

PowerBuilder

6.0

入门与提高

李薇 董恒 等编
西南交通大学出版社

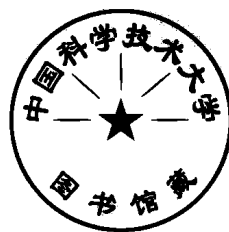
入门与提高

系列丛书

PowerBuilder 6.0 入门与提高

明星谷电脑工作室 策划

李 薇 董 恒 等编



西南交通大学出版社

内 容 提 要

PowerBuilder 6.0 是 Sybase PowerSoft 系列产品家族的一个重要组成成员, 致力于为新兴计算领域的企业级商业提供基于组件的应用开发模式, 包括: 客户机/服务器环境、分布式环境与 Internet 环境。PowerBuilder 满足了那些希望在 4GL RAD 环境中创建完整应用和应用组件的开发人员的需要。

全书共分十六章, 从实用角度出发, 循序渐进地向读者介绍如何使用 PowerBuilder 6.0, 并以大量的实例贯穿其中, 内容深入浅出, 图文并茂, 使读者能在较短时间内迅速掌握 PowerBuilder 6.0 这一强有力的开发工具。

本书适用于准备或已经使用 PowerBuilder 作为开发工具的程序设计人员和数据库管理人员。

本书无反盗版标识不得销
售。违者必究, 举报有奖。举
报电话: (028) 7600560 7600564

JK346/03

PowerBuilder 6.0 入门与提高

——入门与提高系列丛书

李 薇 董 恒 等编

责任编辑 成 鹏

*

西南交通大学出版社出版发行

(成都二环路北一段 610031)

郫县印刷厂印刷

*

开本: 787×1092 1/16 印张: 23.5

字数: 574 千字 印数: 1—5000 册

1998 年 8 月第 1 版 1998 年 8 月第 1 次印刷

ISBN 7-81057-181-8/T·271

定价: 38.00 元

目 录

第一章 客户机/服务器计算模式简介	(1)
§ 1.1 史前计算机	(1)
§ 1.2 大型主机	(1)
§ 1.3 PC 机	(2)
§ 1.4 客户机/服务器集成	(3)
§ 1.5 趋 势	(4)
§ 1.6 PowerBuilder 简介	(6)
§ 1.7 PowerBuilder 6.0 的新特性	(7)
第二章 PowerBuilder 快速入门	(12)
§ 2.1 Painter (描绘器)	(12)
§ 2.2 准备工作	(12)
§ 2.3 运行 PowerBuilder	(12)
§ 2.4 创建一个新库	(14)
§ 2.5 创建新的应用对象	(16)
§ 2.6 创建数据库与数据表	(18)
§ 2.7 创建数据窗口	(21)
§ 2.8 创建应用程序窗口	(25)
§ 2.9 第一次运行	(28)
§ 2.10 第二次运行	(31)
§ 2.11 第三次运行	(33)
§ 2.12 美化外观	(33)
§ 2.13 生成可执行程序	(34)
第三章 PowerBuilder 开发方式综述	(36)
§ 3.1 与传统开发环境的区别	(36)
§ 3.2 理解 PowerBuilder 的对象	(37)
§ 3.3 描绘器	(38)
§ 3.4 应用程序的开发过程	(43)
§ 3.5 工具条的使用	(44)
§ 3.6 联机帮助	(45)
§ 3.7 快捷键	(46)

第四章 创建应用程序	(47)
§ 4.1 Application 描绘器	(47)
§ 4.2 Application 描绘器的主窗口	(48)
§ 4.3 改变当前应用程序	(48)
§ 4.4 浏览对象	(49)
§ 4.5 创建新的应用程序	(50)
§ 4.6 指定应用程序的属性	(51)
§ 4.7 应用程序的事件	(55)
§ 4.8 保存和退出	(57)
第五章 创建窗口	(58)
§ 5.1 启动 Window 描绘器	(58)
§ 5.2 Select Window 对话框	(58)
§ 5.3 Window 描绘器的主窗口	(59)
§ 5.4 窗口的属性表	(60)
§ 5.5 窗口的弹出菜单	(66)
§ 5.6 窗口的脚本程序	(66)
§ 5.7 保存窗口	(67)
§ 5.8 打印窗口	(67)
§ 5.9 预览和运行	(67)
§ 5.10 增加控件	(68)
第六章 菜单制作	(90)
§ 6.1 启动 Menu 描绘器	(90)
§ 6.2 Menu 描绘器的主窗口	(91)
§ 6.3 菜单制作	(92)
§ 6.4 预 览	(93)
§ 6.5 创建级联菜单	(93)
§ 6.6 菜单项属性表	(94)
§ 6.7 事件与脚本	(98)
§ 6.8 保存菜单	(99)
§ 6.9 与窗口相联	(99)
§ 6.10 创建弹出式菜单	(100)
第七章 PowerScript 编程	(101)
§ 7.1 访问 PowerScript 描绘器	(101)
§ 7.2 编辑文字	(103)
§ 7.3 粘贴操作	(103)

