

PowerBuilder 8

Web 数据库开发指南

010101010100101010101010101010101001
010101010110101010100101010101010101001
010101010101001010101010101010101010010
01010101011010101010010101010101010100
010010101010101010101010101010101001

徐松林 王军 等编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>

PowerBuilder 8 Web 数据库 开发指南

徐松林 王军 等编著

JS888/01

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书由浅入深、循序渐进地介绍了最新版本的可视化客户端应用程序开发工具——PowerBuilder 8。它首先介绍了 PowerBuilder 8 的新特性、应用程序开发方法以及作为程序开发前提的数据库管理和脚本编写语法。然后分别介绍了 PowerBuilder 8 面向对象开发过程中涉及到的各种对象和组件,包括应用程序对象、窗口对象、窗口控件及应用、数据窗口对象及应用。接下来介绍了 PowerBuilder 应用程序开发过程的一些技巧和方法,如自定义用户对象和事件、数据管道、库文件管理等。最后介绍了应用程序的调试、创建与发布。本书内容翔实,逻辑清晰,讲解透彻,程序示例丰富,适合各类 PowerBuilder 用户阅读,也可供其他从事数据库应用程序开发和管理工作的人员参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: PowerBuilder 8 Web 数据库开发指南

作 者: 徐松林 王军 等编著

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 杨海儿

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 21.75 **字数:** 516 千字

版 次: 2001 年 7 月第 1 版 2001 年 7 月第 1 次印刷

书 号: ISBN 7-302-04544-5/TP·2689

印 数: 0001~6000

定 价: 34.00 元

前 言

PowerBuilder 是一种可视化客户/服务器数据库应用程序开发工具,具有功能强大、开发效率高、连接传输快、开放程度高等诸多优点,深受国内外广大数据库管理人员和数据库应用程序开发人员的喜爱。最新推出的 PowerBuilder 8,对开发方法和界面作了较为重大的改进,进一步增强了网络开发功能,新增了数据库接口并增强了原有接口,对 EAServer 集成等其他方面也进行了改进。

本书由浅入深、全面而系统地介绍 PowerBuilder 8 的应用程序开发方法。在结构编排上,按照 PowerBuilder 8 应用程序开发过程中所必需的知识构成来逐步讲述的。这样,既便于应用程序开发人员的对照参考,又充分照顾到国内广大初学者的学习习惯。首先让读者了解 PowerBuilder 应用程序开发的必备知识——脚本语法和数据库管理;再依次介绍窗口等各种基本对象的创建与使用方法,然后介绍开发过程中的一些技巧和方法。在内容编写上,既详细讲解了各种相关的属性、方法、事件、函数等,又注重通过典型的操作示例和可视化开发方法介绍具体的应用,便于读者迅速学会和掌握软件。

参加本书编写、制作的人员有徐松林、王军、李万红、麻瑞朝、薛焱、乔洪坤、王斌、郭卫泳、胡阅兵、陈卫东、王武东、胡腾、王琳、张哲峰、胡辰浩、李克坤、周志远、周曙光、马利涛、田辉等人。由于我们水平有限,加之创作时间有限,错误和疏漏之处在所难免,欢迎广大读者多多加以批评指正。

作 者

2001 年 5 月

目 录

第 1 章 PowerBuilder 简介	1
1.1 PowerBuilder 概述	1
1.1.1 跨平台开发	1
1.1.2 开放的数据库连接系统	2
1.1.3 面向对象编程	2
1.1.4 事件驱动的应用程序	3
1.1.5 功能强大的编程语言与函数	3
1.2 PowerBuilder 8 的新特性	4
1.2.1 新的开发方法和用户界面	4
1.2.2 网络功能的增强	6
1.2.3 数据库连接新特性	6
1.2.4 其他新特性	7
1.3 PowerBuilder 开发环境	8
1.3.1 菜单	8
1.3.2 工具栏	9
1.3.3 系统视图窗口和工作区	11
1.4 PowerBuilder 编程的对象特性	11
1.4.1 数据抽象性、继承性和多态性	12
1.4.2 属性、事件和函数	13
1.5 PowerBuilder 应用程序开发步骤	14
1.5.1 系统分析和系统设计	14
1.5.2 系统开发	15
1.5.3 调试	15
第 2 章 数据库连接	16
2.1 数据库连接	16
2.1.1 数据库连接方式	16
2.1.2 连接数据库的操作	17
2.2 通过 ODBC 连接数据库	18
2.2.1 ODBC 简介	18
2.2.2 ODBC 的硬件和软件需求	19
2.2.3 ODBC 的各层软件及作用	19
2.2.4 ODBC 的配置	20
2.2.5 连接数据库	22

