

POWERBUILDER

Client/Server

Internet/Intranet

应用开发用户手册 卷 I

晓通网络数据库研究所

第三章 PowerBuilder 的编程

3.1 PowerBuilder 的编程方法

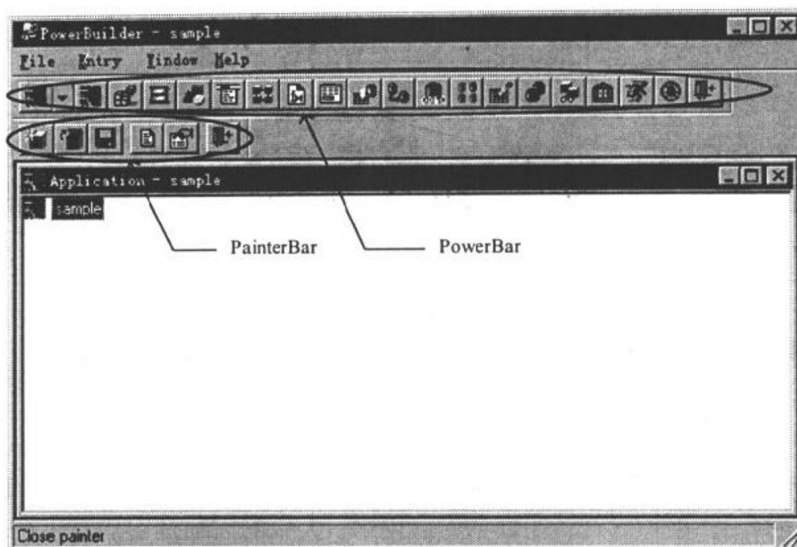
PowerBuilder 是一种面向对象的、具有可视图形界面的、快速的交互式开发工具。它的基本的思想是“事件驱动”，在编程过程中没有预定义的路径，因为在 Windows 环境下无法预知用户要做什么。在数据库应用中，大量的问题是用户发出某个动作（如提出查询请求，选中查询到的某条记录，操作菜单等），应用系统对此动作做出相应的回答（给出查询结果，对选中的记录做特定的处理，执行菜单规定的动作等等）。在 PowerBuilder 中，用户发出的动作叫事件，这个动作作用在哪个对象上就是哪个对象的事件，对此动作的回答就是此事件的事件处理程序。

PowerBuilder 提供的对象有：应用对象 (Application Object)、窗口 (Window)、数据窗口 (DataWindow)、菜单 (Menu)、函数 (Function) 以及用户对象等。开发人员可以对对象的各事件进行编程（如在主窗口的 open 事件中加入事件处理程序，运行时，主窗口打开，触发 open 事件，执行 open 事件的处理程序）。

编程的过程是先定义对象，如窗口及窗口上的控制等，再添加各对象的事件处理程序。PowerBuilder 提供了 PowerScript 编程语言，在各对象的 Script 窗口中用 PowerScript 语言定义事件处理程序。PowerScript 是一种类似 C 语言的编程语言，语言简洁，可读性强，具有内嵌的 SQL 语言，编写好的 PowerScript 语句封装在对象的事件中，在运行时与对象一起被调用。

3.1.1 PowerBuilder 的编程方法

PowerBuilder 编程的所有过程都是在交互式图形界面下进行的。PowerBuilder 提供了两个画笔栏，即 PowerBar 和 PainterBar，如下图所示：

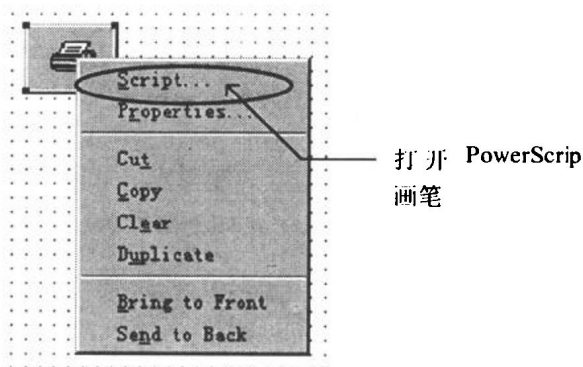


PowerBar 是对 PowerBuilder 的所有对象进行操作的（如打开窗口、新建菜单、连接库、运行当前的应用等等）；PainterBar 只对当前打开的对象进行操作，对于不同类型的对象，PainterBar 是不一样的。

