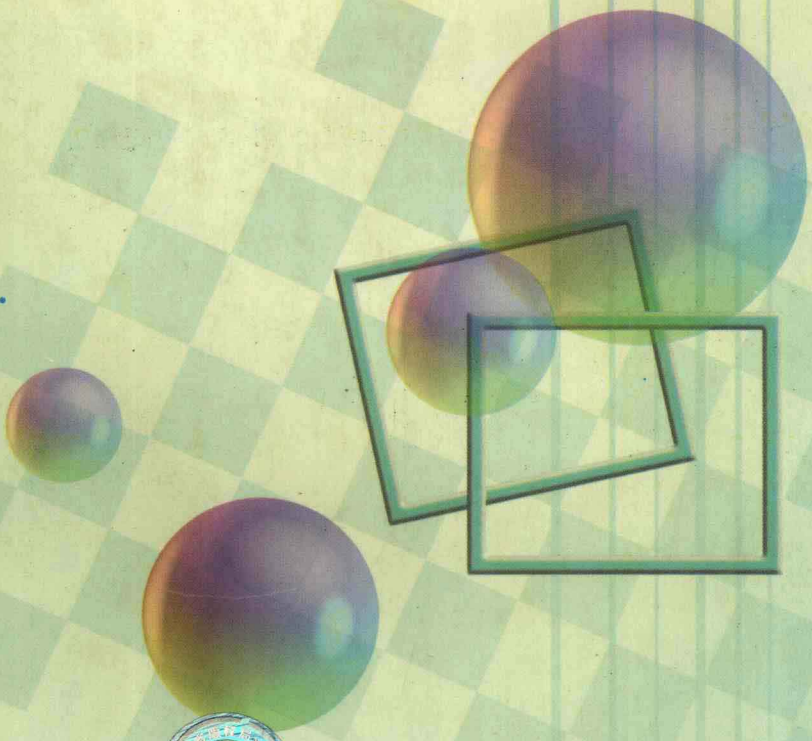
PowerBuilder 5.0

使用指南

主 编 罗 松
副主编 宣 力 赵夏阳 屈 进



电子科技大学出版社
UESTC PUBLISHING HOUSE

PowerBuilder 5.0 使用指南

主　编　罗　松

副主编　宣　力　赵夏阳　屈　进

电子科技大学出版社

内 容 提 要

本书是一本介绍 PowerBuilder 使用方法的快速指南，用于指导数据库应用系统开发人员迅速地熟悉 PowerBuilder，并且使用这个优秀的工具开发出实用的应用程序。

本书内容包括 PowerBuilder 概述、数据库设计、连接数据库、窗口的设计和优化、数据窗口设计、图表设计、程序调试以及一些编程的技巧，并且通过一个实用的股票分析系统说明了以上各个部分的实际操作。在附录中，收录了大部分常用的函数，并列出了其中文说明，以便用于快速查阅。

本书结构合理，语言流畅，附有大量的图解，相信对有志于学习数据库应用系统开发的朋友有很大的帮助。

PowerBuilder 5.0 **使用指南**

主　编　罗　松

副主编　宣　力　赵夏阳　屈　进

出　　版：电子科技大学出版社　（成都建设北路二段四号，邮编：610054）

责任编辑：谢应成

发　　行：新华书店经销

印　　刷：四川峨影印刷厂

开　　本：787×1092　1/16　　**印张**　16.75　　**字数**　380 千字

版　　次：1998 年 8 月第一版

印　　次：1998 年 8 月第一次印刷

书　　号：ISBN 7-81043-955-3/TP・424

印　　数：1—4000 册

定　　价：20.00 元

前　言

随着计算机在国内的迅速普及，各行各业的信息化改造也在如火如荼地进行，其中绝大部分是数据库应用。可以说，在未来的几年内，数据库应用程序的开发将和目前非常热门的Internet并驾齐驱，成为信息产业重要的增长点。

