

计算机新技术应用速成培训

*PowerBuilder*应用开发 速成培训

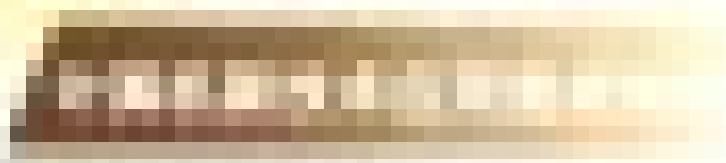
余三明 张如健 陈 旭 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.com.cn>



PowerBuilder 应用开发 速成培训

PowerBuilder 应用开发速成培训



PowerBuilder 应用开发速成培训

计算机新技术应用速成培训教材

PowerBuilder 应用开发速成培训

余三明 张如健 陈 旭 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书系统地介绍了目前数据库开发工具中最流行的可视化客户机/服务器应用程序开发工具——PowerBuilder 6.0/6.5。内容包括面向对象概念介绍、PowerBuilder 6.0 的应用对象、窗口、控件、菜单、PowerScript 语言、数据库的链接、数据管理、应用程序发布及常用事件脚本示例等。全书内容由浅入深，既有基本操作说明，也有开发技巧探讨，同时注重文字通俗易懂、操作步骤明确、理论和实践相结合。书中精心设计了实例程序，以便于读者理解相应章节内容。

可作为 PowerBuilder 应用程序开发人员的自学参考书，也可作为大专院校面向对象程序设计、可视化数据库开发工具教材和参考书，也非常适合作为培训班的培训教材。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，翻版必究。

图书在版编目 (CIP) 数据

PowerBuilder 应用开发速成培训/余三明等编著. —北京：电子工业出版社，1999.10

(计算机新技术应用速成培训教材)

ISBN 7-5053-5536-8

I.P… II.余… III.数据管理系统-软件工具, PowerBuilder-技术培训-教材 IV.TP311.56

中国版本图书馆 CIP 数据核字(1999)第 46025 号

丛 书 名：计算机新技术应用速成培训教材

书 名：**PowerBuilder 应用开发速成培训**

编 著 者：余三明 张如健 陈 旭

责任编辑：张 毅

特约编辑：秋 阳

印 刷 者：北京东光印刷厂

装 订 者：三河市新伟装订厂

出版发行：电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编:100036

经 销：各地新华书店

开 本：787×1092 1/16 印张:11.25 字数:288 千字

版 次：1999 年 10 月第 1 版 1999 年 10 月第 1 次印刷

书 号：ISBN 7-5053-5536-8
TP · 2798

印 数：5000 册 定 价：15.00 元

凡购买电子工业出版社的图书，如有缺页、倒页、脱页、所附磁盘或光盘有问题者，请向购买书店调换。若书店售缺，请与本社发行部联系调换。电话：68279077

出版说明

计算机的迅速发展,不仅使计算机普及到了各行各业,而且应用到了各个办公室,深入到了家庭住户。不久的将来,电脑将如家用电视一样得到普及。人们对电脑的依赖将与日俱增。学习电脑、掌握计算机应用新技术不仅已经成为人们的普遍愿望,而且也是社会发展的客观需求。

人人都要学电脑,人人需要电脑新技术。

但是,相当多的电脑书籍对于大多数非计算机专业的人员来说都显得太难、太深、太厚、太贵了。非计算机专业人员首先需要的不是令人望而却步的“大而全”,而是能够帮助他们化难为易,尽快跨入计算机应用的复杂天地的入门书。

目前,大中专学校各专业包括计算机专业的教学计划与计算机技术的发展速度之间不可避免地存在一定的时间差。为了使学生初步掌握各种软、硬件新技术,各校开设了各种形式的选修课和第二课堂,拓宽学生的知识面。但是大多数计算机书籍对于选修课和第二课堂同样是太难、太深、太厚、太贵了。《计算机新技术应用速成培训教材》正是瞄准了这方面的需求,希望能为学生拓宽知识面、开设选修课、开展第二课堂教育提供合适的教材。

由电子工业出版社出版的《计算机新技术应用速成培训教材》介绍的是计算机应用的最新技术。目的是使众多的最新软硬件技术尽快为各行各业的应用人员所掌握。这套丛书不是大全、不是命令索引,也不是简单的阐述,而是作者对新技术消化整理后的经验总结。篇幅虽然精练,但仍保证了系统性和完整性,体现了以实用为目标的编写原则,旨在将读者引进某项应用技术的大门,初步掌握这项新技术(软件或硬件)的使用方法,为进一步深入学习铺平道路。