§ 7.4	文本的导入和导出	(107)
§ 7.5	语言帮助	(108)
§ 7.6	脚本的编译	(108)
§ 7.7	建立自己的函数	(109)
§ 7.8	结 构	(113)
§ 7.9	PowerScript 基础知识	(116)
§ 7.10	变量及其作用域	(117)
§ 7.11	常 量	(122)
§ 7.12	常用函数和方法	(122)
§ 7.13	选择结构	(125)
§ 7.14	循环结构	(126)
§ 7.15	嵌入 SQL	(128)
第八章	数据库定义	(129)
§ 8.1	启动 Database 描绘器	(129)
§ 8.2	Database 描绘器主窗口	(130)
§ 8.3	数据库管理系统 (DBMS)	(131)
§ 8.4	数据库	(132)
§ 8.5	表	(133)
§ 8.6	视 图	(137)
§ 8.7	Data Manipulation 描绘器	(138)
§ 8.8	Database Administration 描绘器	(143)
§ 8.9	显示格式	(146)
§ 8.10	编辑样式	(147)
§ 8.11	创建和编辑有效性规则	(147)
§ 8.12	Data Pipeline 描绘器	(150)
第九章	SQL 简介	(154)
§ 9.1	SQL 历史	(154)
§ 9.2	SQL 现状	(154)
§ 9.3	SQL 基本结构	(155)
§ 9.4	简单查询	(155)
§ 9.5	复杂问题	(157)
§ 9.6	数值计算和统计函数	(158)
§ 9.7	模式匹配测试	(159)
§ 9.8	排 序	(160)
§ 9.9	用 GROUP BY 分组	(160)
§ 9.10	分组搜索条件	(162)
§ 9.11	删除重复行	(162)

§ 9.12	多表查询.....	(163)
§ 9.13	数据插入.....	(164)
§ 9.14	数据更新.....	(165)
§ 9.15	数据删除.....	(166)
第十章	数据窗口.....	(168)
§ 10.1	DataWindow 对象	(168)
§ 10.2	创建 DataWindow 对象	(169)
§ 10.3	DataWindow 的增强功能	(200)
§ 10.4	行的操纵.....	(207)
§ 10.5	更新特性.....	(211)
§ 10.6	使用 DataWindow	(213)
第十一章	制作用户对象.....	(217)
§ 11.1	创建新的用户对象.....	(217)
§ 11.2	自定义用户对象.....	(218)
§ 11.3	标准用户对象.....	(220)
§ 11.4	外部用户对象.....	(221)
§ 11.5	自定义类.....	(222)
§ 11.6	标准类.....	(222)
§ 11.7	自定义事件.....	(223)
第十二章	面向对象编程.....	(225)
§ 12.1	类和实例.....	(225)
§ 12.2	创建窗口实例.....	(225)
§ 12.3	继 承.....	(225)
§ 12.4	多态性.....	(229)
§ 12.5	重载函数.....	(230)
第十三章	项目管理.....	(231)
§ 13.1	库准则.....	(231)
§ 13.2	打开 Library 描绘器	(232)
§ 13.3	创建和删除库.....	(234)
§ 13.4	更新库说明.....	(235)
§ 13.5	库中的对象.....	(236)
§ 13.6	控制对对象的访问.....	(238)
§ 13.7	引入和引出对象.....	(240)
§ 13.8	搜索库.....	(241)
§ 13.9	打印库报表.....	(242)

