

PowerBuilder 6.0 入门

刘兴初 朱永光等 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

面向对象开发工具 PowerBuilder 是目前最具有代表性的数据库前端开发工具之一,它具有图形化界面的全部处理能力和完全的 OLE 特性。本书简要介绍了 PowerBuilder 6.0 的特点、安装和使用,并以较多的篇幅介绍了学习使用 PowerBuilder 6.0 所需要掌握的基础知识。读者可以通过本书的学习,迅速掌握和了解 PowerBuilder 的基础知识,并能熟练地使用该工具。

本书的内容安排从易到难,既适合刚开始学习的初学者,也适合中级 PowerBuilder 技术人员。

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。

图书在版编目(CIP)数据

PowerBuilder 6.0 入门/刘兴初,朱永光编著. —北京:清华大学出版社,1999

ISBN 7-302-03443-5

I . P… II . ①刘… ②朱… III . 数据库管理系统-软件工具, PowerBuilder 6.0

IV . TP311.56

中国版本图书馆 CIP 数据核字 (1999) 第 09203 号

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)

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

印刷者: 北京丰华印刷厂

发行者: 新华书店总店北京发行所

开 本: 787 × 1092 1/16 **印张:** 20.75 **字数:** 488 千字

版 次: 1999 年 5 月第 1 版 1999 年 5 月第 1 次印刷

书 号: ISBN 7-302-03443-5/TP·1871

印 数: 0001 ~ 8000

定 价: 27.50 元

前 言

面向对象开发工具 PowerBuilder 是目前最具有代表性的数据库前端开发工具之一。据 Meta Group 最新统计,PowerBuilder 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。它几乎已发展为客户/服务器前端开发工具的标准。相对于其他的客户/服务器开发环境,PowerBuilder 可以使得开发的时效更快、成本更低、质量更高、功能更强。

PowerBuilder 具有图形化界面的全部处理能力和完全的 OLE 特性。通过智能的数据窗口对象,PowerBuilder 开发人员无需手工编写 SQL 语句便可进行数据库操作。PowerBuilder 可以操纵众多大型数据库和桌面数据库,支持多种硬件平台;还可以跨平台开发和交付,并且开发的应用程序可以生成机器代码提交用户直接运行,而无需其他运行环境。最新推出的 PowerBuilder 6.0 在以前版本的基础上又增加了许多新的功能,使它一经推出就受到广大开发人员的热烈欢迎。与以往版本相比,PowerBuilder 6.0 具有如下一些新的特性:

- 结合了多线程的优势,在分布式应用程序中,PowerBuilder 提供了对共享对象的支持。
- 数据窗口的 HTML 发布现在不仅支持附加属性让你确定数据窗口在 Web 页中的格式,而且还支持最新的 Dynamic HTML。
- PowerBuilder 6.0 包括了组件生成器,通过它可以更方便地添加自己的组件。
- PowerBuilder 调试器支持新的断点、单步执行和多视图的界面。当发现系统出错或者运行出错时,可以随时打开 Debug 窗口进行调试。

本书简要介绍了 PowerBuilder 6.0 的特点、安装和使用,并以较多的篇幅详细介绍了学习使用 PowerBuilder 6.0 所需要掌握的基础知识。读者可以通过本书的学习,迅速掌握和了解 PowerBuilder 的基础知识,并能熟练地使用它。

本书的内容安排从易到难,既适合刚开始学习的初学者,也适合中级的 PowerBuilder 技术人员。本书是集体创作的结晶,主要由刘兴初、朱永光执笔,参加编写工作的还有刘斌、王浩军、杨光、范醒哲、许庆丰、田守瑞、刘洋、曹海郡等,本书在编写过程中得到了叶青、伍瑛、徐晓峰的大力支持和帮助,在此表示衷心的感谢!

