

活用学活

POWERBUILDER 5.0

最新流行软件丛书



王伟 郭新明 等编
西南交通大学出版社

tp311.56
WW/1

活学活用 PowerBuilder 5.0

王 伟 郭新明 等编
伍盛坤 欧 阳



西南交通大学出版社

044286

内 容 提 要

PowerBuilder 是当今最先进的数据库开发工具之一，它以开放性、可移植性、易用性而闻名于世。

近年来，PowerBuilder 在中国市场的 RAD 工具中占有很大份额，主要用于政府部门、企业集团和大中型企业的管理信息系统的开发。

PowerBuilder 的突出特点是给应用开发人员提供了非常方便的开发环境和工具，这些工具简化了分析和设计应用系统的构造工作。可以借助它们完成复杂的事务处理、报表和用户界面系统。因此大大缩短了工作周期。

本书介绍运行在 Windows 95 环境下的 PowerBuilder 5.0 应用程序的开发，并以大量的实例贯穿其中，使初学者能够在较短时间内迅速掌握 PowerBuilder 5.0 这一强有力的开发工具。

JS259/24

活学活用 PowerBuilder 5.0

王 伟 郭新明 等编

责任编辑 吴晓黎

*

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：20.5

字数：497 千字 印数：1—5000 册

1997 年 8 月第 1 版 1997 年 8 月第 1 次印刷

ISBN 7-81057-098-6/T·269

定价：30.00 元

目 录

第一章 客户机/服务器方式简介	(1)
1.1 原始计算机	(1)
1.2 大型主机	(1)
1.3 PC	(2)
1.4 客户机/服务器	(3)
1.5 趋势	(3)
1.6 PowerBuilder 简介	(5)
第二章 快速入门	(8)
2.1 Painter	(8)
2.2 准备工作	(8)
2.3 运行 PowerBuilder	(8)
2.4 库 Painter	(9)
2.5 应用 Painter	(11)
2.6 数据库 Painter	(13)
2.7 数据窗口 Painter	(17)
2.8 窗口 Painter	(20)
2.9 第一次运行	(25)
2.10 第二次运行	(28)
2.11 第三次运行	(29)
2.12 进一步完善	(29)
2.13 创建可执行文件	(30)
第三章 PowerBuilder 方式	(32)
3.1 与传统环境的区别	(32)
3.2 理解 PowerBuilder 对象	(33)
3.3 Painter	(34)
3.4 工具条的使用	(38)
3.5 联机帮助	(39)
3.6 快捷键	(40)

第四章 创建应用	(41)
4.1 应用 Painter	(41)
4.2 应用 Painter 主窗口	(41)
4.3 更改当前应用	(42)
4.4 查找对象	(43)
4.5 创建新应用	(44)
4.6 指定应用属性	(45)
4.7 应用事件	(48)
4.8 存盘和退出	(50)
第五章 创建窗口	(51)
5.1 启动窗口 Painter	(51)
5.2 Select Window 对话框	(51)
5.3 窗口 Painter 主窗口	(52)
5.4 窗口	(52)
5.5 保存窗口	(57)
5.6 打印窗口	(58)
5.7 预览和运行	(58)
5.8 增加控件	(58)
第六章 制作菜单	(78)
6.1 启动菜单 Painter	(78)
6.2 菜单 Painter 主窗口	(78)
6.3 制作菜单	(79)
6.4 预览	(80)
6.5 创建级联菜单	(80)
6.6 菜单项属性表	(81)
6.7 事件与脚本	(83)
6.8 保存菜单	(83)
6.9 与窗口相联	(84)
6.10 创建弹出菜单	(84)
第七章 Power Script 编程	(85)
7.1 Power Script Painter	(85)
7.2 编辑文字	(86)
7.3 粘贴	(86)
7.4 文本输入与输出	(88)