§ 13.10	对象重建	(243)
§ 13.11	从早期版本移植	(243)
第十四章	多文档界面	(245)
§ 14.1	什么是 MDI 应用程序	(245)
§ 14.2	MDI 与 SDI 应用程序的比较	(246)
§ 14.3	MDI 应用程序的组成	(247)
§ 14.4	创建 MDI 框架	(247)
§ 14.5	创建底稿	(249)
§ 14.6	MDI 应用程序中的菜单	(249)
§ 14.7	工具栏	(250)
第十五章	连接到 Web	(252)
§ 15.1	PowerBuilder 6.0 中 Internet 工具的改进	(252)
§ 15.2	Window ActiveX	(252)
§ 15.3	编写数据窗口 HTML	(269)
§ 15.4	用于插件和 ActiveX 的安全模式	(278)
§ 15.5	环境信息	(280)
§ 15.6	在 PowerBuilder 内定制 Web 跳转	(287)
第十六章	编程实例	(288)
§ 16.1	系统设计	(288)
§ 16.2	创建应用程序对象	(290)
§ 16.3	创建数据库与数据表	(290)
§ 16.4	DataWindow 对象	(292)
§ 16.5	窗口与脚本	(300)
§ 16.6	菜单及其脚本	(310)
§ 16.7	应用对象	(313)
§ 16.8	一点技巧	(314)
附录 A	PowerBuilder 命名规则	(317)
§ A.1	对 象	(317)
§ A.2	控 件	(317)
§ A.3	变 量	(318)
附录 B	PowerBuilder 函数参考	(320)

第一章 客户机/服务器 计算模式简介

Client/server (客户机/服务器) 这一术语常常引起困惑,这是因为在不同的地方,这一术语被用于指代不同的事情,如果不根据上下文来理解,就很难说出它的具体含义。那么,它到底指的是什么?它源于哪里?现在它又能为我们做些什么呢?

为了回答这些问题,本章将快速回顾计算机的发展历史。本章的目的不是讲述枯燥乏味的计算机科学史,而是讨论一下计算机的各个发展阶段,以及在不同时期人们怎样利用计算机来解决问题,并对当今的潮流进行详细讲述。最后,将讨论 PowerBuilder 是如何把多种新技术集成为一个强大、尖端的开发工具的。

§ 1.1 史前计算机

从算盘到 20 世纪 40 年代,被称为史前计算机时代。随着科学的发展和工业的进步,人们需要处理越来越多的实验和生产数据,这就导致了各种早期的计算工具的出现,这些工具包括算盘、计算尺等等。尽管它们的功能有限,但在它们的年代中被广泛使用,有些至今仍在使用。

§ 1.2 大型主机

从 20 世纪 40 年代到现在,被称为大型主机时代。

战争的爆发,使战争双方的政府都需要大量快速而复杂的计算。战争使进行研究的科学家的早期工作导致出了一庞大的计算工作,使得计算器械成为了一个庞大的工业。战争无疑使计算机技术在短短的几年中向前推进了几十年。

战争过后,人们发现和平的世界中出现了一种新式的功能强大的工具——用来进行科学研究和商业运转的大型计算机。虽然此时的“电脑”体积庞大、耗费惊人,但它们可以完成以前任何工具都不能完成的功能,取得了前所未有的成功。

随后,这些机器飞快地发展,以至于每隔几年就有新的型号出现。它们变得越来越快速,越来越容易使用,也越来越可靠。

一、硬件

大型计算机通常体积很大,作为一个巨大的盒子放在专门的建筑物里,而使用大型机的用户则通过终端来对它进行控制,早期的终端看上去就像一个带有键盘的电板打字机,现在通常已被显示器和键盘所代替。

二、软 件

大型机是批处理式机器。所有运行在主机上的软件都是集中式的，而终端只是作为观察机器内部操作情况的一个窗口。MVS 和 VM 等操作系统使多个用户能够在一台机器上同时运行多个程序，数据集中进行存取和更新。而像打印机和硬盘这样的资源请求通过请求队列进行访问。

三、编 程

早期，科学家使用 FORTRAN 作为编程语言。而 COBOL 则是大多数购买了计算机的商业用户的唯一选择。

对于数据存储和检索的需求导致了分层和网状数据库的设计，它们使数据按照商业规则建模。这些数据库使用硬件连接把表组织起来。分层数据库要求设计者必须按父/子关系建立数据表，而网状数据库的设计则去掉了这种限制，可以更自由地建立表间关系。

四、缺 点

大型机体积巨大，购买和维护都要耗费大量的资源。随着用户数量的增加，它们的速度变得越来越慢。而公司通常无法支付处理器的升级费用，以满足人们对运算能力的需求。

另外，如果出现问题，或者进行维护而必须关掉计算机时，整个公司的工作就不得不停顿下来。整个公司完全依赖于这台计算机，而它又是十分昂贵的商品，以至于不能再购买一台作为备份。

§ 1.3 PC 机

从 20 世纪 70 年代至今是 PC 机时代。

由于微处理器的发明，计算机的威力得到了真正的发挥，现在每个人都能拥有一台自己的个人计算机（以下简称 PC）。最初有人怀疑是否每个人都需要一台 PC，而电子表格和字处理软件的广泛使用打消了这种顾虑。一般的工作人员在 PC 上就能工作自如。