2.2.6 数据库描述文件	24
2.3 通过专用接口连接数据库	30
2.3.1 专用接口简介	30
2.3.2 连接数据库	31
第3章 PowerBuilder 编程	32
3.1 PowerScript 基本概念	32
3.1.1 标识符	32
3.1.2 代词和保留字	33
3.1.3 保留字	35
3.1.4 标签和 ASCII 字符	35
3.1.5 书写代码注意事项	36
3.2 数据类型	38
3.3 声明	41
3.3.1 变量	41
3.3.2 常量	42
3.3.3 数组	42
3.3.4 外部函数	43
3.3.5 DBMS 存储过程	43
3.4 运算符和表达式	44
3.4.1 算术运算符	44
3.4.2 关系运算符	45
3.4.3 逻辑运算符	45
3.4.4 连接运算符	46
3.4.5 运算符的优先级	46
3.4.6 表达式	47
3.5 基本语句	48
3.5.1 赋值语句	48
3.5.2 条件控制语句	49
3.5.3 循环控制语句	51
3.5.4 其他语句	53
3.6 函数	55
3.6.1 常用标准函数	55
3.6.2 自定义函数	57
3.6.3 编译和保存函数	61
3.6.4 调用函数	63
3.7 结构	66
3.7.1 定义结构	66
3.7.2 使用结构	67

3.8 编写脚本	68
3.8.1 打开脚本视图区	68
3.8.2 粘贴脚本	69
3.8.3 编译脚本	71
第4章 嵌入式 SQL	72
4.1 嵌入式 SQL 语句简介	72
4.1.1 连接与断开数据库	73
4.1.2 提交与回滚事务	73
4.2 嵌入式 SQL 语句的使用	74
4.2.1 SELECT 语句	74
4.2.2 INSERT 语句	75
4.2.3 DELETE 语句	75
4.2.4 UPDATE 语句	76
4.2.5 SELECTBLOB 语句	76
4.2.6 UPDATEBLOB 语句	76
4.2.7 SQL 语句的执行情况	76
4.2.8 游标操作	77
4.3 动态 SQL 语句	79
4.4 粘贴 SQL 语句	83
4.4.1 粘贴 SELECT 语句	83
4.4.2 粘贴 INSERT 语句	86
4.5 BLOB 数据的应用	87
4.5.1 与 BLOB 相对应的数据类型	88
4.5.2 在数据窗口中使用 OLE 列	88
4.6 OLE 列的更新	89
4.6.1 使用 OLEActivate 函数	89
4.6.2 使用 OLE 方式对 BLOB 进行操作	90
4.6.3 BLOB 字段的函数	91
第5章 应用程序对象	93
5.1 应用程序简介	93
5.2 应用程序对象画板	94
5.2.1 画板的菜单和工具栏	94
5.2.2 应用程序对象工作区	97
5.3 创建应用程序对象	98
5.3.1 新建应用程序对象	98
5.3.2 创建模板应用程序	100
5.4 应用程序的属性、事件和函数	108
5.4.1 应用程序对象的属性	108