编 者

1998 年 6 月

目 录

第一章 PowerBuilder 简介	1
1.1 什么是 PowerBuilder	1
1.1.1 客户/服务器的体系结构	1
1.1.2 PowerScript 语言与函数	3
1.1.3 面向对象的编程	6
1.1.4 跨平台开发	7
1.1.5 与数据库的连接性	8
1.2 PowerBuilder 6.0 的新特性	8
1.2.1 分布式计算	8
1.2.2 与 Web 集成	9
1.2.3 开放技术的增强	10
1.2.4 开发工具的增强	10
1.3 小结	11
第二章 PowerBuilder 的使用	12
2.1 与数据库管理系统的连接方式	12
2.1.1 什么是 ODBC	12
2.1.2 ODBC 的组成	12
2.1.3 通过 ODBC 连接到数据库	13
2.2 PowerBuilder 6.0 环境简介	15
2.2.1 启动 PowerBuilder 6.0	15
2.2.2 PowerBuilder 画板	16
2.2.3 PowerBuilder 工具栏	17
2.2.4 PowerBuilder 的窗口操作	20
2.2.5 PowerBuilder 的联机帮助	21
2.3 小结	23
第三章 第一个 PowerBuilder 应用程序	24
3.1 创建应用程序库	24
3.1.1 PowerBuilder 库简介	24
3.1.2 库画板	25
3.1.3 创建新库	25
3.2 创建应用程序	26
3.2.1 应用画板	27
3.2.2 创建应用	27

3.3	第一次运行	29
3.4	加入命令按钮	30
3.4.1	窗口画板简介	30
3.4.2	加入命令按钮	32
3.4.3	加入脚本	33
3.5	第二次运行	33
3.6	小结	34
第四章	创建数据库	35
4.1	Database 画板	35
4.1.1	打开 Database 画板	35
4.1.2	Database 画板的工具栏	36
4.2	创建本地数据库	37
4.2.1	数据库管理系统(DBMS)	37
4.2.2	创建本地数据库	37
4.2.3	删除本地数据库	38
4.3	创建表	38
4.3.1	创建表的 4 种方法	38
4.3.2	创建表列	38
4.4	定义列的扩展属性	40
4.4.1	说明(Comment)	40
4.4.2	题头(Header)和标签(Label)	40
4.4.3	显示方式(Display)	42
4.4.4	有效性验证规则(Validation Rule)	42
4.4.5	编辑风格(Edit Style)	44
4.5	定义关键字	46
4.5.1	创建主关键字	46
4.5.2	创建外键	47
4.5.3	删除关键字	49
4.6	定义索引	49
4.6.1	创建索引	49
4.6.2	浏览索引	50
4.6.3	删除索引	51
4.6.4	修改索引	51
4.7	定义视图	51
4.7.1	创建视图	51
4.7.2	视图画板	52
4.7.3	连接表	52
4.7.4	选择视图的列	53

4.7.5 定义各种标准	53
4.7.6 预览视图	54
4.7.7 删除视图	54
4.8 数据操纵	55
4.8.1 数据操纵画板简介	55
4.8.2 数据操纵	56
4.9 数据库管理器	58
4.9.1 数据库管理画板	59
4.9.2 输入 SQL 语句	59
4.9.3 解释 SQL 语句	61
4.9.4 执行 SQL 语句	62
4.10 小结	62
第五章 创建应用	63
5.1 应用程序对象	63
5.1.1 创建一个新的应用程序	63
5.1.2 打开应用对象	63
5.2 指定应用程序属性	64
5.2.1 为应用程序指定图标	65
5.2.2 为应用程序指定库的搜索路径	67
5.2.3 为应用程序指定字体	67
5.2.4 应用程序的默认全局对象	68
5.3 应用程序事件	69
5.3.1 打开(Open)事件	70
5.3.2 关闭(Close)事件	70
5.3.3 系统错误(SystemError)事件	71
5.3.4 空闲(Idle)事件	71
5.4 查看应用程序的结构	71
5.5 运行应用程序	73
5.6 小结	74
第六章 创建窗口对象	75
6.1 常用窗口对象	75
6.1.1 主窗口(Main)	75
6.1.2 弹出式窗口(Popup)	75
6.1.3 子窗口(Child)	76
6.1.4 应答窗口(Response)	76
6.1.5 多文档框架窗口(MDI Frame)	76
6.2 创建一个新的窗口对象	76
6.2.1 创建窗口对象	76