《计算机新技术应用速成培训教材》也是为各个专业包括计算机应用专业的广大在校学生编写的。我们希望《速成培训教材》能成为在校学生步入社会及时与工作中应用需要接轨,加速对工作适应的有力工具。

电子工业出版社

前 言

笔者一直从事数据库及其开发工具的教学、研究和应用开发工作。我们发现，一系列的数据库开发环境，Foxpro（for DOS&Windows）、VFP、VC++、VB 和 PowerBuilder 中，不管是系统开发效率，还是对数据库的操作能力，PowerBuilder 都是独树一帜的。

目前，市面上虽然有不少关于 PowerBuilder 的书籍，但一部分书籍只粗略介绍了 PowerBuilder 中的操作，不便于读者提高；另一部分书籍只是程序员参考手册，专门讨论对象的属性、事件和脚本，起点太高，没有基础的人员无法使用。笔者在本书中将上述两类书籍的优点有机结合起来，既可帮助非 PowerBuilder 的技术人员轻松入门，又可帮助有一定基础的 PowerBuilder 程序员进一步提高。

本书共分十六章，第一章是 PowerBuider 应用及其环境的基础知识；第二章到第十六章是本书的核心，详细阐述了 PowerBuider 应用开发的方法、步骤、技巧及要注意的问题；第十六章给出了一个简单的程序实例，供开发者参考。

本书第 1、9、10、11、12、14 章及附录由陈旭编写，第 3、4、6、7、16 章由余三明编写，第 2、5、13、15 章由张如健编写，第 8 章由余三明、陈旭共同编写。全书由陈旭统稿。

限于编者水平，又加之时间仓促，书中难免有许多疏漏之处，敬请广大读者不吝指正。

作 者

第 1 章 PowerBuilder 概论

1.1 PowerBuilder 简介

1.1.1 PowerBuilder 简介

随着数据库技术在各行各业的广泛应用，作为企业级数据库前端开发工具的 PowerBuilder，日益成为开发人员的得力助手。PowerBuilder 以其开放的体系结构，友好的用户界面和简洁高效的开发环境赢得了众多程序员的喜爱。

PowerBuilder 是一个面向对象、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，我们可以利用 PowerBuilder 提供的可视化的工具来构造面向对象的、具有图形用户界面（GUI）的数据库应用。

同时 PowerBuilder 还具有 Windows 风格，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。这些控件包括按钮、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。利用 PowerBuilder 这个优秀的数据库前端开发工具来开发访问服务器上的数据库应用程序非常方便。

1.1.2 PowerBuilder 的基本特点

- PowerBuilder 应用由菜单和窗口等用户界面构成。
- PowerBuilder 应用采用事件驱动。用户可以通过在事件中编写脚本来控制应用程序的运行。
- PowerBuilder 面向对象编程。PowerBuilder 的基本组成元素是对象，每个用户菜单或窗口就是一个对象。因此，在 PowerBuilder 中，使用封装、继承和多态等面向对象的编程技术，可以使应用具有可重用性和可扩展性。PowerBuilder 应用程序里的各种对象，如窗口和菜单，都是保存在 PowerBuilder.PBL 文件里的。当应用程序运行时，PowerBuilder 从库里取得这些对象。
- PowerBuilder 支持跨平台的开发。用户所开发的 PowerBuilder 应用能够移植到不同的平台上。
- PowerBuilder 提供了多种专用数据库的接口，同时还支持 ODBC，使得它能够很方便地连接各种不同的数据库。
- PowerBuilder 提供了强有力的开发工具进行客户机/服务器方式的数据库计算。
- PowerBuilder 提供了数据窗口（Data Window）。使用数据窗口能非常方便地从数据库中提取和显示数据。
- PowerBuilder 应用 PowerScript 进行编程。PowerScript 语言又称脚本，它提供了丰富的内部函数，利用这些函数可以处理应用程序的各种对象和控件。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。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。