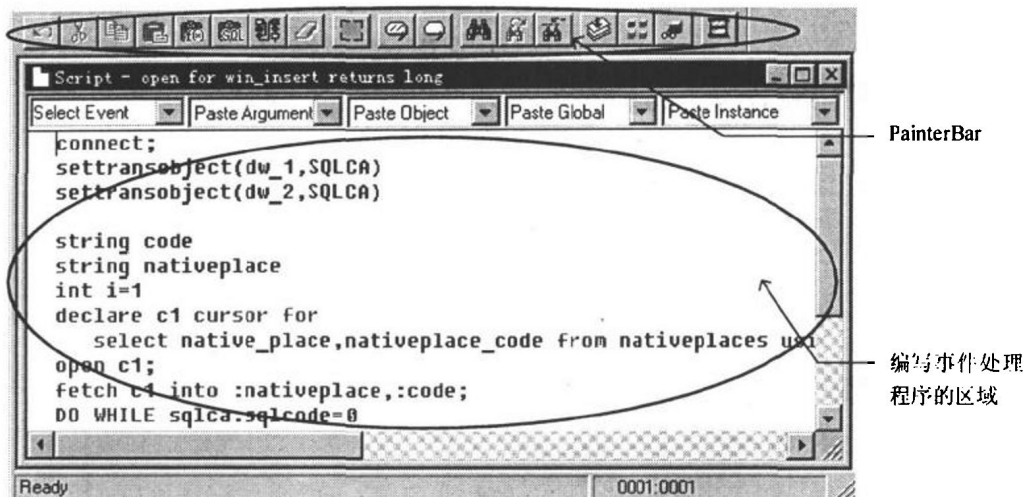
3.1.2 打开 PowerScript 画笔

在 PowerScript 画笔中可以添加和修改对象的事件处理程序。只有应用对象、窗口、菜单等有事件的对象才有此操作，下面我们以窗口对象为例说明。

- 1) 用鼠标选择要处理的对象，然后再单击 PainterBar 中的  图标，或是用鼠标右键单击要处理的对象，选择 Script 项，见下图，打开该对象的 Script 窗口。



- 2) 下面是打开的 PowerScript 画笔：



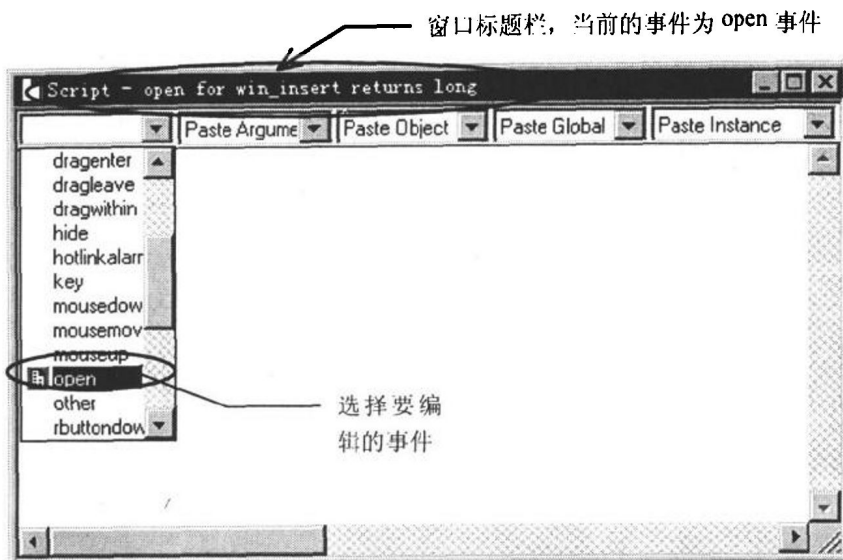
Script 窗口上有五个下拉列表框：

Select Event	选择事件
Paste Argument	粘贴系统参数
Paste Object	粘贴对象
Paste Global	粘贴全局变量
Paste Instance	粘贴实例变量

列表框下面的区域是编写事件处理程序的编程工作区，在编写事件处理程序前必须先选择事件。

3.1.2.1 选择事件、查看已编程的事件

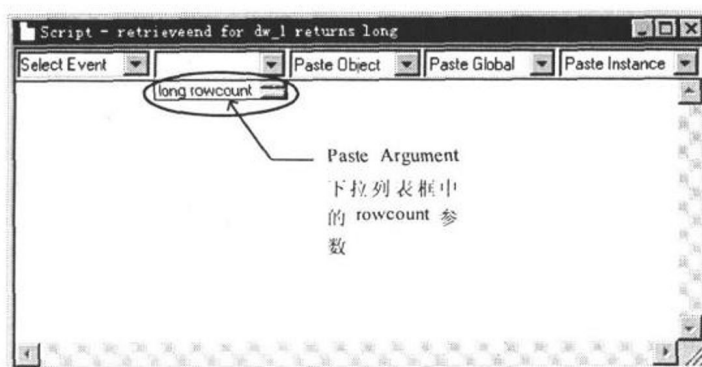
从 Select Event 中选择该对象或控制的事件，打开 Select Event 下拉列表框，用鼠标选择准备编辑的事件。如下图：



窗口的标题栏上显示当前的事件，如上图，当前的事件是 open 事件，这时就可以对当前事件编程了。当 Select Event 列表框打开后，我们能够看到有的事件前有一个小图标，如上图的 open 事件，这就表示此事件中已经编写程序了。选择此事件，此事件的事件处理程序显示在屏幕上，我们可以对它进行修改和删除操作。如果事件前没有图标，则表示没有对此事件编写过程序。