6.2.2	保存窗口对象	77
6.3	指定窗口属性	78
6.3.1	指定窗口类型	78
6.3.2	指定窗口关联菜单	79
6.3.3	指定窗口颜色	79
6.3.4	指定窗口位置与大小	79
6.3.5	指定窗口的滚动属性	79
6.3.6	指定窗口的鼠标形状	81
6.3.7	指定窗口的图标	81
6.3.8	指定窗口的工具栏	82
6.4	窗口控件	82
6.4.1	按钮类控件	84
6.4.2	静态文本控件	85
6.4.3	编辑控件	86
6.4.4	列表框控件	87
6.4.5	复选框与单选按钮控件	89
6.4.6	图片类控件	90
6.4.7	视图类控件	91
6.4.8	分组框控件	92
6.4.9	数据窗口控件	92
6.4.10	选项卡控件	93
6.4.11	用户对象控件	93
6.4.12	滚动条控件	94
6.4.13	OLE 2.0 控件	95
6.5	操作窗口控件	96
6.5.1	在窗口中增添控件	96
6.5.2	选择控件	96
6.5.3	定义控件属性	97
6.5.4	选择控件字体与颜色	97
6.5.5	对齐控件	97
6.5.6	指定控件的可及性	99
6.5.7	定义 Tab 次序	100
6.5.8	定义加速键	100
6.5.9	使用 3D 效果	101
6.6	为新建窗口增加控件	102
6.6.1	增加控件	102
6.6.2	对齐控件	102
6.6.3	指定控件属性	102

6.7 加入脚本	103
6.7.1 为 Open 事件加入脚本	103
6.7.2 为命令按钮加入脚本	103
6.8 运行窗口	104
6.8.1 预览窗口	104
6.8.2 运行窗口	105
6.9 窗口的继承	105
6.10 小结	107
第七章 创建数据窗口	108
7.1 创建数据窗口对象	108
7.2 选择数据源	110
7.2.1 QuickSelect 数据源	110
7.2.2 SQL SELECT 数据源	111
7.2.3 Query 数据源	114
7.2.4 External 数据源	114
7.3 数据窗口的显示风格	115
7.3.1 freeform 显示风格	115
7.3.2 Grid 显示风格	115
7.3.3 Crosstab 显示风格	116
7.3.4 Label 显示风格	116
7.3.5 N-UP 显示风格	116
7.3.6 Group 显示风格	118
7.3.7 Graph 显示风格	118
7.3.8 Composite 显示风格	119
7.3.9 RichText 显示风格	119
7.3.10 Ole 2.0 显示风格	119
7.3.11 Tabular 显示风格	120
7.4 设计 DataWindow	120
7.4.1 设计区及说明	120
7.4.2 DataWindow 工具栏	122
7.4.3 调整 DataWindow 中的数据列	124
7.4.4 改变 Tab 浏览顺序	125
7.5 对数据列的操作	125
7.5.1 检验规则	126
7.5.2 删除和增加数据列	127
7.5.3 计算字段	127
7.6 在窗口中加入一个数据窗口	128
7.6.1 为 Student 表创建一个数据窗口	128

7.6.2	加入数据窗口控件	130
7.6.3	编写脚本	130
7.6.4	运行应用程序	134
7.7	小结	135
第八章	美化数据窗口	136
8.1	数据窗口的属性	136
8.1.1	数据窗口的设置	136
8.1.2	数据列的属性	137
8.2	美化数据窗口	139
8.2.1	颜色和字段边框的设置	139
8.2.2	数据列的位置、大小和对齐方式	141
8.3	显示格式	142
8.4	编辑样式	144
8.4.1	Edit 编辑样式	144
8.4.2	编辑掩码样式	146
8.4.3	下拉式列表框	147
8.4.4	单选按钮和复选按钮	147
8.4.5	下拉数据窗口	148
8.5	在数据窗口中加入对象	149
8.6	数据窗口实例	150
8.6.1	创建数据窗口	150
8.6.2	保存数据窗口	152
8.6.3	创建一个组	152
8.6.4	在数据窗口内加入计算字段	154
8.6.5	定义检索参数	155
8.6.6	建立雇员详细情况的数据窗口	156
8.6.7	增强数据窗口的功能	158
8.7	小结	164
第九章	连接数据库和数据的显示	165
9.1	事务处理对象	165
9.1.1	事务处理对象及其属性	165
9.1.2	设置事务处理对象的属性	166
9.2	连接应用程序和数据库	166
9.3	选择和显示特定数据	167
9.3.1	仅取需要的数据	169
9.3.2	数据窗口缓冲区	170
9.3.3	插入、删除和保存数据	170
9.3.4	数据窗口控件的事件	172