1.2 面向对象程序设计的基本概念

1.2.1 PowerBuilder 对象

PowerBuilder 对象有三个元素：属性、事件和函数。

- 属性。也就是数据。包括系统属性和用户定义属性两类，描述该对象的各种特性。如在窗口对象中的系统属性包括标题、高度、宽度等，用户定义的属性可以是实例变量或共享变量，对这种属性的访问同系统属性相同，只是在封装性上有所不同。
- 事件。PowerBuilder 中的事件与 Windows 的事件存在着映射关系。当用户的操作或系统本身产生了 Windows 的标准事件，就转化成 PowerBuilder 中的事件。例如命令按钮的 Clicked 事件，当用户按按钮时触发，操作系统首先检测到鼠标被点击，并把 Clicked 这一消息转给该按钮，PowerBuilder 运行时响应这一事件，并执行其中的 PowerScript 语句。PowerBuilder 中大部分事件都是可以映射成由用户激发的 Windows 事件。另一类事件是用户定义事件，这种事件是用户声明并可以在任何时刻由用户触发的，PowerBuilder 中保留了多个用户事件由程序员使用。
- 函数。用户通过函数来使对象完成某些操作。Powersoft 公司建议用户通过调用对象事件和函数来修改对象而不要直接修改对象属性，以满足对象封装的要求，例如采用 Window.hide()，而不要用“Window.visible=false”使窗口不可见。

1.2.2 面向对象程序设计的主要原则

1. 继承性

PowerBuilder 中，窗口、菜单和用户对象是可以继承的，而其他对象则都不能。当你继承了一个对象，那你得到的子类将具有父类的属性、实例变量、共享变量、控件、用户自定义事件、对象级函数、事件和代码（Script）。也就是说当继承了一个类，就几乎得到了这个类的全部。

在继承了祖先类后，可以在子类中扩展或覆盖祖先的元素。

2. 封装性

封装的目的是为了实现数据隐藏和数据保护，封装的目标是为对象提供一个对外操作的接口，使其他对象通过函数来访问，而不允许直接操纵对象的属性。在 PowerBuilder 中有三种访问类型 Public、Protect、Private，这三种访问控制类型可以用在对象的变量和函数上，缺省的实例变量和对象函数都是 Public 类型。为了保护数据，应尽可能多地使用 Private 和 Protect 类型，前者只允许对象内部的元素来访问，后者可以接受对象内部和继承类的元素访问。

3. 多态性

在 PowerBuilder 中有大量的多态函数如 print()、TriggerEvent() 等。在运行过程中，只需要指出对象和函数名即可。在有些函数中，即使不知道对象类，也可以用 Class Name() 函数得到对象类，或得到实例名，将对象名作为函数参数调用该函数。

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。面向对象技术使应用更加模块化，更具有可重用性和扩展性，更灵活。

1.3 事件驱动程序设计概念

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 环境中，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。这样，应用程序才能更好地成为用户拥有的工具。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。

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都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。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多态性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。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。

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三个动作之一：

- 触发另一个事件。
- 调用一个函数以完成某些处理。
- 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-Windows 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 CloseQuery 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。

1.4 PowerBuilder 的安装方法

PowerBuilder 的安装方法很简单，其安装步骤如下：

(1) 将安装光盘放入光驱中，运行 Setup.exe 程序。此时屏幕上出现如下窗口：

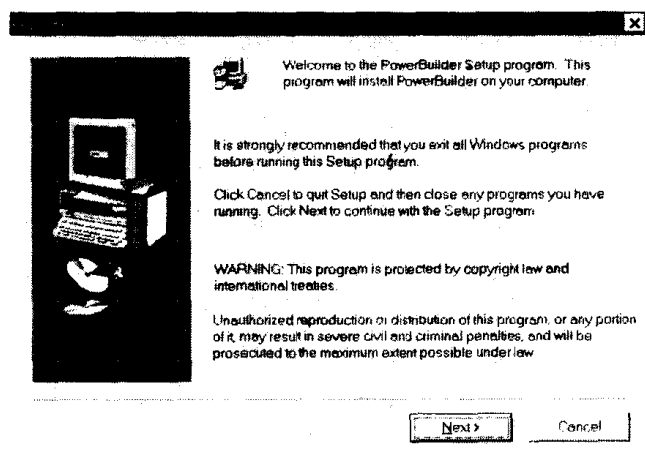


图 1-1

单击 Next 按键，进行下一步的安装。

(2) 输入你的姓名、公司名称，然后单击 Next 按键。

(3) 确认安装目录，系统默认的安装目录为 C:\Program File\Powersoft，如果需要改变安装目录，可单击 Browse 按键，然后在 Browse 对话框中输入新的安装目录。当然也可使用默认的安装目录而直接单击 Next 钮。