7.5	语言帮助	(88)
7.6	编译	(88)
7.7	建立自定义函数	(89)
7.8	结构	(92)
7.9	Power Script 基础	(93)
7.10	变量和作用域.....	(94)
7.11	常量.....	(97)
7.12	常用函数和方法.....	(98)
7.13	选择结构.....	(100)
7.14	循环.....	(102)
7.15	嵌入 SQL	(102)
第八章	数据库定义.....	(103)
8.1	启动数据库 Painter	(103)
8.2	数据库 Painter 主窗口	(103)
8.3	数据库管理系统 (DBMS)	(105)
8.4	数据库	(105)
8.5	表	(106)
8.6	视图	(110)
8.7	数据操作	(111)
8.8	数据库管理 Painter	(116)
第九章	SQL 简介	(119)
9.1	SQL 历史	(119)
9.2	SQL 现状	(119)
9.3	SQL 基本结构	(120)
9.4	数值计算和统计函数	(122)
9.5	模式匹配测试	(123)
9.6	排序	(124)
9.7	分组查询	(125)
9.8	分组搜索条件	(125)
9.9	重复行剔除	(126)
9.10	多表查询.....	(128)
9.11	数据插入.....	(128)
9.12	数据更新.....	(128)
9.13	数据删除.....	(131)
第十章	开发数据窗口.....	(133)

10.1	数据窗口对象	(133)
10.2	启动数据窗口 Painter	(133)
10.3	显示格式	(134)
10.4	数据源	(140)
10.5	数据窗口布局	(145)
第十一章	数据窗口应用	(146)
11.1	数据窗口控件	(146)
11.2	事务处理对象	(147)
11.3	为数据窗口指定事务处理对象	(149)
11.4	获取数据	(149)
11.5	事务处理	(149)
11.6	数据更新	(150)
第十二章	制作用户对象	(152)
12.1	自定义的可视用户对象	(153)
12.2	标准用户对象	(154)
12.3	外部用户对象	(154)
12.4	自定义类	(154)
12.5	标准类	(155)
12.6	使用用户对象	(155)
12.7	自定义事件	(155)
第十三章	面向对象编程	(157)
13.1	类和实例	(157)
13.2	创建窗口实例	(157)
13.3	继承	(157)
13.4	多形	(160)
13.5	函数重载	(160)
第十四章	项目管理	(161)
14.1	库 Painter	(161)
14.2	注释	(162)
14.3	创建和删除库	(163)
14.4	对象	(164)
14.5	库的优化	(165)
14.6	对象重建	(166)
14.7	引入和引出对象	(166)

第十五章 多文档界面	(168)
15.1 创建 MDI 框架	(168)
15.2 创建子窗口	(169)
15.3 MDI 中的菜单	(169)
15.4 工具条	(169)
第十六章 实例	(171)
16.1 创建应用程序对象	(171)
16.2 创建窗口	(173)
16.3 菜单	(178)
16.4 数据库	(182)
16.5 数据窗口	(187)
16.6 脚本	(194)
第十七章 数据库高级技术	(218)
17.1 锁定	(218)
17.2 事务处理	(222)
17.3 数据窗口的事务函数	(228)
第十八章 数据窗口高级技术	(236)
18.1 SQL 预览	(236)
18.2 仅取所需	(237)
18.3 数据窗口缓冲区	(237)
18.4 记录/字段的状态	(239)
18.5 使用子数据窗口	(240)
18.6 动态数据窗口	(242)
18.7 Modify () 和 Describe () 函数	(245)
18.8 把数据源由 SQL SELECT 改变为存储过程	(254)
18.9 查询模式	(255)
18.10 数据窗口层次	(258)
18.11 属性的条件表达式	(259)
附录 A PowerBuilder 命名规则	(262)
A.1 对象	(262)
A.2 控件	(262)
A.3 变量	(263)
附录 B PowerBuilder 函数参考	(265)

第一章 客户机/服务器方式简介

客户机/服务器一词常常引起困惑，从字面上很难看出它的确切含义。那么它到底指的是什么，它的来源是什么，又能为我们做些什么呢？

为了回答这些问题，让我们先看看计算机发展的历程。这里并不打算讨论计算机科学史，而只是简单回顾一下计算机发展的几个阶段，在不同时期人们怎样利用它来解决问题的，以及当今的潮流，最后看一看 PowerBuilder 是怎样把多种新技术集成为一个强大、尖端的开发工具的。

1.1 原始计算机

随着科技和工业的进步，人们需要处理越来越多的实验和生产数据，这就导致了诸如算盘、计算尺、手摇计算机等早期计算器械的出现。尽管功能有限，它们在当时还是得到了广泛的应用，有些至今仍在使用。