9.3.5 CloseQuery 事件	174
9.4 SetActionCode 函数	174
9.5 数据排序	175
9.6 数据过滤	177
9.7 数据打印	178
9.8 小结	180
第十章 数据窗口的深入	181
10.1 图形数据窗口的基本知识	181
10.2 图形数据窗口简介	182
10.2.1 图形对象的属性	183
10.2.2 直方图	187
10.2.3 饼图和面积图	188
10.2.4 线图和散点图	190
10.2.5 堆积图	190
10.3 设置图形对象的属性	191
10.4 交叉表	192
10.4.1 交叉表的定义	193
10.4.2 交叉表的属性	195
10.4.3 改变交叉表的定义	196
10.5 动态数据窗口	198
10.5.1 创建动态数据窗口对象	198
10.5.2 提供查询功能	199
10.5.3 修改数据窗口的属性	199
10.6 组合报表	201
10.7 小结	204
第十一章 创建菜单	205
11.1 菜单与菜单项	205
11.2 Menu 画板	206
11.2.1 打开 Menu 画板	206
11.2.2 Menu 画板主窗口	206
11.3 编辑菜单	208
11.3.1 添加菜单	208
11.3.2 插入菜单项	210
11.3.3 移动菜单项	210
11.3.4 删除菜单项	211
11.4 菜单的属性页	211
11.4.1 General 选项卡	211
11.4.2 Style 选项卡	212

11.4.3	Shortcut 选项卡	215
11.4.4	ToolBar 选项卡和 Pictutes 选项卡	215
11.5	通过继承创建菜单	218
11.6	保存菜单	220
11.7	菜单的使用	220
11.8	小结	221
第十二章	用户对象	222
12.1	用户对象画板	222
12.2	创建用户对象	223
12.2.1	自定义用户对象	223
12.2.2	外部用户对象图形对象的属性	224
12.2.3	标准用户对象	225
12.2.4	标准类	227
12.2.5	自定义类	228
12.3	通过继承创建用户对象	229
12.4	保存用户对象	230
12.5	可视用户对象的使用	230
12.5.1	准备工作	230
12.5.2	开始运行	233
12.6	类用户对象的使用	233
12.7	窗口和用户对象间消息传递	234
12.7.1	使用函数	235
12.7.2	使用用户事件	236
12.8	小结	238
第十三章	PowerScript 语言	239
13.1	PowerScript 基础	239
13.1.1	大小写	239
13.1.2	语句的断行、续行、分隔	239
13.1.3	注释	240
13.1.4	空白符和 NULL	240
13.2	变量及作用域	240
13.2.1	基本数据类型	240
13.2.2	变量的声明	241
13.2.3	变量的引用	241
13.2.4	变量的作用域与私有性	241
13.3	操作符	241
13.3.1	算术操作符	242
13.3.2	关系操作符	242

13.3.3 逻辑操作符	242
13.3.4 操作符优先级	243
13.4 标识符	243
13.4.1 标识符的组成	243
13.4.2 匈牙利命名法	243
13.5 代词	245
13.5.1 This	245
13.5.2 Parent	247
13.5.3 ParentWindow	247
13.6 基本语句	246
13.6.1 条件控制语句	246
13.6.2 循环控制语句	247
13.6.3 其他语句	250
13.7 结构	251
13.7.1 定义结构	251
13.7.2 使用结构	254
13.8 用户自定义函数	255
13.8.1 定义函数	256
13.8.2 使用函数	260
13.9 常用标准函数	261
13.9.1 MessageBox	261
13.9.2 Is 族函数	262
13.9.3 类型转换函数	263
13.9.4 Open()和 Close()	263
13.9.5 Run()	264
13.10 小结	264
第十四章 编写脚本	265
14.1 打开 PowerScript 画板	265
14.2 使用 PowerScript 画板	266
14.2.1 PowerScript 画板的组成	266
14.2.2 PowerScript 画板的文本操作	267
14.2.3 定制 PowerScript 画板	268
14.2.4 粘贴操作	271
14.2.5 编译脚本	278
14.2.6 退出 PowerScript 画板	279
14.3 小结	280
第十五章 调试工具的使用	281
15.1 准备工作	281