PowerBuilder是一个基于客户机/服务器(Client/Server)，具有优良用户界面，采用面向对象技术的数据库前端开发工具。它由PowerSoft公司经过七年的耕耘，到目前获得了极佳的评价，其用户数量在美国的前端开发工具中占第一位。在国内，PowerBuilder也因其卓越的性能而受到欢迎。国内诸多BBS上讨论数据库开发的专题中，有关PowerBuilder的话题总是占了大部分。由此可见一斑。

笔者认为，PowerBuilder的强大，在于它的跨平台能力、友好的用户界面、面向对象技术、以及很有特色的数据窗口。PowerBuilder支持多达近二十种数据库平台，囊括所有主流的数据库系统。在这些数据库系统之间切换，只需简单地改变数据库接口文件的配置(这个工作可以在操作系统下做，也可以在程序动态地进行)，并且基本上不需要修改程序代码。PowerBuilder具有友好的用户界面，即便是习惯在字符模式下工作的程序员也能轻易地转换到图形界面中。它支持面向对象的开发，程序员可以使用“继承”来工作，从而节省时间和金钱。PowerBuilder的PowerScript语言简单易学，即使不懂C++，不会SDK也没有关系，照样能够开发出功能强大的数据库应用系统。数据窗口(DataWindow)技术独树一帜，开业界先河，集数据处理、图形、表格于一体，并且内置数据库连接工具，使得开发人员可以更加灵活地连接、使用和表达数据。另外，PowerBuilder 5.0新增了WEB功能，可以将数据窗口的内容直接转化成WEB网页，从而令PowerBuilder具有了Internet发布的功能。

笔者专业从事数据库应用系统的开发工作，使用过各种主流的开发工具，经过比较，认为PowerBuilder确实是一个在各方面都堪称优秀的数据库前端开发工具，有必要将它介绍给有志于进行数据库应用系统开发的朋友。本书以一个简单的股票分析系统为例，介绍了如何迅速地使用PowerBuilder开发数据库应用程序，包括数据库设计、连接数据库、窗口的设计和优化、数据窗口设计、程序调试以及一些编程的技巧。在附录中，给出了常用函数的中文说明

和用法。本书包括了进行PowerBuilder 程序设计的所有步骤，相信可以使读者快速掌握开发一个小型系统的方法。

宣力、赵夏阳、陈锡明、丁香荣、姜海鹏、张圣辉和屈进参加了本书的部分编写工作；罗姝小姐负责了附录中函数的大部分编写工作；徐强、罗忠海对本书例子程序进行了测试。这里一并致谢。

在本书付梓之时，特别要感谢电子科技大学804 教研室的邱会中老师，他在数据库应用系统的开发中给予了宝贵的经验；更要感谢我的女友和家人在精神上的鼓励和支持，让我能心无旁骛地完成这本书。

本书撰写虽力求结构完整，内容详尽，但是因为笔者水平有限，其中不免有疏漏和失误，敬请各方面人士不吝赐教。

编 者

目　录

第一章 PowerBuilder 初步

1.1 PowerBuilder 简介

PowerBuilder 是一个基于客户机/服务器的图形界面的应用程序开发环境，在这个环境中，你可以构造面向对象的具有图形用户界面（GUI）的数据库应用。GUI 应用具有共同的外观和使用方法。比如，GUI 应用都有标准的窗口外观、菜单，以及对话框。这些标准特性提供的一致性使得 GUI 应用很容易被最终用户学习和使用。PowerBuilder 能使你轻松地制作出 GUI 应用。

客户机/服务器的计算方式把用户界面和数据库访问分开操作。PowerBuilder 允许你能制作可与各种各样的服务器和桌面数据库通信的应用程序。假如你的数据库支持的话，你还能够使用一些高级的技术，例如存储过程和远程过程调用等。

面向对象的程序设计是基于以下三条主要的原则：

（1）继承性 （2）封装性 （3）多态性

程序设计人员可以用 PowerBuilder 的窗口，菜单，以及用户对象来定义具有封装好了的属性、事件和函数的祖先对象，并通过继承来创建子对象。面向对象技术使你的应用更加模块化，更具有可重用性和扩展性，更灵活，更强壮。

