.12 窗口举例	184
9.4.2 高级属性用法举例	158	9.12.1 w_center窗口	184
9.5 窗口对象事件	160	9.12.2 w_splash窗口	185
9.5.1 Activate和Deactivate事件	160	9.13 小结	186
9.5.2 Open事件	160	第10章 库管理	187
9.5.3 Close和CloseQuery事件	160	10.1 库的组织原则	187
9.5.4 Resize事件	161	10.2 Library画板	188
9.5.5 用户自定义事件	162	10.3 定制Library画板	188
9.6 窗口对象函数	162	10.4 Library画板的使用	189
9.6.1 ChangeMenu函数	162	10.4.1 库的操作	189
9.6.2 SetRedraw函数	162	10.4.2 对象的操作	190
9.6.3 Resize函数	163	10.5 对象安全控制	194
9.6.4 WorkSpaceHeight函数	163	10.5.1 注销对象	194
9.6.5 用于MDI窗口的函数	164	10.5.2 注册对象	195
9.7 Open与Close函数	164	10.5.3 撤销注销	195
9.7.1 Open函数	164	10.5.4 查看注销信息	195
9.7.2 Close函数	166	10.6 升级低版本应用程序	196
9.8 使用窗口控件	167	10.7 查看工具	196
9.8.1 命令按钮	168	10.8 小结	197
9.8.2 图形命令按钮	169	第11章 数据窗口对象	198
9.8.3 静态文本	169	11.1 概述	198
9.8.4 单行编辑框	169	11.2 数据窗口对象创建向导	199
9.8.5 多行编辑框	170	11.3 数据窗口的显示样式	201
9.8.6 掩码编辑框	170	11.3.1 Freeform样式	201
9.8.7 富文本编辑框	171	11.3.2 Grid样式	201
9.8.8 选择按钮和复选框	173	11.3.3 Tabular样式	201
9.8.9 列表框和下拉列表框	173	11.3.4 Label样式	202
9.8.10 选项卡控件	174	11.3.5 Crosstab样式	203
9.8.11 数据窗口控件	176	11.3.6 Group样式	203
9.8.12 OLE控件	177	11.3.7 N-Up样式	204
9.8.13 修饰作用的控件	178	11.3.8 Graph样式	204
9.8.14 用户自定义控件	179	11.3.9 Composite样式	205
9.9 PowerBuilder 7新增控件	179	11.3.10 RichText样式	206
9.9.1 HtrackBar和VtrackBar控件	179	11.3.11 OLE 2.0样式	207
9.9.2 HprogressBar和VprogressBar控件	182	11.4 数据源	207
9.9.3 StaticHyperLink 和PictureHyperLink 控件	183	11.4.1 Quick Select数据源	208
		11.4.2 SQL Select数据源	209

11.4.3 Query数据源	212	11.12 小结	246
11.4.4 External数据源	213	第12章 数据窗口控件	247
11.4.5 Stored Procedure数据源	213	12.1 数据窗口控件和数据窗口对象	247
11.5 数据窗口画板	214	12.2 数据窗口控件属性	247
11.5.1 子窗口简介	214	12.3 事务对象	248
11.5.2 定制数据窗口画板	215	12.3.1 事务对象属性	248
11.6 设计数据窗口对象	216	12.3.2 SQLCA	249
11.6.1 数据窗口属性	216	12.3.3 连接数据库	250
11.6.2 区域	218	12.3.4 事务管理	251
11.6.3 带的属性	219	12.4 为数据窗口控件分配事务对象	251
11.6.4 大小、对齐和间距	219	12.5 数据窗口控件事件	252
11.6.5 对象属性	220	12.5.1 Clicked事件	253
11.6.6 Tab次序	228	12.5.2 错误处理事件	253
11.6.7 创建分组	228	12.5.3 记录处理事件	256
11.6.8 压缩重复值	229	12.5.4 检索与更新事件	257
11.6.9 压缩字段间空间	229	12.5.5 打印事件	259
11.7 行操纵	230	12.6 数据窗口控件函数	260
11.7.1 过滤	230	12.6.1 缓冲区	260
11.7.2 排序	231	12.6.2 可编辑控件	262
11.7.3 检索	231	12.6.3 数据窗口控件函数	264
11.8 控制更新属性	232	12.7 小结	275
11.8.1 Key Modification组框	233	第13章 菜单和菜单画板	276
11.8.2 Where Clause for Update/Delete 组框	233	13.1 菜单设计原则	276
11.8.3 Identity Column选项	235	13.2 菜单术语	276
11.9 预览数据窗口	235	13.3 菜单画板	277
11.9.1 Describe对话框	235	13.3.1 菜单画板简介	277
11.9.2 Zoom属性	236	13.3.2 创建菜单对象	277
11.9.3 数据操纵	236	13.3.3 添加菜单项	278
11.9.4 导出和导入数据	236	13.3.4 Script子窗口	279
11.10 使用存储过程更新	237	13.4 菜单项属性	279
11.11 数据窗口增强对象	238	13.4.1 General属性页	279
11.11.1 Button对象	238	13.4.2 Toolbar属性页	281
11.11.2 Computed Field对象	239	13.5 菜单事件	282
11.11.3 Report对象	240	13.6 菜单函数	282
11.11.4 Group Box对象	241	13.7 菜单实例	283
11.11.5 Graph对象	241	13.7.1 Undo菜单的实现	283
11.11.6 OLE Database Blob	242	13.7.2 使用API菜单函数	284
11.11.7 其他对象	246	13.8 小结	286
		第14章 调试	287