(4) 选择安装的部件，PowerBuilder 提供三种选择方案：

- Compact（压缩安装）：最小的一种安装方法，只安装最基本的部分。
- Custom（习惯安装）：用户定制安装方法，可根据需要选择安装。
- Typical（典型安装）：安装常见部分。

我们可根据需要选择其中之一，然后单击 Next 钮。如果选择了 Custom 安装方法，屏幕上将出现如图 1-2 所示窗口：

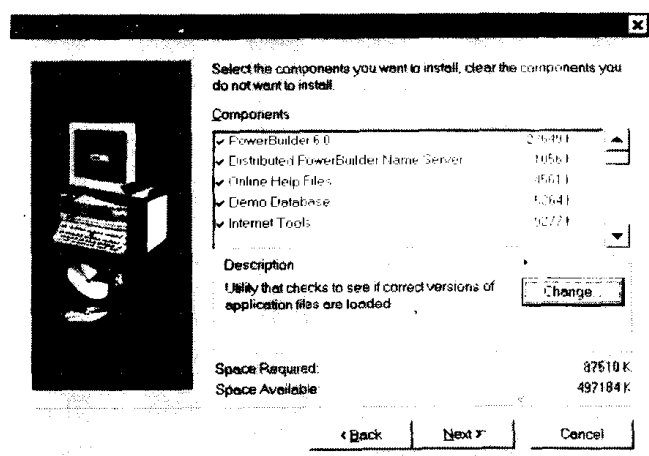


图 1-2

在这个窗口中，用户可根据具体情况对 PowerBuilder 的各部分进行取舍。单击 Change 按钮可修改被选的项目。

(5) 安装开始菜单。如图 1-3 所示。

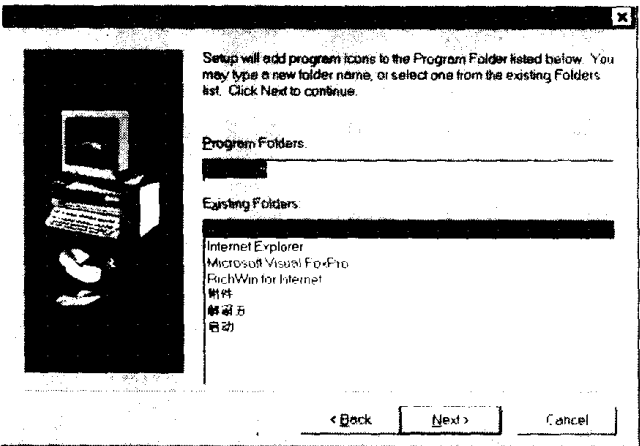


图 1-3

(6) 进行备份选择，保存以前的 PB.INI 及 TUTOR_IM.PBL 文件。
最后在安装完成窗口中单击 Finish 按钮，安装即告完成。

1.5 PowerBuilder 开发环境

1.5.1 PowerBuilder 的启动

启动 PowerBuilder 只须单击开始菜单中的 PowerBuilder 6.0 图标，PowerBuilder 6.0 的主窗口如图 1-4 所示。

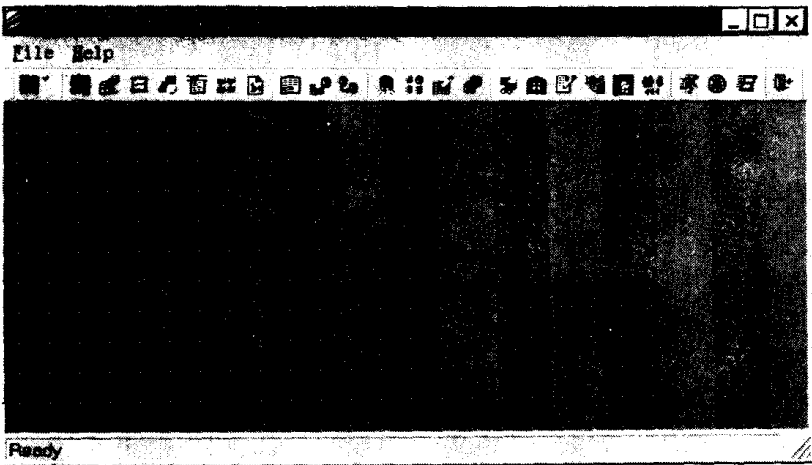


图 1-4

从图 1-4 中可以看到 PowerBuilder 的主窗口包含菜单条和 PowerBar。
退出 PowerBuilder 6.0，只需单击 PowerBuilder 主窗口中的关闭按钮即可。