3.1.2.2 选择并粘贴事件上已定义的参数

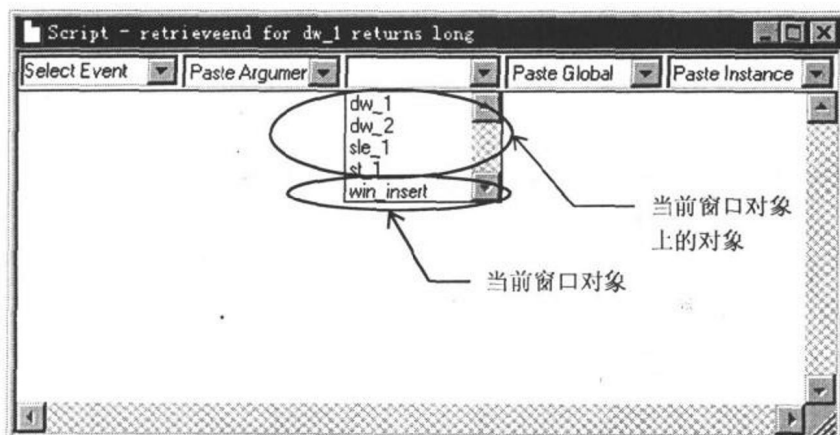
Script 窗口上的 Paste Argument 下拉列表框用来粘贴事件上系统已定义的参数。对于不同的事件，系统定义参数不一样，如窗口对象的 doubleclicked 事件中定义的参数有 xpos,ypos,row,two；数据窗口对象的 retrieveend 事件中定义的参数有 rowcount。下面我们以数据窗口对象的 retrieveend 事件中定义的参数 rowcount 为例说明粘贴事件上已定义参数的方法，见下图：



如上图所示，打开 Paste Argument 下拉列表框，单击 rowcount 参数，该参数被粘贴到光标所在处。rowcount 参数是系统给出的参数，它表示在触发该事件时数据窗口中记录的条数，我们可以通过引用该参数获得数据窗口中记录的条数。

3.1.2.3 选择并粘贴对象

Script 窗口上的 Paste Object 下拉列表框是用来粘贴当前窗口对象和该窗口对象上的对象的：



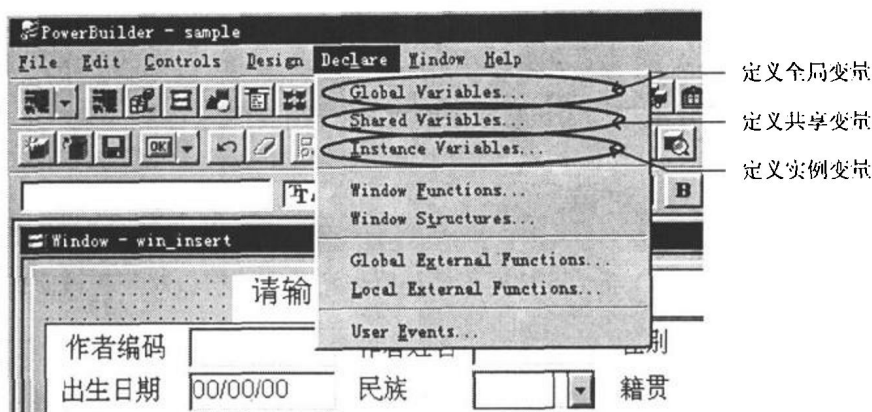
打开 paste object 下拉列表框，选择所需的对象，该对象名就被粘贴到了光标所在处。粘贴当前窗口对象可以方便编程，因为有时窗口上的控制对象比较多，不容易记住，通过粘贴窗口对象功能可以快速、准确地把所需的对象名放在处理程序中。

3.1.2.4 定义和粘贴变量（全局变量、实例变量）

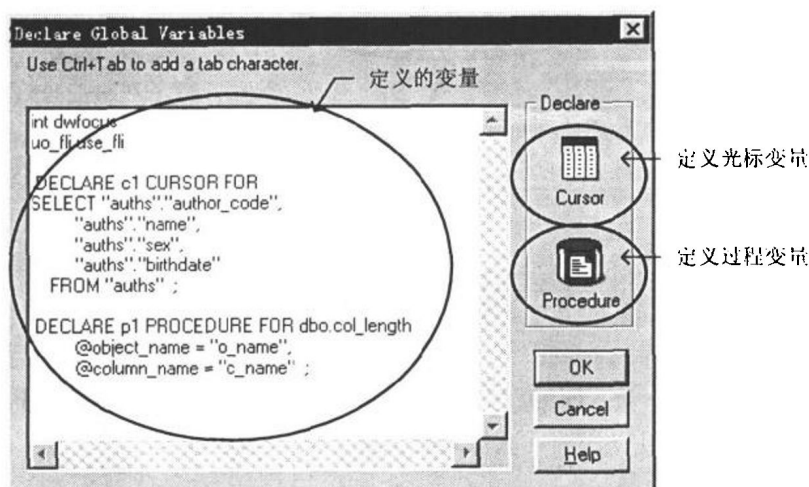
1. 定义变量

在 PowerScript 语言中，可以定义各种类型的变量。变量有三种：Global Variables（全局变量）、Shared Variables（共享变量）、Instance Variables（实例变量）。全局变量定义后可以在此应用的所有对象的事件中引用；共享变量可以在本对象内的所有事件中调用，实例变量可以在本对象及其继承后代的所有事件中调用。变量的定义方法如下：