PC 世界的变化相当快——比大型机的变化要快得多。这些机器的功能每隔几年就会成倍增长。PC 的用户界面，从一开始就比大型机要好得多，现在还在不断增强。通信程序使得用户可以把他们的计算机连接到其它的 PC 或大型机上。图像和平面设计工具导致产生了桌面印刷系统。

一、硬 件

一个长约两英尺、高约八英寸的机箱，上面放一台显示器，前面有一个键盘，这就是大多数人进入计算机世界所看到的一切。中央处理器放在机箱里面，输入/输出设备要么嵌在机箱面板上，要么放在机箱旁边。不用等候，所有的资源在任何时间对用户都是可用的。

二、软 件

PC 既能用于批处理环境，又能用于联机环境。PC 的所有软件都是本地的。如果用户想运行另一台机器上的一个程序，那么它必须拥有该程序的一个拷贝。数据是非共享的。PC 的执行速度较慢，而响应时间却较短。尽管 PC 的中央处理器的功能远远不如大型机，但所有的资源都被用户独占，所有处理都直接发生在用户面前。单用户操作系统 MS-DOS 控制着 PC 的

运行。

三、编程

PC 的软件开发与主机相比有很大不同。大型机功能非常强大,因此对于编程来说,内存、外存、运行速度都不是什么太大的问题。而这些对于 PC 来说,却是编程时应考虑的非常重要的因素。最初,大多数编程由汇编语言完成。随后,C 语言逐渐流行起来。C 语言速度快,效率高,比较短小,同时层次也比较高,适于大项目的编写和维护。只要拥有合适的库函数,它几乎可以和任何硬件或外设进行通信。早期,数据以 ASCII 文件形式存在硬盘上。随着数据和程序变得越来越复杂,像 dBASE、Paradox 以及最近的 Microsoft Access 登上了历史舞台。与大型机上的分层或网状数据库相比它们更灵活,更简单,但比 ASCII 文件要强大得多。

四、缺点

当然,PC 也有问题。比如,某公司的销售部的数据库存有一个客户的姓名和地址,相应的客户部和公关部也有同样的记录。但是,改变一个地方的信息并不能保证其它地方的相应信息被更新。数据的重复同时困扰着顾客和雇员。数据的共享几乎无法实现。

§ 1.4 客户机/服务器集成

从 20 世纪 90 年代开始,出现了客户机/服务器集成。

客户机/服务器 (Client/Server) 方式是主机与 PC 的完美结合。它不是一种计算机硬件或软件,而是一种体系结构,是一种使用硬件和软件的方式。在这种方式中,数据、应用程序甚至处理能力都分布在集中存储仓库和所有使用它们的计算机之间。这个中央存储仓库被称为服务器 (Server) 或后台 (back-end)。它接收其它计算机 (客户机或者前台) 的请求。然后服务器处理这些请求并返回结果。它能很好地工作得益于所有客户机都能共享服务器所提供的有限的服务,而且服务器和客户机之间用同一种语言进行通讯。

例如对于这样的请求:“给出五月份销售额超过 10 000 元的推销员的名字。”回答是“张某、李某、王某”。这是一个典型的由服务器提供的数据库查询,这时使用的语言是结构化查询语言 (SQL)。在 SQL 中请求是这样表述的:

```
SELECT Name FROM Salesmen WHERE Month='MAY' AND Sales>10000
```

如果你不懂 SQL,没有关系,后面将会用专门的一章来讲述它。现在只需要知道服务、请求、响应和通讯方式这几个概念。

一、硬件

整个系统既不是大型机,也不是主机,而是一组通过局域网 (LAN) 进行通讯的计算机组成的。一种标准配置是:一组不同型号的 PC 通过 Novell NetWare 网络连接到一台作为服务器的高档 PC 上。当然,小型机 (如 Sun 或 Sequent)、中型机 (如 DEC VAX、AS1400) 甚至大型机 (如 IBM 3090) 均可以作为服务器。也可以有多台服务器,每台提供不同的服务 (如文件服务器、数据服务器、传真服务器等等)。

