

PowerBuilder 6.0

基础教程



Official PowerBuilder® 6 Fundamentals



[美] Steve Erlank, Craig Levin 著
抖斗书屋 译



中国水利水电出版社
www.waterpub.com.cn

TP311.13
ELK/1

万水 PowerBuilder 从入门到精通系列

PowerBuilder 6.0 基础教程

[美] ^{S. Erlank C. Levin} Steve Erlank, Craig Levin 著

抖斗书屋 译

史惠康 审校



中国水利水电出版社

052940

内 容 提 要

PowerBuilder 是一种强大的数据库开发工具, 因其善于客户机/服务器的体系结构、面向对象的开发方法以及可视化的开发环境, 使它赢得了越来越多的用户。

本书从介绍 PowerBuilder 6.0 的基础出发, 详细描述了如何使用 PowerBuilder 6.0 做数据库方面的开发, 不仅讨论了简单的基本知识, 还花大量的篇幅讲述了如何充分利用 PowerBuilder 6.0 提供的功能进行复杂的, 功能强大的数据库设计。

"Original English Language edition published by The Coriolis Group, Inc., 14455 N. Hayden Drive, Suite 220, Scottsdale, Arizona 85260 USA, telephone (602)483-0192, fax(602)483-0193. Copyright ©1998 by The Coriolis Group. All rights reserved."

北京市版权局著作权登记号: 图字 01-98-2078 号

JS357/02

书 名	PowerBuilder 6.0 基础教程
著 者	[美] Steve Erlank, Craig Levin 著
译 者	抖斗书屋
审 校	史惠康
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: sale@waterpub.com.cn 电话: (010) 63202266 (总机) 68331835 (发行部)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂印刷
规 格	787×1092 毫米 16 开本 29.5 印张 671 千字
版 次	1999 年 7 月北京第一版 1999 年 7 月北京第一次印刷
印 数	0001—5000 册
定 价	60.00 元 (1CD, 含配套书)

引 言

计算机工业正处在蓬勃发展的状态当中。对于千万用户来讲,不论是在家庭还是办公室,个人计算机(PC)已经发展成为颇受欢迎的计算机环境。Microsoft Windows 是目前这千万台计算机可选择的操作系统。Microsoft Windows 目前面临着来自为数不多但甚为重要的其他操作系统的竞争。新的多媒体技术(诸如音频、CD-ROM、视频和图形图像技术)日渐成为标准而不再是昂贵的附加产品了。随着愈来愈多个人和企业加入到日益增长的国际用户群中,Internet 及其他全球性网络正以每月一百万用户的速度激增。

在工作场合,当 IS 经理们想要将遗留下的大系统和上亿字节的数据转化成新的有用的形式时,他们遇到了严峻的挑战。局域网(LAN)在商务领域的广泛应用以及开放系统的发展趋势使局域网能够与其他类型的硬件及软件相联接,联结范围上至金融业中使用的大型主机,下至工厂中应用的微型计算机。外部资源化,尺寸减小化,合同程序化,商业再造,2000 问题以及其他许多繁冗的问题都在施加压力迫使传统的 IS 部门改变其从事商务活动的方式。

(注:外部资源化是联系外部企业从事以前的内部组织功能的政策,比如人力资源管理,系统开发的财务管理等。尺寸减小化是使计算机应用从昂贵的大型主机环境转移到相对便宜、更灵活的中型服务器上。商业再造是评估商业如何运行,重新设计内部商业活动以提高效率的过程。2000 年问题是指过去的计算机或计算机应用程序不能正确处理 2000 年这个日期的问题。)

与新兴技术相伴而来的是彻底的新系统发展模式,这种模式得益于以更富创造性和更有用的方式给用户信息的新兴技术。面向对象技术和客户机/服务器模式是本书要更深一步讨论的两方面问题。

不过本书实质上是一本关于 PowerBuilder 的书,PowerBuilder 是一种已经占领 GUI (用户图形界面)编程领域的语言。PowerBuilder 的普遍使用是由于增强了几个关键方面。PowerBuilder 是用于开发客户机/服务器系统的第四代图形工具,它带有可在多个平台上运行的用户图形界面(GUI)。和其他类似的语言一样,它允许程序员快速而有效地开发一个应用程序运行在像 Microsoft Windows 这样的图形操作系统下。但 PowerBuilder 中数据窗口(直接联向后面的数据库表并允许用户浏览和控制使用表中窗口的信息)的实现是一个极有力的概念,在这一点上其他程序语言只到目前才开始追赶。对那些在处理 VB 源程序时受挫,或者花费了很多时间编写低水平的 VC++代码的程序员来说,使用 PowerBuilder 数据窗口工作无疑是一件乐事。