PowerBuilder 提供了强有力的开发工具来进行客户机/服务器方式的数据库计算。PowerBuilder 的独特之处在于它的数据窗口（DataWindow）。使用数据窗口能非常方便地从数据库中提取和显示数据。数据窗口的卓越之处在于：

（1）构造一个数据窗口只需要很少或根本不需要 SQL 知识；

（2）使用数据窗口可以有效地减少将数据表达在表格和报表中的系统资源；

（3）数据窗口还能提供扩展的报表特性，包括计算域、图形和嵌套报表。

1.2 事件驱动程序的设计方法

由于是在 GUI 环境中编程，事件驱动的编程环境与传统的线性编程环境完全不同。在线性编程方法中，用户经过一个线性路径运行程序，并且只能完成很少的功能。这时程序将决定在任意给定的时间内，哪些选项对用户是有效的。而在 GUI 环境，向用户开放的功能选项数目很多，从而给用户和开发人员留有更多的自由度。然而，由于用户选项组太大，这种应用程序不再像传统的、基于字符的应用程序那样容易被用户控制。显然，人们可以将 GUI 程序设计成与任何基于字符的应用程序一样简单而紧凑，但这样将违背 GUI 的最基本目的——在应用程序上授予用户高度的权利。所以，GUI 应用程序不应给用户施加过多的约束，而应该提供一个宽松的环境，使用户有足够的自由去控制程

序的运行。这样，应用程序才能更好地成为用户拥有的工具，而不是相反：用户被应用程序所控制。

实践表明，当用户能更多地控制应用程序时，他们就能更有效地使用应用程序。在一个设计良好的应用程序中，用户常常会找到连其设计及开发人员都始料未及的操作方法，这会使开发人员非常高兴，同时也是一个设计精良和实用的应用程序的标志。

当开发一个事件驱动的应用程序时，要力求实现几个重要的目标。实现这些目标要付出较大的努力，以使用户完全掌握这个应用程序：

（1）应用程序中不应包括对用户无用的控制；

（2）用户不能违背从事的行业规则；

（3）应用程序应该响应用户要求，而不是强迫用户进入限定的路径；

（4）用户可用的每个操作都必须是“安全”的。

在开发事件驱动的应用程序时，开发人员必须编写能够响应大量用户操作的程序。显然，不是所有可能的事件可以或应该编写相应的程序代码。最好的办法是认真地挑选出用户（或系统）可能触发的某些事件，然后专门对这些事件编写程序。请记住，事件可以由任何动作触发。事件本身不应关心是什么触发了它，它只需要知道需要作出响应的相关对象的当前状态。（对象状态是指存放在对象的变量或属性参数的当前值）当以这种方式编码时，事件本身就成为实现上述第三个目标的强大而又灵活的机制。

当一个或多个属性参数发生改变时，对象状态就改变了（术语“变量”和“属性参数”可以互换使用）。PowerBuilder 图形对象的属性参数包括“Visible”及“Enable”，这描述了对象状态的一部分。开发人员经常选择某些变量来表示一个对象的常用状态。PowerBuilder 使用布尔窗口变量 bi_UpdateMode 来表示窗口正处于更新过程。改变变量也就改变了对象状态。变量不能改变其它变量的值，因为变量只存储信息。通过参数或事件 script 程序可以改变变量的值。

PowerBuilder 中的所有处理过程都是通过事件或函数参数中的 script 程序来执行的。每个对象均可包括事件、函数及属性参数。事件可以由任何对象的事件或函数参数中的任意 script 程序触发。

一个函数可以包括任意数目的函数参数和一个针对任何有效的 PowerBuilder 数据类型或类的返回值。事件所能传递的参数数目和类型是有限制的，同时 TriggerEvent 函数的返回值也受到了限制。当在某个 PowerBuilder Painter 环境中建立一个用户定义函数时，所说明的参数是调用函数时的必选参数，PowerBuilder 所定义的函数利用了“重载函数”的优点，这种技术允许多个函数具有相同的名称。重载函数之间的差别在于它们所需要的参数集不同。例如，MessageBox（）函数可以在不同的变元组中被调用：

```
MessageBox (" Title"," Message")
MessageBox (" Title"," Message", Information!)
MessageBox (" Title"," Message", Information!, OKCancel!)
MessageBox (" Title"," Message", Information!, OKCancel!, 2)
```