二、软件

大型机的优势在于它的存储能力和处理大量数据的能力。PC 的优势在于它与用户的交互能力。为了充分利用两者的长处,应把应用程序分割为两部分:服务器端提供数据处理服务

和大量数据的存储/检索功能；客户端控制用户界面以及进行特定的数据分类和过滤，这样就能取得最佳的效果。

三、编程

要得到最佳效果就必须付出代价。用户界面、数据库管理、数据共享之间连结的复杂性使人生畏，并且当今的商业活动要求更加快速的系统开发。那么怎么来做到这一点呢？软件开发的一些新趋势也许能给出答案。

§ 1.5 趋势

下面我们将介绍几种不断发展的激动人心的新技术，并讨论 PowerBuilder 是如何把它们集成在一个统一环境里的。

一、软件工程/CASE 工具

尽管已不是什么新概念，软件工程近几年仍然是热门话题。它是软件满足用户的特定需要和避免设计编码过程中出错的保证。它把一种严格的规则引进了软件设计和开发过程。

编写程序之前，应先绘制框图、流程图，再写出详细的伪代码，这就是自顶向下方法，即是从一般到具体，从最高层次的抽象到最低层次的细节。研究表明：这种方法对于大型软件可以产生精确的结果，但对于中大项目却还有很多争论。

计算机辅助软件工程（CASE）工具是一些软件包，用来帮助人们在开发过程中设计规格定义、图表，甚至能为软件工程项目自动编写代码。可以实现从简单的图表软件包到复杂的分析、设计和代码生成模块。对于大型项目来说，CASE 工具是必备的。尽管 PowerBuilder 不是 CASE 工具，但它可以和许多 CASE 工具集成使用，因而也是一种理想的工具。

二、面向对象编程（OOP）

传统编程将程序和数据分开对待，而 OOP 方法却把二者结合到一个称为类（Class）的可重用单元中。类就像 Pascal 语言中的记录和 C 语言中的结构，但它除了数据（OOP 中称为属性）之外还含有过程（OOP 中称为方法）。你可以用定义特殊结构或记录类型变量的方法来实例化（生成一个实例）一个类。实例化的类被称作对象（Object）。由于对象中的代码像变量一样容易复制和访问，因此可重用性很容易实现。

OOP 方法的另一个好处是继承。当创建一个类以后，就可以创建这个类的派生类，派生类与父类相似，但包含有更多的数据和功能。使用继承，就可以建一个基础类作为许多子类的祖先类。

由于面向对象编程与传统编程有很大不同，因此程序员要花大量的时间才能掌握它。

三、开放系统

“开放系统”这一术语通常被用来指硬件系统具有可扩充性和允许与其他厂商提供的系统进行连接。对程序员来说更关心的是软件领域的开放系统。

就在不久以前，几乎所有的计算机公司都认为他们是世界上唯一的计算机公司，写出的程序只能在自己的机器上运行。除非你只使用一家的产品，否则数据共享几乎是不可能的。这就迫使你必须选择一家公司，坚持使用它的产品，并期望它不会很快过时。

令人欢欣鼓舞的是，这样的日子已经一去不复返了。现在，大部分的软件都提供很强的

数据共享能力,有的甚至拥有与其它软件商的软件包进行集成的能力,甚至还要求与其它厂商的软件包组合使用。

作为最开放的系统之一,PowerBuilder 脱颖而出。它不仅能与所有的流行后台数据库连接,还可以无疑地集成众多开发工具。

四、OLE 2

Windows 用相同的用户界面和剪贴板把各个应用程序集成在一起,通过它们,用户可以把数据从一个应用程序传递到另一个应用程序。现在,由于有了 OLE 2,多个应用程序自动集成成为了现实。

OLE 2 可以使其他应用程序(如 Word 或 Excel)的文档被链接或嵌入到你的应用程序中,以增强它的功能。另外,OLE 功能允许一个应用程序通过发送命令的方式控制一个或多个在后台运行的应用程序。

PowerBuilder 支持 OLE 2 功能。

五、图形用户界面

GUI(图形用户界面)出现之前,使用 PC 是一件非常困难的事。在图形用户界面下,不再需要用户在提示符下输入命令,用户只需使用鼠标在图形环境中选择图标即可完成操作。这是计算机历史上的转折点。今天,Microsoft 公司的 Windows 和 Windows 95 是最有名的、占统治地位的图形用户界面,它们给人们的工作带来了极大的便利。

PowerBuilder 允许创建图形用户界面,帮助用户理解和操纵复杂的数据结构。这在基于字符的系统中是无法实现的。

六、事件驱动编程

在传统的编程过程中,你编写一个过程然后在别的地方调用它。而事件驱动的编程方式完全改变了这种状态。因为 Windows 是一个事件驱动的操作系统,编程必须按它的规则行事,没有别的选择。