14.1 调试画板	287	17.1 ListView控件	332
14.2 调试画板的使用	288	17.1.1 ListView控件属性	333
14.2.1 选择脚本	288	17.1.2 ListView函数	335
14.2.2 设置断点	289	17.1.3 ListView控件事件	337
14.2.3 调试应用程序	291	17.1.4 ListView控件实例	337
14.2.4 其他进入调试状态的方法	294	17.2 TreeView控件	340
14.3 PBDebug	295	17.2.1 TreeView控件属性	341
14.3.1 .dbg文件	295	17.2.2 TreeView控件函数	341
14.3.2 .pbp文件	297	17.2.3 TreeView控件事件	344
14.4 其他调试技术	302	17.2.4 TreeView控件实例	345
14.4.1 MessageBox函数	302	17.3 图形控件	349
14.4.2 Beep函数	303	17.3.1 PowerBuilder图形基础	349
14.4.3 SqlPreview事件	303	17.3.2 图形控件属性	350
14.5 小结	304	17.3.3 图形控件函数	355
第二部分 高级应用		17.4 小结	359
第15章 用户对象	305	第18章 数据窗口进阶	360
15.1 创建用户对象	305	18.1 访问属性的直接语法	360
15.1.1 设计用户对象	305	18.1.1 属性值	361
15.1.2 使用用户对象	307	18.1.2 表达式值	361
15.2 用户对象分类	308	18.1.3 示例	362
15.2.1 创建标准用户对象	308	18.2 Describe函数	363
15.2.2 外部用户对象	309	18.2.1 函数语法	363
15.2.3 自定义类	309	18.2.2 SQL应用	365
15.2.4 标准类	312	18.2.3 运算表达式	365
15.3 运行时创建用户对象	313	18.3 Modify函数	366
15.3.1 OpenUserObject函数	314	18.3.1 函数语法	367
15.3.2 OpenUserObjectWithParm函数	314	18.3.2 创建对象	367
15.3.3 CloseUserObject函数	315	18.3.3 删除对象	367
15.4 动态用户对象示例	315	18.3.4 修改对象属性	368
15.5 小结	320	18.3.5 使用表达式	370
第16章 MDI应用程序	321	18.4 DataWindow Syntax工具	370
16.1 MDI基本概念	321	18.4.1 Destribe窗口	370
16.2 工作表管理器对象	323	18.4.2 Modify窗口	371
16.3 重置用户工作区	325	18.4.3 SyntaxFromSQL窗口	372
16.4 工具栏的控制	326	18.5 访问数据的直接语法	373
16.5 MDI应用程序实例	328	18.5.1 按列名访问数据	373
16.6 小结	331	18.5.2 按列号访问数据	374
第17章 使用高级控件	332	18.5.3 访问选中行数据	375
		18.5.4 访问整行数据	376

18.5.5 数据访问小结	376	21.3.2 通过函数自动配置数据源	409
18.6 数据存储对象	376	21.4 小结	411
18.6.1 使用数据存储对象	377	第22章 PowerBuilder 7的邮件功能	412
18.6.2 使用数据存储对象的函数	377	22.1 PowerBuilder 7对MAPI的支持	412
18.7 动态创建数据窗口	377	22.1.1 Mailsession对象	412
18.7.1 Create函数	378	22.1.2 Mailsession对象具体用法	413
18.7.2 LibraryExport函数	378	22.2 发送数据窗口对象	424
18.7.3 SyntaxFromSQL函数	379	22.3 小结	424
18.8 在字段间移动焦点	381	第23章 拖放技术	425
18.9 小结	382	23.1 拖放技术的概念	425
第19章 PowerBuilder 7与API调用	383	23.2 拖放组件和规则	426
19.1 声明外部函数	383	23.2.1 拖放组件	426
19.2 数据类型转换	384	23.2.2 拖放规则	427
19.3 传递变量参数	384	23.3 实现拖放功能	427
19.3.1 传址方式传递参数	385	23.3.1 拖放对象的属性	427
19.3.2 传递结构	385	23.3.2 拖放对象的事件	428
19.4 使用外部函数用户对象	386	23.3.3 拖放对象的方法	429
19.5 调用外部函数实例	388	23.4 拖放编程示例	433
19.5.1 设计用户对象	388	23.4.1 示例简介	433
19.5.2 使用用户对象	389	23.4.2 设计方法	433
19.6 小结	389	23.5 小结	438
第20章 DDE与OLE	390	第24章 数据管道	439
20.1 共享资源	390	24.1 数据管道	439
20.2 动态数据交换	390	24.2 数据管道画板	439
20.2.1 DDE概述	390	24.2.1 源表	442
20.2.2 DDE概念	390	24.2.2 目标表	442
20.2.3 编写DDE应用程序	391	24.2.3 数据管道选项	443
20.2.4 DDE应用程序举例	395	24.2.4 执行数据管道	444
20.3 对象的链接和嵌入	399	24.2.5 数据管道错误	444
20.3.1 OLE概述	399	24.2.6 保存数据管道对象	445
20.3.2 编写OLE应用程序	400	24.2.7 处理Blob数据	445
20.4 小结	404	24.3 在应用程序中使用数据管道	446
第21章 PowerBuilder 7与注册表	405	24.3.1 管道用户对象	447
21.1 Windows 注册表	405	24.3.2 数据管道用户对象编码	449
21.2 在PowerBuilder 7中修改注册表信息	406	24.4 小结	451
21.2.1 PowerBuilder 7提供的函数	406	第25章 面向对象的概念与PFC编程	452
21.2.2 函数的具体用法	406	25.1 面向对象编程	452
21.3 应用示例	408	25.1.1 类	452
21.3.1 自动配置数据源的重要性	408	25.1.2 抽象	453

25.1.3 多态	454	25.5 添加服务	472
25.1.4 封装	455	25.6 小结	472
25.1.5 继承	455	第26章 应用程序的创建和分配	473
25.1.6 超类和子类	456	26.1 创建各组件	473
25.1.7 初始化类对象	456	26.1.1 可执行文件	473
25.1.8 析构类对象	456	26.1.2 资源文件	473
25.1.9 垃圾收集函数	456	26.1.3 动态库文件	474
25.2 面向对象的实例	457	26.2 Project画板	475
25.3 PFC组件	461	26.3 EXE文件和动态库文件的性能	477
25.3.1 PFC库	461	26.4 应用程序的分配	477
25.3.2 命名约定	462	26.5 小结	478
25.3.3 PFC对象服务	463	第27章 用InstallShield创建安装程序	479
25.4 设计PFC应用	463	27.1 InstallShield	479
25.4.1 应用程序服务	463	27.2 创建安装程序	480
25.4.2 菜单	465	27.2.1 使用项目向导建立项目	480
25.4.3 窗口	466	27.2.2 添加文件并修改代码	484
25.4.4 数据窗口与服务	468	27.3 卸载应用程序	487
25.4.5 标准可视用户对象	472	27.4 小结	488

第一部分 基本概念与使用

第1章 了解PowerBuilder

自1991年6月由Powersoft公司推出数据库应用开发工具PowerBuilder以来，它已逐渐发展成为开发客户机/服务器体系应用程序的首选开发环境。本章从传统的开发工具存在着的明显的不足开始，介绍PowerBuilder的主要特点，使没有接触过该开发环境的读者有一些总体上的了解。

1.1 传统开发工具的不足

随着技术的不断进步，用户对系统功能的要求越来越高，应用开发人员使用传统的开发工具很难满足这些要求。这主要表现在：

- 开发方法陈旧，兼容性差，难以适应变化。
- 功能弱，性能较低。
- 大多数应用程序只能使用特定的数据库接口，移植相当困难。

1.2 PowerBuilder的特点

总的说来，PowerBuilder是具有图形界面的客户/服务器模式以及分布式数据库应用程序的前端开发工具，它可以快速简洁地开发出功能强大的、充分满足用户需要的数据库应用软件。