5.4.2	应用程序对象的事件	110
5.4.3	应用程序对象的函数	111
5.4.4	应用程序出错对象	112
5.5	应用程序配置参数	113
5.5.1	应用程序初始化文件	113
5.5.2	应用程序注册表	114
5.5.3	开发应用程序涉及到的主要注册表主键	115
5.5.4	PowerBuilder 的注册表函数	116
5.6	事务对象及其应用	118
5.6.1	事务对象的概念	118
5.6.2	创建自定义事务对象	119
5.6.3	事务对象的属性	120
第 6 章	管理数据库	121
6.1	创建和删除数据库	121
6.1.1	在 PowerBuilder 中创建和删除 ASA 数据库	121
6.1.2	使用 Sybase Central 管理数据库	123
6.1.3	数据库信息的存储位置	128
6.2	数据库画板	129
6.2.1	数据库画板的功能	129
6.2.2	数据库画板工具栏	131
6.2.3	数据库画板工作区	133
6.3	操作数据库表	134
6.3.1	表的创建、修改与删除	135
6.3.2	定义列扩展属性	135
6.3.3	创建索引和主键	137
6.3.4	删除表、主键、外部键与索引	138
6.3.5	定义表属性	138
6.4	表数据的浏览、修改与输出	139
6.4.1	结果视图区及基本操作	139
6.4.2	更新数据	140
6.4.3	创建视图	141
6.5	定义数据库扩展属性	143
6.5.1	显示格式	143
6.5.2	显示掩码	144
6.5.3	编辑风格	146
6.5.4	创建有效性规则	147
第 7 章	窗口与菜单设计	151
7.1	窗口画板	151

7.1.1	窗口对象简介	151
7.1.2	窗口画板的菜单	151
7.1.3	窗口画板的工具栏	153
7.1.4	窗口画板的视图区	153
7.2	窗口的属性	154
7.2.1	窗口类型	154
7.2.2	MDI 框架窗口	156
7.2.3	窗口属性	156
7.2.4	窗口的大小和位置	157
7.3	窗口事件	159
7.3.1	Open 事件	159
7.3.2	Active 事件	160
7.3.3	Clicked 事件	160
7.3.4	Close 事件	161
7.3.5	CloseQuery 事件	161
7.3.6	Key 事件	161
7.3.7	RButtonDown 事件	162
7.3.8	Timer 事件	162
7.3.9	Other 事件	164
7.4	窗口函数	164
7.4.1	窗口操作	165
7.4.2	MDI 窗口具有的操作	169
7.4.3	自动化窗口事件	172
7.5	窗口变量	173
7.6	创建菜单	175
7.6.1	菜单类型与特点	175
7.6.2	菜单画板	176
7.6.3	创建菜单	177
7.6.4	修改菜单	179
7.7	设置菜单属性	179
7.7.1	General 选项卡	180
7.7.2	ToolBar 选项卡	181
7.8	创建弹出式菜单	182
7.8.1	窗口弹出菜单	182
7.8.2	其他弹出菜单	183
7.9	菜单事件及其编程	183
7.9.1	菜单事件及其处理程序的编写	183
7.9.2	创建通用的弹出式菜单	184

第 8 章 窗口控件	187
8.1 添加控件	187
8.1.1 控件概述	187
8.1.2 窗口控件的选择与复制	189
8.1.3 窗口控件的布局	190
8.2 控件属性、事件和函数	192
8.2.1 控件属性	193
8.2.2 控件名称	195
8.2.3 控件事件及其脚本	197
8.2.4 控件函数	198
8.3 控件的应用	199
8.3.1 命令按钮	199
8.3.2 文本编辑及显示控件	201
8.3.3 单选按钮、复选框与单选组	204
8.3.4 修饰控件	206
第 9 章 高级控件	210
9.1 列表框控件	210
9.1.1 列表框与图片列表框	210
9.1.2 下拉列表框和下拉图片列表框	214
9.2 列表视图	215
9.2.1 列表视图控件风格	215
9.2.2 列表视图控件属性	216
9.2.3 列表项的添加与删除	218
9.2.4 列表视图控件的事件	221
9.3 树形视图	222
9.3.1 树形视图的创建与属性	223
9.3.2 添加树形视图项	224
9.3.3 删除树形视图项	225
9.3.4 树形视图控件的常用事件	226
9.4 选项卡控件	228
9.4.1 选项卡控件的组成	228
9.4.2 选项卡控件的创建与操作	228
9.4.3 选项卡控件的属性	230
9.4.4 选项卡控件的应用	232
9.4.5 选项卡控件的事件	234
9.5 OLE 控件	234
第 10 章 数据窗口	238
10.1 数据窗口概述	238