PowerBuilder 实现了客户机/服务器模式,支持多数据库模式,这意味着 PowerBuilder 适宜作大型主机数据库环境的开发工具。PowerBuilder 简单易用,尤其对于那些已经熟悉开发单一主机终端系统的程序员更是如此。

本书的写作目的

1994 年, Cape Town 大学明智迅速地决定教学生在客户机/服务器模式和 GUI 环境下开发系统。该实验在一个有 100 名学生的毕业班上进行, 这些学生已经有一年用一种叫作 Clipper 的面向文本的程序语言开发 PC 系统的经验。学生的学习过程虽然是很快的, 但也有很多困难。因此, 必然需要一本教材来帮助这些利用 PowerBuilder 编程的新手来加快他们的学习过程。本书围绕着在 PowerBuilder 教学中所碰到的问题而展开, 并且包含了我们在教学过程中所萌发的一些新想法以及一些逐步积累并完善起来的经验。

本书之所以成书实际上具有两方面的原因, 首先我们的学生需要这样一个课本来帮助他们更有效地学习如何在 PowerBuilder 下开发系统, 特别是他们要求我们将那些在实践中摸索出的技巧和技能编辑整理出来。

第二个原因是虽然已经有很多好的 PowerBuilder 书籍, 但发现它们都不能满足我们的具体需求: 尽可能快地使有文本编辑技巧和熟悉 Windows 的学生熟悉使用 PowerBuilder, 同时又不必丢掉以前那些好的技术和开发习惯。

许多书一步步介绍如何用 PowerBuilder 中各种各样的设计工具(称为描绘器), 但是大多数入门书除了介绍怎么使用按钮外, 很少再深一步介绍什么, 我们的学生需要的是一本能使他们理解那些好的设计的书, 也就是说他们不仅需要这本书帮助学习 PowerBuilder 的使用, 而且这本书应在如何开发具有复杂专业知识的 GUI 系统这一点上对学生有所帮助。

基于此, 本书除适用于学生外, 还可广泛应用于其他人员。大多数使用一门新语言的程序员愿意被告之那些编程的技巧、陷阱等而不愿再亲自实践去发现它们。我们的经验是: 初次接触 PowerBuilder 的学生主要在以下三个方面遇到困难:

- 对于事件驱动编程和客户机/服务器开发的潜在发展范式的理解。
- 对像“面向对象”、“继承性”、“消息”、“封装”这些概念的接受。
- 对于 PowerBuilder 提供的开发工具, 诸如描绘器、SQL 产生器、调试器、库管理器的学习。

本书将试着具体解决这些问题, 帮助编程新手理解 PowerBuilder 应用程序之中的概念和设计决策。已经有不少关于写 GUI 系统的好的书籍, 但他们都是针对 VB 或 VC++ 的。本书是专门帮助读者学习 PowerBuilder 编程的, 但本书还有一个更重要的目标, 那就是帮助读者从一开始就以正确的方法开发 GUI 系统。

学习写 PowerBuilder 程序并不难, 如果读者只有在基于文本环境下的编程经验, 会发现和其他环境一样, GUI 环境是很有乐趣的。但开发一个好的系统却是有难度的。PowerBuilder 开发者很有可能为用户提供一个真正精良而又便于使用的界面。但遗憾的是, 那么多编程者都被颜色和图表的选择搞得晕头转向, 以至于忘记首先考虑要选用它们的原因。不可避免地, 他们刚开始做的几个系统本来是为了使用户用起来简单点, 结果却像是未来派画廊中展示的愚蠢的艺术品, 而不是一个有用的专业系统。

在从基于文本编程向 GUI 环境、从面向功能的结构化的编程向事件驱动及面向对象的转变过程中，学生们在风格和方法上不得不进行概念性的跳跃。那些能从过去环境跃入新环境的学生处理得很好，学得很快，而那些不愿或不能摆脱过去编程习惯的学生则很难适应 PowerBuilder。我希望用户能自愿摆脱其他语言的束缚，带着 PowerBuilder 需要的开放式思维进入这个功能强大而又令人激动的语言中来。