1.5.2 PowerBuilder 工具栏

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 画笔、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。

要打开画笔或工具，可以单击 PowerBar 上代表该画笔或工具的图标，或者从 File 菜单里打开它们。除此之外，还可以使用快捷键打开相应的画笔或工具。如图 1-5 所示：



图 1-5

以下按顺序列出了这些常用的图标和它们的意义。

- 画笔列表 (Application)。
- 应用对象画笔 (Application)：用于创建应用对象及保存所有对象的应用库。
- 工程画笔 (Project)：用于建立工程文件、编译应用程序、生成可执行文件等。
- 窗口画笔 (Window)：可以利用它来创建应用程序里的各种窗口。
- 用户对象画笔 (User Object)：用于创建及管理各种可重用的用户自定义对象。
- 菜单画笔 (Menu)：用于创建菜单。
- 结构画笔 (Structure)：用于定义结构 (即变量组)。
- 函数画笔 (Function)：用于创建对应各种级别的函数。
- 数据窗口画笔 (DataWindow)：用于创建数据窗口对象。
- 查询画笔 (Query)：用于定义和保存可 SQL SELECT 语句。
- 数据管道画笔 (Pipeline)：可以用它在数据库之间、数据库内部传递表结构和数据。
- ODBC 定义画笔 (Configure ODBC)：通过 ODBC, PowerBuilder 可以连接各种支持 ODBC 的数据库。
- 数据库接口参数画笔 (DB Profile)：用于配置连接各数据库的参数。
- 表操作画笔 (Table)：创建及修改数据表。
- 数据库画笔 (Database)：用于维护数据库，控制用户对数据库的访问和操纵数据库里的数据。
- 对象浏览器 (Browser)：可以使用它来快速浏览各种对象。
- 库画笔 (Library)：用于创建和维护 PowerBuilder 库。
- 文本编辑器 (Edit)：用于编辑各类文本。
- Web.PB 向导 (Web.PB Wizard)：生成激活分布式对象的 HTML 页面。
- OLE 注册画笔 (OLE Generate Registration)：用于创建 OLE。
- 同步 (Synchronize)。
- 运行画笔 (Run)：运行当前的 PowerBuilder 应用程序。
- 调试画笔 (Debug)：跟踪调试当前的 PowerBuilder 应用程序。
- 窗口运行画笔 (Run Window)：运行当前窗口程序。
- 退出画笔 (Exit)：退出 PowerBuilder。

1.5.3 定制工具栏

1. 改变工具栏位置

下面介绍如何利用 Toolbars 窗口来控制工具栏的显示。

(1) 从 Window 菜单里选择 Toolbars, 于是出现 Toolbars 窗口, 如图 1-6 所示:

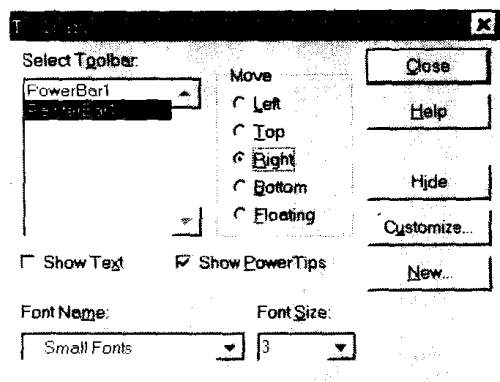


图 1-6

(2) 单击 Select Toolbar 框里的工具栏, 如 PainterBar1。

(3) 单击 Move 组里的某个按钮, 如 Right。

(4) 单击 Close 按钮。

这样工具栏 PainterBar1 的位置发生了改变, 原来在顶部, 现位于右边。

2. 将图标加入工具栏

要想将某个图标加入工具栏, 可以遵照下列步骤:

(1) 将光标移动到该工具栏上, 然后单击鼠标右键。

(2) 从弹出菜单里选中 Customize, 此时出现 Customize 窗口, 如图 1-7 所示:

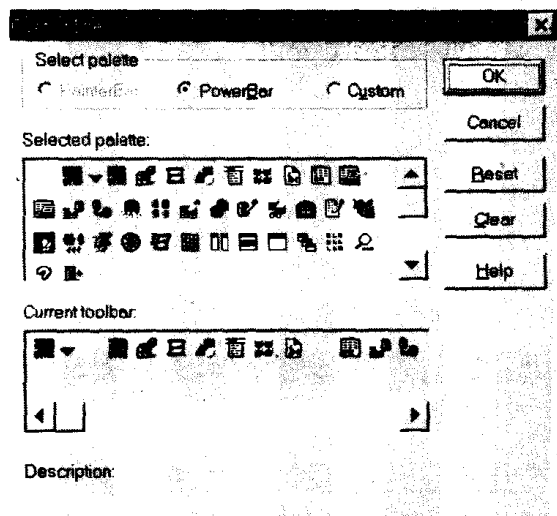


图 1-7

(3) 单击 Select Palette 组里的某个选项, 如 PowerBar 等。

(4) 从 Selected Palette 框里选择某个图标。

如果从 Select Palette 组里选择的是 Custom 项, 则会弹出另外一个对话框, 如图 1-8 所示。

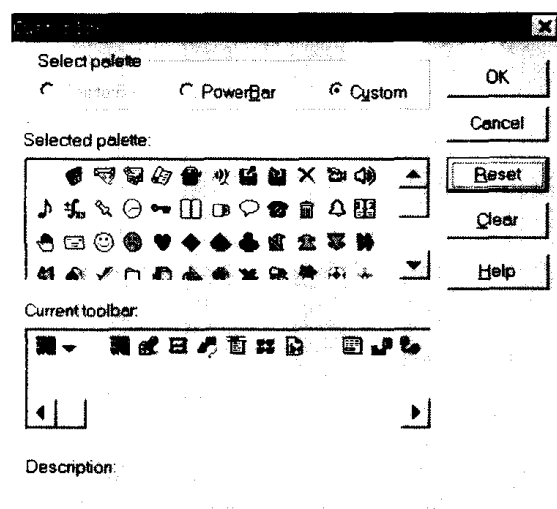


图 1-8

(5) 将选中的图标拖到 Current toolbar 框的合适位置上，然后按 OK。

3. 从工具栏上删除图标

要从工具栏上删除某个图标，可以遵照下列步骤：

- (1) 打开 Customize 窗口。
- (2) 将光标移到 Current toolbar 框里的某个想要删除的图标上。
- (3) 按住鼠标左键，将该图标拖到 Current toolbar 框之外的任何位置。
- (4) 单击 OK。

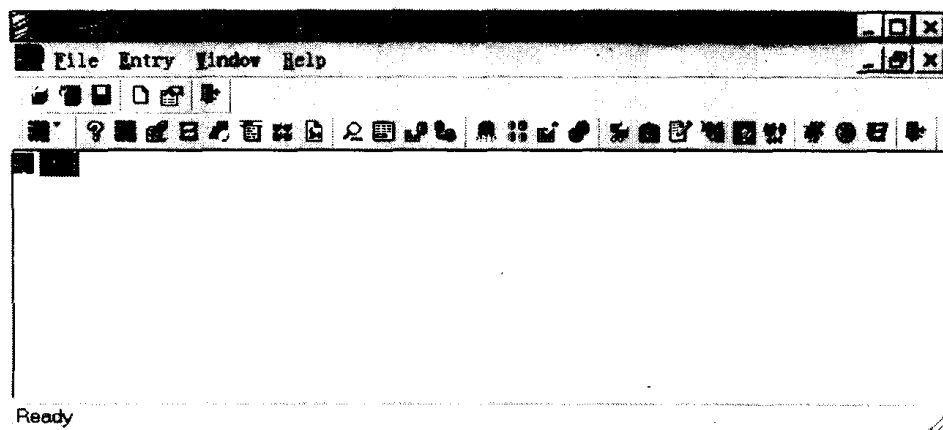
4. 恢复工具栏的原始设置

在任何时候，都可以将工具栏上的图标恢复成原始设置。步骤如下：

- (1) 将光标移到工具栏上，然后单击鼠标右键。
- (2) 从弹出菜单里选中 Customize。
- (3) 单击 Customize 窗口里的 Reset 按钮。
- (4) 单击 Yes。
- (5) 单击 OK。

练习题

启动 PowerBuilder6.0。定制如下图所示的工具栏。



第2章 创建应用对象

应用对象是 PowerBuilder 开发应用程序的一个入口点。每个 PowerBuilder 应用程序都必须有一个，而且只能有一个应用对象，程序从应用对象开始运行。除了作为应用程序的入口点这一功能外，应用对象还定义了应用程序的系统级行为。例如，应用程序使用哪些应用库（一种 PowerBuilder 定义的特殊格式的文件）来保存用到的各种对象，对象的缺省字体、应用程序的图标等。

PowerBuilder 是个面向对象的开发环境，用它开发的应用程序以对象和事件驱动为基本特征。和其他 PowerBuilder 的对象一样，应用对象也有一组事件，当用户运行应用程序时，首先触发应用对象的 Open 事件，在这个事件的事件处理程序中我们可以设置应用的初始状、装入其他对象、连接数据库等，完成必要的系统清理工作，例如断开与数据库的连接、清除用户自定义对象等。当程序运行过程中发生严重错误时，应用对象的 SystemError 事件被触发，这时可以报告错误类型和性质，以便及时排除错误。由此可见，应用对象对开发完美的 PowerBuilder 应用程序至关重要。

2.1 应用对象和应用库

2.1.1 应用对象

应用对象 (Application Object) 是 PowerBuilder 的众多对象之一，它是每个 PowerBuilder 应用程序都必须具备的对象。它标识应用程序，是应用程序的入口点。应用对象在应用对象画笔中创建和修改。下面介绍应用对象的用途，应用对象画笔的组成以及创建应用对象的方法。

应用对象是 PowerBuilder 应用程序启动时装入的第一个对象，除了标识应用程序和作为应用程序的入口点外，应用对象还保存和定义了下述信息：

- 缺省字体。应用对象是个非可视对象，它的四个字体属性用于定义应用程序中其他对象使用的缺省字体。把应用对象的字体属性设置成应用程序中使用最频繁的字体，将有助于应有程序开发过程（避免了为每个对象设置所需字体）。
- 应用程序图标。指定用哪个图标代表运行中的应用程序。
- 应用库搜索路径。该路径指明了在哪些应用库中能查找到应用程序中包含的各种对象。先查找前面的应用库。
- 缺省的全局变量。PowerBuilder 为每个应用程序都提供五个缺省的全局变量，分别为 SQLCA、SQLDA、SQLSA、Error、Message。在应用程序中既可以直接使用这些预定义的全局变量，也能够在应用对象画笔中更改这些全局变量的类类型，只要新的类类型从相应的原始类中继承得到即可（使用用户画笔创建新的类类型）。
- 应用对象结构、函数和实例变量。
- 其他全局变量。
- 全局外部函数。当我们想使用 PowerBuilder 本身没有提供的功能时，可以通过访问外部函数（比如 Windows 系统中的动态链接库）来解决。

2.1.2 应用库

用 PowerBuilder 开发应用程序时，我们建立的所有对象（包括对象函数、事件处理程序）函数、结构，定义的所有变量，最终都存储在磁盘文件中，该文件以 pbl 为后缀，我们称这些文件为应用库。一个应用程序既可以存放在一个应用库中，也可以分散存储在多个库中，但是，单个应用库不要太大，通常不宜超过 800 K，否则将增加系统的查找时间。库文件不能直接使用文本编辑器查看，但通过库管理画笔（Library Painter）可以清楚地浏览、操纵应用库，具体操作请参阅以后章节。

2.2 建立应用对象

建立 PowerBuilder 应用程序的首要任务就是建立新的应用对象，其操作步骤为：

- (1) 单击画笔栏的“应用画笔”图标进入应用画笔工作区。
- (2) 从“File”菜单中选择“New”菜单项，打开应用库选择对话框，如图 2-1 所示。

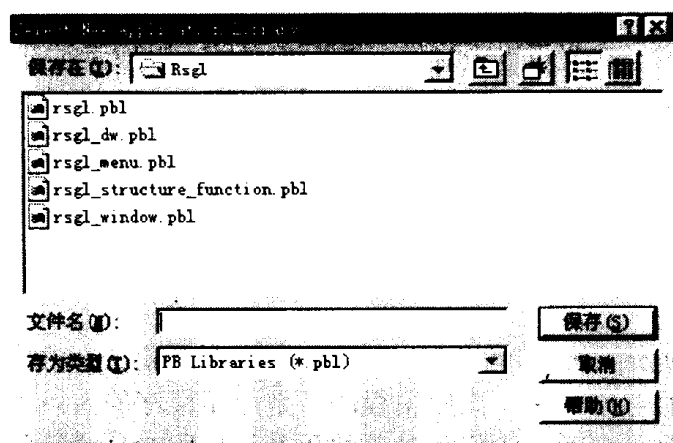


图 2-1 “Select New Application Library”对话框

- (3) 在“保存在”下拉列表框中选择应用库的保存路径，在“文件名”后面键入应用库的名称（如，Rsglk）单击“保存”按钮，打开如图 2-2 所示的“Save Application”（保存应用）对话框。

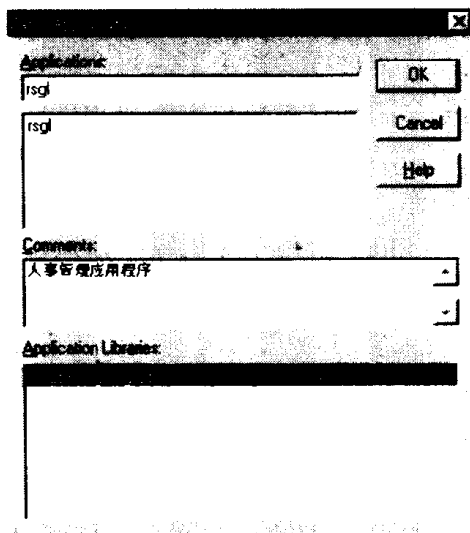


图 2-2 “Save Application”对话框

(4) 在“Application (应用)”编辑框中键入一个字符串 (如 Rsgl) 作为应用对象的名称。以后打开应用程序后对象名称将出现在 PowerBuilder 的标题栏。

(5) 在“Comments (注释)”编辑框中键入一些说明文字作为应用对象的注释, 该注释将在库管理画笔中显示, 可以帮助开发人员理解该应用对象。

(6) 单击“OK”按钮, 系统弹出“Application”对话框询问是否希望让 Powerbuilder 建立应用样板, 如图 2-3 所示。作为初学者, 可以单击“否”, 不生成应用样板, 这样我们就建立了一个应用对象。如果选择“是”, 则在创建新的应用对象时能够自动生成应用程序框架。该框架实际上是个多文档界面 (MDI) 的应用程序, 它包括了框架窗口、MDI 子窗口、菜单和工具栏。通过查看该框架程序, 我们可以学到多文档界面应用程序的设计方法。

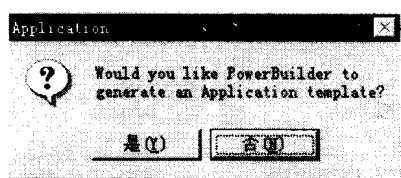


图 2-3 “Application”对话框

2.3 设置应用对象的属性

应用对象是开发 PowerBuilder 应用程序必须用到的对象, 应用对象的属性对整个应用程序的开发起着举足轻重的作用。它的字体属性决定了应用程序开发过程中其他对象的文本字体, 因此, 在设计应用的开始就选择一种适合绝大多数对象使用的字体将会减轻开发压力, 缩短开发时间。利用应用对象可以为应用程序设置如下属性: 缺省字体属性 (包括字体、风格、大小和颜色)、应用程序的库搜索路径、应用程序的图标、缺省的全局对象的类型。本节简述这些属性的用途与设置方法。

应用对象的四个字体属性决定了开发应用程序过程中经常使用的四种字体, 这四个字体属性分别为 Text Font、Column Font、Header Font、Label Font。Text Font 决定窗口中的控件、用户对象、数据窗口中静态文本的缺省字体; Column Font 指定数据窗口中检索出的数据使用的缺省字体; Header Font 定义列表风格 (tabular) 和网格风格 (grid) 数据窗口对象中列标题的缺省字体; Label Font 确定自由风格 (Free form) 数据窗口对象中列标签的缺省字体。在开始开发应用程序时, 根据需要先设定这些字体, 以后设计其他对象时就免除了反复设置字体的重复劳动。这里所说的字体包括四方面的内容: 字体类型 (即宋体、楷体、行楷等), 字体大小 (如 10、12、16、18 等), 字体风格 (即常规、斜体、黑体、黑斜体等), 字体颜色 (包括文字本身的颜色和背景颜色)。下面以设置 Text Font 属性为例说明缺省字体属性的设置方法, 其他字体属性的设置过程与此相似。

(1) 单击“应用画笔”图标进入应用画笔工作区。

(2) 单击应用画笔工具栏上的属性 (Propertise) 图标, 系统打开“Application”对话框。

(3) 单击“Text Font”标签, 系统显示如图 2-4 所示的标签页。