不幸的是，用户自定义函数（在 PowerBuiler 函数画板中说明的函数）不允许重载。当调用函数时所有已定义的参数都必须有给定值。不管怎么说，PowerBuilder 所支持的

参数数目和类型非常广，从而提供了一个功能强大的开发环境。

函数参数可以直接对应一个值或一个参数指针。当通过参数指针给参数传递值时，函数就访问初始参数并且可以直接改变这个初始参数。这实际上是给参数值赋予了一个引用指针。当直接将数值赋给参数时，实际上是将一个临时存放在函数局部变量的值赋给函数。函数对函数内的参数值所做的任何变化都不影响在调用 script 程序时的值。函数也能返回一个值，这个值可以是任何有效的 PowerBuilder 数据类型或类。不返回值的函数常被称为子程序。

事件由一个窗口事件或其它 script 程序中的事件函数触发（如 TriggerEvent ()）。事件可以直接传递一个限定数目的值—— 一个或两个以上数字或一个数字加一个字符串。这些值实际上是通过系统消息对象来传递的。

事件不像函数那样返回数值。为了将一个参数传递给某个事件，必须依靠变量（全局变量或实例变量）。PowerBuilder 消息对象可用于当事件触发时将数值传递给事件，但这种方法必须被一致的编码才能正确运行。每当调用一个倚赖于消息对象值的事件时，需要的值都应该在事件被触发前载入消息对象。在触发事件的 script 代码的开始，任何通过消息对象传递给事件的值都应该立即载入局部变量。请记住，存放在消息对象的值可通过任意的事件和函数改变。事件中包含的值只有在开始时是可靠的，在处理整个事件的过程中消息变量很可能会发生变化。

例如，鼠标单击一个命令按钮可激活一个 Clicked 事件。如果这个命令按钮存在相应的 script 程序，则这段程序可能实施以下 3 个动作之一：

（1）触发另一个事件；

（2）调用一个函数以完成某些处理；

（3）执行最少的处理。

一个事件驱动的应用程序提供给用户一定数量的对象，用这些对象可以与计算机交互。用户动作被对象以事件的形式捕捉（用户动作一般都显示在屏幕上）。这些事件的触发将使应用程序知道发生了某些事件并需要作出响应。这些事件中有许多都在 PowerBuilder 对象中存在相应 script 程序。例如，双击基本 MS/Window 对象左上角的控制菜单可以关闭窗口。用于关闭窗口的代码从 PowerBuilder 对象继承而来。PowerBuilder 窗口关闭过程还激发了窗口事件 Close Query 和 Close，这些事件的 script 代码可以在窗口 Painter 中编写。

事件驱动环境下的大部分 script 程序都是用于在事件发生时进行响应的程序。开发人员需要从以下两方面考虑：可能发生什么？动作的结果应该是什么？不是每个动作都有意义，故不必为每个动作编写程序。窗口内的控件表示当用户操作这些控件时可能会发生某些动作。一个不具备任何实际功能的命令按钮没有必要出现在窗口，它只会给用户造成混乱，而且占用空间。

开发人员还应该考虑根据对象状态允许哪些处理功能。例如，当把一条新记录加入到一个单记录 Freeform 格式的数据输入窗口时，无疑应设置一个有效的“删除”按钮，而当没有记录可删除时，“删除”按钮应该无效或非可视。另一方面，在数据窗口 PickList 的环境下，如果没有选择任何记录，则窗口中所有控制 PickList 窗口内的数据记录的命

令按钮（如“打开”按钮）应该无效。对所有这些情况应包括某些一贯的想法，并对这些想法进行测试和润色，最后结果应该是一个面向用户的环境。在这个环境下，不可用的事件不会被激发。这对于开发人员来说简化了编码工作，对用户来说建立了一个更合乎逻辑的应用程序。