本书的范围

PowerBuilder 6.0 适用于 Wintel 环境，诸如 Windows 3.1、Windows NT 和 Windows 95，在 Macintosh 环境下也可以使用，还可以运行在 Sun Solaris、IBM AIX 和 HP-UX 上的 UNIX 环境中。

虽然本书是在基于 Sybase SQL Anywhere 数据库（由 PowerBuilder 所载）Windows 95 系统下 PC 机上开发而成，尽管有些图表的外观会有些明显但不太重要的差异，但书中所涉及例子、法则对运行在其他系统下的 PowerBuilder 也是非常有效的。这些例子也同样适用 PowerBuilder 所普遍使用的数据库。

总的说来，图表与 PowerBuilder 6.0 的默认安装有关。如果用户已经安装了 PowerBuilder，并且已经定制过了，有可能会发现工具条的设计和屏幕的外观有轻微的不同。如果在网上使用 PowerBuilder，用户还会发现对话框默认的文件和目录不同。

在 Microsoft 的操作环境中，Windows NT Workstation 4.x、Windows 95 的用户会发现那些例子和环境匹配得非常好。对旧版本 Windows NT、Windows 3.x 的用户来说，各种图表的窗口和对话框的外观稍有差别。

Macintosh 环境下 PowerBuilder 用户会发现界面有些不同，GUI 标准这一章所涉及个别标准在该环境下不适合。

本书还提供了综合性 PowerBuilder 练习教程，包含 PowerBuilder 介绍教程中的大多数材料，此外还向读者介绍一些只有高级 PowerBuilder 教程才有的主题。

面向对象的设计包括书写面向对象的系统的方法，使用继承性和框架的方法。

- 如何使用更加复杂的数据窗口。
- GUI 设计规则。
- 数据库设计概念。
- 可以在网上使用的程序介绍。

本书最有用的一个方面是它提供了可用的范例即代码样本，并不是所有的教程都提供这些代码和范例的。

不应把正文和参考书当作练习教程的替代品，应该说它们是练习教程的产物。如果您觉得本书内容非常容易明白，那就没必要做 PowerBuilder 的练习教程了。因此，一旦您掌握书中的材料，就可以应付更高级的技术性的 PowerBuilder。

此外当用户已经学习完 PowerBuilder 的介绍教程后，本书还可以作为一本很好的参

考书，加深记忆，指导用户进行更好的编程实践。

为了真正精通这门有多种强大功能的语言，应该尽可能做练习，如果有时间参加各种 PowerBuilder 会议，或者上网的话，您会发现这是一条获得知识和经验的良好途径。

如何使用本书

本书不是一本综合性的手册，一本综合性的手册至少要有目前这本书的四倍那么厚，这对于作者和读者来说都是非常困难的。

本书设计供那些想学习用编程语言写系统的用户使用。如果您只是想大概了解 PowerBuilder，本书也是可用的。我们希望书中关于高级技术部分能给您带来额外收获，而这在用户手册中是很难找到的。

书中的介绍部分详细叙述了关于客户机/服务器和 GUI 编程环境的概念，以及在 PowerBuilder 如何实现它们的问题。如果您时间短促，可以直接跳到教程部分阅读，但是当您需要对所做的是有什么有一个深入了解时，请再回过头来阅读这部分。

教程部分将教您逐步完成第一个 PowerBuilder 应用程序。顺便我们还会给出一些技巧和难点，否则的话用户要花费很多时间才能发现这些技巧和难点。别忘了，除非有人先告诉编程新手 MDI 界面是什么，MicroHelp 是什么，否则他们肯定会去查 PowerBuilder 的用户手册。

本书的高级技术部分帮助读者为系统增加一些特性以使系统看起来更加专业化。另外还有数据库部分帮助读者更好地使用 SQL 和数据库设计。

本书关于方法论、风格和窗口显现标准这部分帮助读者沿着正确的方向开始思考。

阅读本书所需要的知识

我们估计大多数用户是第一次接触 GUI 编程环境，但我们假设读者从用户角度已经熟悉自己的 GUI 环境。本书是用 Windows 95 环境下的 PowerBuilder 6 编写的，涉及诸如注册、任务条、快捷方式之类的组件。