1.2 大型主机

二战的爆发，使得计算器械成为了一个庞大的工业。交战双方都需要快速地进行诸如火箭弹道这类复杂数据的计算，而不仅仅是简单的加减法计算。战争使计算机技术在短短的几年间得到了突飞猛进的发展。

战后，人们发现科学和商业数据的处理有了一个功能强大的新工具——主机计算机，此时的“电脑”体积庞大、耗资惊人，但也取得了前所未有的成功。

1.2.1 硬件

主机计算机通常体积很大，安装在专门的建筑物里，用户通过终端对它进行操纵，早期的终端是带键盘的电传打字机，现在已被显示器和键盘所取代。

1.2.2 软件

主机上运行的软件都是集中管理，终端的作用只是主机对外部的窗口。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。打印机和硬盘等资源通过请求队列进行访问。

1.2.3 编程

早期，科技人员使用 FORTRAN 语言，而对于众多的商业用户来说，唯一的选择是

COBOL。

对数据存取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。这些数据库通过硬连接把表组织起来。分层数据库迫使程序员按父/子关系建立数据表，而网状设计取消了这一限制，使连接更加自由。

1.2.4 缺点

购买和维护主机耗费巨大。随着用户数量的逐渐增加，它们的速度变得越来越慢，公司通常无法支付处理器的升级费用，以满足对运算能力需求。批处理的方式也使得用户界面很难掌握。

最大的问题是一旦机器出了故障停机检修，整个公司的工作也就不得不停顿下来。这样，对计算机的依赖性太强，而全面备份又太昂贵。

1.3 PC

由于微处理器的发明，使计算机的威力真正得到了发挥，现在每个人都能拥有一台 PC。最初有人怀疑 PC 的用途，而电子表格和字处理软件的广泛使用打消了这种顾虑。电子表格完成了许多几年前还需要主机才能完成的工作，无需专门的技术人员指导，一般的工作人员就能使用自如。而字处理程序给人们的写作方式带来了革命性的变化。此外还有各种电子游戏……，PC 的变革异常迅速，远远超过了主机。PC 的性能每隔几年就会翻番。用户界面远胜于主机，而且还在不断改进，小商小贩也能使用功能强大的数据库软件进行数据存取；通讯软件使用户能将 PC 与 PC、PC 与主机连接起来；图象和平面设计工具把出版业带到了桌面。

1.3.1 硬件

一个机箱、上面放一台显示器、前面有一个键盘，这就是无数人的工作环境。处理器在机箱里面，一部分输入/输出设备嵌在机箱面板上（驱动器）；另一部分放在旁边（打印机），无需等待，所有资源立等可取。

1.3.2 软件

PC 的所有软件都是本地的，一个程序运行在一台机器上，想要在另一台上运行就必须拷贝，数据是非共享的，运算速度慢，而响应时间却较短。尽管 PC 的中央处理器比起主机来，功能明显要差，但所有的资源都由用户独占。单用户操作系统 MS-DOS 控制了一切。

1.3.3 编程

PC 的软件开发与主机相比有很大不同。主机功能强大，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 PC 却很重要。最初，大多数编程由汇编语言完成，随后 C 语言流行起来，C 语言快速、高效、短小，同时又比较高级，便于大的项目的编写和维护。只要有合适的库，几乎能与所有的硬件和外设通讯。早期，数据以 ASCII 文件形式存在硬盘上，随着用户和程序的要求越来越复杂，关系数据库 dBase、Paradox 以及最近的 Microsoft Access 登上了舞台。与主机上的分层或网状数据库相比，它们更灵活、更易于使用。

1.3.4 缺点

当然，PC 也有问题。比如，一个公司市场部的数据库存有一个客户的姓名和地址，相应的信贷部和客户部也有同样的记录，改变其中的一个并不能使其他的得到更新。数据重复同时困扰着顾客和雇员，数据共享几乎无法实现。