事件驱动的基本思想是这样的:程序开始运行并初始化后就停顿下来,只有当用户作出动作,例如选择菜单、点击鼠标,程序才会作出反应,是用户而不是程序员在控制程序。程序员必须对事件编写代码,告诉程序“当用户点击这个按钮,就调用这一过程;当用户选择菜单,调用那一个”。当你熟悉这种编程方式后,会发现这样很容易进行分段编码和测试。

七、可视化开发环境

Windows 3.0 刚发布时,除了用 Windows SDK 以 C 语言编写 Windows 程序外没有别的办法,即使是编写最小的程序也要耗费很多时间。现在,一批新的开发环境使得编程序就像使用 Windows 一样简单有趣,可以像画画一样画出应用程序窗口。

PowerBuilder 是可视化开发环境,它的一个重要优点就是明显降低了开发所耗费的时间。

八、组件/集成开发人员

放眼未来,在不久的将来会出现这样的景象:只有两种类型的开发人员,即组件开发人员和集成开发人员。

组件开发人员使用像 C/C++ 和汇编这样的语言开发一些许多程序都可以使用的常用对象。这些对象包括组合框、拼写检查器和任何可以用于多种环境下的具有特定功能的对象。

集成开发人员在可视化开发环境中将这些组件连在一起,用一种简单的语言为特定需求创建一个完整的应用系统,例如某公司的工资系统或者客户服务系统等等,这样的可视化开发环境使应用程序能够很快地利用各种预制件按照具体需求建造起来。

PowerBuilder 适合作为集成开发人员的工具来进行 Client/Server 应用程序的开发。

§ 1.6 PowerBuilder 简介

在前面各节中,我们介绍了什么是客户机/服务器以及一些新技术,下面我们介绍 PowerBuilder 的强大功能和威力。

一、可视化开发环境

PowerBuilder 本质是一个面向客户机/服务器开发的完全的可视化开发环境。使用 PowerBuilder,你可以用一种可视的直观的方式来创建应用程序的用户界面和数据库窗口。只要会用鼠标单击和拖曳,就可以创建自己的用户界面。

二、面向对象的工具

PowerBuilder 创造性地把面向对象技术应用到用户界面之中。例如,你可以为主窗口建立一个带有基本控件的父窗口,当需要一个新的主窗口时,直接从该父窗口继承一个,再稍加修改就可以使用了。这样做的好处在于,一旦项目的用户界面要做大的改动,只须对父窗口进行修改,而所有的子窗口就自动地得到了升级。

请务必牢记,PowerBuilder 用户界面的各个组成部分都是对象,它们都有属性、事件、方法。想要改变窗口颜色,就要改变颜色属性;想要在用户点击某按钮时执行某个功能,就要对点击按钮事件编程;想要在运行时移动某个控件,就要执行移动方法。一开始会有些迷惑,但熟悉之后你就会极大地缩短开发时间。

三、功能强大的语言

设计好用户界面以后,你只需右击某控件(如按钮或列表框),从弹出菜单中选择 Script,你就会看到一个编辑器,你可以在这里输入 PowerBuilder 的高级语言——PowerScript。

每个对象有特定的事件,你需要分别为它们编写代码。例如,你可以为刚刚发生在窗口中的按钮的 Clicked 事件编写脚本,这样当用户单击该按钮时就会选择这段代码,你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。

使用这种分段代码,你就不能按你所习惯的那样同时看到某个程序的所有代码行。不过,这种分段代码的使用非常方便,你不需要对大量的代码搜索,就能直接找到所需代码,而且也能自动形成好的程序结构。

四、开放的体系

没有任何开发环境能满足你的全部需求,这就是为什么 PowerBuilder 是一个开放的、协作的环境的原因。作为一个开放系统,PowerBuilder 的突出特点是可以访问任何一个常用的后台数据库系统,例如:Oracle、Sybase、Informix 或 DB2。此外,PowerBuilder 可以通过一个叫做客户机/服务器开放开发环境(CODE)的规范与其它 PC 产品进行集成。

五、多平台开发环境

这是一个纷繁复杂的世界,你的公司中的用户可能使用不同类型的硬件和不同类型的操

作系统,这一事实使得企业级的开发相当困难。不过,由于 PowerBuilder 有了 Macintosh 和 UNIX 版本,PowerBuilder 使这一切变得简单了。它不仅意味着你可以在不同操作系统平台上使用相同的开发环境,还意味着在一个操作系统中开发的应用程序,只需把它拿到别的操作系统中重新编译,就可以直接运行了。