- 1) 点取菜单项 Declare->Global Variables(这里以全局变量的定义为例，共享变量、实例变量的定义方法与全局变量的定义方法一样)：



2) 打开全局变量定义对话框:



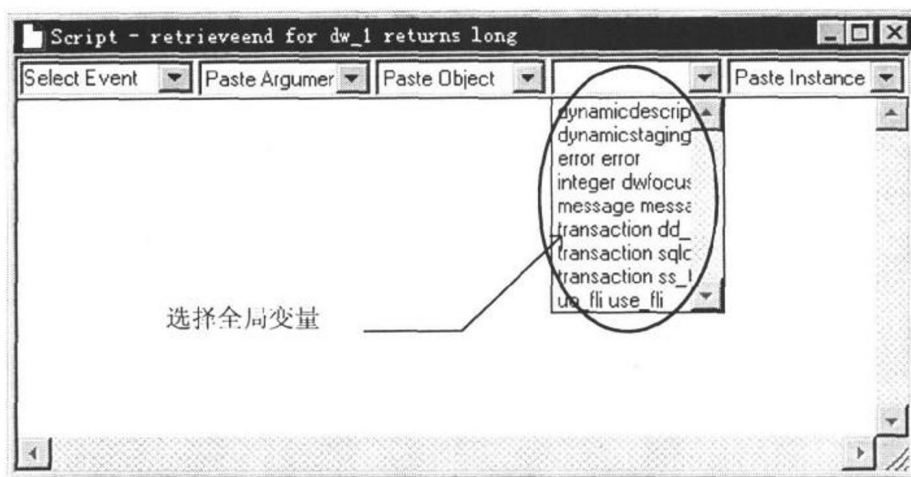
如上图，左边的列表框内可以定义各种类型的变量，如图所示已经定义的变量有：整型变量 dwfocus、用户对象 uo_fli 型变量 use_fli、光标型变量 c1、过程型变量 p1。光标是 PowerBuilder 中特有的变量，是把一条 SQL 语句定义为一个变量，过程型变量是把一个存储过程定义为一个变量，以后只要通过特定的方法使用这个变量就可操作这条 SQL 语句或过程。在上图中可以利用 Declare 中的 Cursor 和 Procedure 项来定义光标变量和过程变量。