读者如果已经使用过一些 Windows Explorer (Windows 3.1 中的文件管理器) 这样的 Windows 程序，而且已经熟知打开窗口、改变窗口尺寸、或者在自己的桌面上定位窗口 (当然是用鼠标器而不是键盘命令)，那么您就具有足够的知识开始学习 PowerBuilder 了。如果有关于 Windows 应用程序的良好经验，那就更好了。

本书假定读者对编程环境有足够的了解，并且能够管理自己的文件和文件夹。

虽然非程序员也可以使用本书，但有一些编程经验对学习是有帮助的。如果用户已经用过像 Clipper 这样的 PC 语言，dBase 或者其他的 xBase 语言，本书对这类读者最有用。因为这样的读者更熟悉一些功能更强大的屏幕处理方法，这些方法在大型机环境中是找不到的。有诸如在 xBase PC 语言或 Oracle、Sybase、较大型计算机上的 DB2 之类的关系数据库经验也将是有用的。

我们还涉及了完整性和背景这些领域的一些概念，但如果您没有这些领域的经验，学习 PowerBuilder 之前有必要花些时间看看 Windows 的手册和帮助系统。

致谢

在本书的准备过程中，承蒙许多人帮助，感谢这些给予我们帮助的人。Mike van Dijken 帮助设计了正文原型的封面。特别感谢 Michael-John Turner, Elroy Heynes, Alan Abrahams 和 Dana Gordon-Davis，他们花费了大量时间校对和检查本书早期的几种草案。由于作者水平有限，书中定有不足之处。

感谢 Sandy Williamson 的支持与鼓励，她可口的食物缓解了作者由于睡眠不足而导致的困顿。

特别感谢 Sybase/Powersoft 公司的 Jose Cartergena、Julie Cooney-Doan 等许多朋友所给予的支持、建议，他们多次提供给作者所需要的培训教材及其文档。Jim deWolf，Theron Shreve 为本书的出版工作走上正轨做了大量的工作，Jo-Ann Campbell 在 mle 设计中做了大量的幕后工作。

除本书之外，许多朋友还支持帮助我们在大学进行教学。我们尤其感谢 Van Vuuren 和有关朋友，他们主动耗费了大量时间给予我们教学上的支持。Sybase 公司的 Rob Simpson 和 Bryn Davies 两位朋友，给我们的学生提供了教学素材和支持。没有所有这些朋友的支持，我们不可能如此成功地引入 PowerBuilder。

同时我们也感谢 Cape Town 大学的低年级学生们所做的不可估量的贡献，他们花费了许多时间在实验室学习写 PowerBuilder 的代码，他们是本书中的一些特殊方法的原动力和灵感所在。如果没有他们的热情参与，如果不是他们热切地想扩展知识范围，本书就不可能产生。

同时感谢本书旧版本的读者，他们通过电话、传真和电子邮件向我们提出了他们的批评、建议、要求和忠告。我们已经将读者的不少建议都写进了新书中。请读者继续保持这种优良的传统。

最后我们感谢那些不知名朋友，他们组成 PowerBuilder 讨论小组和服务小组，并把讨论的问题和解决问题的方案无偿提供给本书使用，给本书增色不少。

STEVE ERLANK
GRAIG LEVIN

目 录

引言

第一部分 概 念

第 1 章 时刻变化的系统特性	1
1.1 图形用户界面 (GUI)	3
1.2 Windows 应用程序	7
1.3 客户机/服务器体系结构的概念	9
1.4 面向对象编程	15
1.5 事件驱动编程	21
1.6 本章回顾	23
第 2 章 PowerBuilder 的概念	25
2.1 描绘器	25
2.2 应用程序、应用程序对象和库	27
2.3 窗口	29
2.4 控件	34
2.5 数据库和表单	35
2.6 数据窗口	37
2.7 脚本	38
2.8 PowerBuilder 帮助项	39
2.9 本章回顾	41