1.4 客户机/服务器

客户机/服务器方式是主机与 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，一种同时使用硬件和软件的方法。应用程序、数据甚至运算能力集中放置在服务器一端，服务器接收并处理其他计算机（客户机）发来的请求，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，并且服务器和客户机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 100 000 元的推销员的名字。”，回答是：“张三、李四、王五”。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 > 100000
```

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。

1.4.1 硬件

整个系统不是一台 PC 或主机，而是由一组通过局域网（LAN）进行通讯的计算机组成的。标准配置是：一组不同型号的 PC 通过 Novell NetWare 网连接到一台作为服务器的高档 PC 上。当然小型机（如 SUN）、中型机（如 DEC VAX、AS/400）甚至大型机（如 IBM 3090）也可以作为服务器。也可以配置数台服务器，每台提供不同的服务（比如文件服务器、数据库服务器、传真服务器等等）。

1.4.2 软件

主机的强项是大规模数据的存储和处理，PC 的强项是交互性。为了充分利用两者的长处，把应用程序分为两部分，服务器端进行大量数据的访问和存取；客户机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。

1.4.3 编程

要得到这一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。怎样解决这一问题呢？软件开发的一些新趋势也许能给出答案。

1.5 趋势

下面我们将讨论刚刚涌现出来的几种激动人心的新技术，以及 PowerBuilder 是怎样把它

们集成在一个统一环境里的。

1.5.1 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，它是软件符合用户需求，及无错设计和编码的保证。它把其他行业的工程学原则引入了软件设计和开发。

编写程序之前，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：从普遍到特殊；从抽象到具体。研究表明对于大规模项目，自顶向下方法取得了良好的结果，对于中小项目却还有很多争论。

计算机辅助软件工程（CASE）工具是一种软件包，可以完成需求设计、流程图，甚至编码的辅助开发工作。对于大型项目来说，CASE 工具是必备的。尽管 PowerBuilder 不是一个 CASE 工具，但它包含了几个 CASE 工具（例如 ERWin 和 LBMS）。

1.5.2 面向对象编程（OOP）

传统编程将程序和数据分开对待，而 OOP 把两者结合成一个称为类（class）的可重用单元。类与 C 语言中的结构和 Pascal 语言中的记录很相似，但它除数据（OOP 中称为属性）外还含有过程（OOP 中称为方法）。说明类与说明结构或记录变量类似，对象（object）是类的实体。由于对象中的代码象变量一样容易复制和访问，因此可重用性很容易实现。

OOP 的另一个优点是继承。当建立一个类以后，就可以在它的基础上建立其他的与它类似但含有更多数据和功能的派生类。通过这一方法，就能把一个基础类作为其他子类的父类。

面向对象编程与传统编程有很大不同，因此掌握它需要花一定时间，详细的讨论已超出了本书的范围，下面举一个简单的例子加以说明。

假设一个大学的信息系统。有一个称为“人员”父类，它有一些属性（变量），如名字、地址、籍贯，和一些方法（过程）用来更新、打印属性字段。你可以从“人员”类派生出一个称为“学生”的类，它有一些附加的属性，比如课程表、成绩单，以及相应的方法。还可以从“人员”派生出“职员”类，它除了有“人员”的全部数据和方法外，还有工资、职位等属性及相应方法。从“职员”类又能派生出“教授”类，“教授”又有学科、教龄等属性。通过这样的继承就能够建立直观易懂的应用程序。

1.5.3 开放系统

“开放系统”一词更多的是指不同厂商的硬件产品可互连和可扩充。对程序员来说更关心的是软件领域的开放系统。