现在假设有一个包括一个“打开”按钮的 PickList 窗口，这个“打开”按钮的用途是为当前记录打开维护窗口。由于编码不完善，即使窗口没有检索任何记录，“打开”按钮也仍然保持有效。这时用户可能会感到困惑：为什么屏幕上不能打开任何对象，而“打开”按钮却一直有效？如果用户决定单击这个按钮以了解将出现什么现象，这时的结果可能是一个应用程序错误，具体决定于按钮的 script 程序的内容。如果 script 程序盲目地认为存在一个记录，那么将出现一个严重的应用程序错。从一定意义上说，应用程序应该能够避开致命错误。从另一个角度来说，事件没有必要测试每个可能出现的问题。

一个良好的设计通过在必要时只允许执行某些 script 程序来减少许多代码，而一个劣质的设计需要很多代码行，以防止出现意外情况。劣质的设计中，许多意外情况并不能马上被发现，而且它还需要大量代码，这导致劣质设计需要较长的开发时间和更多的重复工作。这并不是说不应该检测错误情况，而是正好相反。然而，劣质设计需要额外增加条件代码行去检查各种意外事件，而良好的设计则不需要。

1.3 PowerBuilder 开发环境简介

在这一节中，我们将对 PowerBuilder 的开发环境作一个大致的了解，将介绍 PowerBuilder 提供的各种开发工具。内容包括：

(1) PowerBuilder 基础；
(2) 怎样使用 PowerBar；
(3) 什么是画板；
(4) 怎样使用工具栏；
(5) 怎样使用弹出菜单；
(6) PowerBuilder 窗口；
(7) 怎样使用文件编辑器；
(8) 怎样使用联机帮助。

1.3.1 PowerBuilder 基础

PowerBuilder 是一个基于 PC 机客户机/服务器结构的图形界面应用程序开发环境。

利用 PowerBuilder 来开发涉及访问服务器上的数据库的 Windows 应用程序非常方便。

PowerBuilder 是真正意义上的 Windows 应用程序，它由包含各种控件的窗口组成，用控件跟应用程序交互作用。这里的控件包括标准的 Windows 控件，如按钮、复选框、下拉列表框和编辑框等，同时还包括 PowerBuilder 所特有的各种控件。

PowerBuilder 还提供了各种便于应用程序开发的工具。

画板

在开发 PowerBuilder 应用程序的过程中，需要用到各种画板（Painter）。画板其实就是建造某一类型的 PowerBuilder 对象时用到的工具箱。

例如，为创建窗口，可以打开窗口画板，从这里可以定义窗口的各种属性和加入各种控件，如按钮和编辑框。

事件和脚本

PowerBuilder 应用程序是事件驱动的，也就是说，用户可以通过各种动作来控制应用序的流程。例如，当用户单击某个按钮、从菜单里选中某个菜单项或在某个编辑框里键入数据时，都会触发相应的事件。表 1- 1 列出了 PowerBuilder 的各种对象所拥有的事件。

可以通过编写脚本（script）来指定响应这些事件的处理过程。例如，可以为某个按钮上的 Clicked 事件编写脚本，以指定用户单击该按钮时该如何处理。同样地，也可以为某个编辑框的 Modified 事件编写脚本，当用户修改编辑框中的内容时，这段脚本将被执行。

脚本实际上就是程序。像编写其它的程序一样，编写脚本也要用到语言，PowerBuilder 所使用的语言称为 PowerScript 。脚本由 PowerScript 所提供的各种命令、函数和语句构成，它们构成了响应某个事件的处理过程。

例如，为某个按钮上的 Clicked 事件编写的脚本，可能要执行的处理是从数据库获取并显示数据。而为某个编辑框上的 Modifide 事件所编写的脚本，可能要对修改后的数据进行某种处理。

脚本本身也可以触发事件。例如，在脚本里打开另一个窗口时，就要触发该窗口上的 Open 事件。如表 1-1。

表 1-1　PowerBuilder 各种对象的脚本可响应的事件

对　象	事　件
窗　口	dranenter，dragdrop，dragleave，dragwithin，clicked，constructor，dberror，destructor，doubleclicked，editchanged，getfocus，itemchanged，itemerror，itemfocuschanged，losefocus，other，printend，printpage，printstart，rbuttondown，resize，retrieveend ，retrieverow，retrievestart，rowfocuschanged，scrollhorizontal，scrollvertical，sqlpreview，updateend，updatestart
菜　单	clicked，selected
用户对象	constructor，destructor，dragdrop，dragenter，dragleave，dragwith，other，rbuttondown，clicked，timer

续 表

对 象	事 件
数据窗口	clicked, constructor, dberror, destructor, doubleclicked, dragdrop, dragenter, dragleave, dragwithin, editchanged, getfocus, itemchanged, itemerror, itemfocuschanged, losefocus, other, printend, printpage, printstart, rbuttondown, resize, retrieveend , retrieverow, retrievestart, rowfocuschanged, scrollhorizontal, scrollvertical, sqlpreview, updateend, updatestart
命令按钮、图像按钮、复选框和单选按钮	clicked, constructor, destructor, dragdrop, dragenter, dragleave, dragwithin, getfocus, losefocus, other, rbuttondown
单行编辑框、编辑屏蔽和多行编辑框	constructor, destructor, dragdrop, dragenter, dragleave, dragwithin, getfocus, losefocus, other, rbuttondown
列表框、图像和图表	constructor, destructor, doubleclicked, dragdrop, dragenter, dragleave, dragwithin, getfocus, losefocus, other, rbuttondown, selectionchanged
下拉列表框	constructor, destructor, doubleclicked, dragdrop, dragenter, dragleave, dragwithin, getfocus, losefocus, modified, other, rbuttondown, selectionchanged
滚动条	constructor, destructor, dragdrop, dragenter, dragleave, dragwithin, getfocus, losefocus, lineleft, lineright, moved, other, pageleft, pageright, rbuttondown

函数

PowerScript 提供了丰富的内部函数，可以利用这些函数来对应用程序的各种对象和控件进行处理。如打开窗口的函数、关闭窗口的函数、使按钮有效的函数、获取数据的函数和更新数据库的函数，等等。

除此之外，我们还可以定义自己的函数来执行某种特殊的处理过程。

库

PowerBuilder 应用程序里的各种对象，如窗口和菜单，都是保存在 PowerBuilder 库（即 .PBL 文件）里的。当应用程序运行时，PowerBuilder 从库里取得这些对象。

PowerBuilder 提供了一个库画板，可以借助该画板来管理库里的内容。

用户对象

用户对象是一种用户自己创建的、用来执行应用程序需要频繁使用的处理过程的对象。它们就像其它的 PowerBuilder 控件一样，可以在窗口中使用它们，也可以在另外的用户对象中使用它们。

用户对象通常执行公用处理过程。当使用用户对象时，应该为这个对象编写一些脚本，以使它能专门为应用服务。例如，假使有一个用户对象的功能是显示一个文件列表，而在一个目录控制应用程序的窗口中使用它，那么就应该为这个用户对象编写脚本，限制用户对象只显示目录文件。

结构

一个结构是一个或多个相互联系的值的集合。这些值具有相同或不同的数据类型，但它们都是在同一个名字下被集合起来。结构可以使你把相关的多个对象当成一个单元来使用，而不是分别单独来使用它们。例如，可以将用户的 ID 号、地址、访问级别、图像等信息定义成一个名叫 user_struct 的结构，接着就可以使用 user_struct 来引用上述值的集合。

创建可执行文件

在开发完 PowerBuilder 应用程序之后，可以创建该应用程序的可执行文件，也就是能脱离 PowerBuilder 环境而运行的文件。

PowerBuilder 提供了用来包装应用程序的简单的方法，包装后的应用程序就可以分发了。

1.3.2　怎样使用 PowerBar

PowerBar 是 PowerBuilder 开发环境里的主工具栏。从 PowerBar 上可以进行以下活动：打开各种 PowerBuilder 画板，跟踪调试或运行当前应用程序，请求帮助，以及根据需要定制 PowerBuilder。