10.1.1 数据窗口对象	239
10.1.2 数据窗口控件	240
10.2 创建数据窗口	240
10.2.1 创建步骤	240
10.2.2 数据源与表现风格	240
10.2.3 创建一个新的数据窗口	242
10.3 数据窗口数据源	245
10.3.1 Quick Select 数据源	245
10.3.2 SQL Select 数据源	247
10.3.3 Query 数据源	250
10.3.4 External 数据源	250
10.3.5 Stored Procedure 数据源	251
10.4 数据窗口的表现风格	252
10.4.1 Freeform 风格	252
10.4.2 Tabular 风格	252
10.4.3 Grid 风格	253
10.4.4 Label 风格	254
10.4.5 N-UP 风格	254
10.4.6 Group 风格	255
10.4.7 Crosstab 风格	256
10.4.8 Graph 风格	256
10.4.9 Composite 风格	257
10.4.10 OLE 2.0 风格	257
10.4.11 RichText 风格	257
第 11 章 使用数据窗口	258
11.1 放置数据窗口对象	258
11.1.1 连接函数	258
11.1.2 添加数据窗口控件	259
11.1.3 编辑数据窗口控件	260
11.1.4 连接数据窗口控件与数据窗口对象	260
11.1.5 动态连接数据窗口对象	261
11.1.6 设置数据窗口属性	262
11.2 数据窗口控件与数据库	263
11.2.1 事务对象	264
11.2.2 为数据窗口控件设置事务对象	264
11.2.3 创建事务对象	265
11.2.4 检索和更新数据	267
11.3 访问数据窗口缓存区	268

11.3.1 通过数据窗口控件管理数据	269
11.3.2 处理数据窗口控件中的文本数据	270
11.4 数据窗口项的引用	271
11.4.1 使用表达式引用数据窗口项	271
11.4.2 使用函数访问数据窗口项	272
11.5 数据窗口事件	274
11.5.1 ButtonClicked 事件	274
11.5.2 Clicked 事件	274
11.6 数据存储	275
11.6.1 使用数据存储对象	275
11.6.2 数据存储对象的方法和属性	276
11.6.3 定制数据存储对象	277
第 12 章 用户对象与用户事件	278
12.1 用户对象的类型	278
12.1.1 可视用户对象	279
12.1.2 类用户对象	280
12.2 创建用户对象	281
12.2.1 创建标准可视用户对象	281
12.2.2 创建定制可视用户对象	283
12.2.3 创建外部可视用户对象	283
12.2.4 创建标准类用户对象	284
12.2.5 创建定制类用户对象	284
12.2.6 使用继承创建用户对象	285
12.3 使用用户对象	285
12.3.1 使用可视用户对象	285
12.3.2 使用类用户对象	286
12.3.3 窗口与用户对象间的通信	286
12.3.4 用户对象的命名规则	287
12.4 用户事件	287
12.4.1 用户事件的适用场合	287
12.4.2 定义用户事件	288
12.4.3 注意事项	290
第 13 章 数据管道	292
13.1 数据管道概述	292
13.2 创建数据管道对象	293
13.2.1 启动数据管道画板	293
13.2.2 源表	295
13.2.3 目的表	296

13.2.4 选择管道操作	296
13.2.5 灌入 Blob 类型的数据	297
13.2.6 改变源数据库和目的数据库	298
13.2.7 运行数据管道	299
13.2.8 数据管道出错处理	299
13.3 数据管道的应用	300
13.3.1 创建数据管道用户对象	300
13.3.2 数据管道用户对象的属性	301
13.3.3 数据管道的事件	302
13.3.4 数据管道的函数	302
第 14 章 库文件管理	306
14.1 库文件组织方式	306
14.2 库画板	307
14.2.1 库画板的工具	307
14.2.2 库画板工作区	308
14.3 维护库文件以及其中的对象	308
14.3.1 库操作	308
14.3.2 维护库中对象	310
14.3.3 导出与导入对象	313
14.3.4 重新生成对象	314
第 15 章 应用程序的调试、编译与发布	315
15.1 应用程序的调试	315
15.1.1 应用程序测试过程	315
15.1.2 调试器	317
15.1.3 断点设置	319
15.1.4 调试过程	322
15.2 其他调试手段	324
15.2.1 运用 PowerBuilderDEBUG 功能	324
15.2.2 追踪数据库访问	325
15.2.3 其他调试技术	327
15.3 应用程序的编译	328
15.3.1 创建可执行文件的一般步骤	328
15.3.2 创建可执行文件时要考虑的问题	331
15.4 应用程序的发布	333
15.4.1 PowerBuilder 运行时库及数据库接口	333
15.4.2 配置 ODBC 数据源	333