读者可以从下面小节得到更为深入的了解。

1.2.1 专业的客户机/服务器开发工具

客户机/服务器工作方式已经成为企业应用程序开发的主流方式，PowerBuilder是用来进行客户/服务器开发的专业工具。

在传统的两层客户机/服务器体系中，用户界面和商业规则在PowerBuilder中创建并放置在客户工作站上，数据访问与操作在数据库管理系统中。虽然这种结构体系获得了相当的成功并在许多部门中得到应用，但是在大型企业或系统中，这种两层结构却存在以下两个需要解决的问题。

- 性能问题。在大型企业或系统中，应用程序生成的事务量是巨大的。为满足商业规则需要使用了大量的存储过程及触发器访问数据库，从而对数据库服务器造成了极大的压力。同时，在一些对反应时间要求较高的系统中，性能问题就更加突出。
- 应用程序发布问题。将商业规则放在三层结构的中间层可以解决应用程序发布时存在的问题，因为这样可以保证在主要使用某一版本的商业规则时所有的用户都使用这一相同版本。

为解决上述问题，从6.0版本开始，PowerBuilder支持所谓的三层结构或n层结构。三层客户机/服务器模型以构建分割式应用程序的能力为基础。对一个应用程序进行了分割以后，可将代码分为不同的逻辑组件。一旦用这种模型和实现技术开发了一个应用程序，每个组件都可以分配给发挥最佳性能的机器。具体分配取决于系统方案及当前的商业要求。这种结构的优点至少包括以下4个方面：

1) 可重复使用。投入组件设计和实施的时间与精力不会白费，因为以后可在不同的应用程序间共享。

2) 性能提升。由于可在除客户工作站以外的其他机器上配置组件，所以能将计算机负担从功能不强的客户机电脑转移到强大的服务器。利用这种配置与设计的灵活性，作为开发者，我们可充分利用计算资源，使应用程序的执行达到最佳状态。由此便能获得更好的性能。

3) 易于管理。将应用程序服务封装到各种组件以后，可将大型、复杂的应用程序划分为易于管理的模块。

4) 易于维护。由于将组件集中起来，以便重复使用，所以一旦需要对某种组件进行修改，便可以更容易地重新进行配置。这样便能随时适应商业需求的变化。

1.2.2 面向对象的开发工具

面向对象是软件开发的一种方法。这种方法产生的目标是增强应用程序代码的可重用性和可靠性，同时减少应用程序的维护工作。面向对象并不是一个很新的概念，它的许多方面都建立在早期的开发方法的基础之上，例如结构化的设计方法。

理解面向对象的思想很重要，在PowerBuilder中我们可以这样认为，面向对象的方法也就是基于组件的开发方法，开发人员可以使用已有的或者前面使用过的组件组装新的应用系统而不是重新开发一个新系统。

在PowerBuilder中有许多种对象，例如应用程序、窗口、控件、菜单等等，每一个对象就是一个应用程序组件，其中有许多称为属性的特性和称为方法（包括事件和函数）的行为。开发人员对这些对象及其属性和方法了解之后，使用事件驱动的编程方式，就能很快地得到应用程序的雏形，这样大大缩减了开发周期。

面向对象的开发方法的主要特点是：

- 继承性。继承性允许根据已经存在的祖先对象定义子孙对象，子孙对象继承得到其祖先对象定义的所有属性和方法。
- 封装性。对象的封装性使被封装的对象之间保持了最大的独立性，对一个对象的修改一般不会影响到其他对象。这种特性大大提高了对象的可重用性并减少系统维护的代价。
- 多态性。多态性是指不同的对象可以对相同的消息作出不同的反应，它的主要作用是消息的发出者不需要知道接受者的确切信息就可以发送消息。

在本书后面的章节里还要详细讨论面向对象方法的思想。

1.2.3 交叉平台的开发环境

今天的企业用户有很多的硬件选择，如果不是统一型号配置的话，开发能运行于多种平台的应用程序是一件非常麻烦的任务。也许C++一类的语言能解决这样的问题，但精通C++的程序员太少了。

PowerBuilder可以在Window 3.X、Window 95、Windows NT、Apple Macintosh以及Unix Motif上运行，开发者只须在一种平台上学习PowerBuilder，就能够在多个环境中开发应用程序。也就是说，你不仅可以在不同的操作系统中使用相同的开发环境，而且在一个系统中开发的应用程序，只须拿到另外的系统中重新编译，就可以直接运行了。