PowerBar 本身也可以被定制。例如，可以将经常使用的图标加入该工具栏。

PowerBar 上的内容还能以 Power 画板的形式出现，可以在这两种形式之间来回切换。两者之间的区别是：PowerBar 是工具栏，它上面的内容可以重新定制，并且该工具栏可以放到窗口的任何位置；而 Power 面板是窗口，它里面显示的是一组固定的图标。

要想切换到 Power 画板，只需从 Power 菜单里选中 PowerPanel，于是 Power 画板出现，如图 1-1，图 1-2 所示。

要想回到 PowerBar，只需从 File 菜单中选中 PowerBar。

PowerBar 上的各种图标分别代表了 PowerBuilder 里经常用到的画板和工具。表 1-2 列出了这些常用的图标和它们的意义。

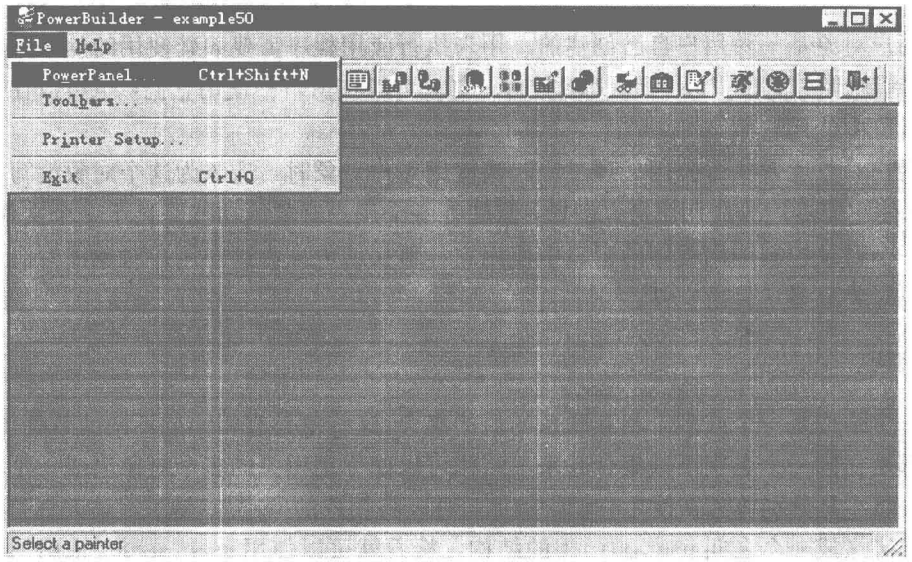


图 1-1

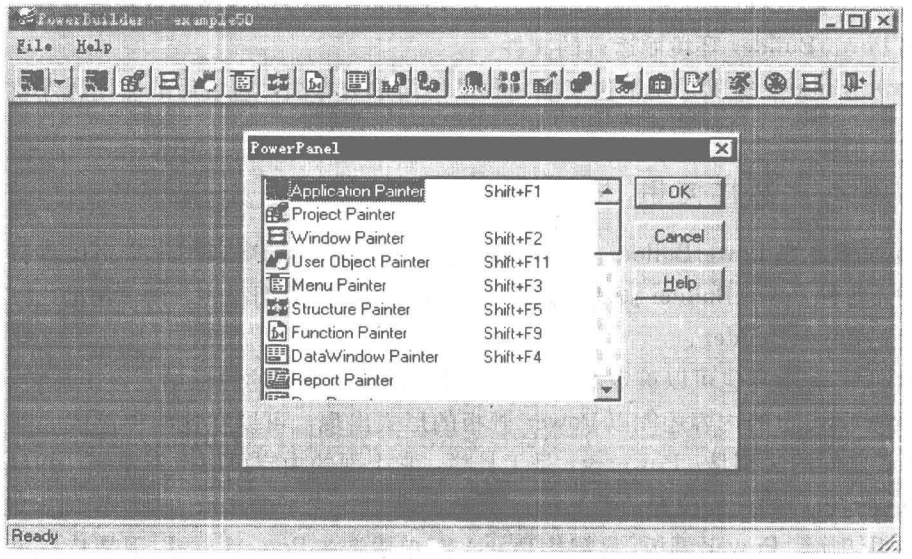


图 1-2