第 1 章 PowerBuilder 简介

科学技术的不断推陈出新,使得当今的商业市场正在以前所未有的速度变化着。随着 Internet 和浏览器技术的成熟与普及,形成了一种简便、低成本的信息传播方式,该方式正在改变着传统的产品销售方式和服务方式。

作为能够适应企业全部需求的信息管理系统解决方案,首先应该能够支持客户/服务器 (Client/Server) 结构,并且能够把它扩展成为包括 HTML 和 Java 的分布式体系结构。PowerBuilder 正是符合该趋势的一种分布式数据库前端开发工具,通过其图形界面可以快速简捷地开发出面向对象的数据库应用程序。

1.1 PowerBuilder 概述

使用 PowerBuilder 之前,首先应该了解 PowerBuilder 是什么,它又是如何工作的。概括而言,PowerBuilder 是一个功能超群、使用方便、易于开发复杂应用系统的前端数据库开发工具。利用 PowerBuilder 所提供的丰富而完善的开发工具,可以轻松地开发大型数据库应用系统。这些大型数据库应用系统的数据库和用户可以分布在范围极广的地理空间上,通过 PowerBuilder 对网络的巨大支持功能,可以将地理上分散的数据库及其用户组织到一起。由于 PowerBuilder 是具有图形界面的分布式数据库前端开发工具,它所采用的图形界面使得程序员能够快速方便地开发出相互独立的对象,而这些对象可供程序员共享或重复使用。

PowerBuilder 并不能被简单地认识为一种开发工具,它比其他开发工具功能多得多,也强得多,可以说是一个强有力的开发环境。开发人员不仅能利用它来开发各种具有友好界面的应用程序,还可以通过 PowerBuilder 修改数据库。同时,400 多个预定义的内置函数为开发适应各种需要的应用程序提供了极大的方便。

PowerBuilder 正在成为客户/服务器应用程序开发的标准。与其他客户/服务器开发环境相比,PowerBuilder 可以使开发人员的开发进程更快、成本更低、质量更高、功能更强。PowerBuilder 为应用开发提供了全面综合的支持,可以概括为如下几点:

- 跨平台开发。
- 开放的数据库连接系统。
- 面向对象的编程。
- 事件驱动的应用程序。
- 功能强大的编程语言与函数。

1.1.1 跨平台开发

PowerBuilder 是一个客户/服务器结构的可视化图形界面应用程序开发环境。利用 PowerBuilder 开发需要访问数据库服务器上数据库的 Windows 应用程序,不仅开发效率高,

而且开发完成的应用系统能够切实保证数据的安全可靠、正确有效,能够为最终用户提供一个界面友好、数据访问便捷高效、功能齐备的基于网络的分布式数据库应用系统。

PowerBuilder 应用系统可以在 Windows 95/98/NT、Macintosh 和 Sun Solaris 等多种平台上开发和运行。PowerBuilder 支持跨平台的开发和分布。例如,在 Windows 系统上利用 PowerBuilder 开发应用程序,可以将该应用程序分布到 UNIX 平台上运行,反之亦然。全部开发人员可以组成一个跨平台的开发小组,其中一些人员在 UNIX 平台开发,另一些则在 Windows 平台上开发。对于应用程序中的 PowerBuilder 对象,也可以在不同的计算平台之间,由全体开发人员自由地共享。

1.1.2 开放的数据库连接系统

PowerBuilder 是一个开放的应用程序开发环境,它可以访问很多常见的后台数据库系统,例如 Oracle、Sybase、DB2 及 Informix。

在开发应用程序时,数据库内部的细节对用户是不可见的。用户也无需关心每一个连接所使用数据库的特点和特性,这主要是因为 PowerBuilder 提供了一个通用的前端用户界面,使用户可以集中精力进行客户端的应用程序开发,而与后端数据库的连接工作则由 PowerBuilder 来完成。