15.2	脚本调试	282
15.2.1	打开调试器	282
15.2.2	设置断点	285
15.2.3	断点运行	287
15.2.4	显示变量	289
15.2.5	观察变量	290
15.2.6	其他操作	292
15.3	小结	294
第十六章	库和工程	295
16.1	库的管理	295
16.1.1	库的组织	295
16.1.2	使用注释	295
16.1.3	搜索库中的对象	296
16.1.4	复制、移动和删除库中的对象	297
16.1.5	对象的检入与检出	299
16.1.6	对象的导入与导出	300
16.1.7	库的优化	300
16.1.8	库的重新生成	301
16.2	工程画板	301
16.2.1	打开工程画板	301
16.2.2	可执行文件命名	302
16.2.3	选择外部资源文件	302
16.2.4	选择编译连接方式	304
16.2.5	进行编译连接	304
16.2.6	工程的保存	305
16.2.7	工程画板的 Design 操作	306
16.3	PowerBuilder 应用程序的发布形式	307
16.3.1	PDB 文件	307
16.3.2	使用资源	307
16.3.3	应用程序的发布形式	307
16.4	使用动态库	308
16.4.1	在 Project 画板中创建动态库	308
16.4.2	在 Library 画板中创建动态库	309
16.5	小结	309
附录	PowerBuilder 的安装与卸载	310

第一章 PowerBuilder 简介

1.1 什么是 PowerBuilder

PowerBuilder 是由美国著名的数据库开发应用厂商 PowerSoft 公司于 1991 年 6 月推出的基于客户/服务器应用的快速数据库开发工具。它是最具代表性的新一代数据库前端开发工具。PowerBuilder 以其独特的设计思想一经上市就赢得了广大用户的好评,并在客户/服务器前端开发市场获得了一席之地。到 1997 年,根据 Meta Group 公司的统计,它已占有了全球客户/服务器前端开发市场 40% 的份额,雄踞第一。

PowerBuilder 充满活力,它不仅在美国的多次评比中荣获第一,而且升级速度极快,从 1991 年 6 月推出 1.0 版到如今已发展到 6.0 版。在每次升级过程中,PowerBuilder 都不断融入新技术,使其在功能上不断完善。作为新一代的应用前端开发工具,PowerBuilder 6.0 具有以下特征:

- 独立于特定数据库管理系统的应用程序开发
- 透明地访问异种数据库
- 面向对象的程序设计
- 可视化图形界面设计
- 开放性
- 真正的基于客户/服务器的集成应用开发

1.1.1 客户/服务器的体系结构

随着近两年网络技术的发展,客户/服务器结构已在越来越多的地方被采用。客户指的是用来提供用户界面和前端处理的应用程序,服务器则提供由客户使用的资源和服务。客户一般要利用服务器上的某些资源,例如数据库资源、程序、打印机等,来完成特定任务。

在客户/服务器结构的数据库中,一般由客户负责应用处理,而由数据库服务器负责数据访问和事务管理。多数情况下,客户、服务器软件分别运行在网络的不同计算机上,客户软件一般运行在 PC 机上,服务器软件则运行在大型计算机上。

在这种体系结构中,应用的处理一般由客户软件完成,例如交互的界面及各种逻辑表示;而服务器端则处理数据,管理事务。一个典型的查询过程是这样的:客户软件提供界面,接受用户输入,响应用户事件;当用户提出查询请求时,客户软件将查询请求发送给服务器,服务器接到查询请求后,通过数据库完成查询任务,将结果返回给客户。这样的过程具有以下优点:

- 由于网络上仅仅传输查询的请求和结果,因此与传统的集中式系统相比,减少了网络的传输量。

- 由于客户软件提供了部分数据处理功能,所以使得服务器可以有更多的能力来完成数据访问和事务处理,从而可以支持更多的用户,提高系统的性能。

客户/服务器的体系结构如图 1-1 所示。

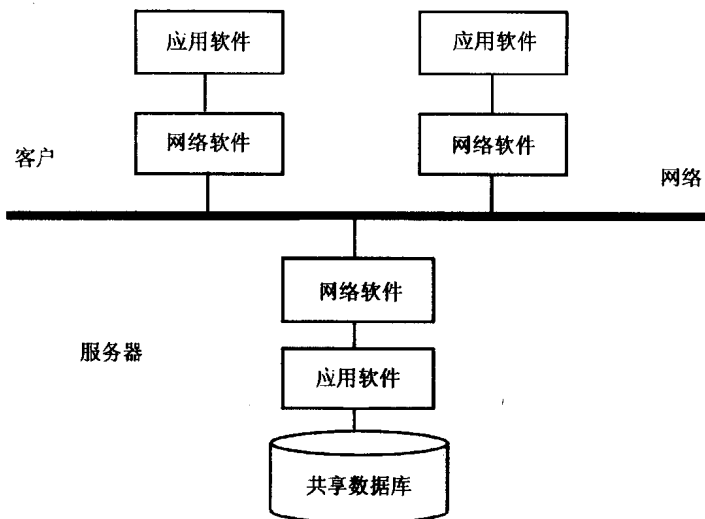


图 1-1 客户/服务器体系结构

PowerBuilder 即是基于上述客户/服务器模型的客户端应用开发工具。客户/服务器的这种结构使得 PowerBuilder 具有高效率、高性能、高可靠性、开放、易扩充等优越的性能,主要表现在以下几方面:

- 数据的无缝集成

PowerBuilder 不仅支持 Microsoft 的 ODBC,而且还自带一些专用数据库接口。通过它们,PowerBuilder 可以透明地访问多个数据库,使得在一个应用内访问多个数据库成为可能。

- 开放的系统结构

客户/服务器结构通过各种语言和连接接口的国际标准或工业标准,例如 ANSI/ISO SQL、开放式数据库连接标准(ODBC)、网络协议 TCP/IP 等,使得由 PowerBuilder 产生的应用可以方便地与异构网络的数据源连接,透明地访问多个异构数据源,并且可以自由地与别的厂家的服务器产品相连。这不仅可以方便用户,而且使得系统的扩充变得容易。

- 高效的生产

由于客户端接管了部分数据处理任务,所以客户端的开发工具就变成了重要的组成部分。像 PowerBuilder 这样优秀的开发工具,不仅提供了一组可视化的、高效的图形界面开发工具,而且提供了非过程化的开发语言,支持开发小组的开发方式。在这样的应用开发环境下,不仅应用效率可以大大提高,而且系统的质量也可以得到充分保证。

1.1.2 PowerScript 语言与函数

PowerScript 是 PowerBuilder 的编程语言,它不是一般意义上的 Script 语言,而是一种高级的、结构化的、面向对象的编程语言,它的特点介于 C/C++ 和 Basic 之间,有点像 Visual Basic。同许多高级语言一样,PowerScript 也是一种编译型的语言,即应用程序要经过编译和连接才能生成可执行文件。如图 1-2 所示,所有程序代码都在脚本画板内编写。

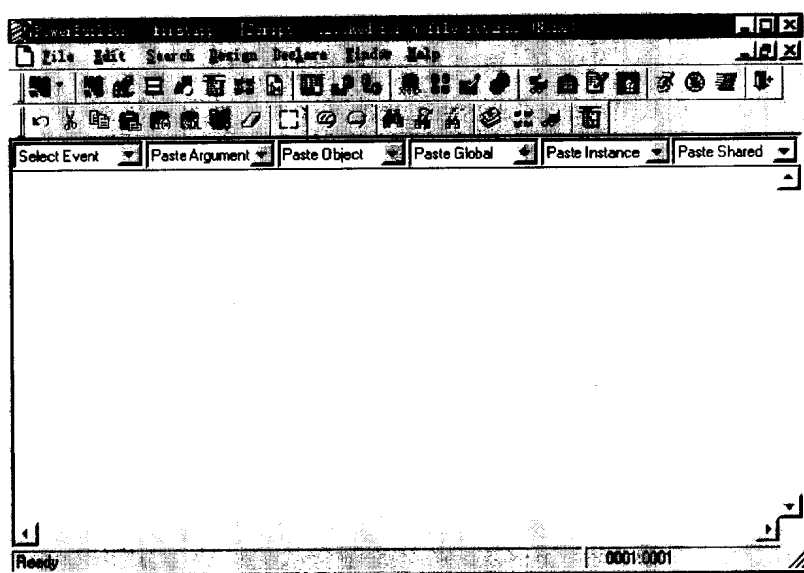


图 1-2 脚本画板

PowerScript 不仅提供了功能强大的内部函数,同时也允许用户使用 PowerBuilder 提供的函数画板来编写自己的函数,以完成特定的任务。

下面简要介绍 PowerScript 的一些知识。

1. 基本数据类型

- Blob 二进制型,用来存取图片和文本等大型数据文件,长度不限。
- Boolean 布尔型数据,只有真(TRUE)和假(FALSE)两种取值。
- Character 字符型数据,存取单个字符。如果用来操作字符串中的单个字符,可以用字符数。
- Date 日期型数据,包括年(1 000 ~ 3 000)、月(01 ~ 12)、日(01 ~ 31)。例如 1997-07-01。
- Datetime 日期时间型数据,只有对数据库进行操作时才能使用。
- Decimal 带符号数据,可有 18 位有效数字。
- Double 双精度浮点数据,绝对值从 $2.2E-308$ 到 $1.7E-308$ 。
- Integer 16 位带符号数据,从 $-32\,768$ 到 $+32\,767$ 。
- Long 32 位长整型数据,从 $-2\,147\,483\,648$ 到 $+2\,147\,483\,648$ 。
- Real 带符号浮点数据,绝对值从 $1.17E-38$ 到 $3.4E-38$ 。

- String 字符串型数据。
- Time 时间型数据,包括小时(00~23)、分(00~59)、秒(00~59)。例如 09-23-05。
- Unsigned integer 无符号整型数据,从 0 到 65 535。
- Unsigned long 无符号长整型数据,从 0 到 4 294 967 295。

2. 变量的作用域与私有性

PowerScript 中的变量根据作用域可分为四类:

- 全局变量(Global Variables): 可以在应用程序的任何地方访问;
- 实例变量(Instance Variables): 它与对象的某个特定实例相关联,只有在该对象的事件脚本或函数中才能使用,并且随着对象的打开或关闭而被创建或删除。
- 共享变量(Shared Variables): 它与对象的定义相关联,不管对象的状态如何,它总是存在的。
- 局部变量(Local Variables): 它只能在所属的程序或函数中声明和引用,其他任何地方都不能访问。

变量按照访问权限又可分为四类:

- 公有变量(Public): 应用于整个应用程序。
- 私有变量(Private): 用在变量定义所在对象的每一个事件,但在其派生对象的程序中不可用。
- 保护变量(Protected): 在该函数所在对象和其后代的脚本中使用。
- 共享变量(Shared): 可用在变量定义所在对象及其派生对象中。

3. 基本语句

PowerScript 常用的语句有条件控制语句、循环控制语句等,用户可以使用如图 1-3 所示的脚本编辑器的粘贴板来进行粘贴。其中,If...Then 和 Choose...Case 是条件控制语句,For...Next 和 Do...Loop 则属于循环控制语句。

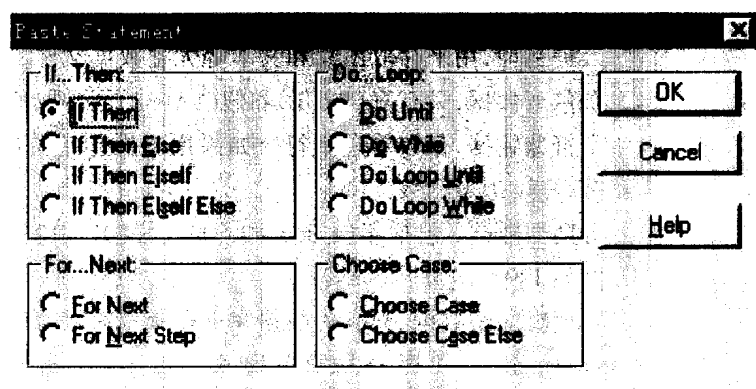


图 1-3 PowerScript 的基本语句

此外还有一些常用的语句,如 Continue、Exit、Return、Create 和 Destroy 等,在粘贴板里是没有的,用户必须自己输入。