不久以前的一段时期，似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。这样一来，你不得不绑死在一种系统上，并期望它不会很快过时。

幸好这样的日子已经过去了。如今，多数的软件包都提供很强的数据共享能力，甚至有些还要求与其他厂商的软件包组合使用。

作为最开放的系统之一，Powersoft 公司的 PowerBuilder 脱颖而出，它不仅能与所有的流行数据库后端连接，并且无缝地集成了众多开发工具（参见 1.6.4 节）。

1.5.4 OLE 2

通过用户界面和剪贴板, Windows 让使用者能够把数据从一个应用程序传送到另一个应用程序。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 使得不同应用程序(如 Word、Excel)的文档能够链接或嵌入到你的程序中去以增强它的功能。另外, OLE 自动化使一个应用程序能够通过指令控制一个或多个其他程序。

1.5.5 图形用户界面

GUI(图形用户界面)出现之前, PC 难于使用, 第一个 GUI 是由施乐公司的帕洛阿图研究中心开发的, 最早应用这一成果的是苹果公司的 Lisa 和 Macintosh 机。

用户不再需要键入命令, 而是在图形环境下用鼠标选择图标, 这是计算机历史上的转折点。今天, 微软的 Windows 和 Windows 95 是最有名的 GUI, 它们给人们的工作带来了极大的便利。

1.5.6 事件驱动编程

在传统编程中, 你编写一个过程然后在别的地方调用它, 事件驱动编程则完全不同, 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。

事件驱动的基本概念是这样的: 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, 例如点击鼠标、选择菜单, 程序才会作出反应, 是用户在控制程序而不是程序员。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。

1.5.7 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。现在出现了一些开发环境, 使 Windows 编程变得轻松有趣, 可以像画画一样画出应用程序窗口。

1.5.8 部件/方案程序员

我们可以预见, 在不久的将来会出现这样的景象: 只有两种程序员——部件开发者和方案开发者。

部件开发者用 C/C++ 或汇编语言编写不同程序的通用对象部件, 例如组合框、拼写检查器等等。而方案开发者在可视化开发环境里用一种简单的语言把各种部件组合成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。

1.6 PowerBuilder 简介

前面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。

1.6.1 可视化的开发环境

PowerBuilder 本质上是一个面向客户机/服务器方式的完整的可视化开发环境。通过 PowerBuilder 你可以可视地、直观地为应用程序设计用户界面和数据库接口。只要会点击和拖曳你就能制作出色的用户界面。

1.6.2 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。

需要牢记的是 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。

1.6.3 功能强大的语言

设计好用户界面之后，要做的就是右击某控件（如按钮或列表框），从弹出菜单中选择 Script，得到一个编辑器，这就是输入 PowerBuilder 的高级语言——PowerScript 的地方。

每个对象有特定的事件，你需要分别为它们编写代码。比如为一个按钮的点击事件编写代码后，当用户点击该按钮这段代码就会被执行。

如果你习惯于一次看到整个程序的全部代码，现在就需要改变观念，分段编码更容易，而且使程序有更好的结构。

1.6.4 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 的突出特点是能够访问任何通用数据库后端，例如 Sybase、Informix、Oracle 或者 DB2。此外 Powersoft 还与其他许多客户机/服务器开发工具商结成了长期战略伙伴关系（CODE—客户机/服务器开放开发环境），使 PowerBuilder 能与其他 PC 产品集成，这就极大地方便了开发者的工作。