另外，也可以在 Script 窗口下直接定义私有变量，此变量只能在该事件中调用。

2. 粘贴变量（全局变量、实例变量）

paste global 与 paste instance 的功能基本一样，paste global 是用来粘贴全局变量的，

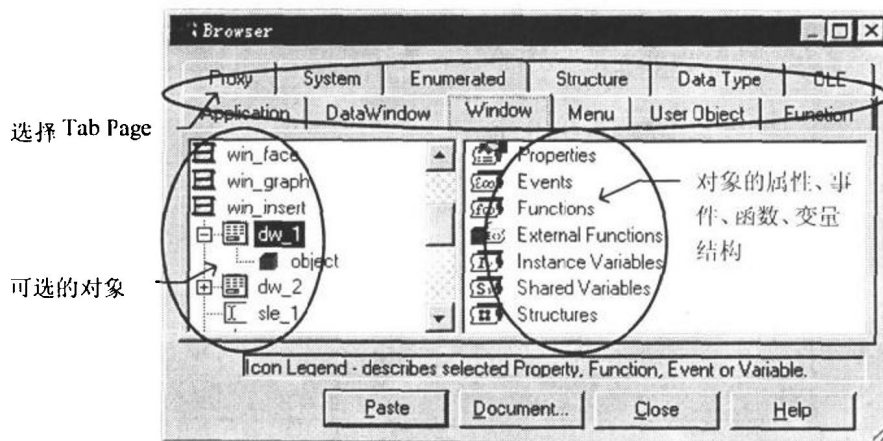
paste instance 是用来粘贴实例变量的，见下图的粘贴全局变量：



打开列表框后用鼠标选择要粘贴的变量，该变量就会被粘贴到当前光标所在处，粘贴实例变量的方法与粘贴全局变量的方法一样。

3.1.2.5 利用对象浏览工具

利用对象浏览工具可以粘贴应用中的数据类型、函数、结构、变量和对象的属性。单击 PainterBar 中的  图标，对象浏览工具打开，见下图：

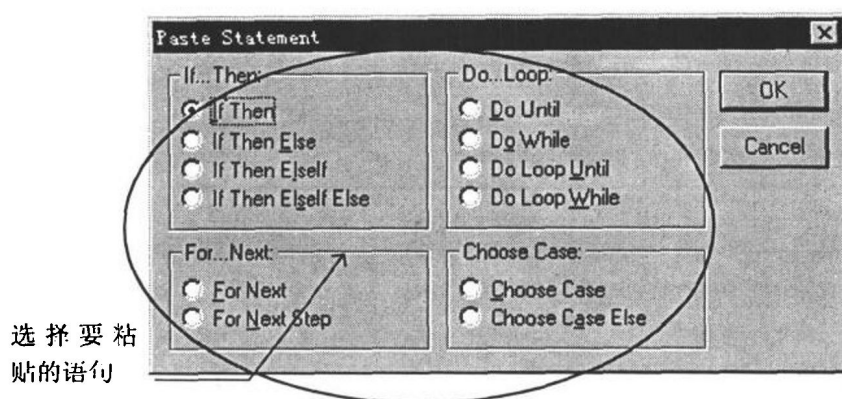


此窗口采用了当前流行的 Tab 形式，将许多的设置工作都集中在一个窗口上，用户只要点取相应的 Tab 按钮，对应的 Tab Page 即被打开。左边列表框内为可选的对象，右边列表框内为所选对象的属性、事件、函数、变量、结构等。如果列表框内某一项内

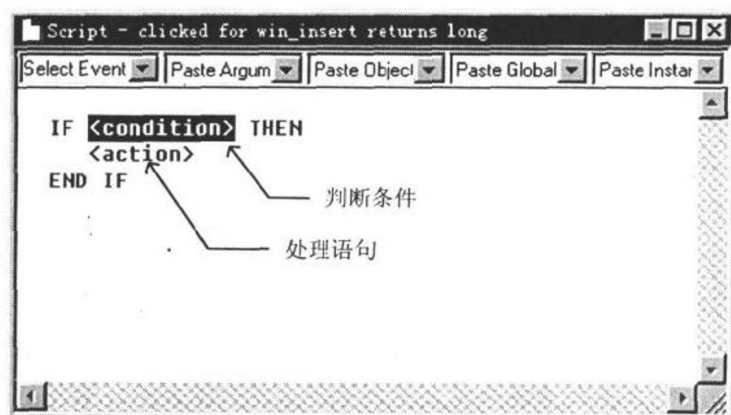
还有内容，可以双击该项，打开该项下一级的内容，选择要粘贴的项，点取 Paste 按钮，要粘贴的内容就会被粘贴到了光标所在处。

3.1.2.6 粘贴语句

利用 PowerBuilder 粘贴语句功能可以粘贴 PowerScript 语句的结构（如 IF...THEN, DO...LOOP, FOR...NEXT, CHOOSE...CASE）。单击 PainterBar 上的粘贴语句按钮  图标，粘贴语句对话框打开，见下图：

