

PowerBuilder 4.0
使用精解

徐东晖 桑大勇



PowerBuilder 4.0

使用精解

西安电子科技大学出版社

PowerBuilder 4.0 使用精解

徐东晖 桑大勇

西安电子科技大学出版社

1996

(陕)新登字 010 号

内 容 简 介

PowerBuilder 是一种客户/服务器系统的前端开发工具。本书首先以一个开发实例说明了 PowerBuilder 4.0 的基本使用方法和开发过程,然后阐述了窗口、控制部件、菜单及用户对象的设计和事件处理,介绍了 PowerBuilder 编程语言 PowerScript 的语法、库函数以及用户结构和用户函数,描绘了数据库的操作和连接,并对数据窗口对象作了详尽的论述。书中还介绍了 MDI 应用程序、库管理、程序调试、可执行程序的生成、OLE2、DDE 以及拖放控制等高级应用技术。

本书内容详实且重点突出,可作为初学者的培训教材,更可作为需要深入了解和掌握 PowerBuilder 4.0 开发技术的各类读者的参考资料。

PowerBuilder 4.0 使用精解

徐东晖 桑大勇

责任编辑 霍小齐

西安电子科技大学出版社出版发行

西安市长青印刷厂印刷

新华书店经销

开本 787×1092 1/16 印张 24 8/16 字数 581 千字

1996 年 11 月第 1 版 1996 年 11 月第 1 次印刷 印数 1—6 000

ISBN 7-5606-0497-8/TP·0233 定价: 32.00 元

前 言

随着客户机能力的不断增强以及服务器性能的相对下降,原来集中在服务器上的许多操作逐渐转移到客户机上来,由此促进了客户机/服务器系统的蓬勃发展。

开发传统的客户机/服务器应用需要繁琐的编程工作,尤其是进行面向数据库的操作或操作集时,用户面临的工作不但繁杂,而且缺少调试和即时观察结果的工具。这一切随着大量客户机/服务器系统前端开发工具的诞生和不断发展,有了显著的改变。

PowerBuilder 作为这些开发工具中极其优秀的一员,自进入中国市场以来,获得了广泛的应用。本书将由浅入深地介绍 PowerBuilder,引导读者了解和掌握这一优秀的开发工具。

全书共分十四章,继第一章概述之后,第二章以一个完整的例子介绍了 PowerBuilder 开发的全过程。第三章到第五章详细讲述了 PowerScript 语言以及窗口设计和消息处理这些 PowerBuilder 中程序设计的基础知识。第六章、第七章针对数据库应用,详细介绍了利用 PowerBuilder 进行数据库操作、管理以及和各种数据库的直接接口、ODBC 接口,另外还详细讨论了数据管道和嵌入式 SQL 语言。第八章、第九章着重阐述了数据窗口,这一 PowerBuilder 独有的强大的数据管理工具。第十章的用户对象,突出了 PowerBuilder 的另一个重要部分。第十一章帮助你生成可独立执行程序。十二章的库管理技术可使你更有条理,并有可能与同伴合作开发和管理大型的应用程序。第十三章进述的调试技术能帮助你尽快地发现和解决应用程序中存在的问题。最后一章讨论了部分高级应用技巧,主要侧重于 PowerBuilder 系统的支持技术。

本书既偏重于基础知识的讲述,对必要的操作步骤亦细心提供了大量的有参考价值的资料备查。

本书适合于下列几种用户的情况:

- 正在选择一个合适的数据库前端工具,而对于每种工具的特性不甚了解。你可以通过阅读这本书来对 PowerBuilder 有一个深入了解,看看它是否适合你的需要。
- 你的客户要求使用 PowerBuilder,而你作为公司的管理人员,认为掌握适当的背景知识也是必须的。
- 作为编程人员,你正在寻求一种快速、高效的方法完成一个大型客户机/服务器系统的开发,而你不想从每一行汇编语句写起。

本书由徐东晖主编,徐东晖、桑大勇合编,其中第一、二、六、七、十一、十三、十四章由徐东晖编写,第三、四、五、八、九、十章由桑大勇编写。

本书涉及的内容较新,这些内容也正处在不断的变化发展之中,作者本身也在不断学习之中,限于本身的能力和手头的资料,书中一定存在不少错误、疏漏之处,敬请各位读者不吝赐教。

作者于西电软件所

1996 年 10 月

目 录

第一章 概述	1	2.7.2 定义窗口的初始显示位置	34
1.1 客户机/服务器结构简介	1	2.8 创建一个数据窗口	35
1.2 利用 PowerBuilder 开发应用程序 的过程	2	2.8.1 创建新的数据窗口对象	35
1.3 PowerBuilder 的功能和作用	4	2.8.2 预演数据窗口对象	39
1.3.1 PowerBuilder 的功能	4	2.8.3 保存数据窗口对象	39
1.3.2 PowerBuilder 的作用	4	2.8.4 创建一个组(Group)	39
第二章 PowerBuilder 快速入门	6	2.9 加入一个数据窗口	43
2.1 安装和运行 PowerBuilder	6	2.9.1 在窗口中放置一个数据 窗口类型的控制	43
2.2 使用联机帮助	9	2.9.2 设置控制中的数据窗口对象	44
2.3 创建一个表	10	2.9.3 给窗口的 Open 事件增加 控制程序	45
2.3.1 打开数据库画板 (Database Painter)	11	2.9.4 给 SQLCA 赋值	48
2.3.2 创建一个新的数据库	11	2.9.5 运行	49
2.3.3 工具条特性的设置	11	2.10 创建第二个数据窗口对象	49
2.3.4 创建表	13	2.10.1 选择数据源和表现风格	49
2.3.5 定义主关键字	15	2.10.2 选择表及表中的列	50
2.3.6 请求创建表	15	2.10.3 定义一个检索变量	50
2.3.7 已创建的表	15	2.10.4 指定 WHERE 子句	51
2.4 为表设置扩展属性并添加数据	16	2.10.5 预演数据窗口对象	52
2.4.1 设置扩展属性	16	2.10.6 保存数据窗口对象	53
2.4.2 在表中增加数据	20	2.10.7 改进第二个数据窗口对象	53
2.4.3 测试这个表	22	2.11 增加第二个数据窗口对象	56
2.5 创建应用对象	24	2.11.1 增加新的数据窗口控制	56
2.5.1 打开应用程序画板 (Application Painter)	24	2.11.2 预演窗口	57
2.5.2 创建并保存应用对象	25	2.11.3 给数据窗口写控制程序	58
2.5.3 为应用对象设置图标	26	2.11.4 扩展窗口的 Open 事件的 处理程序	59
2.6 创建窗口对象	27	2.11.5 运行	59
2.6.1 创建一个新的空白窗口	27	2.12 增加数据库维护功能	60
2.6.2 增加一个命令按钮 (Command Button)	29	2.12.1 增加图形按钮	61
2.6.3 给按钮增加程序代码	30	2.12.2 使创建按钮可以工作	61
2.6.4 保存这个窗口	32	2.12.3 使删除按钮可以工作	62
2.6.5 给应用对象加上控制代码	32	2.12.4 使保存按钮可以工作	62
2.6.6 执行应用程序	32	2.12.5 调整按钮的定位(Alignment)、 大小(Size)和间距(Space)	63
2.7 改进我们的窗口	33	2.12.6 运行	63
2.7.1 定义窗口的风格	33	2.13 增加菜单	65

2.13.1 打开菜单画板	66	4.1.4 增加 MicroHelp 功能	112
2.13.2 加入菜单项	67	4.1.5 MDI 主窗口的函数	112
2.13.3 增加控制代码	67	4.1.6 MDI 用户区(MDI-1)的 属性和函数	114
2.13.4 增加更多的菜单	67	4.1.7 MDI 应用的键盘支持	115
2.13.5 保存菜单	68	4.2 自定义工具条的实现	115
2.13.6 把菜单挂接到窗口上	68	4.2.1 定义 w_mdi3 窗口	115
2.13.7 测试菜单的功能	69	4.2.2 定义 m_mdi3 菜单	116
2.14 创建单独的可执行的 EXE 文件	69	4.2.3 关联工具图标与菜单项	116
2.14.1 创建 EXE 文件	69	4.2.4 在窗口画板中连接菜单与窗口	118
2.14.2 测试 EXE 文件	69	4.2.5 运行时工具条显示的控制	118
第三章 窗口设计及消息处理	71	4.3 带工具条的用户定制 MDI 主窗口	123
3.1 窗口属性及消息处理	71	第五章 PowerScript 程序设计语言	126
3.1.1 窗口类型	71	5.1 语言基础	126
3.1.2 窗口属性	72	5.1.1 注释	126
3.1.3 窗口事件	74	5.1.2 标识符	126
3.1.4 窗口函数	75	5.1.3 字符串常量	128
3.2 控制部件特性及消息处理	78	5.1.4 空白符和 NULL 值	129
3.2.1 窗口中添加控制部件	78	5.1.5 语句的续行	129
3.2.2 各种控制部件及其用途	83	5.1.6 PowerBuilder 坐标单位(PBU)	130
3.2.3 控制部件的属性	86	5.2 数据类型	131
3.2.4 控制部件的事件	90	5.2.1 标准数据类型	131
3.2.5 控制部件的函数	91	5.2.2 系统对象数据类型	132
3.3 菜单设计及消息处理	96	5.2.3 枚举数据类型	133
3.3.1 菜单设计	97	5.3 变量的作用域及变量声明	134
3.3.2 菜单项的属性	98	5.3.1 不同作用域的变量类型	134
3.3.3 菜单项的消息处理	99	5.3.2 局部变量的声明及初始化	135
3.4 窗口与菜单的继承	101	5.3.3 数组的声明	136
3.4.1 通过继承创建窗口	101	5.4 基本语句	137
3.4.2 通过继承创建菜单	102	5.5 结构及其定义	141
3.4.3 类层次结构	102	5.5.1 结构的定义	141
3.4.4 代码的继承、重置与扩充	103	5.5.2 结构的修改	143
3.5 用户事件	104	5.5.3 结构的使用	144
3.5.1 用户事件的定义	105	5.6 库函数	145
3.5.2 用户事件的使用	105	5.6.1 blob 类型的库函数	145
3.6 Message 对象和事件触发	106	5.6.2 文件支持库函数	146
3.6.1 Message 对象	106	5.6.3 系统与环境库函数	148
3.6.2 事件的触发	107	5.6.4 其它杂类库函数	152
第四章 MDI 界面及自定义 工具条	108	5.7 用户函数及其定义	154
4.1 多文档界面设计	108	5.7.1 用户函数的定义	154
4.1.1 MDI 界面简介	108	5.7.2 用户函数的编译和保存	156
4.1.2 创建 MDI 主窗口	109	5.7.3 用户函数的修改	157
4.1.3 用户定制(Custom)MDI 主窗口	111	5.7.4 用户函数的调用	159

第六章 数据库操作	160	6.8 数据操纵画板	179
6.1 概览	160	6.8.1 打开数据操纵画板	179
6.1.1 可管理的数据库部件	160	6.8.2 检索数据	180
6.1.2 数据库管理的跨平台特性	160	6.8.3 修改数据	180
6.1.3 相关的画板	161	6.8.4 数据排序和过滤	181
6.2 数据库画板	161	6.8.5 查看行的信息	182
6.2.1 键盘操作	161	6.8.6 输入(Importing)数据	183
6.2.2 记录你的操作	162	6.8.7 打印数据	183
6.2.3 创建和删除数据库	163	6.8.8 保存数据	184
6.3 改变 DBMS 或数据库	163	6.8.9 返回数据库画板	184
6.3.1 通过提示改变连接参数	163	6.9 数据库管理画板	184
6.3.2 定义和使用数据库配置	164	6.9.1 打开数据库管理画板	184
6.3.3 共享数据库配置	164	6.9.2 控制数据库存取	185
6.4 表的操作	165	6.9.3 输入 SQL 语句	185
6.4.1 打开表	165	6.9.4 解释 SQL 语句	186
6.4.2 创建表	166	6.9.5 执行 SQL 语句	186
6.4.3 提交表	166	6.10 数据管道画板	186
6.4.4 指定字体	167	6.10.1 创建数据管道	187
6.4.5 指定扩展属性	168	6.10.2 数据管道设计画板	188
6.4.6 删除表	170	6.10.3 改变源数据库和目的数据库	189
6.4.7 输出表或视图的语法串	170	6.10.4 执行数据管道	190
6.5 索引	170	6.10.5 保存数据管道	190
6.5.1 创建索引	170	6.10.6 删除数据管道	190
6.5.2 浏览索引	171	6.10.7 多个表的数据管道	190
6.5.3 删除索引	172	6.10.8 检索变量	190
6.6 主键和外键	172	6.11 数据管道对象	191
6.6.1 查看关键字	172	6.11.1 数据管道对象属性	191
6.6.2 打开一个特定的外键 引用的表	173	6.11.2 数据管道对象函数	191
6.6.3 打开所有引用了特定 外键的表	173	6.11.3 数据管道对象事件	192
6.6.4 定义主键	174	6.11.4 对检索变量的支持	192
6.6.5 定义外键	174	6.11.5 使用数据管道对象	192
6.6.6 删除关键字	175	6.11.6 数据管道函数错误代码	193
6.7 视图	175	6.12 嵌入式 SQL 语言	193
6.7.1 打开视图	175	6.12.1 连接语句	194
6.7.2 创建视图	175	6.12.2 事务处理语句	194
6.7.3 显示一个视图的 SQL 语句	176	6.12.3 数据操作语句	195
6.7.4 从视图中选择列	177	6.12.4 游标语句	196
6.7.5 表的连接	178	6.12.5 存储过程语句	197
6.7.6 指定 WHERE, GROUP BY 和 HAVING 子句	179	6.12.6 BLOB 列操作语句	198
6.7.7 删除视图	179	6.12.7 SqlCode	198
		6.13 动态 SQL 语言	199
		6.13.1 动态 SQL 语句的第一种格式	199
		6.13.2 动态 SQL 语句的第二种格式	200

6.13.3 动态 SQL 语句的第三种格式	201	8.3 修改基本数据窗口对象	248
6.13.4 动态 SQL 语句的第四种格式	202	8.3.1 数据窗口画板工作区简介	248
第七章 连接数据库	207	8.3.2 改变数据窗口对象的 一般属性	249
7.1 管理数据源	207	8.3.3 添加输出对象	251
7.1.1 PowerBuilder 支持的 所有数据源	207	8.4 显示格式	254
7.1.2 使用 ODBC 数据源	209	8.4.1 为列数据指定显示格式	254
7.1.3 配置 Powersoft 数据库接口	210	8.4.2 显示格式的定义	257
7.2 使用 ODBC 数据源驱动程序	213	8.5 编辑风格	259
7.2.1 Microsoft Access	213	8.5.1 为数据列指定编辑风格	259
7.2.2 Microsoft Btrieve	214	8.5.2 编辑风格的定义	261
7.2.3 Q+E Btrieve	215	8.6 码表	264
7.2.4 Microsoft dBASE	216	8.7 输入数据的有效性检验	265
7.2.5 Q+E dBASE 和 Clipper	218	8.7.1 有效性检验规则	265
7.2.6 Microsoft Excel	219	8.7.2 在数据库画板中定义有效性 检验规则	265
7.2.7 Microsoft FoxPro	220	8.7.3 在数据窗口画板中定义有效 性检验规则	266
7.2.8 Q+E Netware SQL	221	8.8 滤波器、行排序和分组	267
7.2.9 Microsoft Paradox	222	8.8.1 滤波器	267
7.2.10 Q+E Paradox 5.X	223	8.8.2 行排序	268
7.2.11 Q+E Paradox	224	8.8.3 行分组	269
7.2.12 DEC Rdb	225	8.9 数据窗口对象的使用	270
7.2.13 Microsoft Text File	227	8.9.1 关联数据窗口对象和 数据窗口控制部件	270
7.2.14 WATCOM SQL	229	8.9.2 应用程序与数据库的通信	271
7.3 使用 Powersoft 数据库接口	231	第九章 数据窗口的高级特性	275
7.3.1 ALLBASE/SQL	231	9.1 图形对象	275
7.3.2 Database Manager 和 DB2/2	233	9.1.1 图形对象概述	275
7.3.3 INFORMIX	233	9.1.2 在窗口对象中使用图形对象	277
7.3.4 MicroDecisionware Database Gateway Interface for DB2	234	9.2 图形对象与数据窗口	282
7.3.5 ORACLE	235	9.2.1 在数据窗口中使用图形对象	282
7.3.6 SQL Server	236	9.2.2 数据窗口采用图形表现方式	284
7.3.7 SQLBase	237	9.3 数据窗口控制部件的 属性、事件和函数	285
7.3.8 Sybase Net—Gateway Interface for DB2	238	9.3.1 数据窗口控制部件的属性	285
7.3.9 XDB	239	9.3.2 数据窗口控制部件的事件	285
第八章 数据窗口	240	9.3.3 数据窗口控制部件的函数	287
8.1 选择数据源	240	9.4 动态数据窗口	305
8.1.1 Quick Select 数据源	240	9.4.1 创建一个动态数据窗口对象	306
8.1.2 SQL Select 数据源	241	9.4.2 动态修改数据窗口对象	307
8.1.3 Query 数据源	241	9.4.3 向用户提供查询能力 的数据窗口	308
8.1.4 External 数据源	242		
8.1.5 Stored Procedure 数据源	243		
8.2 选择表现方式	244		

9.4.4 数据窗口对象的重用	309	12.1 库概览	335
第十章 用户对象	311	12.1.1 使用多个库	336
10.1 用户对象及其建立	311	12.1.2 库的组织	336
10.1.1 用户对象的种类及其用途	311	12.2 使用库画板	337
10.1.2 用户对象的建立	312	12.2.1 树的操作	337
10.1.3 通过继承建立用户对象	317	12.2.2 使用弹出式菜单	338
10.2 用户对象的使用与编程	318	12.3 限制库中对象的显示	338
10.2.1 用户对象的使用	318	12.3.1 指定被显示的对象	338
10.2.2 向画板工具条中添加可视 用户对象图标	320	12.3.2 指定被显示对象的显示信息	338
10.2.3 用户对象的编程	321	12.3.3 保存设置	339
10.3 窗口中的用户对象和窗口 之间的通信	322	12.4 选择库中对象	339
10.3.1 使用用户函数实现通信	323	12.5 使用注释	339
10.3.2 使用用户事件实现通信	324	12.5.1 更新库的注释	339
第十一章 生成可执行程序	325	12.5.2 更新库中对象的注释	340
11.1 概览	325	12.6 创建和删除库	340
11.1.1 EXE 文件	325	12.6.1 库的创建	340
11.1.2 PBD 文件	325	12.6.2 库的删除	341
11.1.3 使用资源	325	12.7 复制、搬移和删除库中对象	341
11.1.4 四种基本的应用包装方式	326	12.7.1 向其它库复制库中对象	341
11.2 创建 EXE 文件	326	12.7.2 搬移库中对象到其它库	342
11.2.1 创建 EXE 文件的步骤	326	12.7.3 删除库中对象	342
11.2.2 PowerBuilder 拷贝对象 的原则	327	12.8 浏览库中对象	342
11.3 散发资源文件	328	12.9 转移到对象画板	343
11.3.1 单独散发资源文件	328	12.10 浏览类层次结构	343
11.3.2 使用 PowerBuilder 资源 描述文件	329	12.10.1 在库画板中浏览类 层次结构	344
11.3.3 运行时如何查找资源	329	12.10.2 浏览类层次结构的 另一种方法	344
11.4 使用动态库	329	12.11 使用检出/检入功能	344
11.4.1 创建动态库的源库	329	12.11.1 检出操作	345
11.4.2 创建一个动态库	330	12.11.2 查看被检出的对象	345
11.5 使用工程对象(Project)	330	12.11.3 检入对象	346
11.5.1 创建工程对象	330	12.11.4 清除对象的被检出状态	346
11.5.2 工程对象画板	331	12.12 库的优化	346
11.5.3 执行工程对象	332	12.13 库的重新生成	347
11.5.4 保存工程对象	333	12.13.1 重新生成库中对象	347
11.5.5 删除工程对象	333	12.13.2 重新生成某个对象的 所有后代	347
11.6 运行你的应用	333	12.14 输出和输入对象 (Exporting and Importing)	347
11.6.1 完整的应用	333	12.14.1 输出对象到文本文件	348
11.6.2 跟踪执行	333	12.14.2 将文本文件输入到库中	348
第十二章 库管理	335	12.15 创建动态库	349

12.16 创建库内容的报告	349	第十四章 其它高级应用技巧	360
12.16.1 创建库中对象报告	349	14.1 OLE 列	360
12.16.2 创建库目录报告	350	14.1.1 创建 OLE 列	360
第十三章 调试	351	14.1.2 预演 OLE 列	362
13.1 打开调试窗口	351	14.1.3 在应用中使用 OLE 列	363
13.2 加入断点	351	14.2 OLE 2.0 支持	363
13.3 编辑断点	353	14.2.1 OLE 2.0 容器(Container)	364
13.3.1 打开断点编辑窗口	353	14.2.2 OLE 2.0 控制的属性	364
13.3.2 断点编辑窗口中显示的内容	354	14.2.3 OLE 2.0 控制的函数	366
13.4 在调试方式下运行程序	354	14.2.4 OLE 2.0 控制的事件	368
13.4.1 单步或连续运行	354	14.2.5 OLE 2.0 自动作(Automation)	
13.4.2 停止运行时查看信息	354	函数	368
13.5 显示变量值	355	14.3 Drag and Drop(拖放技术)	371
13.5.1 打开变量显示窗口	355	14.3.1 拖放技术基础	371
13.5.2 显示各种变量	355	14.3.2 拖放属性	372
13.5.3 变量窗口中的操作	355	14.3.3 拖放事件	373
13.6 使用观察变量	355	14.3.4 拖放函数	373
13.6.1 创建观察变量表	355	14.4 DDE	374
13.6.2 删除观察变量表	357	14.4.1 DDE Server 端函数	374
13.7 改变变量的值	357	14.4.2 DDE Client 端函数	377
13.8 打印变量值	357	14.4.3 DDE Server 端事件	380
13.9 运行时错误处理	357	14.4.4 DDE Client 端事件	380
13.9.1 如果不处理 SystemError		14.5 HELP	380
事件	357	14.5.1 给其它 PowerBuilder 开发者	
13.9.2 如果包含 SystemError		提供在线帮助	380
事件处理程序	358	14.5.2 给最终用户提供在线帮助	381

第一章 概 述

我们处于一个不断变化发展的世界,计算机作为我们向新的文明进步的见证人和好伙伴,陪伴我们走过了一程又一程。在计算机的世界里,等待只是意味着放弃,一切都在不停地奔流,从平均 1.5 年一代的计算机硬件的更新换代,到 Internet 上每个月数百乃至上万个新用户的加入,每天数以亿计的新的信息的产生,世界真的要爆炸了。

我们需要不断注入新的血液,就像这本书希望带给你的那样。我们将详细地论述 PowerBuilder,引导你在快速入门之后,不断地加深理解,了解 PowerBuilder 的方方面面,帮助你使用 PowerBuilder 这个优秀的可视化客户机/服务器系统的前端开发工具开发你自己的应用,最后你会发现,客户机/服务器系统并不神秘,你也可以开发客户机/服务器系统。

1.1 客户机/服务器结构简介

数据处理是一个古老而广泛的话题,但是,从来没有一个体系结构像客户机/服务器(Client/Server,简称 C/S)结构那样能引起这样的关注。我们到处都可以看到有关客户机/服务器技术的文献,硬件和软件商也正在向我们鼓吹他们的产品是如何适合于客户/服务器环境。但是,什么是 C/S 结构呢?

本书不准备对这个问题作深入的研究,只是在这里作一个粗浅的介绍。

客户/服务器并不是一个物理概念。也就是说,客户机和服务器并非一定是两台机器。可能客户机和服务器均位于同一个机器上,也可能客户机和服务器不止两台机器,更有可能在不同的场合和条件下,客户机和服务器的地位可能互换。精确地说,C/S 技术是同时执行的软件过程之间相互作用的范例或者模型。

客户机进程发送请求给服务器进程,服务器进程对这些请求作出响应,并给出对应结果。一般而言,服务器进程应该能完成客户机进程所不能完成的特殊处理,而客户机进程由于免除了复杂的进行这些处理的开销,从而能进行其它有益的工作。客户机进程和服务器进程之间的相互作用是协作式的,事务处理是交互式的。其中,客户机为主动方,发出请求;服务器为被动方,接受请求进行服务。

客户机/服务器的典型应用还是在网络上。例如,在计算机管理信息系统(MIS)中,如果按照非客户/服务器结构的模式进行设计,那么每当前台工作站需要什么信息时,就会打开服务器上的数据库,通过网络将数据库传送到前台。在前台处理完后,再将数据库传回服务器。同样,当前台工作站需要更新数据时,也会将更新涉及到的数据库在网络上来回传递。用户的消极等待和网络大量无谓数据的传输就成了鲜明的对照。另外一种情况是,如果所有请求均需要发回服务器才能进行处理,则会形成一方面是服务器的极重负担,而另一方面是客户机无所事事,用户无谓等待。

对于 C/S 技术而言,这些问题太简单了。对于前者,客户机只要将请求发给服务器,而服务器在处理完客户机发来的请求之后,只将结果发送回客户机,而对于后者来说,将

一部分商业逻辑(business logic)分散到客户端,将大大提高人机交互的效率,只有在必要时才和服务器进行网络通讯。这样,网络的数据传输量大大减少,效率大为提高。在这期间,客户机负责同用户打交道,用丰富友善的 GUI 获取和表示数据,而服务器则专门负责数据库的操作、维护,各司其职,各专其能。同时,一个服务器也可以为多个客户机进程服务,而一个客户机也可以同时向多个服务器申请服务。

客户机/服务器结构与传统的数据处理体系结构相比,有以下几个优点:

① 服务器应用程序的规模明显缩减,某些和具体应用相关的部分被分散到客户机端,从而使服务器的相对性能有所提高,可以同时处理更多的请求。

② 客户机/服务器结构便于在不同应用程序之间进行通讯。使用不同通讯协议的两个客户机应用程序不能直接通讯,却可以通过同时支持两种协议的服务器进行通讯。

③ 客户机/服务器结构可以使应用程序的各个部分独立开发,每一部分可以修改或替换,却不影响其它部分。

④ 使有限的资源得到最大限度和合理的运用。客户机/服务器技术的迅猛发展和 PC 机性能的飞速增长是分不开的。这是由于客户机性能的不断提高和服务器相对性能的下降,使得 C/S 技术成为可能并迅速发展起来。相信随着新产品新技术的不断涌现, C/S 技术也会更上一层楼,发挥更大的作用。

1.2 利用 PowerBuilder 开发应用程序的过程

利用 PowerBuilder 开发应用程序,同样必须遵循软件工程的开发原则,同时针对它所面临的应用,又有其自身的特点,我们将之归纳成 PowerBuilder 开发十步曲:

1. 明确用户的需求

这一步是非常基础的一步,因而也十分重要。许多开发机构和开发人员往往在用户的需求尚未完全明确之时就仓促上马,结果自己费了不少力气,用户还不满意。

这样做对两方面都有好处:对用户而言,明确地提出需求有利于将来对产品的验收,对交付的产品中应该具备的特征可以理直气壮地进行检验;对开发者而言,不会因为无休止的需求变更,劳而无功,苦恼不堪,使整个开发周期无限期地延长。总之,初期对需求的明确也同时明确了双方的责任,并且在互相沟通中交流了思想,避免了将来由于理解偏差等原因造成的互相推诿,责任不明。

2. 数据库设计

由于客户机/服务器系统往往和数据库系统联系在一起,数据库设计的重要地位也就不言而喻了。这部分有专著专门讨论,它包括选取所需的数据库管理系统,数据库和表、视图、关键字等数据库要素及相互关系的划分、确定等,请读者自行参阅有关书籍。

3. 选择数据库管理系统及配套的数据库开发工具如 Sybase 的 Open Server, DB - Library 等,管理工具如 SQL DBA 等。

这部分内容并不随 PowerBuilder 软件包带上,而由各数据库厂商提供。PowerBuilder 能够支持的数据库管理系统很多,本身也附带了一个 WATCOM SQL 数据库管理系统。由于 PowerBuilder 良好的跨平台特性,用户可以选择先在本地的 WATCOM SQL 上进行开发、设计,待模型成熟后再移植到其它大型数据库管理系统上。

4. 选取所采用的网络通讯协议

可供选取的网络通讯协议很多,通常使用的有 TCP/IP, IPX/SPX, Named Pipe 等。具体选用何种协议需要考虑客户机和服务器两端所支持的协议,尤其是客户机端的数据库开发工具包和服务器的数据库管理系统所支持的协议以及系统需求,如安全性等而决定。

这部分软件同样不包括在 PowerBuilder 软件包内,一般随数据库管理系统提供。

5. 划分程序的功能模块,并进行各模块逻辑功能的设计

这一步是完成系统核心所必不可少的一步。模块划分正确,逻辑功能准确而且完备,就会给系统运行的可靠性提供最大的保障。除了各模块内部需要进行很好的设计外,模块之间的接口的定义也非常重要。

一个大型的客户机/服务器应用系统往往不是一个人、两个人就能够完成的,它需要很多人的协作努力。这样,接口定义得是否清晰明了,完全起到只需要了解接口,而尽可能不需要了解其它模块的内部实现就可以完成本模块的设计,或者反过来,不需要给模块外提供除了接口以外的信息就可以完成模块之间的通讯,就成了能否有效地协调多个开发者之间的工作的基础。

6. 设计每个模块所需的数据输入/输出表现方式

在各模块逻辑功能的设计完成后,紧接着需要进行的就 GUI 的设计。这一部分包括窗口、菜单、数据窗口、图标、背景图像等。GUI 的设计与逻辑功能的设计紧密相关。在应用系统中,不但要有丰富的内在功能,而且需要有美观的、恰如其分的输入/输出表现方式。

输入界面应该做到简洁,明快,操作简便易学,符合常规和一般的习惯,减少用户学习和操作的难度,而且要适当考虑美观和配色的协调,使用户长期工作不致于疲劳。好的输入界面设计应留有余地,允许用户按照自身的某些习惯进行配置,例如颜色、工具条的位置、形状等。

输出界面既要包含足够的信息,用图表、数据窗口等信息含量大的表现手段给予体现,又不能使用户淹没在漫无边际的数据中,无法找到所需的数据。这里有一个组织和层次划分的问题。

合理运用 PowerBuilder 及 Windows 系列平台给予的输入/输出、通讯和数据传递手段,有效地组织数据的收集、分析、归纳、重现等,需要用户不断地进行摸索。

7. 对 5、6 设计内容的分别实现与结合

在设计完成后,要着手进行实现。如果设计步骤完成得好,这一步的工作要容易得多。实现步骤要进行具体的编码、调试,包括单模块的编码、调试与多模块的联调。

在程序完成之前,要再一次检验用户的需求是否有变化,这种变化是否有可能体现在本次版本中,更要检查对已有的需求和设计是否达到要求。

8. 自身测试

这个自身测试是指开发部门内部的测试,但又有别于开发人员自己在开发过程中进行的测试。开发人员应会同组织人员制订测试文本,详细交待测试步骤、需测试的内容及反映的实质问题。具体测试应另外组织人员进行,测试过程中发生的问题应该有详细的记录,区分问题性质,拟定解决方案。

9. 用户试运行

自身测试完成后(指对应的问题也已解决后),还应该交付用户试运行一段时间。这样做有两个目的:使用户熟悉系统的运行、操作以及在本开发周期内最后一次听取用户的意见和反馈。对用户的培训等也可在此阶段进行。根据用户反馈的意见,退回到前面某一步或完成本步骤。

10. 最终交付用户

在全部系统完成后,整理必备的文档,制作用户说明书等相关资料,制作安装盘,交付用户使用。

在向用户移交相应的文档的同时,整理系统的开发文档也相当重要。一方面为系统维护打好基础,另一方面也作为一种积累,以免由于人员流动等各种原因,造成系统维护和再开发的困难。

以上简要介绍了用 PowerBuilder 开发客户机/服务器系统的全过程。应该着重指出的是,PowerBuilder 是客户机/服务器系统的前端开发工具,能够提供对客户端部分开发的完整支持,但是对服务器端的支持显然是不足的。但由于大多数客户机/服务器系统,尤其是 MIS 系统,需求往往集中在客户机端,因此 PowerBuilder 成为炙手可热的开发工具也就不足为奇了。

1.3 PowerBuilder 的功能和作用

讲完了客户机/服务器技术,咱们再来谈谈本书的主题:PowerBuilder。

PowerBuilder Enterprise 是美国著名的数据库应用开发工具厂商 Powersoft 公司最具代表性的产品,自 1991 年以来获得多次大奖,被评为 1991 年~1994 年世界风云产品,在工具市场占有 40% 的份额。而 Powersoft 公司本身,被美国《Fortune》杂志评为十大杰出高科技公司之一。它在 1993 年进入我国后,在金融、期货交易、环保、水利、商业、邮电、外贸、电力、石化等行业均有长足发展,1994 年 4 月还成立了 Powersoft 中国有限公司。

1.3.1 PowerBuilder 的功能

PowerBuilder Enterprise for Windows 是开发客户/服务器体系结构下的基于 Windows 3.1 或 Windows NT 上的应用程序的集成开发环境,它提供了可视化的开发工具以全面支持 Windows 3.1 或 Windows NT,提供了 PowerScript 语言以及对 OLE 2.0、DDE、VBX 等的支持。

此外,它特有的 DataWindow 对象是一种智能对象,它知道所要操纵的不同数据库的不同功能,可完成对数据库的直接管理。

PowerBuilder Enterprise 提供可直接与当前流行的大型数据库相连的本地驱动程序,如 ORACLE、SYBASE、Informix、DB2、Gupta 等。对于桌面级数据库,PowerBuilder 可通过 ODBC 与之相连,这方面的例子有 FoxPro、Microsoft Access、Excel 等。

PowerBuilder Enterprise 自带一个单用户的 32 位 WATCOM SQL 数据库,使开发者可以在拥有 PowerBuilder Enterprise 之后,立即着手工作。PowerBuilder 本身自带的例子和教学程序均使用其自带的 WATCOM 数据库,PowerBuilder 通过 ODBC 接口与这个数据库相连。用

单用户 WATCOM 数据库开发的应用程序可以很容易地扩展到多用户 WATCOM 数据库上。

PowerBuilder 还提供了完全可视化的查询生成器和多个表的快速选择器来建立查询对象,并把查询结果作为各种报表和应用的数据源。PowerBuilder 提供的报表生成器为你提供了丰富的报表格式,同时还有一个功能强大的商业图形绘制模块帮助你以丰富的表现形式,如二维、三维饼图、柱状图、线图 etc 来表现你的数据。

PowerBuilder 为应用程序提供了一个非常有效的管理方法。在应用程序设计环境中,整个应用程序的结构用图形方式分层地显示出来,使你对整个应用的结构一目了然,同时可以方便地进入相应对象的编辑环境进行编辑。

1.3.2 PowerBuilder 的作用

PowerBuilder 是一个客户机前端开发工具。所谓前端开发工具,就是说它能帮助你完成在客户机上面对最终用户所需要的功能,包括 GUI 设计和数据库连接,当然,还有各种可视化的操作手段。它是完全按照你的想法去做的。因此,不可能指望它会“自动”去做什么事,这里指的是除了 Windows 所提供的系统功能外。同样,它只是客户机上的开发工具,而不是服务器上的,因此服务器上的管理必须使用其它工具。另外,客户机和服务器是同时运行的两个进程,而且,客户机是主动方。现在,主动权在你手里,你可以命令它,让它帮助你完成什么,但是,并不是说,有了 PowerBuilder,一切就解决了,就像人人都有一双手,却不能表明人什么时候都很能干一样。

第二章 PowerBuilder 快速入门

2.1 安装和运行 PowerBuilder

安装 PowerBuilder 的基本环境要求如下：

- 386SX 以上(推荐为 486DX 以上)
- 内存 4MB 以上(推荐为 8MB 以上)
- PC-DOS 或 MS-DOS V3.30 以上
- Microsoft Windows 3.1 以上(中西文均可)
- 一个 3.5" 高密软驱
- 25MB 以上的硬盘空间(推荐为 40MB 以上)
- VGA 显示器
- 鼠标

PowerBuilder 的 Setup 程序位于第一张盘上，它将 PowerBuilder 安装到你指定的驱动器 and 路径中。如果不存在，PowerBuilder 将自动创建这个目录。

安装过程中会请你输入用户名和公司名。

安装 PowerBuilder 时可供选择的项目有 PowerBuilder Enterprise, Code Example, Demo Database, Building Applications Sample, WATCOM SQL 4.0, Open Database Connectivity Drivers, Native Database Drivers 以及 Source Control Interfaces 等 8 个部件，部件名前的 x 标记表示需要安装这个部件。如图 2-1 所示。

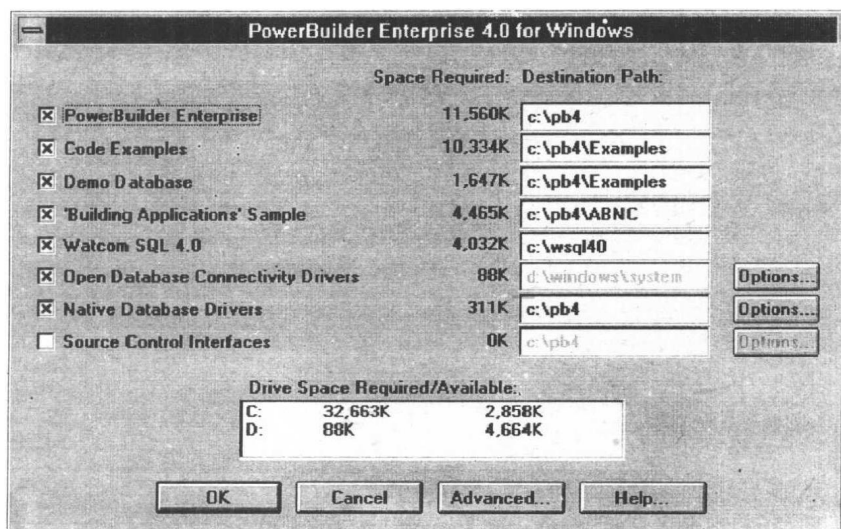


图 2-1 安装 PowerBuilder

各个部件后还有两项内容：所需磁盘空间和安装目录。所需磁盘空间由 PowerBuilder 安装程序计算后给出，安装目录可由用户指定。后三个部件的右端还带有一个 Options 按钮，分别用于安装不同的 ODBC 驱动程序、直接数据库接口和源代码控制接口。屏幕的下方显示了安装所用到的各个驱动器上所需空间和目前剩余空间的数量。如果你是重装 PowerBuilder，显示的剩余空间可能不够，你可以不予理睬。

可供选择的 ODBC 驱动程序有(见图 2-2)Btrieve (Microsoft Driver), Btrieve (Q+E Driver), dBase (Microsoft Driver), dBase (Q+E Driver), DEC RDB, Excel (Microsoft Driver), FoxPro (Microsoft Driver), Microsoft Access 1.0 (Microsoft Driver), Netware SQL (Q+E Driver), Paradox (Microsoft Driver), Paradox (Q+E Driver), Paradox 5 (Q+E Driver), Text (Microsoft Driver), WATCOM SQL 等 14 种。

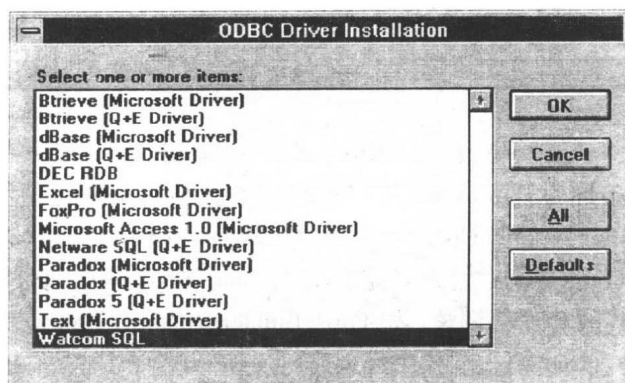


图 2-2 可供选择的 ODBC 驱动程序

可供选择的直接数据库接口(见图 2-3)有 Gupta SQL Base, HP Allbase/SQL, IBM DB Manager/DB2, Informix V4. x, Informix V5. x, MicroDecisionware Inc. DB2 Gateway, ODBC, ORACLE V6. X, ORACLE V7. X, SQL Server V4. x, Sybase Net Gateway, Sybase System 10, XDB 等 13 种。

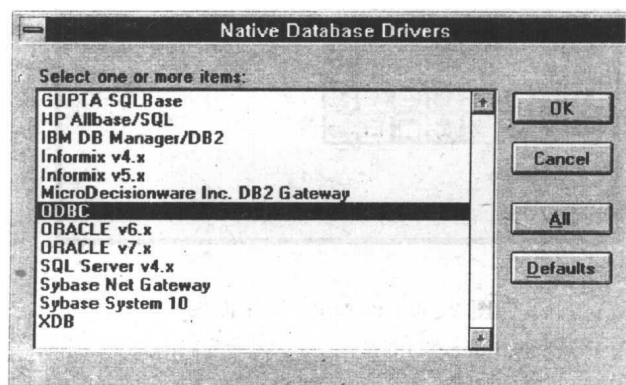


图 2-3 可供选择的直接数据库接口

PowerBuilder 中包括一个例子，演示了许多 PowerBuilder 的技术，推荐安装这个例子。