第二部分 PowerBuilder 入门教程

第 3 章 第一个 PowerBuilder 应用程序	43
3.1 PowerBuilder 起步	45
3.2 练习 1: 创建一个数据库	50
3.3 练习 2: 创建一个应用程序的对象	66
3.4 练习 3: 创建一个窗口	71
3.5 练习 4: 创建一个数据窗口	77
3.6 练习 5: 加入数据窗口	84
第 4 章 增强应用程序	93
4.1 练习 6: 创建另一个数据窗口	93
4.2 创建一个检索变元	95
4.3 练习 7: 在应用窗口里加入第二个数据窗口	100

4.4	练习 8: 加入或去掉数据窗口的更新能力	105
4.5	练习 9: 创建一个 MDI 框架	115
4.6	练习 10: 书写报表	124
第 5 章	增强 GUI 设计	140
5.1	练习 11: 应用良好的 GUI 设计原则	141
5.2	练习 12: 使应用程序更加健壮	159
5.3	练习 13: 用脚本来实现有效性验证	176
5.4	练习 14: 建立防错(BulletProof)系统	190
5.5	练习 15: 生成一个可执行的.EXE 文件	194
5.6	部分回顾	200

第三部分 PowerScript 编程

第 6 章	PowerScript 语言基础	202
6.1	PowerBuilder 中的命名规则	205
6.2	数据类型	208
6.3	数组	212
6.4	变量的作用域和私有性	213
6.5	PowerScript 语言组成部分	217
6.6	PowerBuilder 函数	222
6.7	用户自定义函数	225
6.8	本章回顾	229
第 7 章	应用 PowerScript 描绘器	230
7.1	PowerScript 描绘器中的下拉式列表框	232
7.2	配置 PowerScript 描绘器	233
7.3	打印脚本	234
7.4	使用语句的粘贴特性	234
7.5	应用 SQL 的粘贴特性	235
7.6	使用对象浏览器	236
7.7	本章回顾	238
第 8 章	使用调试程序	239
8.1	使用 Debugger 描绘器	240
8.2	在脚本中插入停止点	242
8.3	运行 Debugger	244
8.4	理解 Debugger 用户界面	244
8.5	在 Debugger 中检查变量和对象的属性	246
8.6	逐步测试脚本 (以步进制通过脚本)	246
8.7	系统运行时应用程序中的数据更新	248

8.8 即时调试	249
8.9 本章回顾	249

第四部分 图形用户界面(GUI)

第 9 章 GUI 界面的组件	250
9.1 窗口	251
9.2 菜单	267
9.3 本章回顾	271
第 10 章 GUI 的风格和标准	272
10.1 GUI 设计的原则	273
10.2 把优秀的原则应用于界面设计	279
10.3 图形界面和万维网 (WWW)	288
10.4 本章回顾	290
第 11 章 界面增强	291
11.1 创建 MicroHelp	291
11.2 使用颜色	293
11.3 创建弹出式菜单	295
11.4 改变指针	298
11.5 创建 PowerBuilder 类型的工具条	299
11.6 高亮显示数据窗口行	300
11.7 排序、过滤器与软查找	304
11.8 用表格构造列表框	309
11.9 拖放方式	314
11.10 创建 ToolTips	319
11.11 关于数据窗口的更多信息	322
11.12 本章回顾	334
第 12 章 为你的应用程序增加值	335
12.1 使用统计图	335
12.2 在会话中存储信息	340
12.3 在 PowerBuilder 中运行另一个应用程序	342
12.4 输出和输入数据	343
12.5 链接另一个应用程序	348
12.6 用剪贴板来剪切、拷贝和粘贴	355
12.7 建立 Internet 体系	357
12.8 本章回顾	366

第五部分 数 据 库

第 13 章 构造、维护数据库	367
13.1 连接数据库	369
13.2 定义一个新的数据源	376
13.3 使用 DBMS 的设计特性	376
13.4 在 PowerBuilder 中使用主键和外键	377
13.5 在 PowerBuilder 的表中使用索引	378
13.6 建立表后立即进行修改	379
13.7 使用 SQL 进行高级数据库管理	384
13.8 本章回顾	386
第 14 章 数据库编程	387
14.1 PowerBuilder 处理数据库方式	388
14.2 本章回顾	392
第 15 章 结构化查询语言	393
15.1 结构化查询语言	394
15.2 数据库管理员命令	398
15.3 本章回顾	400

第六部分 PowerBuilder 开发策略

第 16 章 PowerBuilder 开发周期	401
16.1 系统开发生命周期	402
16.2 本章回顾	407
第 17 章 库的管理	408
17.1 库描绘器	409
17.2 多用户开发环境	411
17.3 本章回顾	414
第 18 章 分发应用程序	415
18.1 PowerBuilder 动态库 (PBD)	416
18.2 PowerBuilder 资源文件	417
18.3 创建可执行文件	419
18.4 分发应用程序	421
18.5 本章回顾	422
第 19 章 继承	423
19.1 使用继承的原因	423
19.2 继承的内容	424
19.3 建立自己的窗口类继承	425

19.4	继承菜单	428
19.5	处理继承的脚本.....	429
19.6	继承的限制	431
19.7	本章回顾	431
第 20 章	PowerBuilder 高级开发技巧	432
20.1	使用应用程序框架.....	432
20.2	PowerBuilder 基础类库	433
20.3	PowerBuilder 分布应用程序的执行	434
20.4	PowerBuilder 提供的其他工具	435
附录 A		437
	设计数据库	437
A.1	实体关系图解.....	439
A.2	数据建模.....	448
A.3	数据通用化.....	452
A.4	附录回顾.....	457

第一部分 概念

- ◆ 第 1 章，时刻变化的系统特性。讲述了诸如个人计算机和图形用户界面等新技术的产生，这些技术正在改变组织内系统的外观和功能。
- ◆ 第 2 章，PowerBuilder 的概念。描述了 PowerBuilder 环境中诸如窗口和控件等基本组成部分，以及用来创建它们的描绘器。

第 1 章 时刻变化的系统特性

程序设计已经经历了许多的变革。一些变革是由于新类型硬件的开发所引发的。另外一些变革是出于更好地编写与维护程序的需要。所有这些革命在应用程序和系统结构以及在程序员编写和测试程序的方法上都有深远的意义。

早期的程序设计

在早期的程序设计中，代码的编写（按照今天的标准看）或多或少有点随意性。所投入的大量工作只是为了使指令的数目减少到最小。为确保程序尽可能高效率地利用资源，而忽略了风格、结构和可读性。

被称为千年虫的 2000 年问题（Y2K）就是这种编程风格所导致的沉重代价之一。实际上，这根本不是一条“虫”，而是一个潜在的系统中的设计缺陷，是那些在数据区内只保存两位构成世纪信息的编程人和设计者造成的。用 10/09/65 保存一个日期而不是用 10/09/1965 来保存，他们可以编写更高效率的系统。显然，当日期从 12/31/99 变成 01/01/00（或者其他一些任意嵌入在微芯片的值），每一个用简缩日期进行的计算都将是错误的。

这个问题给所有编程人员以有益的教训：那就是先仔细考虑和规划一个实际问题，即当系统、人员和组织改变或超过规定的时间时该怎么办。

早期的程序主要是批处理程序，对于联机终端根本不会出现这类问题。大多数程序是用机器语言或用汇编语言编制的，这些程序除非是受过训练的编程人员能够看懂，其他人是不容易理解的。虽然那时的计算机物理形式看起来是很大的，但是它们的功能根本无法与现在的计算机进行比较。甚至个人电脑的功能比那些早期的巨大电子怪物要强好几倍。

结构化设计和框架

计算机的功能逐渐变得越来越强大了，一些类似 ALGOL、COBOL 和 FORTRAN 新的

程序设计语言产生了。这些语言在利用计算机资源方面比汇编语言低效。它们被称为“高级语言”。这种语言可以使用一些类似 IF、PERFORM 和 COMPUTER 这些讲英语的人可以看懂的词组和单词来编写。它们需要编译，也就是将这种语言转换成计算机可以识别的代码。

用这种语言编写一些很大的程序已经变成一种习惯的方式。这种语言经历了好几代编程人员的发展，直到今天，许多系统还是用 COBOL 语言编写的。一种被称为结构化编程的技术逐渐发展起来，它可使程序员在一个有上千行代码的大型程序中保持某种规则和可读性。结构化编程最基本的概念就是将一个大程序分成具有各自功能的部分或模块。每个程序的结构是层次化的，上一层模块可以调用下一层模块去执行一特定的任务。典型的任务可以是执行某种计算、打印报表表头或更新文件中的一个记录。

PC 机的出现

虽然 PC 机很多年前就已出现，但 80 年代早期推出 IBM PC 机因为满足了商业用途的需要而得到普及。

PC 机永远地改变了计算的面貌，它们的特性对于编写应用软件带来新的挑战。PC 机应用软件的编写主要是针对单用户环境，而不受那些同时为多个用户共享的大型机软件的限制，诱人的彩色用户界面产生了许多编制系统的新途径。PC 机所拥有的独立处理功能的特性意味着更加强大的前端数据访问和组织工具可以被开发出来，而这些特点在那种可以为许多联机用户共享中央机的大型机环境下是不可想象的。PC 机很快在最终用户办公自动化领域获得主导地位，尤其当联接到局域网时更是如此。这给传统的系统施加了一种压力，用户开始要求大型机上的数据应该与 PC 机上的数据一样容易访问。

新的命令

传统开发程序的方法是使用结构化程序设计技术和库函数，这种方法不能满足 PC 机用户所要求的新的和灵活的接口。在这种要求下，出现了面向对象的设计方法，它能很理想地符合事件驱动的程序设计要求，程序员不用预测用户活动的可能顺序，依旧能够给用户提供任何适合他们活动顺序的执行任务的能力。

同时，客户机/服务器体系结构的概念，即应用程序和数据可以存储在完全独立的计算机平台上，越来越受到业界人士的喜欢。在客户机/服务器体系结构下，主机可以做一些它所擅长的工作，也就是用受约束的方式管理大量数据，而不用涉及在用户端屏幕怎样显示。换句话说，它可以作为数据库服务器，在任何客户机要求下发送数据包。客户端通常是一台 PC 机，它可以接受自己申请的数据，也可以处理数据并以一种用户所喜欢的风格显示。

客户/服务器体系结构也非常适用多媒体等显现技术。影像和声音文件占用了大量的空间。它们可以很容易地存储在主机中的数据库中，当一台能够显示图形和播放声音文件的客户机有要求时，就下载这些数据，而这些在主机环境中是不可能实现的。所以，这些新类型的数据可以存储在主机中，但是在 PC 机中使用。

正是 PowerBuilder 这种编程语言适应了在这些既新又复杂的环境下开发系统的挑战。

INTERNET 的影响

至今在有组织的计算机领域,最令人激动和最富有挑战性的技术也许是互联网络。在过去的几年中,很多组织已经意识到这个全球网络在商务,与顾客、商业伙伴及雇员的交流,发行公司信息等多方面的巨大潜能。当今互联网络浏览程序与真实的全球性范围的国际网络结合在一起,它们提供的全球性界面的魅力是难以抗拒的。

随之而来的是各个公司对互联网络的产品和信息市场的争夺(有时是不体面的)。比较典型的是各公司对互联网络所做的第一阶段工作,包括公司信息的发行和产品信息的下载。但是不久大多数的组织希望超越早期阶段,它们希望组织内不同管理者能够通过互联网络到达一些内部系统并获得这个组织的数据(至少是其中一部分)。用这种方法,顾客能够定货并能够跟踪定单。雇员可以从他们的公文包中获得邮件和任务书。供给者能够直接和快速地获得供给指令。

在快速变化的技术革新和竞争的压力下,许多组织不再大量投资去重新开发原有的系统。取而代之的是用一些创新的方法去修改旧系统、数据库和网络。PowerBuilder 6.0 系统以其特有的性能运行在复杂的硬件和软件平台,既能和互联网络天衣无缝地集成在一起,又能被配置在多级客户机/服务器方式中,它提供的这样一种“粘和剂”让所有那些隔离的原有系统互相紧密地结合在一起。

以下几节将更详细地描述这些概念和术语。

1.1 图形用户界面(GUI)

在很早的计算时代,计算机图形就已经存在了,但是主要用于一些专门的应用程序,而且一般需要昂贵的专用设备。直到 PC 机出现后,在一个用户界面上显示颜色和图形才渐渐变得可行。

和图形用户界面相对的是一个文本和按钮驱动接口。MS-DOS 操作系统的早期版本是命令驱动,以后的一些版本,虽然有色彩及可选择的菜单,但是仍旧是面向文本的。许多早期的应用程序例如电子制表和文字处理软件,它们的运行都是通过在操作系统的提示下键入一个命令而实现的。许多用户都习惯了在 MS-DOS 下的 C:\>命令提示。一旦执行应用程序,就会显示一个某种类型的图形界面,但是往往是显示用户从基于文本的菜单上选择的一些功能。

现在的大多数用户已经熟悉了图形用户界面(GUI)。第一个图形用户界面是由 Xerox Palo Alto 研究中心开发的。苹果公司的 Macintosh 计算机很早以前就已经拥有图形用户界面了,但是直到 Microsoft 公司的 Windows 操作系统的发布之后,用户图形界面才渐渐成为一种和符合 IBM PC 机的商业标准。Microsoft 的 Windows 并不是完美无缺的操作系统,只是它的许多局限性因素被很好地归档处理了。尽管如此,它可能是世界上第一个实实在在的标准操作系统,不仅仅是因为有很多人在使用它,而且因为很多专业的应用程序工作在它的环境下。Windows 95 对广告的巨大投入在简短的计算机历史中当然是件空前的事件。

从没有一种操作系统的发布，对上千百万使用不同硬件的计算机用户有这种影响力。

图形用户界面用一种图形的方式来模仿一个用户“桌面”。图形用户界面有时用作一种 WIMP 界面，即标准的窗口、图标、菜单和指点设备。最普通的指点设备是鼠标。图标是一些图片，用来表示最常用的功能和应用程序。菜单的设计也是出于同样的目的，只不过是基于文本的。因为菜单占用的空间比图标少，所以它们通常包括了所有可能的功能，而图标仅仅用于常用的功能。

在图形用户界面中使用窗口技术，可以同时桌面上运行多个被激活的程序，每个程序都运行在独立的窗口中。用户可以在一个窗口中运行文字处理程序，同时在另一个窗口运行电子数据表格软件。用户可以在屏幕中扩展或缩小窗口，并排显示窗口，或者将一个窗口叠放在另一个窗口上面，就好像用户可以在一个真正的桌面上处理纸质文档一样。所有这些 GUI 的概念都在 Microsoft 的 Windows 中得到体现。

1.1.1 使用图形用户界面

尽管大多数用户熟悉 GUI 的基础用途，但很少有用户能够真正知道在用户界面中显示的所有特性。用户喜欢去学习 GUI 或者应用程序中那些让他们感兴趣的方面，或者是能够提高生产效率的那些方面，却忽视了当时并不需要的其他特性。这显然是一种个人选择，而不是一个问题。

但是应用程序的开发者需要彻底了解 GUI 的所有特性，因为开发者需要提供有许多不同用户的应用程序，而每个用户都有自己喜欢的工作方式。

GUI 主要是点操作，通常是用鼠标实现的，但也用到轨迹球键盘和其他一些各异的灵敏的指示设备。键盘是最重要的输入设备，因此图形用户界面也允许用户用键盘访问应用程序的一些特性。这自然就意味着在界面中有冗余重复的地方，但是因为这样做可以提供很灵活地界面，所以这样做是必要的。

所有的 GUI 用户都已经熟悉基本的鼠标操作，比如指示屏幕上的一个对象，单击鼠标的左键进行选择和执行操作。但是还有更强大的功能活动，例如拖动和下拉，鼠标的右击和双击一个对象。

1.1.1.1 典型的鼠标操作

指点意味着移动鼠标让屏幕的光标指向感兴趣的对象。当光标移至某个对象时，高级的程序可以显示一个文本提示，简要地说明这个对象是什么或者能干什么。这个文本提示被称为 PowerTip 或者 ToolTip。单击鼠标左键来选择一个在屏幕中被指点的对象。有时，单击还可以激发系统的某个功能。

经常被用户忽略的右按钮单击用于显示被选择对象的附加信息。在 Windows 95 环境下（在 PowerBuilder 6.0 的广大领域里），鼠标的右按钮用来拉下“属性工作表”，它用来更具体地描述被选择的对象，还可以执行某些操作来改变所选择的对象。比如，在 Windows 95 桌面下，单击鼠标右键来更换墙纸或者布局（换句话说，桌面的属性）。

在界面中引入拖动和下拉功能也是一种发展的趋势。用户可以通过单击来选择一个对象，然后按住鼠标的左键将这个对象拖到桌面的另一个对象上。当按钮被释放时，这个被