PowerBuilder使用了Environment对象来识别特定的与平台有关的信息，它是一个系统结构，通过访问该对象的相关属性就能得到需要的有关平台的信息。Environment对象拥有诸如OSType（平台类型）、Machinecode（16位或是32位平台）、Cputype（Cpu类型）、Osmajorrevision（操作系统的主版本号）等等属性。读者要开发适用于多平台的应用程序时应该充分使用Environment对象。

当开发者要使用PowerBuilder开发适用于多平台的商业应用程序时，要注意避免应用PowerBuilder的某些只与某一指定平台相关的特性，即应选择执行平台和调用平台都认可的一般解决方案。例如，一些PowerBuilder的有些函数只支持某一平台。另外，有一些Microsoft/Windows技术，例如MAPI、DDE、OLE、OCX、ActiveX等，在其他平台上并不支持，所以在开发多平台应用程序时也要尽量避免使用这些技术。

除了多平台支持外，现在的PowerBuilder还支持商业应用程序在不同的国家本地化，使应用程序能方便地在不同国家部署。这些功能主要体现在以下几个方面：

- 在Windows NT 4.x平台上的Unicode标准支持。
- 日文版Windows平台的日文DBCS（Double-Byte Character Support）支持。
- 在所有的32位平台上支持阿拉伯和希伯来语言的自右向左方式。

在本地化商业应用程序时需要注意的与语言有关的几个重要方面是：

- PowerBuilder应用程序。
- PowerBuilder运行的DLL。
- 操作系统（OS）。
- 数据库管理系统（DBMS）。

1.2.4 强大、灵活的PowerScript

PowerScript是PowerBuilder使用的面向对象的编程语言。它是由语句、命令、函数、用户自定义对象或类以及嵌入式SQL语句组成的。PowerScript是一种高级的、结构化的语言，它易学易用，对于那些使用过其他高级语言的用户更是如此。

具体地说，它是类似Visual Basic、QuickBASIC这样介于C/C++和现代BASIC之间的一种语言。

1.2.5 操作方便和有机结合的集成开发环境

PowerBuilder为开发人员提供了一个高度集成的工作环境，这个环境由一系列画板（Painter）构成，包括窗口、菜单、数据窗口、数据库、应用程序管理、调试等等。这种集成的环境的一个最大好处是能使上述诸对象有机地结合在一起，相互关联，而且给开发人员以极大的方便性。

在PowerBuilder 6.0中，调试器成为众多画板中首先实现非模态集成环境的画板，它包括多个子窗口和用户自定义子窗口，大大方便了开发人员的使用。在PowerBuilder 7中，所有的

画板都实现了非模态化, 开发环境高度集成, 这必将有效地提高开发人员的效率。

1.2.6 数据窗口

PowerBuilder的成功在很大程度上要归功于它的数据窗口能力。数据窗口是为正在被观察的数据保留数据信息的特有对象, 它为生成一个带有高质量的数据库事务处理的应用程序提供了强大的实现方法。可以说, 数据窗口是PowerBuilder的核心, 在对数据库的开发过程中, 无论是录入、查询、统计还是报表, 都离不开数据窗口, 并且能够以多种类型风格来显示数据。

数据窗口的处理机制首先是根据所需的显示方式将数据源表中的域设计好, 在运行时将所要的数据从数据库服务器上下载到客户机上处理, 然后将更改后的数据进行提交(Commit)或回滚(Rollback)。这种处理机制大大降低了数据的传输量, 减轻了网络的负担, 也提高了数据的处理速度。充分利用数据窗口的特点, 可达到事半功倍的效果。

1.2.7 对Web的完整支持

PowerBuilder对Web提供了完整的支持, 体现在以下方面:

- 对HTML、PB和Java客户机连接的支持。
- 用于Web客户机访问中层逻辑的Web.PB。
- 支持PB Window Plug-in和Window ActiveX。
- PB Web客户机的PowerBuilder VM支持。

1.2.8 配套工具产品包

Sybase公司为开发企业级应用提供了全面的支持, 并提供了以下工具产品, 与PowerBuilder配合工作。

- PowerJ。
- Power++。
- PowerDynamo。
- PowerBuilder Foundation Class(PFC)。
- PowerSite。
- PowerDesigner。
- Jaguar CTS。
- Adaptive Server Anywhere。
- Jaguar 3.0。
- AppModeler 6.1.2。
- Application Integrator 3.0。
- PowerSite 7.0。
- Objectcycle 2.0。
- InstallShield 5。
- Infomaker 6.0。