

第 1 章 PowerBuilder 概述

教学提示：PowerBuilder 是美国 Powersoft 公司推出的一种使用方便、功能强大的数据库应用系统的前端开发工具。最新的 PowerBuilder 9.0 版本扩展并改进了许多功能，增加了对 .NET、PBDOM、XML DataWindow、RAD JavaServer Pages 和 PBNI 的支持。PowerBuilder 可以实现与多种大型数据库管理系统的连接，是一个高度集成的数据库应用系统开发工具。

教学要求：本章首先介绍 PowerBuilder 的主要特点，重点使读者掌握 PowerBuilder 9.0 集成开发环境的组成和基本用法，学会使用 PowerBuilder 9.0 开发简单应用程序的基本过程。

1.1 PowerBuilder 简介

PowerBuilder 是著名的数据库应用开发工具生产厂商 Sybase Inc.的子公司 PowerSoft 公司于 1991 年 6 月推出的产品。它基于客户机/服务器体系结构研制设计，用于数据库应用程序的客户端开发。PowerBuilder 采用了面向对象和可视化技术，提供可视化的应用开发环境，利用 PowerBuilder 可以方便、快捷地开发出利用后台服务器中的数据库管理系统的数据库应用系统。这些应用系统既可以运行于单机上，也可以运行于局域网或因特网平台上。PowerBuilder 自投放市场以来，以其独特的体系结构和强大的功能受到广大开发人员的好评和欢迎，在数据库前端开发工具市场占据了很大的份额。

自 1991 年 6 月 PowerBuilder 1.0 投放市场以来，PowerBuilder 经历了多次升级换代，其性能大大增强和改善，使用效率大大提高。特别是随着因特网的发展，PowerBuilder 开始与 Internet 相结合，分别于 1997 年 12 月和 1998 年 11 月推出 PowerBuilder 6.0 和 PowerBuilder 7.0，提供了 Internet 开发环境所需要的组件、库和工具，增强了 Internet 的应用能力，并提供了更加高效的数据窗口对象。2001 年 7 月推出的 PowerBuilder 8.0 进行了较大的改进，在界面上和功能上都做了较大的调整和提高，新增加了 Workspace 对象、Target 对象、系统视图窗口和拖曳拷贝窗口；支持 JavaScript，可以开发网络数据窗口，可以将控件转换成 HTML 格式并应用于网络开发中。

PowerBuilder 9.0 版本于 2003 年推出，与 PowerBuilder 8.0 版本相比，它集设计、建模、开发、部署和管理等功能于一体，增加了对 .NET、PBDOM、XML DataWindow、RAD JavaServer Pages 和 PBNI 的支持。同时，还提供了与 Sybase 应用服务器(Enterprise Application Server, EAServer)更为紧密的集成，能够在 EAServer 中调用 Enterprise JavaBean(EJB)，并能够将现有的组件连接到 EAServer。

1.1.1 PowerBuilder 的功能特点

数据库的开发工具有很多种,如 Microsoft 公司的 Visual Basic、Inprise 公司的 Delphi、Sybase 公司的 PowerBuilder 等,但 PowerBuilder 是其中较优秀的一种,它具有如下主要功能特点。

1. 基于客户端/服务器的体系结构

客户端/服务器(Client/Server, C/S)体系结构是一种将任务分解并协同解决的计算模式,在目前的数据库应用系统中普遍采用。在这种体系结构中,数据库管理系统和数据存在于数据库服务器上,由于服务器负责数据的统一控制和管理,保证了数据的安全性和完整性,并可以充分发挥服务器的高性能;客户机提供用户的访问界面,通过采用标准的 SQL 语句等方式访问服务器上数据库中的数据。PowerBuilder 正是基于 C/S 体系结构的客户端开发工具,使用 PowerBuilder 集成开发环境可以高效、快捷地开发基于 C/S 体系结构的数据库应用系统。

2. 面向对象的开发方法

PowerBuilder 是一种面向对象的开发工具。在 PowerBuilder 中,构成应用程序的窗口、菜单、控件、数据窗口等都是一个个的对象,除了使用这些 PowerBuilder 系统的预定义对象外,开发人员还可以创建新的对象(称为用户对象)。PowerBuilder 提供了对面向对象程序设计方法中的各种技术的全面支持,利用面向对象方法中的对象的封装性、继承性、多态性等特点,使得所开发的应用程序具有极大的可重用性和可扩展性,而这一点正是软件工程中对应应用程序所提出的重要目标。PowerBuilder 提供了基础类库 PFC,为应用程序的开发提供了大量可重用的预定义类和对象,利用 PFC 可以快速地开发出重用性好、质量高的应用程序。

3. 功能完善的 PowerScript 编程语言

PowerBuilder 具有内置的编程语言 PowerScript 语言。该语言除了提供基本的流程控制语句外,还提供了几百个函数来操纵各种对象和提供对多文档界面(MDI)、动态数据交换(DDE)、对象连接与嵌入(OLE/OCX)以及动态链接库(DLL)等方面的支持。此外,开发人员还可以定义自己的函数、处理特定的事件。另外,在脚本中还可以使用嵌入式 SQL 语句操纵和访问数据库。在 PowerBuilder 的学习过程中,需要逐渐了解和掌握这些 PowerBuilder 函数的使用。

4. 事件驱动工作方式

同大多数的 Windows 应用程序一样,PowerBuilder 开发的应用程序也是采用事件驱动方式工作的。在这种工作方式中,程序的运行没有固定的流程,而是用户通过各种操作来控制程序的执行流程。程序中的代码是为各种可能发生的事件编写的脚本,当程序开始运行之后,可以接受来自系统、用户或者其他应用程序触发的事件,然后执行相应事件的脚本代码。例如,用户单击了某个按钮,则执行该按钮的单击(Click)事件的脚本。PowerBuilder 应用程序开发的大量工作就是编写各种对象的事件脚本。

5. 支持多种关系数据库管理系统

PowerBuilder 提供了对目前流行的大多数关系数据库管理系统的支持, 包括 Oracle、Sybase、SQL Server 和 Informix 等, 提供了连接数据库的多种专用接口和 ODBC 标准接口。在 PowerBuilder 的应用程序中, 对数据库访问的部分一般采用国际化标准数据库查询语言 SQL, 使得用 PowerBuilder 开发的应用程序可以不做修改或者只做少量的修改就可以在不同的后台数据库管理系统上使用。也就是说, 用 PowerBuilder 开发的应用程序是独立于服务器上的数据库管理系统的。