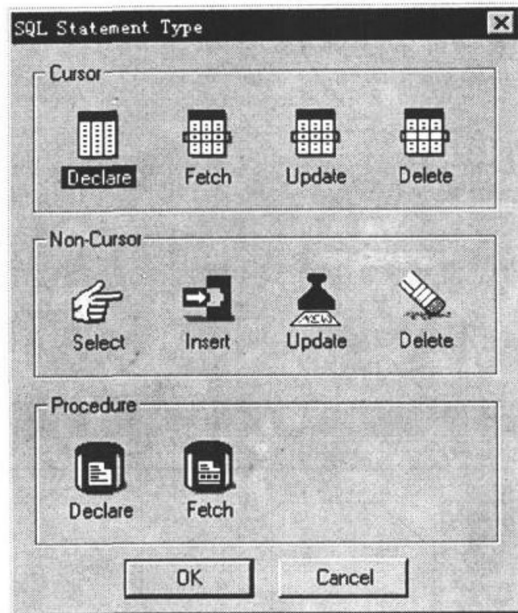


用鼠标选择要粘贴的语句，然后点取 OK 按钮，选中的语句就会被粘贴到光标所在处。如下图：



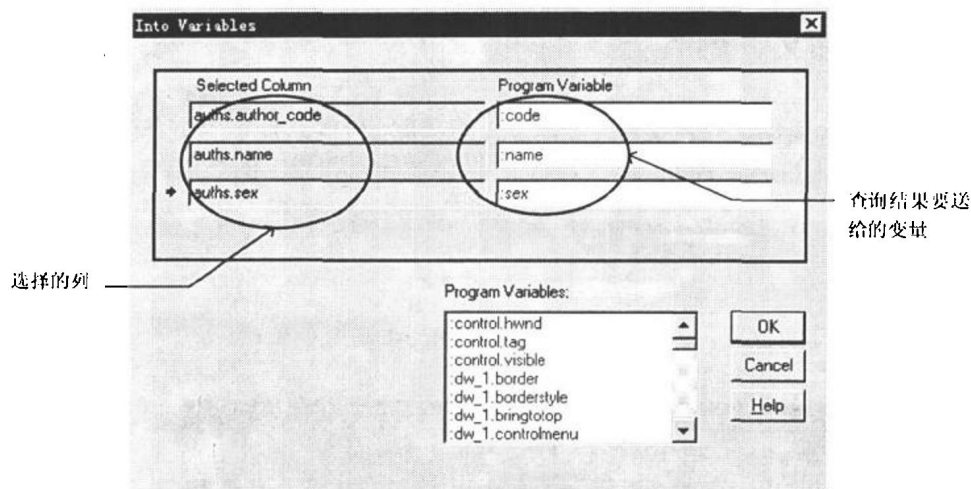
3.1.2.7 粘贴 SQL 语句

- 1) 单击 PainterBar 中的粘贴 SQL 语句按钮  图标，弹出选择 SQL 语句类型的对话框，见下图：



此窗口共有三栏：Cursor 光标栏，Non-CursorSQL 非光标 SQL 语句栏，Procedure 过程栏。

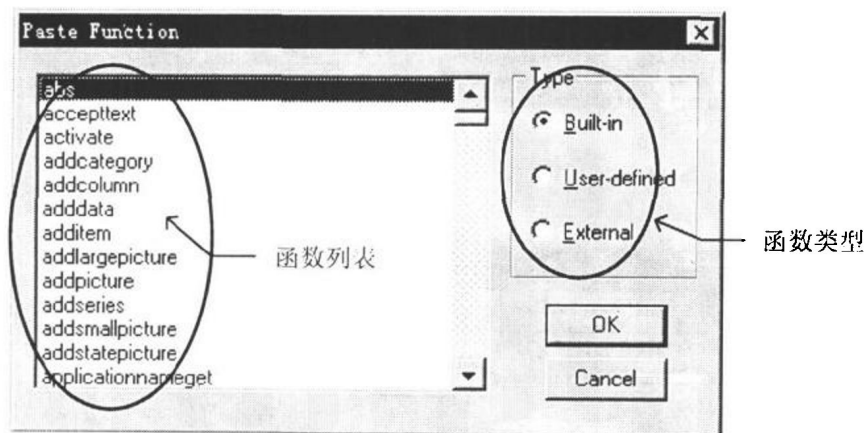
- 2) 下面我们以 Non-Cursor 中的 Select 语句为例讲述粘贴 SQL 语句。先选择 Non-Cursor 中的 Select，单击 OK 按钮，弹出 Select Table 对话框：



在窗口右边的 Program Variable 录入框内输入查询结果要送给的变量，单击 OK 按钮，然后退出定义 Select 语句的窗口，定义好的 Select 语句就被粘贴到光标所在处。

3.1.2.8 粘贴函数

点取 PainterBar 中的  图标，弹出粘贴函数对话框，见下图：



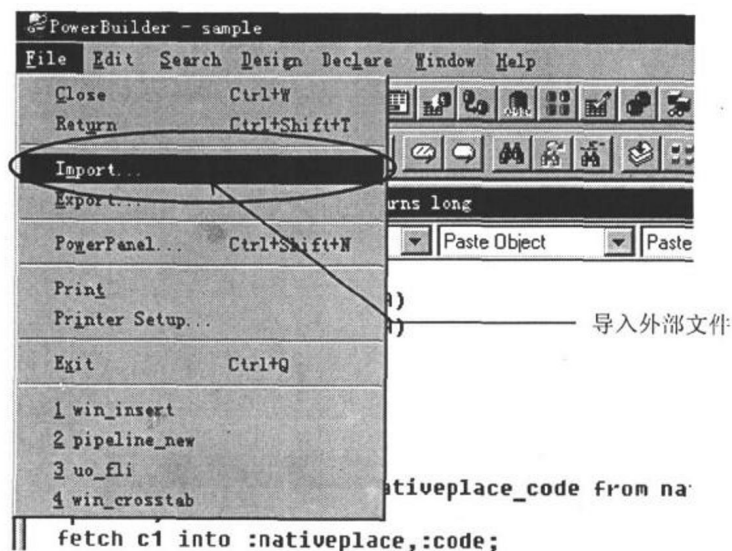
右上角的 Type 框内是函数的类型：(Built-in)系统内部的、(User-defined)用户自定义的和(External)外部的。首先选择函数的类型，然后选择左边列表框中的函数，单击 OK 按钮，所选的函数就被粘贴到光标所在处了。

3.1.2.9 粘贴文件内容

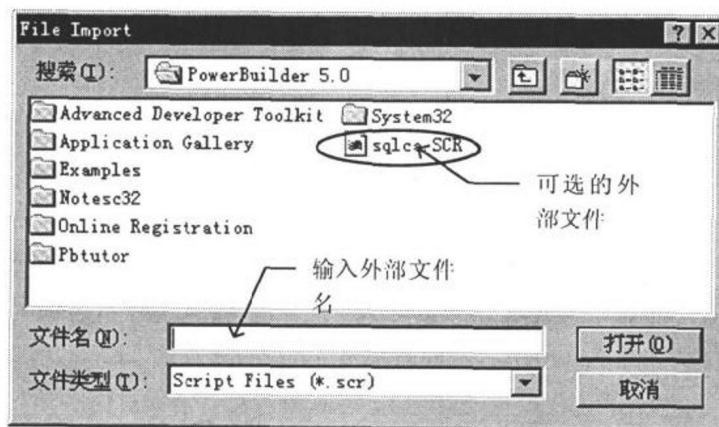
PowerBuilder 提供了导入、导出外部文件的功能，外部文件一般指存放 PowerScript 程序的文件，默认的扩展名为 .scr。下面分别讲述导入和导出的过程。