本书主要讲述 PowerBuilder 6.0 for Windows 95,但对其他环境中的 PowerBuilder 6.0 也是适用的。

六、企业的客户机/服务器开发方案

熟悉 PC 数据库开发工具的人也许会对 PowerBuilder 持怀疑态度。PowerBuilder 和 dBASE、Paradox 或 Access 之间的差别到底有多大呢?回答是:差别很大。

尽管 PC 数据库开发已经历了很长一段时间,但 PowerBuilder 可以真正地、完全地支持企业级开发。dBASE 等数据库软件主要用来开发单用户程序,并且在数据的安全性、完整性方面有较大缺陷。PowerBuilder 与所有流行的客户机/服务器数据库都有接口,因此你可以按照功能、安全性、支持能力等原则选择合适的数据库。并利用 PowerBuilder 在多用户环境下访问数据库。在用 PowerBuilder 进行开发的过程中可以保证安全性和数据完整性。从管理角度看,这些都是区别,是工业化的数据库与其它数据库的区别。

七、整套的工具集

PowerBuilder 是一整套能够满足不同需要的数据库工具集。InfoMaker 是独立于 PowerBuilder 的产品。它向最终用户提供查询和报表功能。它使高级用户可以按自己的需要创建查询和报表。PowerBuilder Desktop (桌面版)是一个完整的 PowerBuilder 版本,去掉与客户机/服务器数据库的连接之后,其价格只有 PowerBuilder Enterprise (企业版)的几分之一。

§ 1.7 PowerBuilder 6.0 的新特性

PowerBuilder 6.0 是 Sybase Powersoft 系列工具产品家族的一个重要组成成员,致力于为新兴计算领域的企业级商业提供基于组件的应用开发模式,包括:客户机/服务器环境,分布式环境与 Internet 环境。PowerBuilder 满足了那些希望在 4GL RAD 环境中创建完整应用和应用组件的开发人员的需要。特别是在 6.0 版中,PowerBuilder 更进一步在如下几个重要方面增强了领先于业界的企业级开发特性。

- Web——支持中间层应用逻辑的快速高效开发,并可使客户通过 Web 访问该逻辑。
- 组件——可利用 4GL RAD 工具制造并集成各种组件。
- 开放——可灵活适用于各种变化的企业计算环境和异构环境。

另外,PowerBuilder 6.0 通过领先的增强的开发界面,使应用的开发效率达到了最高。

以下详细介绍 PowerBuilder 6.0 的新特性。

一、使用组件更加轻松

今天,由于应用的扩展性代码的重用性、灵活性与维护性的需求增加,促使应用的体系结构正逐步向分布式、基于组件的方向发展。PowerBuilder 将充分支持这种应用体系结构的迁移。

1. 结构化的对象生成器

PowerBuilder 6.0 包含了一个可扩展的组件生成器的结构框架。在 6.x 版的生命周期内, PowerBuilder 将会增加相应的功能, 将 PowerBuilder 开发的对象 NVO 生成 C++、CORBA、DCOM, Java 以及其他标准的对象类型。

2. 支持 Sybase Jaguar 组件事务服务器

Jaguar CTS 支持 PowerBuilder 对象 (NVO), OLE 自动服务器。

3. 为 Microsoft Transaction Server 提供基本支持

PowerBuilder 6.0 可以作为 MTS 对象的一个使用者, 也可以作为 MTS 服务器上的服务器 (OLE 自动对象)。目前尚缺乏对事务的支持能力。

4. 异步应用服务器处理

异步处理提供了在应用服务器上客户请求的排队机制, 允许客户发出一个对应用服务器上方法的调用请求, 而且不需要客户端停下等待这个方法执行完毕。

5. Server Push (应用服务器回调)

Server Push 使得分布式 PowerBuilder 应用服务器可以向应用的客户端回送消息, 既可支持以异步的方式传送结果给客户, 也可向客户端发送信号。

6. Name Server 实用工具

Name Server 实用工具, 可以实现物理应用服务器的逻辑划分, 在类似的服务器之间均衡客户的连接负载, 也可以为分布式 PowerBuilder 应用服务器提供连接信息。

7. 应用服务器共享对象

共享对象允许多个客户会话共享在应用服务器上运行的一个实例对象。对象的命名实例可以作为共享对象注册。一旦注册, 共享对象就可被应用服务器会话里的任一客户会话所访问。

8. IBM AIX 与 HP-UX 平台支持

PowerBuilder 6.0 还将提供 IBM AIX 和 HP-UX 平台的支持, 以满足在这些平台上对 PowerBuilder 和 PowerBuilder 应用运行的市场需求。