另外, PowerBuilder 9.0 还自带了一个功能强大的数据库管理系统(Adaptive Server Anywhere 8.0, 简称 ASA 8.0)。ASA 8.0 是 Sybase 公司专为 PowerBuilder 用户设计的, 功能上具备了大型关系数据库的特点。ASA 8.0 可以与 PowerBuilder 运行于同一台计算机上, 作为 PowerBuilder 的数据库服务器, 在 PowerBuilder 集成开发环境中可以直接操纵该数据库管理系统, 为 PowerBuilder 应用程序的开发提供了完整的环境。

6. 功能强大的数据窗口对象

PowerBuilder 一个很大的特点就是提出了数据窗口对象的概念。数据窗口对象也是 PowerBuilder 中的一种对象类型, 与其他对象不同的是数据窗口对象是专门为了访问后台的数据库服务的。在数据窗口对象中可以定义数据的来源和数据的显示风格, 这样在应用程序开发时, 开发人员可以把精力放在程序的运行流程控制上, 而不用关心具体数据的来源。需要使用数据库中不同的数据时, 只需对数据窗口对象进行修改就可以, PowerBuilder 在数据窗口对象中提供了丰富的数据显示方式, 可以满足各种不同的显示和打印格式需要。利用数据窗口对象可以直接实现数据的查询、添加、删除和修改, 大大减少了程序中数据访问的设计工作量。

7. 使用方便的可视化集成开发环境

PowerBuilder 提供的集成开发环境使应用程序的各种组成成分(如窗口、菜单、数据窗口、报表、脚本等)的创建以及各种开发工作(如创建对象、数据库管理、调试、编译、分布等)都可以在其中完成, 而不必转换到其他开发环境中实现, 从而大大加快了开发速度。

8. 支持多平台开发

PowerBuilder 支持跨平台的开发和分布。PowerBuilder 应用系统可以运行在 Windows 95/98/2000/NT、Macintosh、UNIX、Sun Solaris、HP-UX、IBM AIX RS 6000 等多种开发平台上。例如, 在 Windows 系统上, 利用 PowerBuilder 开发的应用程序可以将该应用程序分布到 UNIX 系统上运行, 反之亦然。开发人员可以组成跨平台的开发小组, 可以自由地共享在 PowerBuilder 中的对象, 因为 PowerBuilder 所支持的不同计算环境的对象是完全一样的。

1.1.2 PowerBuilder 9.0 的新特性

PowerBuilder 9.0 是 Sybase 公司最新推出的一个功能超群、使用方便、易于开发复杂应用系统的数据库前端开发工具, 新版本提供了更为开放的开发环境并扩展了很多新的功能。利用 PowerBuilder 提供的功能强大的开发工具, 可以快速开发大型数据库应用系统。

由于对网络功能的巨大支持，PowerBuilder 正在成为客户机/服务器应用程序开发的标准。与其他客户机/服务器开发环境相比，PowerBuilder 可以使开发人员的开发进度更快、成本更低、质量更高、功能更强。PowerBuilder 为应用的开发提供了全面综合的支持。PowerBuilder 9.0 具有以下新的特性。

1. 先进的 4GLPLUS 基础

PowerBuilder 9.0 的 4GLplus 提供了多种全新的开发功能和集成特性，适用于：J2EE™、Microsoft .NET™、客户端/服务器、Web 服务器、语言集成、设计/建模同步、n 层结构、版本控制、脚本等多种情况。PowerBuilder 9.0 允许用户快速创建并配置持久的企业级或者商务级应用程序。

2. 多样的适用性

1) 扩展用户的客户机/服务器组合

利用一个全新的、强大的应用程序编程接口 PowerBuilder Native Interface (PBNI)，可以将现有的 PowerBuilder 应用程序的功能与生命力扩展到 C++ 和 Java 程序中，为它们打开一个全新的世界与市场。现在，用户能够将处于“维护状态”的客户端/服务器程序带入一个崭新的、以前无法企及的环境中。

2) 利用 N-Tier 与 Web 架构取得成功

将 PowerBuilder 的强大优势充分应用到 Web 和分布式应用程序中。在新的 Web 程序中使用已经创建的 DataWindows 或者全新建造的 DataWindows 生成的强大的瘦客户端应用程序，提供用户所熟悉的数据操作、验证、格式化以及表现方式。现在，只需要一个功能强大的集成开发环境 PowerBuilder 9.0，用户就能够构建一个不同种类的、由客户机/服务器、n 层结构以及 Web 程序组成的混合体应用系统。

3) 对 Microsoft.NET 和 J2EE 的支持

PowerBuilder 9.0 的推出标志着首次对 Microsoft .NET 的支持，也是与强大的 J2EE 在更高级别上的集成。作为与 .NET 整合的第一阶段，PowerBuilder 9.0 可以生成并使用 Web 服务，这使得 .NET 程序员能够充分利用 DataWindow 与 DataStore 模式的优势。在未来，与 .NET 的整合计划包括编译成 MSIL 代码的能力，并使程序员能够利用 .NET 自带控件，例如利用预览、选项卡等建立 .NET 应用程序。

通过与 Sybase EAServer 的紧密结合，PowerBuilder 已经能够支持 J2EE 的开发与配置。在最新版本 PowerBuilder 中，用户创建的应用程序既能够访问任何一台 J2EE™ 兼容应用程序服务器(包括 IBM® WebSphere® Application Server、BEA WebLogic™ Server 以及其他 J2EE 应用程序服务器)的 Enterprise JavaBeans，也可以同时使用 PowerBuilder 的 Web 服务。用户还能够使用 PowerBuilder 创建 RAD 类型的 JavaServer Pages™ (JSP) 应用程序。

3. 集成开发与设计的紧密结合

Sybase PowerBuilder 与 Sybase PowerDesigner 更紧密的结合，为客户机/服务器开发工作带来了高水准的精确、速度与智能。PowerDesigner 提供的全新功能可以从 PowerBuilder 9.0 中提取所有的对象与代码，而不再仅仅是 NVO，这大大简化了现有应用程序的维护。

4. 增加的强大功能

Power Builder 9.0 拥有诸多处于技术前沿的新一代开发功能，它们能够将程序开发提升到更高的水平。

1) PowerBuilder 文档对象模型

PowerBuilder 的文档对象模型(PBDOM)是通过使用标准的 PowerScript 来创建、阅读、编写并操作 XML 文档的完善的 API。PBDOM 以非可视化类为基础，使用 PowerBuilder Native Interface(PBNI)技术进行构建。

2) XML DataWindow

DataWindow 是 PowerBuilder 的专利技术，它将 XML 的强大功能融入到开发项目中。DataWindow painter 的一个新视图把 XML 模板图形化地构建为 DataWindow 对象的一部分。这些模板能够导入或者导出各种传统架构内的 XML 数据。DataWindow 演示也能够被保存为 XSL-FO，用于描述页面报告、打印以及 XML-to-PDF 应用程序的 XML 语言。

3) RAD JavaServer Pages™

PowerBuilder 9.0 包括一个便捷的开发环境，用于编写 JSP Web 应用程序，它能够使许多开发工作自动化，并且允许配置 Web 应用程序。用户可以直接将 JSP Web 目标从 PowerBuilder 配置到 EAServer 或 Tomcat 上，并且一个命令行选项使用户能够将之配置到其他流行的 JSP 容器中。

4) Web 服务向导

PowerBuilder IDE 的 Web 服务向导创建了一个代理对象或传统的 tag 库，包含从 PowerBuilder 应用程序或 JSP 页面调用 Web 服务的必要信息。Web 服务是支持 Microsoft .NET 框架计划几个阶段中的第一阶段。

1.2 PowerBuilder 9.0 集成开发环境

本节介绍 Windows XP 环境下 PowerBuilder 9.0 集成开发环境的组成和基本使用方法。

1.2.1 启动和退出 PowerBuilder 9.0

1. 启动 PowerBuilder 9.0 的方法



在 Windows XP 操作系统下完成 PowerBuilder 9.0 安装后，在【开始】菜单的【所有程序】项中生成【Sybase】程序组项，选择其中的【PowerBuilder 9.0】|【PowerBuilder 9.0】程序项执行，即可启动 PowerBuilder 9.0。

首次启动 PowerBuilder 9.0 时，会弹出【Welcome to PowerBuilder】对话框，如图 1.1 所示。

利用欢迎对话框中的功能选择项，用户可以打开或新建工作区和目标，直接单击窗口关闭按钮可以关闭此对话框。若选中【Don't show this dialog again】复选项再关闭窗口，则以后再启动 PowerBuilder 时，不再弹出欢迎对话框。

2. 退出 PowerBuilder 9.0 的方法

退出 PowerBuilder 9.0 的方法有 4 种。

- (1) 在 PowerBuilder 中选择【File】|【Exit】命令。
- (2) 单击 PowerBuilder 主窗口右上角的窗口关闭按钮.
- (3) 单击 PowerBuilder 主工具条的退出按钮.
- (4) PowerBuilder 主窗口为当前活动窗口时，按下快捷键【ALT+F4】。

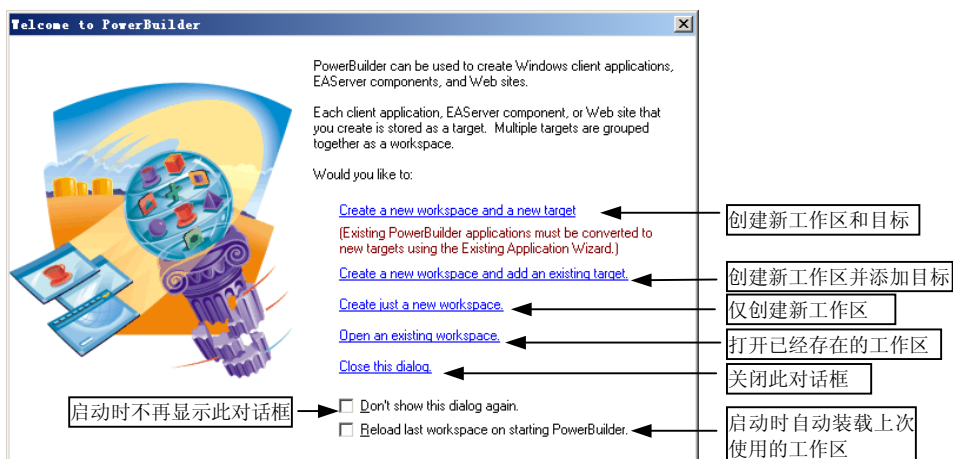


图 1.1 PowerBuilder 的欢迎对话框

1.2.2 PowerBuilder 的主窗口

启动 PowerBuilder 后，在未打开任何工作区时主窗口如图 1.2 所示。PowerBuilder 主窗口是一个 MDI 框架窗口，包含主菜单、主工具条(PowerBar)、系统树窗口(System Tree Window)、剪贴板窗口(Clips Window)、输出窗口(Output Window)、状态栏和工作区。

主菜单和主工具条提供了实现 PowerBuilder 各种功能的菜单项和按钮，多数菜单项都有对应的工具条按钮，这样一来提高了操作效率。

在设计目标对象时，系统树窗口中以树状形式显示所开发的应用系统的资源。系统树窗口有 4 个不同的标签页：Workspace(工作空间)、Page(页面)、Language(语言)和 Components(组件)。页面、语言和组件主要用于 Web Target(Web 目标)的设计。本教程介绍的 PowerScript Target(PowerScript 目标)设计主要使用工作空间标签页，该标签页列出所设计的应用的目标、应用库、各个组成对象及其属性、事件脚本等。

剪贴板窗口用于存放频繁使用的代码片断，代码片断命名存放在剪贴板窗口中，通过拖曳或拷贝，可以将存储的代码粘贴到所需要的脚本编辑器中。

输出窗口显示不同操作时的系统输出信息。

工作区内可以打开不同的编辑器和画板。

通过菜单或工具条按钮可以控制系统树窗口、剪贴板窗口和输出窗口的显示和隐藏。

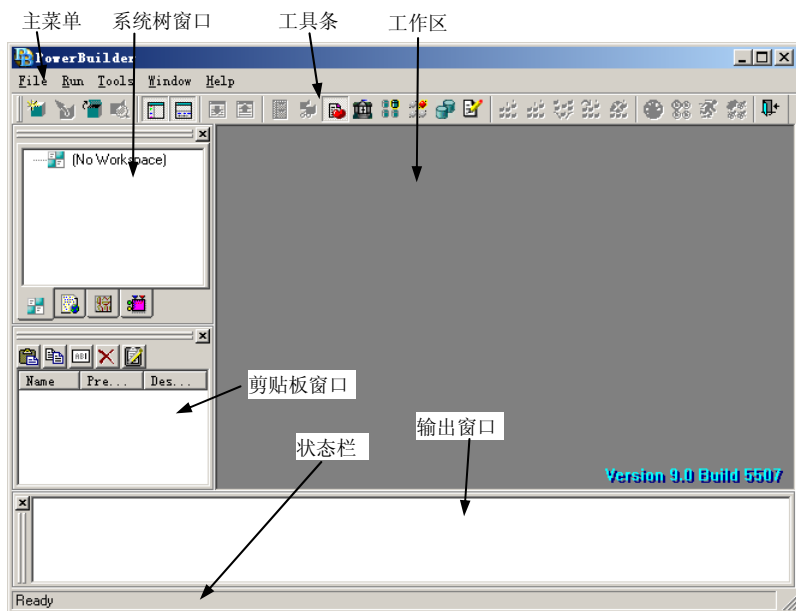


图 1.2 PowerBuilder 9.0 主窗口

1.2.3 PowerBuilder 的主菜单和主工具条





PowerBuilder 的菜单包含了其各项功能，工具条是为了简便操作，对应于常用的菜单项。刚打开 PowerBuilder 时，主菜单包括【File】、【Run】、【Tool】、【Window】和【Help】共 5 个菜单。












PowerBuilder 的主菜单和主工具条如图 1.3 所示。



图 1.3 PowerBuilder 主菜单和主工具条

(1) 【File】菜单：实现对文件的操作。

- ① New: 创建新的对象。对应的工具条图标为 .
- ② Inherit: 由已经存在的对象继承产生新的对象。对应的工具条图标为 .
- ③ Open: 打开已经存在的对象。对应的工具条图标为 .
- ④ Run/Preview: 运行/预览窗口或数据窗口对象。对应的工具条图标为 .
- ⑤ Open Workspace: 打开已经存在的一个工作空间。
- ⑥ Printer Setup: 设置打印机。
- ⑦ Recent Objects: 查看最近使用过的对象，选择某对象可直接打开该对象。
- ⑧ Recent Workspaces: 查看最近使用过的工作空间，选择某工作空间可直接打开该工作空间。
- ⑨ Recent Connections: 查看最近使用过的数据库连接，选择某连接可直接打开该数据库连接。

- ⑩ **Exit**: 退出 PowerBuilder。对应的工具条图标为 。
- (2) **【Run】** 菜单: 对应用程序进行编译、调试、运行等操作。
Run: 运行正在编辑的应用程序。对应的工具条图标为 。
其他菜单项在相应的章节中介绍。
- (3) **【Tools】** 菜单: 提供 PowerBuilder 开发环境的常用工具。
- ① **Toolbars**: 设置工具条。
 - ② **Keyboard Shortcuts**: 设置快捷键。
 - ③ **System Options**: 设置系统选项。
 - ④ **To Do List**: 显示执行列表窗口。对应的工具条图标为 。
 - ⑤ **Browser**: 显示对象浏览器窗口。对应的工具条图标为 。
 - ⑥ **Library Painter**: 打开应用库画板。对应的工具条图标为 。
 - ⑦ **Database Profile**: 打开数据库描述文件(DB Profile)。对应的工具条图标为 。
 - ⑧ **Database Painter**: 打开数据库画板。对应的工具条图标为 。
 - ⑨ **File Editor**: 打开 PowerBuilder 的文件编辑器。对应的工具条图标为 。
- (4) **【Window】** 菜单: 实现对 PowerBuilder 开发环境中的窗口的排列、布局、显示/隐藏等操作。
- ① **Tile Vertical**: 垂直排列打开的多个窗口。
 - ② **Tile Horizontal**: 水平排列打开的多个窗口。
 - ③ **Layer**: 分层排列打开的多个窗口。
 - ④ **Cascade**: 层叠排列打开的多个窗口。
 - ⑤ **Arrange Icons**: 排列图标。
 - ⑥ **Close All**: 关闭所有窗口。
 - ⑦ **System Tree**: 显示或隐藏系统树窗口。对应的工具条图标为 。
 - ⑧ **Output**: 显示或隐藏输出窗口。对应的工具条图标为 。
 - ⑨ **Clip**: 显示或隐藏剪贴板窗口。对应的工具条图标为 。
- (5) **【Help】** 菜单: 提供联机帮助、相关网址和关于 PowerBuilder 的介绍。

1.2.4 PowerBuilder 的画板

在 PowerBuilder 中, Painter(画板)是用于编辑对象的工具。PowerBuilder 为不同的对象设计有不同的画板,如 Application Painter(应用对象画板)、Window Painter(窗口画板)、Menu Painter(菜单画板)、DataWindow Painter(数据窗口画板)等。

除了这些编辑对象的画板外, PowerBuilder 还提供了一些其他操作的画板,如 Library Painter(应用库画板)、Database Painter(数据库画板)。

1. 打开画板的方法


打开编辑对象的画板主要使用下列方法。


(1) 在主工具条上单击 **【New】** 图标或 **【Inherit】** 图标创建新的对象,或者单击 **【Open】** 图标打开已经存在的对象。

(2) 选择 **【File】** 菜单的 **【New】** 或 **【Inherit】** 命令创建新的对象,或者选择 **【Open】**

命令打开已经存在的对象。

(3) 在系统树窗口双击要编辑的对象。

打开数据库画板可以通过单击主工具条上的数据库图标实现。

打开应用库画板可以通过单击主工具条上的应用库图标实现。

PowerBuilder 9.0 提供的画板及其功能见表 1-1。

表 1-1 PowerBuilder 9.0 的画板及其功能

画 板	名 称	功 能
Application Painter	应用对象画板	为应用对象设置属性及编写脚本
Database Painter	数据库画板	管理数据库、创建表、操纵数据
DataWindow Painter	数据窗口画板	创建和编辑数据窗口对象
Data Pipeline Painter	数据管道画板	创建从一个数据源到另一个数据源传递数据的数据管道
Function Painter	函数画板	定义全局函数
Library Painter	应用库画板	创建和管理应用库
Menu Painter	菜单画板	创建和编辑菜单
Project Painter	项目画板	用于创建可执行的文件、动态库或组件
Query Painter	查询画板	图形方式创建并保存 SQL-SELECT 语句
SQL Select Painter	SQL Select 画板	为数据窗口图形方式定义 SQL-SELECT 语句
Structure Painter	结构画板	定义全局结构
User Object Painter (visual)	可视化用户对象画板	创建可重用的可视化用户对象
User Object Painter (nonvisual)	非可视化用户对象画板	创建可重用的非可视化用户对象
Window Painter	窗口画板	创建窗口

2. 画板的组成

画板打开后，以窗口形式出现在主窗口的工作区，菜单栏中增加了对应于此画板的菜单，同时增加了相应的工具条，称为 PainterBar(画板工具条)。如图 1.4 所示为一个打开窗口画板时的主窗口布局。

在画板窗口中打开了若干 Pane(面板)窗口(也称为子窗口)，每个面板窗口由一个或多个叠放的 View(视窗)组成，下端显示视窗名称。不同的画板由不同的视窗组成，在【View】菜单中列出了这些视窗的名称。例如，如图 1.4 所示的窗口画板中有两个面板窗口，左边的面板窗口中叠放着【Layout】、【Script】、【Event List】等视窗；右边的面板窗口中叠

放着【Properties】、【Control List】等视窗。

各种画板的具体用法将在相关的章节中介绍。

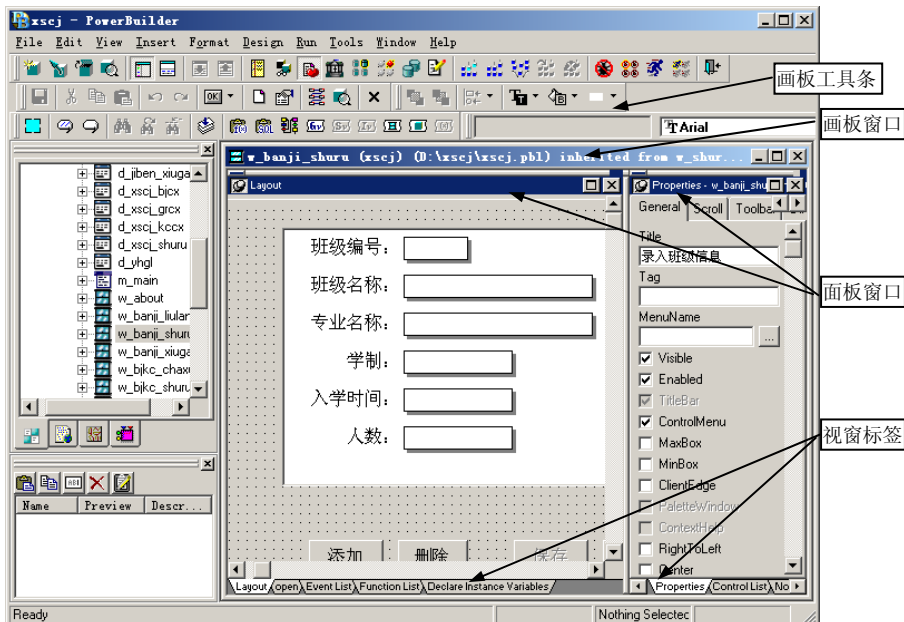


图 1.4 打开窗口画板时的主窗口布局

1.2.5 调整 PowerBuilder 开发环境布局

为了操作方便，经常需要调整 PowerBuilder 开发环境的布局，如改变工具条位置、显示/隐藏工具条、改变窗口位置、显示/隐藏窗口等。下面将介绍如何实现这些操作。

1. 定制工具条

选择【Tools】|【Toolbars】命令，则打开【Toolbars】对话框，如图 1.5 所示。

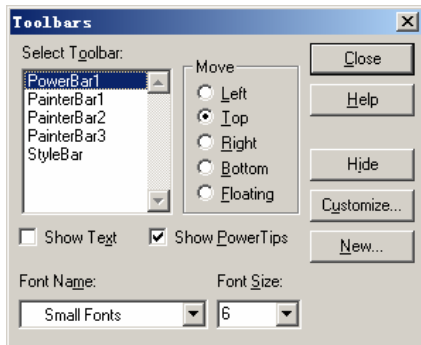


图 1.5 【Toolbars】对话框

在此对话框中，可以设置【Show Text】(工具条图标上是否显示文字)选项以及【Show PowerTips】(鼠标停在某个图标上时是否显示提示文字)选项，还可以设置文字的字体和大小等。在【Select Toolbar】列表中选择一个工具条，在【Move】组框中可以设置该工具条

的位置(左、上、右、下、浮动),单击【Hide】|【Show】按钮可以隐藏/显示该工具条。

单击【Close】按钮,则关闭【Toolbars】对话框。

将鼠标放在主工具条或画板工具条的左端(或上端)双线上,拖动该工具条到要停靠的位置,然后抬起鼠标,则该工具条便可停靠在鼠标释放的位置,同样实现工具条的移动。

另外,在工具条上右击鼠标,则弹出快捷菜单(如图 1.6 所示),使用此快捷菜单也可以定制工具条,设置方法同【Toolbars】对话框类似。

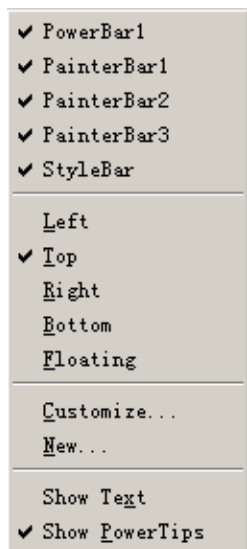



图 1.6 工具条快捷菜单

2. 调整窗口布局

在主窗口中的系统树窗口、剪贴板窗口和输出窗口都是停靠窗口,通过拖动窗口上端(或左端)的双线可以将窗口停靠在主窗口的任意边框处。


单击窗口右上角(或左上角)的关闭按钮 ,可以关闭此窗口。

利用【Window】菜单中的【System Tree】、【Output】、【Clip】命令分别可以显示/隐藏系统树窗口、输出窗口和剪贴板窗口。

3. 调整画板布局

在工作区中打开某种画板时,该画板往往由很多视窗组成,在【View】菜单中列出了这些视窗的名称,如图 1.7 所示。

通过画板窗口右上角的【最大化】、【最小化】、【关闭】、【还原】按钮可以对画板窗口进行相应的操作。

画板中多个视窗叠放在一起形成一个个面板窗口,面板窗口的标题栏是自动隐藏的。鼠标移到标题栏时则自动显示标题栏,单击标题栏左端的图钉按钮 ,则标题栏不再隐藏;再次单击此按钮,标题栏还原为自动隐藏。

通过面板窗口右上角的【最大化】、【关闭】、【还原】按钮可以对该面板窗口进行相应的操作。拖动面板窗口之间的夹缝可以改变面板窗口的大小。

拖曳面板窗口下边某个视窗标签到面板外,可以将该视窗独立显示为一个面板窗口;

将一个面板窗口拖曳到另一个面板窗口上，则合并为一个面板窗口。

选择【View】|【Layouts】|【(Default)】命令，还原为系统默认的画板布局形式。

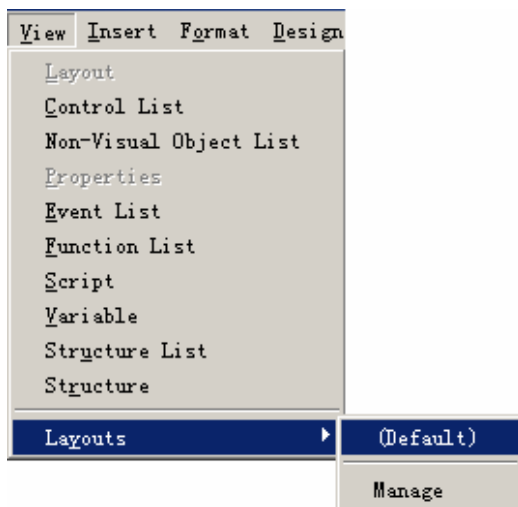


图 1.7 【View】菜单

1.2.6 使用联机帮助

使用 PowerBuilder 提供的联机帮助，对于快速、准确地掌握 PowerBuilder 开发环境的使用方法和编程语言十分有益。选择【Help】|【Contents】命令或者按下【F1】键随时可以调出 PowerBuilder 系统的联机帮助，它有【目录】和【索引】两个标签页。

目录页是以书目的形式列出了帮助主题，如图 1.8 所示。双击某一书目图标，可以将其展开，再双击某一标题则打开其具体帮助内容。

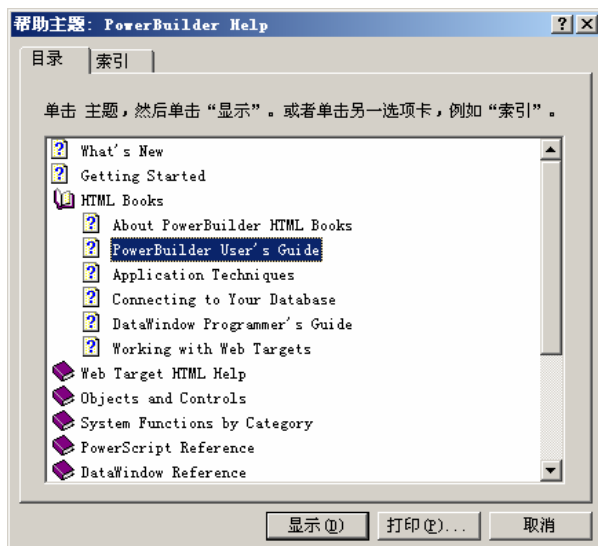


图 1.8 联机帮助目录页

索引页如图 1.9 所示，在输入框中输入要查询主题的头几个字母，列表中立刻定位到

对应的索引项，单击【显示】按钮即可调出有关的帮助内容。



图 1.9 联机帮助索引页

1.3 PowerBuilder 9.0 应用系统的组成

PowerBuilder 9.0 可以开发两种类型的应用程序，一种是 2-tier 或 n-tier 的应用程序，另一种是 Web 应用程序。本教程仅介绍 2-tier 的 C/S 应用程序设计。

在 PowerBuilder 中，一个 C/S 应用程序作为一个 Target(目标)，该目标又由若干 Object(对象)组成，常用的有：应用对象、窗口对象、菜单对象、数据窗口对象、用户自定义对象等，组成目标的所有对象存储在一个或多个 Library(应用库)文件中。各个对象有自己的属性、事件和方法，窗口对象和数据窗口对象又可以包含控件对象。

1.3.1 工作空间和目标

为了管理方便，PowerBuilder 把开发环境中正在设计的应用程序目标以及开发环境的布局看做一个 Workspace(工作空间)，工作空间以文件(扩展名为.pbw)形式存储。在一个工作空间中可以包含多个设计的目标，这样，开发人员就可以同时设计多个应用程序，但需要注意的是任何时刻只有一个目标是活动的。

Target 是 PowerBuilder 开发的一种应用程序，目标文件的扩展名为.pbt。PowerBuilder 的目标对象可以分为两种。

(1) PowerScript Target(脚本目标)对象：包括传统的两层客户机/服务器应用程序和多层的分布式应用程序(客户机可执行程序和服务端组件)。

(2) Web Target(Web 目标)对象：Web 应用程序，包括建立一个网站所需的全部组件，如 HTML 文件、脚本、图像、下载组件等。

工作空间的组成以树状形式显示在系统树窗口中，层次清楚，便于管理和操作。如图 1.10 所示为一个工作空间的结构，工作空间 xscj 包含一个目标 xscj，该目标的对象存储在一个应用库文件(xscj.pbl)中。

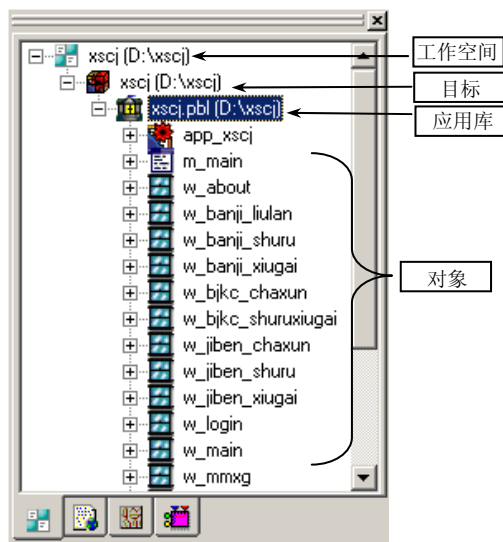


图 1.10 工作空间的结构

1.3.2 对象

PowerBuilder 是采用面向对象技术的开发工具。任何 PowerBuilder 应用程序都是由对象组成的，如脚本目标对象、应用对象等。PowerBuilder 提供了多种类型的对象，有可视化对象，如窗口对象、菜单对象、按钮对象等；还有非可视化对象，如 DataStore(数据存储)对象。除了系统预定义的对象外，用户还可以自定义对象。

各种对象都包含各自的属性、方法和事件。在设计 PowerBuilder 应用程序时，就是要根据需要创建各种对象、设置对象的属性、编写对象的事件脚本。

创建对象既可以新建，又可以由已经存在的对象继承产生。

1.3.3 事件和脚本

PowerBuilder 是事件驱动的工作方式，Event(事件)就是系统或用户向对象发送的消息，对象接收到消息时触发相应的事件。例如，当用户单击一个按钮、选择一个菜单项、在编辑框中输入了文本等，则对象的相应事件就被触发，系统就会执行该事件的脚本。每种不同的对象都有相应的事件集，程序设计时就要为可能触发的事件书写代码，即 Script(脚本)。

例如，为某按钮的 Click 事件书写了脚本，则用户单击该按钮时就执行此 Click 事件脚本。

1.4 开发一个简单应用程序

本节通过一个简单的应用程序介绍 PowerBuilder 程序的开发过程。

1.4.1 程序简介

为简化程序开发的复杂度，使读者了解 PowerBuilder 程序开发的基本过程，本程序不涉及数据库的访问。

本程序功能很简单，实现 100 以内的加法练习。单击【开始】按钮后，随机产生两个 100 以内的加数，练习者填入两数和后，单击【确定】按钮，程序判断结果是否正确，单击【下一题】按钮重新出题。运行界面如图 1.11 所示。

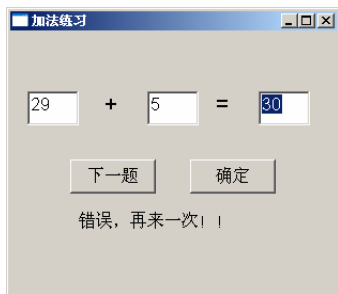


图 1.11 运行界面

1.4.2 新建一个工作空间

首先，新建一个工作空间。当 PowerBuilder 开发环境中打开有工作空间时，尽管可以在当前工作空间中新建应用程序，但一般建议重新建立一个工作空间，具体步骤如下。

(1) 选择【File】|【New】命令或单击主工具条的【New】图标，则打开【New】对话框，如图 1.12 所示。

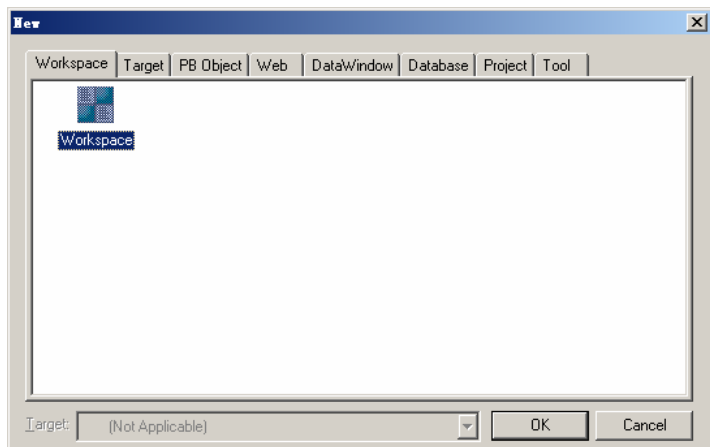


图 1.12 【New】对话框

(2) 选择【Workspace】标签页中的【Workspace】图标，然后单击【OK】按钮，则打开【New Workspace】对话框，如图 1.13 所示。

(3) 选择工作空间的保存位置，输入工作空间文件名 jflx，单击【保存】按钮。完成新工作空间的建立。

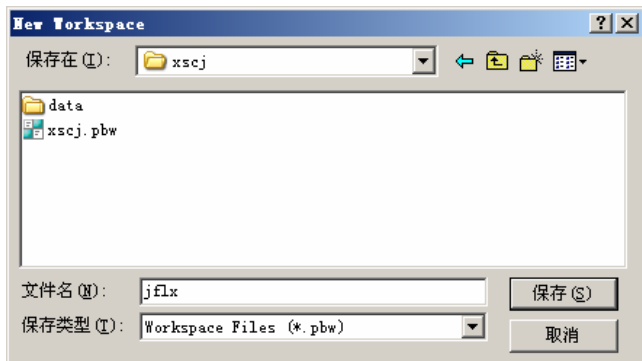


图 1.13 【New Workspace】对话框

1.4.3 新建一个应用对象

在新建的工作空间中新建一个应用程序目标，步骤如下。

(1) 使用菜单或主工具条图标打开【New】对话框，在【Target】标签页中选择【Application】图标，然后单击【OK】按钮，则打开【Specify New Application and Library】对话框，如图 1.14 所示。

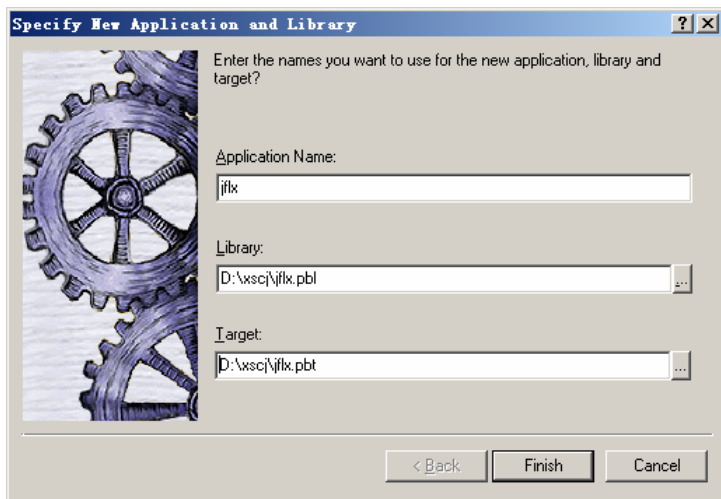


图 1.14 【Specify New Application and Library】对话框

(2) 在【Application Name】输入框中输入应用对象的名称“jflx”，在【Library】输入框中输入应用库文件名“d:\xscj\jflx”，在【Target】输入框中输入目标对象的文件名“d:\xscj\jflx”。输入后两个文件名必须带路径，也可以通过单击按钮打开对话框，选择文件位置并命名保存，最后单击【Finish】按钮完成。

1.4.4 创建窗口对象


创建窗口对象，步骤如下。

(1) 使用菜单或主工具条图标打开【New】对话框，在【PB Object】标签页中选择

【Window】图标，然后单击【OK】按钮，则打开窗口画板。

(2) 在窗口画板的【Layout】视窗中调整设计窗口的大小。在【Properties】视窗中将其 Title 属性设置为“加法练习”。

(3) 在窗口中添加控件，结果如图 1.15 所示，步骤如下。

① 单击画板工具条中图标的右边箭头，拉出控件面板。单击其中的 SingleLineEdit(单行编辑框)控件图标，然后鼠标移到窗口画板的【Layout】视窗的设计窗口单击，则窗口上出现一个单行编辑框控件(sle_1)。调整其大小和位置，将其 Text 属性设置为空，DisplayOnly 属性设置为未选中。同样方法创建另两个单行编辑框控件(sle_2, sle_3)，sle_3 的 DisplayOnly 属性取默认值。

② 添加 3 个 StaticText(静态文本)控件(st_1, st_2, st_3)，调整位置和大小，将其 Text 属性分别设置为“+”、“=”和“空”。

③ 添加两个 CommandButton(命令按钮)控件(cb_1、cb_2)，调整位置和大小，其 Text 属性分别设置为“开始”和“确定”。

(4) 编写控件的事件脚本。

双击【开始】命令按钮，进入【Script】视窗，为其 Click 事件添加以下脚本代码。

```
int li_x, li_y                //定义局部变量
randomize(0)
do                            //产生两个加数
    li_x= Rand ( 100 )
    li_y= Rand ( 100 )
loop while li_x+li_y>100
sle_1.text=string(li_x)      //转换为文本
sle_2.text=string(li_y)
sle_3.text=""
this.text = "下一题"        //改变按钮文本
st_3.text = ""
sle_3.setfocus( )          //输入焦点移到“和”输入框
```

为【确定】命令按钮的 Click 事件添加以下脚本代码。

```
int li_x, li_y, li_z
li_x=integer(sle_1.text)     //转换为整数
li_y=integer(sle_2.text)
li_z=integer(sle_3.text)
if li_x+li_y = li_z then     //判断结果是否正确
    st_3.text="正确!!"
else
    st_3.text="错误，再来一次!!"
    sle_3.SetFocus( )
    parent.sle_3.selecttext(1, 3)
end if
```

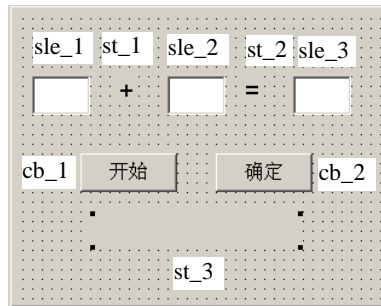


图 1.15 添加控件

(5) 选择【File】|【Save】命令，保存设计的窗口，命名为 w_main。关闭窗口画板，完成窗口设计。

1.4.5 添加应用对象的事件脚本

双击系统树窗口中的应用对象“jflx”，打开应用对象画板，在【Script】视窗中为应用对象的 Open 事件书写以下脚本。

```
open(w_main)
```

关闭应用对象画板，保存修改，完成程序设计。

单击主工具条中的运行图标，就可以运行此程序。

通过本程序只是说明了设计 PowerBuilder 程序的基本过程，当程序功能复杂时，需要创建更多的各种对象，编写复杂的事件脚本。

1.5 小 结

本章介绍了 PowerBuilder 的功能和特点以及新版本 PowerBuilder 9.0 的一些新特性。PowerBuilder 是一种高效、便捷、功能强大的客户机/服务器模式以及分布式的数据库应用程序前端开发工具，新版本中增强了对各种网络功能的支持、开放度更大。

本章还介绍了 PowerBuilder 开发环境的组成和基本用法，PowerBuilder 开发环境是一个高度集成的、可视化的开发工具。特别是 PowerBuilder 的众多画板为程序设计提供了方便、高效的工具。各种画板将在后续有关章节中详细介绍。

本章最后介绍了 PowerBuilder 应用程序的组成部分，并通过设计一个“加法练习”小程序，使读者初步了解了 PowerBuilder 应用程序设计的基本过程。

1.6 实 训

实训目的

- (1) 熟悉 PowerBuilder 开发环境布局。
- (2) 掌握 PowerBuilder 开发环境布局的调整方法。
- (3) 学会设计一个简单 PowerBuilder 应用程序。

实训内容

- (1) 使用 PowerBuilder 开发环境。
- (2) 设计一个“减法练习”程序。
- (3) 练习画板的调整。

实训步骤

1. 使用 PowerBuilder 开发环境

- (1) 启动 PowerBuilder 开发环境/退出 PowerBuilder。
- (2) 调整主工具条的位置。
- (3) 显示/隐藏系统树窗口、剪贴板窗口和输出窗口。
- (4) 改变系统树窗口、剪贴板窗口和输出窗口的位置及大小。

2. 设计一个“减法练习”程序

仿照 1.4 节的程序设计过程，设计一个“减法练习”小程序，实现 100 以内的减法练习，被减数和减数随机产生，差不能为负数。工作空间、应用对象、目标和应用库都命名为：jianfalianxi，具体设计步骤省略，参照 1.4 节。

提示：

- (1) 静态文本控件 st_1 的 Text 属性为“+”。
- (2) 命令按钮控件 cb_1 的 Click 事件脚本如下。

```
int li_x,li_y                                //定义局部变量
randomize(0)
do                                            //产生两个加数
    li_x= Rand ( 100 )
    li_y= Rand ( 100 )
loop while li_x - li_y<100
sle_1.text=string(li_x)                    //转换为文本
sle_2.text=string(li_y)
sle_3.text=""
this.text = "下一题"                      //改变按钮文本
st_3.text = ""
sle_3.setfocus( )                        //输入焦点移到“和”输入框
```

- (3) 命令按钮控件 cb_1 的 Click 事件脚本如下。

```
int li_x,li_y,li_z
li_x=integer(sle_1.text)                  //转换为整数
li_y=integer(sle_2.text)
li_z=integer(sle_3.text)
if li_x - li_y = li_z then                //判断结果是否正确
    st_3.text="正确!! "
else
    st_3.text="错误，再来一次!! "
    sle_3.SetFocus( )
    parent.sle_3.selecttext(1, 3)
end if
```

设计完成后运行程序，测试运行结果是否正确。

3. 练习画板的调整

- (1) 打开上面的“减法练习”程序的工作空间 jianfalianxi。

(2) 在系统树窗口中练习展开/收缩工作空间、目标、应用库、应用对象和窗口对象的层次结构。

(3) 双击窗口对象，打开窗口画板。

(4) 拖曳【Script】视窗成为独立的面板窗口。

(5) 设置【Script】视窗的标题栏为自动隐藏/固定显示。

(6) 拖动面板窗口之间的夹缝，改变面板窗口的大小。

(7) 最大化/还原某个面板窗口。

(8) 拖曳【Script】视窗合并到【Layout】视窗所在的面板窗口。

(9) 设置画板布局为系统默认形式。

(10) 关闭窗口画板。

1.7 习 题

1. 简述 PowerBuilder 的主要特点。

2. 简述 PowerBuilder 的主窗口的组成。

3. 简述 PowerBuilder 开发程序的基本过程。

4. 设计一个 100 以内乘法练习的 PowerBuilder 程序。