1.6.5 多平台的开发环境

世界不是单一的，人们使用不同的硬件和操作系统，这一事实使得企业级开发极为困难，但由于有了 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，还意味着应用程序在不同平台间的移植只需重新编译即可。

本书主要讲述 PowerBuilder 5.0 for Windows 95，但对其他平台的 PowerBuilder 5.0 也是

适用的。

1.6.6 企业级的客户/服务器开发方案

熟悉 PC 数据库开发工具的人也许会用怀疑的眼光看待 PowerBuilder, 它和 dBase、Paradox、Foxpro 或 Access 到底有多少区别? 回答是区别很大。

尽管 PC 数据库开发已经有了一段时间, PowerBuilder 为真正的企业级开发提供了全新的功能。dBase 等软件主要是用来开发单用户程序, 并且在安全性、数据完整性方面有很大缺陷, 而这些对于企业级数据库却是关键的。PowerBuilder 能够与所有流行的客户机/服务器数据库连接, 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, 也可以在多用户环境下访问数据库。最后, 在用 PowerBuilder 进行开发的过程中安全性和数据完整性是作为一个整体来考虑的。这些就是 PowerBuilder 和 dBase 等软件的区别, 从管理的眼光来看, 这些不同之处决定了一个数据库是否具备“工业强度”。

1.6.7 整套的工具

不久以前, PowerBuilder 和 Powersoft 公司还是同义词, PowerBuilder 是该公司的唯一产品, 现在它已经不是一个单独的产品, 而是一整套数据库工具, 能够满足不同的需要。

第二章 快速入门

本章将通过建立一个通讯录应用程序示例，让你粗略地了解一下 PowerBuilder 及其开发环境。在完成这个示例的过程中你也许会有许多疑问，在以后的章节中将对它们作更进一步讨论。

2.1 Painter

PowerBuilder 的所有工作都是由各种 painter（画板）完成的，所谓 Painter 是指开发环境中特定的对象。例如窗口 Painter 拥有创建一个带各种控件的窗口的全部工具；菜单 painter 用来制作下拉菜单。在创建一个应用的过程中将会使用到各种不同的 painter。

2.2 准备工作

在开始这个示例之前，需要在硬盘上为它留出一块空间。

1. 打开 windows 95 中的资源管理器。
2. 在硬盘上建立一个名为 adrsbook 的文件夹。

2.3 运行 PowerBuilder

1. 点击 Windows 95 中的开始（Start）按钮。
2. 选择程序（Program）中的 Powersoft 图标。

注：本书假定你使用的是 Windows 95，并已安装了 PowerBuilder Enterprise 5.0 或 PowerBuilder Desktop 5.0。如果你使用的是 PowerBuilder 的 Windows 3.1、Macintosh 或 UNIX 版本，上述的步骤将会有所不同。

当第一次使用 PowerBuilder，屏幕将会类似图 2.1。

PowerBuilder 的工具条称为 Power 条（PowerBar）在熟悉它们之前，Power 条上的图标显得有些费解。有几种方法可以解决这一问题，其中之一是：

1. 鼠标右击 Power 条上的任意按钮，得到一弹出菜单。
2. 单击弹出菜单底部的 ShowText，这时工具条上的所有按钮都扩展为带有文本的说明。

看起来挺不错的，但这样一来按钮都变大了，以致于右边的图标被推到了屏幕之外，除非关闭 ShowText 使之还原。好在还有另一个简单的方法：

1. 将光标指向一个按钮，停留一两秒钟，一个描述该按钮的小文本框将会弹出。
2. 按下鼠标左键不放，窗口底部的状态条将会出现对该按钮更详细的说明。

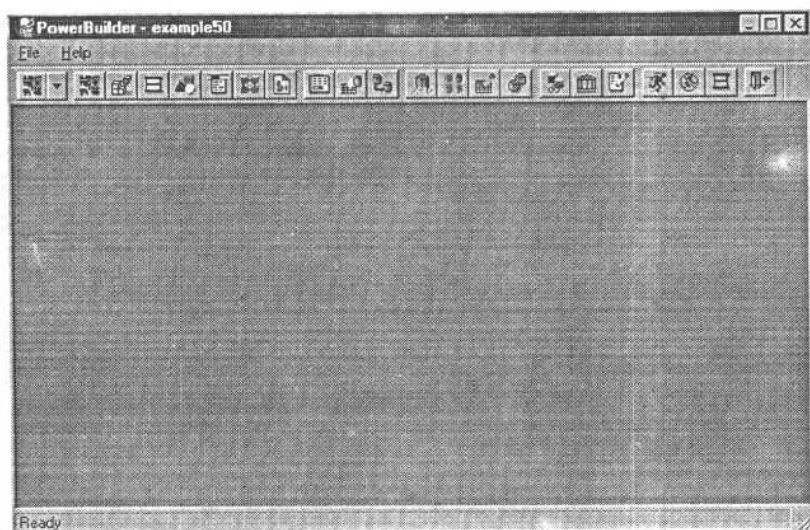


图 2.1 PowerBuilder 主窗口

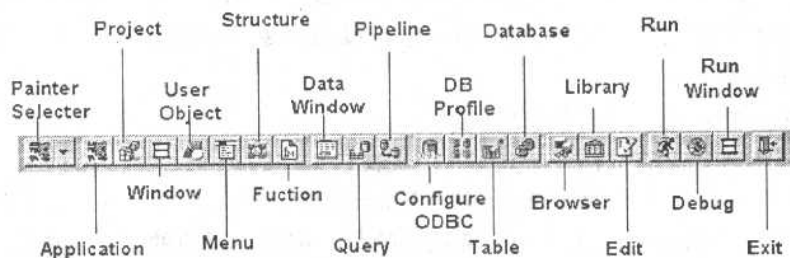


图 2.2 Power 条

2.4 库 Painter

单击 Power 条的 Library 按钮，可以看到屏幕类似于图 2.3。