9. Window ActiveX

Window ActiveX 是一个 PowerBuilder 窗口的 ActiveX 代理, 与 Window Plug-in 类似, Window ActiveX 可以嵌入任意一个支持 ActiveX 的应用当中, 通过 PowerBuilder 虚拟机运行 Window ActiveX 相应的 PowerBuilder 窗口。可提供标准模式与安全模式两种方式的 Window ActiveX。

二、支持 Web

支持开发人员在 Web 上分发的应用的需要, PowerBuilder 6.0 增加了以下新特性:

1. 将 Internet 开发工具集成到 PowerBuilder 的开发环境中

并囊括在 Internet Developer Toolkit for PowerBuilder 5.0 开发包中的所有组件, 这些组件包括:

- Web. PB DLL——用于 CGI、ISAPI 和 NSAPI Web 服务器。
- Web. PB 类库——生成 HTML 以及管理浏览器的连接状态。
- Web. PB 向导——生成用于激活分布式对象服务的 HTML 页面。
- PowerBuilder Window Plug-in DLL——标准和安全模式版本。
- Window ActiveX——标准和安全模式版本, 与 Plug-in 类似, 但可以与 HTML 中的

JavaScript 和 VBScript 交互。

- PowerBuilder Data Window Plug-in DLL——仅有标准版本。
- O'Reilly WebSite Web 服务器。

2. PowerBuilder VM 虚拟机

将 PowerBuilder 执行所需的有关文件和动态连接库形成一个统一的 PowerBuilder 运行环境——PowerBuilder VM 虚拟机，以简化 PowerBuilder 执行环境的分发。

3. 同步工具

这个新增加的功能，可以检验任何一组文件以确定是否包括所有 DLL 的最新版本，如果没有，当这些 DLL 被使用时，新版本的 DLL 会自动地进行更新，Synchronizer 对于检验 PowerBuilder VM 的本地拷贝尤其有用。开发人员可以设置 Synchronizer，使其在机器或应用启动时自动运行。PBSynch 可以支持 16 位和 32 位的 Windows（不支持 Macintosh，UNIX）也可以 ActiveX 的方式进行使用。

4. 安全模式 Window Plug-in/ActiveX

为安全模式的 PowerBuilder Window Plug-in 和 ActiveX 提供了另一套 DLL。安全版本可以防止 Plug-in/ActiveX 对用户的机器上的内容进行任何修改。

5. Context 对象

ActiveX 或 Plug-in 可使用一种称为 Context 的全局对象，该对象提供了 ActiveX 或 Plug-in 组件所能访问的信息和服务。这种对象能够为 PowerBuilder 程序设计人员提供 Internet 和环境信息服务，例如：允许发送、接收消息或发出 HTTP 请求。

6. HTML 增强

PB6 增加了创建 HTML Form 的能力。对于自由格式 (Free form) 或表格式 (Tabular) 的 DataWindow，可以为一行或一定范围的行，也可对所有列或一定范围的列生成相应的 HTML Form。具有非零 Tab Order 的列成为输入域，HTML Form 的 INPUT 域可从列名派生。文本、计算列等成为文本，PB 的按钮对应 HTML 中的按钮，其他的 PB 控件映射为 HTML 中适当的控件（单选钮、复选框、列表框等等）。所有的输入域都可形成相应的 HTML Table 语法。对于嵌入的嵌套 DataWindow，则可生成 HTML 语法中的 HTML Table。

增强 DataWindow HTML 的生成，更好地保留了 DataWindow 的表现风格，更为灵活轻松的生成动态 Web 页面。

7. Web. PB Shopping Carts

允许 Web. PB 在同一参数名下接受多个值。允许多项选择，从而能支持受购类型的应用。

8. JavaBean 代理

在 PowerBuilder 6.0 的产品生命周期里，PowerBuilder 将会支持 JavaBean 代理 (Proxy) 的创建，使分布式 PowerBuilder 对象与分布式 PowerBuilder 服务器对象之间进行通讯。JavaBean 代理是 PowerBuilder 代理写成 JavaBean 的一种形式，使得 JavaBean 代理可以进行下载并运行于任意一台 Java 虚拟机上。通过 JavaBean 代理，Java Applet 能够直接调用分布式 PowerBuilder 服务器上的方法。

三、开放技术

PowerBuilder 6.0 的增强，继续信守产品开放的承诺，并且包括了对使用平台、数据库接口、国际语言和事务服务器等技术的支持。