PowerBuilder 可以支持不同的关系数据库管理系统,并能充分发挥每一个数据库的特长。它可以访问多个数据库信息,而后显示在一个窗口中。应用程序开发人员可以利用 PowerBuilder 提供的高性能数据库引擎 Adaptive Server Anywhere 生成独立的应用程序。PowerBuilder 连接数据库的方式有下面两种,如图 1-1 所示。

- 通过使用 ODBC(开放式数据库连接)。
- 通过使用 PowerBuilder 与数据库连接的专用接口。

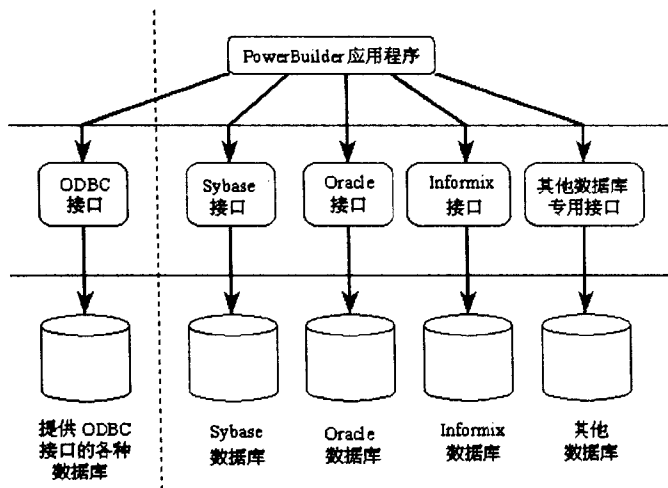


图 1-1 PowerBuilder 通过不同的接口与不同的数据库相连

1.1.3 面向对象编程

在 PowerBuilder 中创建的每一个菜单或窗口都是一个独立的模块,称为对象(Object)。

PowerBuilder 应用程序基本的组件就是用户所创建的对象,每个对象都具有一些特性和特定的行为(属性、事件和功能)。利用诸如封装性、继承性和多态性这样的面向对象的编程技术,开发人员可以有效地利用所创建的对象,使应用程序具有可重用性、可扩展性及其他强有力的功能。

PowerBuilder 应用程序由一系列对象组成,包括窗口、菜单、函数、结构和数据窗口、用户对象、用户事件等。对象中又包含若干控件,如命令按钮、单行编辑框、窗口等,这些对象和控件都可在许多应用中重复使用。因此,提高了程序员的效率和软件质量,又极大地减少了开发时间。

PowerBuilder 还有“画板”环境,可以生成用户定义的对象、事件和函数,其方便之处在于代码的可重复使用性。支持多级继承性可使对象类库(Object Class Library)的开发和维护更简便,用户对象画板(User Object Painter)完全支持第三方的控制对象,如 VBA 和 C++ 的访问。

因为 PowerBuilder 用户界面的元素都是对象,所以它们也都有属性、事件和方法。例如,若要改变窗口的颜色,可以通过修改窗口的 Color 属性实现。当用户按下按钮执行操作时,就可在按钮的 Clicked 事件中添加代码。要想在运行时移动控件的位置,可以执行相应的 Move 事件。

1.1.4 事件驱动的应用程序

PowerBuilder 的应用程序是由事件驱动的,这就是说所有运行于该环境下的程序并不是顺序地执行,即不是一条指令接着一条指令地执行,而是用户通过各种操作来控制应用程序的流程。例如,单击某个按钮,从菜单中选取某个命令,或在某个编辑框中输入数据时,都会触发相应的事件。用户可以通过编写脚本(Script)来说明当事件被触发时所要完成的处理过程,程序被写成许多独立的片段。某段程序只有当与之相关联的事件发生时才会被执行。例如,有一段代码与某个按钮的 Clicked 事件关联,只有当用户用鼠标单击该按钮时才会发生 Clicked 事件,否则代码不被执行。在事件驱动的环境中,是用户控制程序的运行,而不是程序的运行引导用户的操作。