1. 导入文件：

1) 选择导入文件的位置，点取菜单项 File->Import，见下图：



2) 弹出选择外部文件对话框：

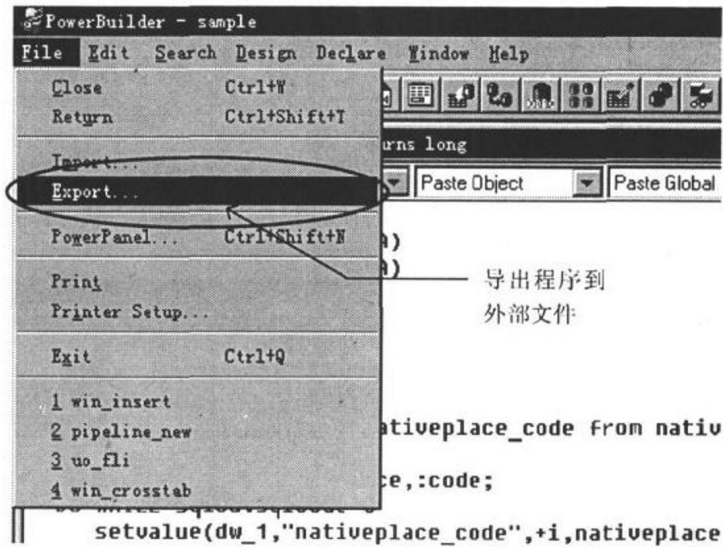


在“文件名(N)”录入框内输入一个已存在的外部文件，单击“打开(O)”按钮，外

部文件的内容被导入到了光标所在处。

2. 导出文件:

- 1) 用鼠标选中准备导出的程序段（如果不选将导出当前事件下的所有程序），点取菜单项 File->Export，见下图:



- 2) 弹出保存外部文件对话框:



在“文件名(N)”录入框内输入要保存的文件名，单击“保存(S)”按钮，所选的程序段被保存到外部文件中。

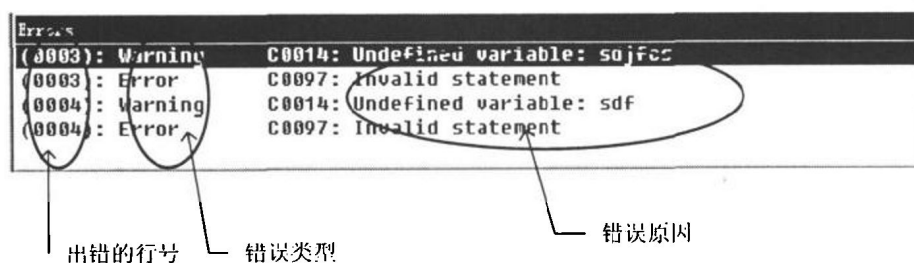
3.1.3 编译程序、错误处理

1. 编译程序

事件处理程序编写完以后，单击 PainterBar 中的编译按钮  图标，系统会对当前的程序段进行编译（单击 PainterBar 中的退出按钮  图标，在退出编程之前系统也会对当前的程序段进行编译）。

2. 错误处理

编译时如果程序有错，系统会把错误信息显示在屏幕上，错误信息有两种：Warning 和 Error。Warning 是警告错误，Error 是语法错误，见下图：



窗口右边显示的是错误的原因，开发人员可以根据错误原因修改程序。

3.1.4 在对象上封装函数和结构

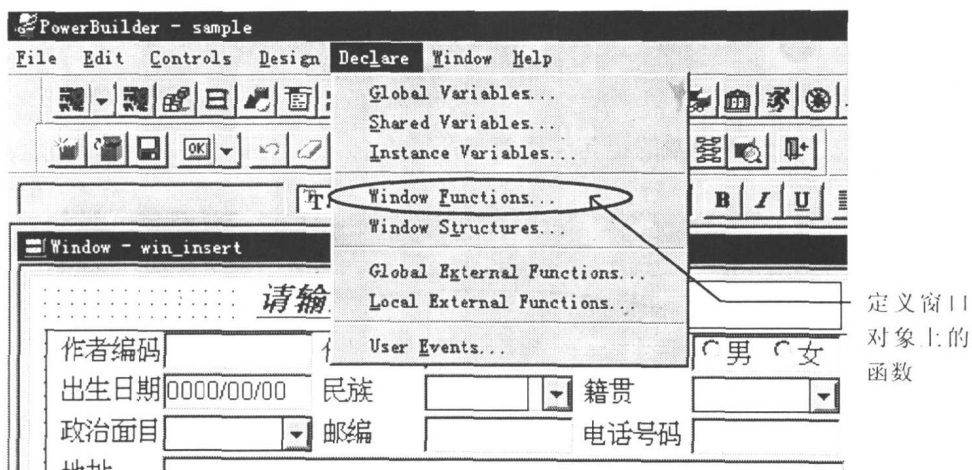
PowerBuilder 中的函数和结构与 C 语言的函数和结构类似。PowerBuilder 中可以定义全局的函数和结构，在编程时随时调用；也可以把函数和结构封装在对象上，仅在此对象中调用。