用户通过 PowerBuilder 创建的窗口、控件和其他的应用程序对象都是一个预定的事件集合。例如,每个按钮都有一个 Clicked 事件,每个编辑框都有一个 Modified 事件。通常情况下,预定义的事件是用户在开发应用程序时经常用到的,而且用户还可以根据开发应用程序的需要定义自己的事件,这些自定义事件与标准事件一样容易使用。

PowerBuilder 为开发人员提供了一个集成环境。应用程序的各种成分,包括窗口、菜单、事件逻辑、数据库存取、数据库生成、图像、报表等全部都能在 PowerBuilder 内开发,而不必再到其他环境去做,这就大大加快了开发过程。

1.1.5 功能强大的编程语言与函数

PowerScript 是 PowerBuilder 内置的功能强大的编程语言。用 PowerScript 语言编写的程序通常称为脚本。脚本往往由 PowerScript 命令、函数和对应某一事件所要完成的处理过程的语句所组成。它能使开发人员很容易地将简单或复杂的事物对象与应用项目配合。它有几百个函数用于操作对象、数字、文本和日期等信息,并且全面支持 DDE 和全文本 I/O。

每个对象都有可以发生在它上面的特定事件。用户编写的脚本是按对象和对象上的事件来分段的。例如,用户可以为刚刚放在窗口中按钮的 Clicked 事件编写脚本,可能要进行的处理是从数据库中检索和显示数据,这样当用户单击该按钮时就会执行这段代码。为某个编辑框上的 Modified 事件所编写的脚本,可能要对修改后的数据进行某种处理。用户也可以为 DragDrop(拖放)事件编写脚本(当拾取另一个对象并将其放到该按钮上时发生该事件),还可以为同一窗口中列表框的 Clicked 事件编写完全不同的脚本。

事件本身也可以触发事件,这是通过在某一事件脚本中的特定函数调用来完成的。例如,在一个按钮的 Clicked 事件脚本中调用 Open 函数,该函数用于打开窗口,Clicked 事件的执行结果触发了目标窗口的 Open 事件。

PowerScript 还提供了丰富的内置函数,可以利用这些函数对应用程序的各种对象和控件进行处理,如打开窗口函数、关闭窗口函数、使按钮有效函数、检索数据函数、更新数据库函数,等等。

此外,在应用程序中用户还可以定义自己的函数来执行某种特定的处理过程。外部函数可以在 PowerBuilder 内声明和访问,就如同内置 PowerBuilder 函数一样,这使得 PowerBuilder 可以在客户或服务器端与外部过程进行通信。

1.2 PowerBuilder 8 的新特性

PowerBuilder 8 是由 Sybase 公司旗下 PowerSoft 子公司推出的 PowerBuilder 系列数据库开发工具的最新版本。除了在整体上进行增强以外,相对于以前版本,PowerBuilder 8 着重在用户界面、开发方法、网络功能、数据库连接和 EAServer 集成等方面进行了改进和扩充。

1.2.1 新的开发方法和用户界面

在 PowerBuilder 8 中,对用户的开发界面和方法作了较大改进,改进后的用户界面和方法可以使应用程序创建更容易、更有效率。具体表现在以下几个方面:

● 引入 workspace 和 target 对象

在 PowerBuilder 以前的版本中,开发人员仅能在一个应用程序对象中进行开发工作。在 PowerBuilder 8 中,引入了 workspace(工作空间)对象,开发人员可以在 workspace 对象中,对多个应用程序对象和 target(目标)对象同时进行开发。

要想使用 target 对象,开发人员必须打开并且仅能同时打开一个 workspace 对象。但是,开发人员可以在 workspace 对象中打开并对多个 target 对象中的对象进行编辑。同时,开发人员能够创建和配置多个 target 对象。

target 对象可以分为两种:脚本 targets 对象和网络 targets 对象。脚本 targets 对象可以生成各种类型的应用程序对象,例如,客户端执行程序或 EAServer 组件等。网络 targets 对象就是网络应用程序,包括建立一个网站所需的全部组件,如 HTML 文件、脚本、图像、下载组件等。

● 完全集成 PowerSite

在 PowerBuilder 8 中,将 PowerBuilder 以前版本开发包中的 PowerSite 网络开发工具完全