3.1.4.1 在对象上定义函数和结构

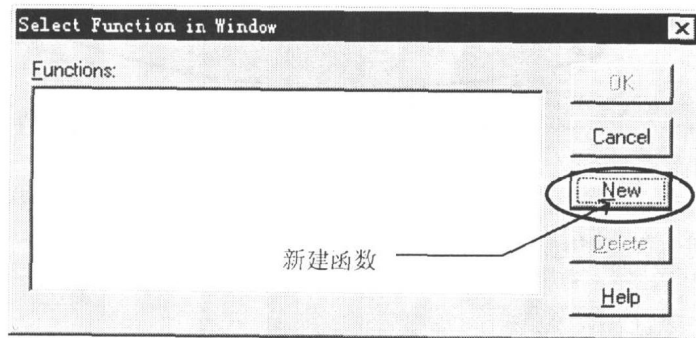
在窗口和用户对象上都可定义函数和结构，下面我们以窗口为例说明在对象上定义函数和结构的方法。

1. 在对象上定义函数

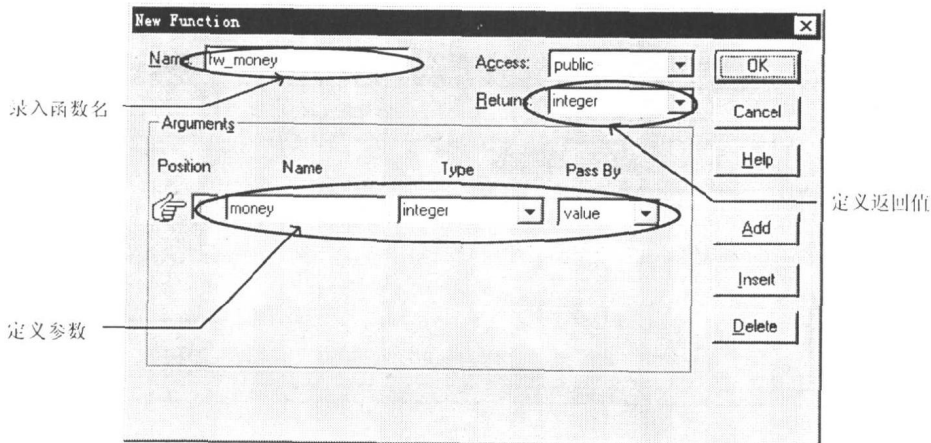
1) 打开下拉菜单 Declare，如下图：



2) 点取上图下拉菜单中的 Window Function 菜单项，弹出选择函数对话框：

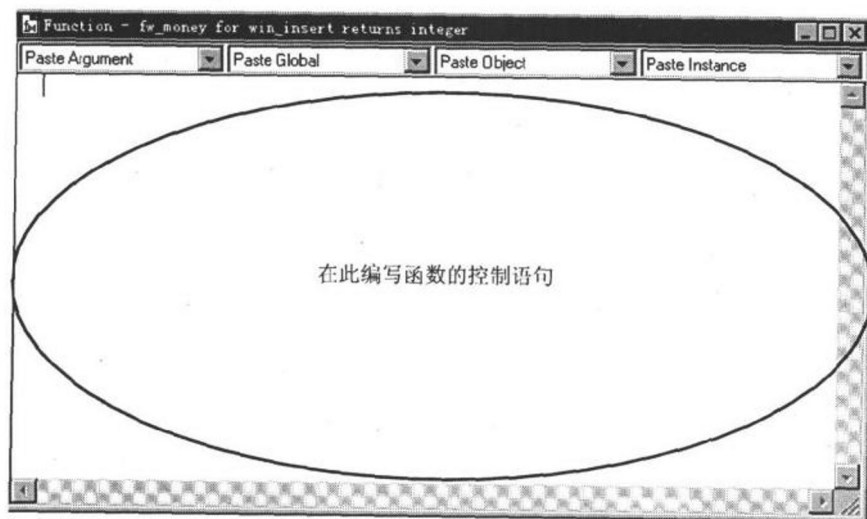


3) 点取 New 按钮，弹出定义函数对话框：



在 Name 录入框内录入函数名, 在 Arguments 框内添加参数, Return 下拉列表框选择返回值类型。

- 4) 点取 OK 按钮, 进入函数的语句编辑窗口:



- 5) 在窗口的编程工作区编写函数的控制语句, 结束后点取 PainterBar 中的  图标按钮返回。详细的函数定义过程请参见用户自定义函数部分。

2. 在对象上定义结构

- 1) 与函数的定义相同, 在菜单上点取菜单项 Declare->Window Struct, 弹出选择对象级结构对话框:

