

POWERBUILDER

Client/Server

Internet/Intranet

应用开发用户手册 卷 II

晓通网络数据库研究所

第十三章 PowerBuilder 应用的 调试/运行/编译/发行

13.1 PowerBuilder 应用的调试/运行

调试与运行是应用开发过程中必不可少的步骤，在应用开发过程中，必须经过调试和运行，改正错误，验证了它的正确性后，方可交付用户使用。PowerBuilder 提供了 Debug（调试）画笔和 Run（运行）画笔作为其调试/运行工具。

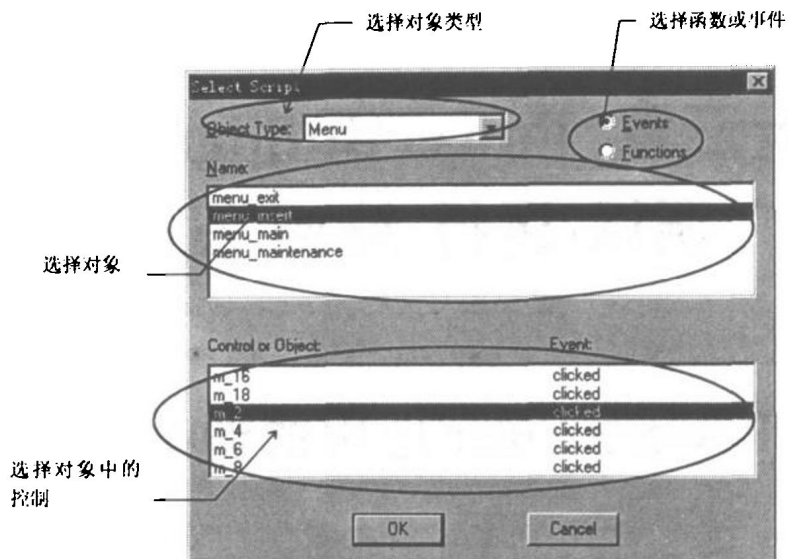
13.1.1 PowerBuilder 的 Debug 画笔

调试 PowerBuilder 应用时，可使用 PowerBar 中的 Debug 画笔  设置断点、单步执行、显示变量值来协助程序员判断和查找错误。其使用方法在后面的各节讲述。

13.1.2 使用 Debug 画笔跟踪执行

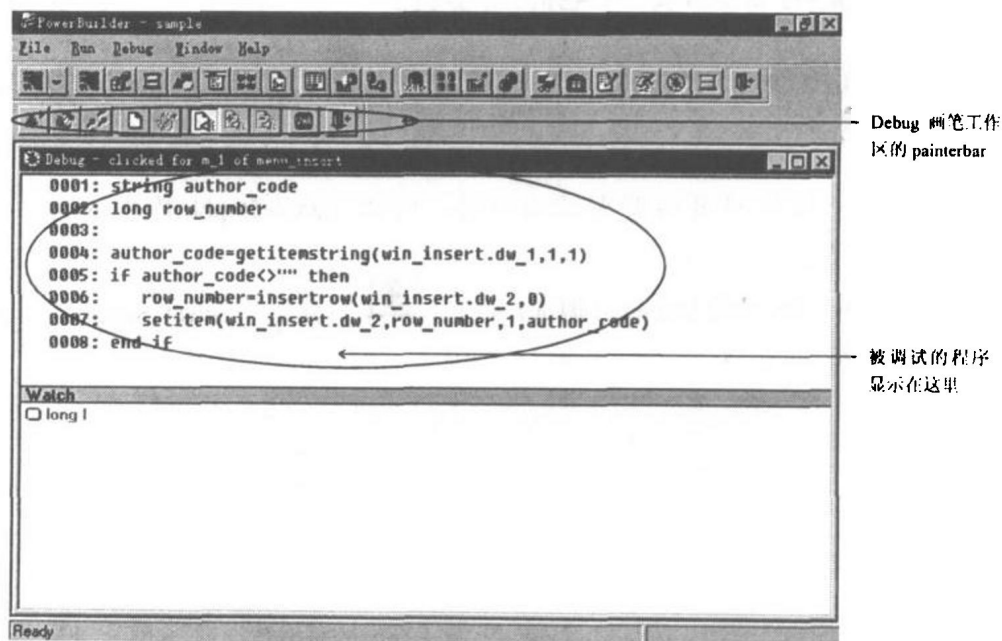
下面介绍用 PowerBuilder 的 Debug 画笔跟踪程序执行并获取跟踪信息的方法。步骤如下：

- 1) 在应用对象中打开需要调试的应用，例如 sample50.pbl 库中的菜单对象 menu_insert。
- 2) 单击 PowerBar 中的 Debug（调试）画笔 ，弹出“Select Script”对话框，如下图所示：



在此对话框中，可选取需要调试的对象或事件中的程序段。

- 3) 在 Object type 下拉列表框中选择对象类型。
- 4) 在 “ Name ” 区中选择对象。
- 5) 在 “ Control or Object ” 区中选择对象的控制及事件（该控制或对象中包含需要设置断点的程序，或对象上定义的函数）。
- 6) 单击 “ OK ” 按钮，打开调试工作区，如下图：



在 Debug 工作区中，PainterBar 把调试画笔菜单中的常用功能设成图标，排成一排以便快速选择这些操作命令，其功能如下：



Start 开始执行应用。点取该按钮即可按调试模式执行当前应用。



Continue 继续执行。当程序遇到断点暂停执行时，点取该按钮即可使程序继续执行直到遇到下一断点。



Step 单步执行。点取该按钮即可使程序单步执行。



Select Script 选择要调试的程序。在 Debug 工作区中选择要调试的程序。



Edit Stop 编辑断点。用此按钮可以显示出程序中设置的所有断点，并根据需要增加或删除断点。



Watch Window 打开或关闭变量观察窗（按下表示打开，弹起表示关闭）。



Add Watch 向变量观察窗中添加变量



Remove Watch 从变量观察窗中移出变量。

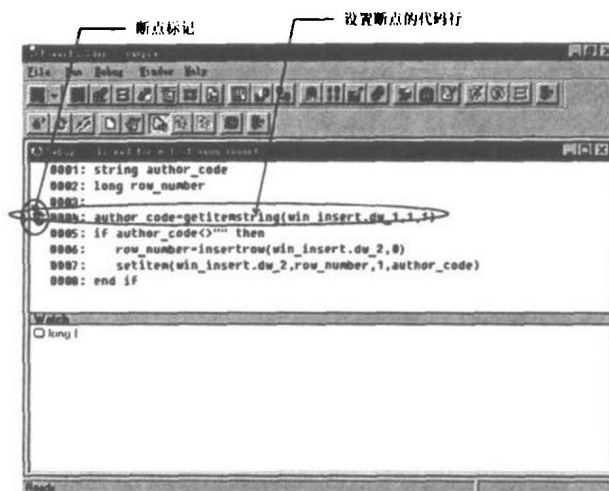


Variable Window 打开或关闭变量窗（按下表示打开，弹起表示关闭）。



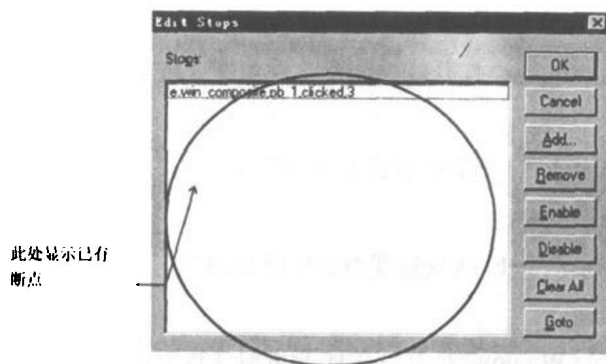
Close 关闭 Debug（调试）画笔。

- 7) 设置断点。在准备设断点的程序代码行上双击鼠标，该行即成为程序中的一个断点，并在工作区中增加一个断点标记，如下图：

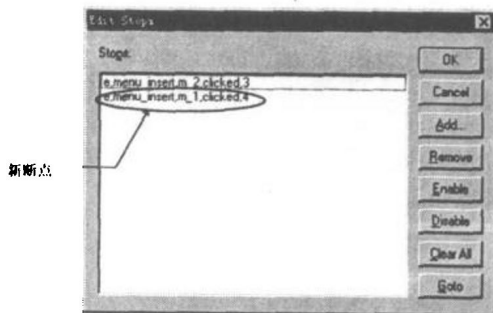


- 8) 断点的增减。在调试程序时，可能需要在不同的地方设置断点，有时又需要消除原来的断点，在应用程序中增加新的断点或消除原有断点称为对断点的编辑。

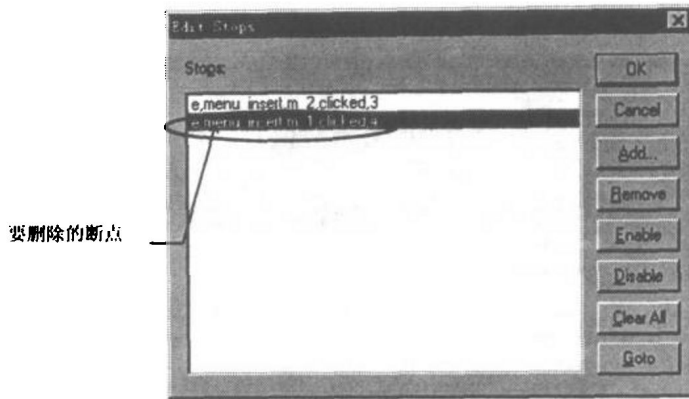
单击 PainterBar 中的 Edit Stop  图标，显示 Edit Stop 对话框如下：



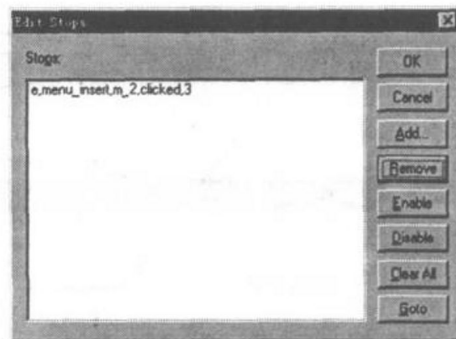
- a) 增加新断点：点取 Add 按钮。重复前面的步骤设置新的断点。例如，下面是新增加的断点：



- b) 消除断点：选择断点后也可消除断点，方法是选取要消除的断点，点取 **Remove** 按钮。例如：要消除图中的断点，首先选取要消除的断点，如下图：



点取 **Remove** 按钮，断点即被消除，如下图：

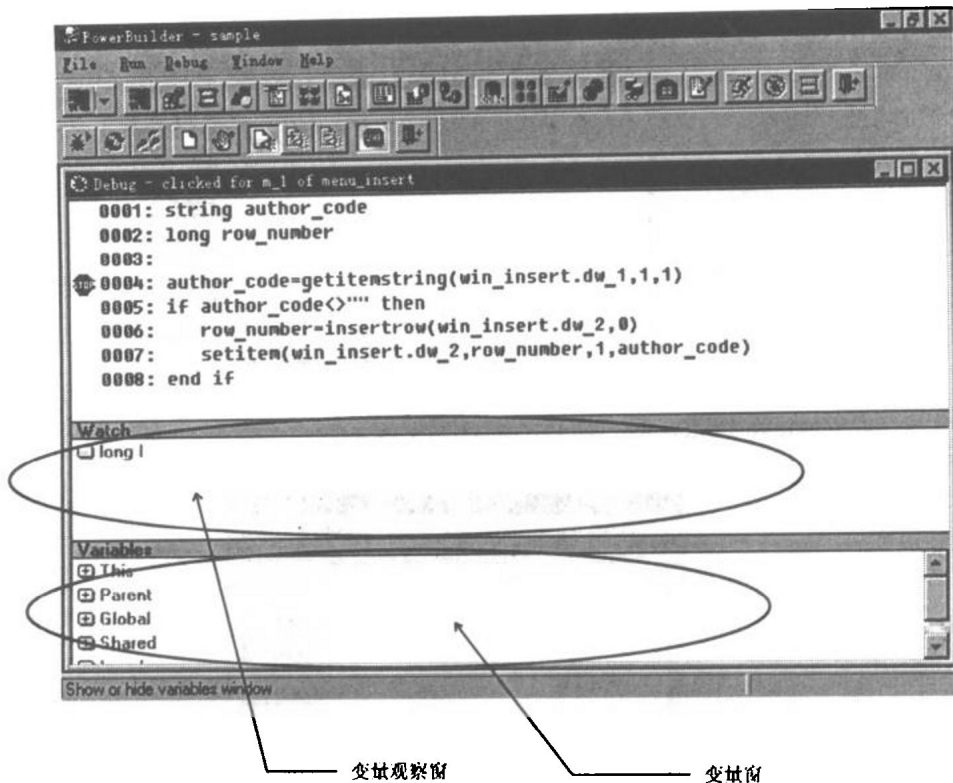


- c) 全部消除：点取 **Clear all** 按钮可消除所有断点。

- 9) 以跟踪模式运行。在 **Debug**（调试）画笔中，可以运行应用程序。此时，应用程序以跟踪模式运行（连续执行遇断点暂停或单步执行）。
- 单击 **PainterBar** 中的 **Start** 按钮  应用开始执行，并在第一个断点处停止。
 - 程序暂停后，单击 **PainterBar** 中的 **Continue** 按钮  应用继续执行，并在下一个断点处停止。
 - 单击 **PainterBar** 中的 **Step** 按钮  应用单步执行。

13.1.3 跟踪信息的获得

在 Debug（调试）画笔工作区中，可以显示调试程序所需的跟踪信息，程序员可以根据这些跟踪信息检查错误。



采用下述方法，可在跟踪过程中显示程序执行的位置及各变量、属性的值，并可据此判断、查找应用中的错误。

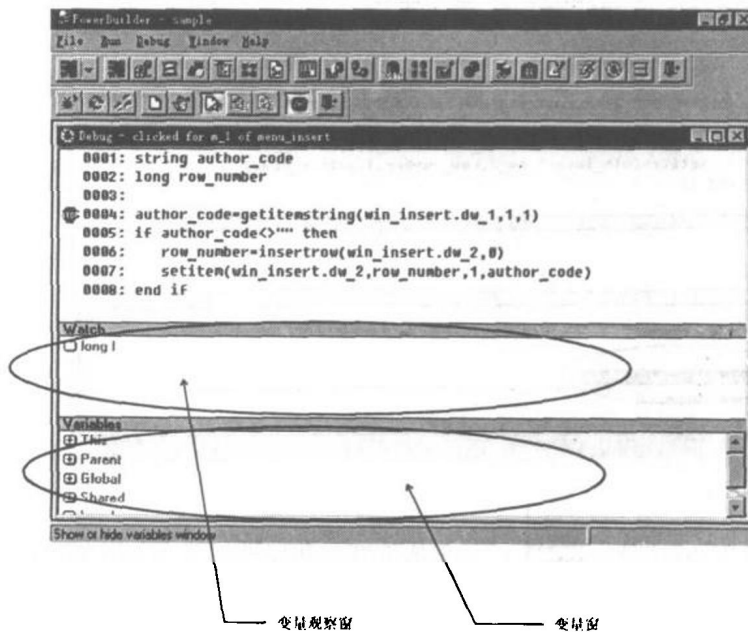
1. 程序执行位置

Debug（调试）画笔用高亮度条显示程序执行的位置。当程序以跟踪模式运行时，在遇到断点或以单步模式执行的时候，Debug（调试）画笔工作区中会以高亮度条表示程序执行的当前位置，如下图所示：

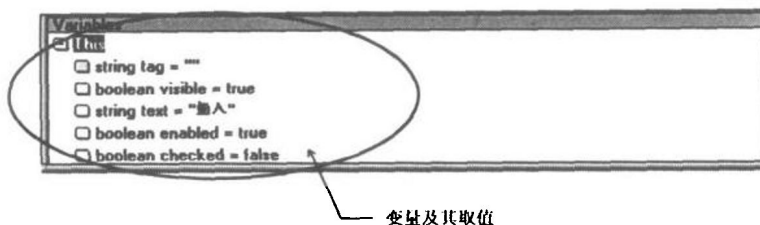


2. 显示变量和属性值

在 Debug（调试）画笔工作区的下端，有两个观察窗用于观察应用变量的取值，一个是变量窗（Variables），在这里可以观察到所有的变量及其取值，另一个是变量观察窗（Watch），可以在这里显示指定的变量及其取值，如下图所示：



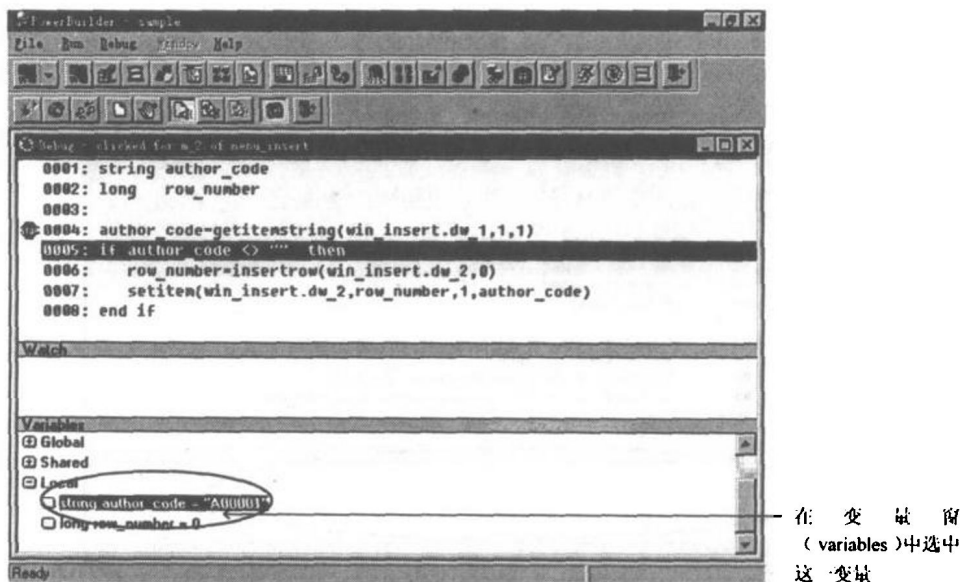
- 单击 PainterBar 中的 var (变量) 图标  ,可以打开或关闭变量窗 (Variables) (按下表示打开,弹起表示关闭)。在变量窗 (Variables) 中双击小标记 “+” ,即可将这类变量展开:



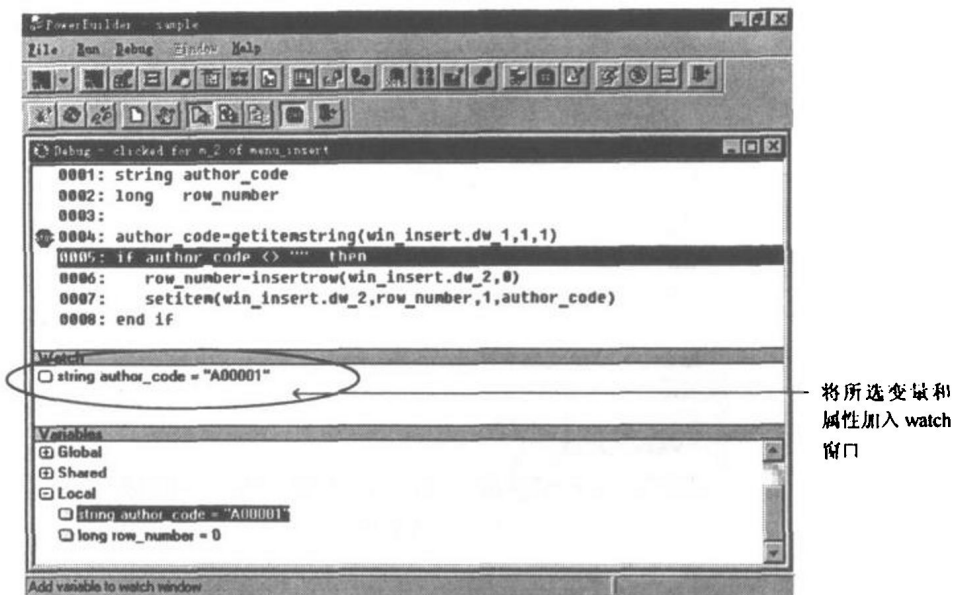
- 单击 PainterBar 中的 Watch Window 图标  , 打开或关闭变量观察窗(Watch) (按下表示打开,弹起表示关闭)。

在变量观察窗中可以指定需要观察的变量, 方法如下:

- 1) 在变量窗 (Variables) 中选中某一变量或某一类型变量, 如下图:



- 2) 单击 Add Watch 图标  , 可将所选变量和属性加入 Watch 窗口, 如下图:



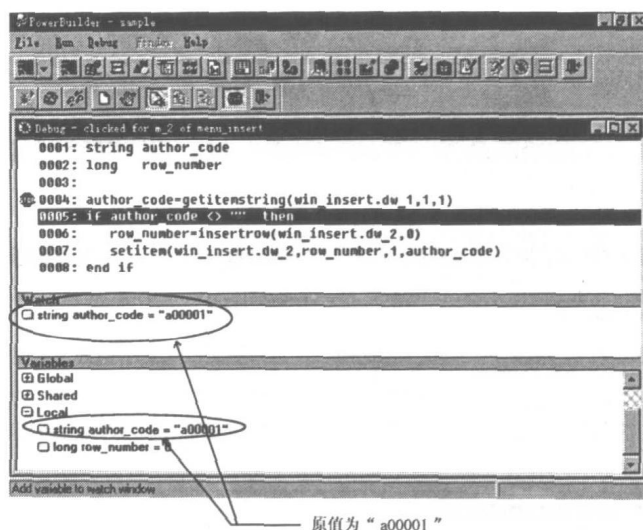
亦可在变量观察窗（ Watch ）中选中某一变量或某一类型变量，然后单击 Remove Watch 图标 ，将所选变量移出 Watch 窗口。

3. 跟踪时改变变量和属性值

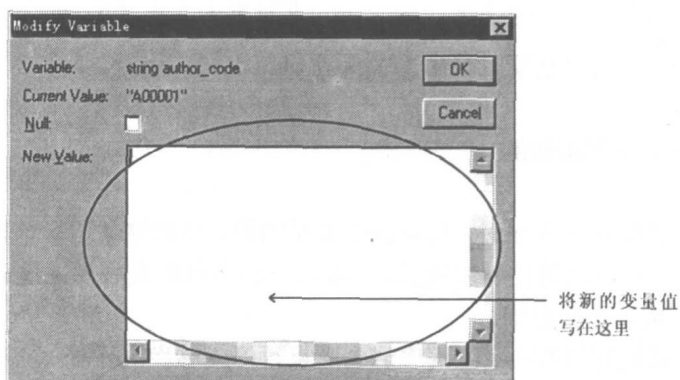
以跟踪模式运行应用程序时，可以改变变量的值，然后继续运行该程序。变量取不同的值时，程序会按不同的控制流执行，利用这一技术可以很容易地检查整个应用程序中的不同控制流执行是否正确。

在以跟踪模式运行过程中，当程序遇到断点暂停或单步执行时，可按下述方法改变变量的值。

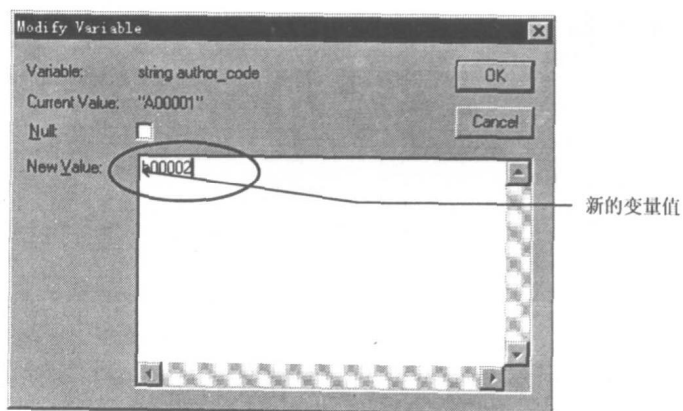
- 1) 在 Variables 窗口双击某一变量（下面以在调试过程中改变变量“ auths_code ”的取值为例），如下图：



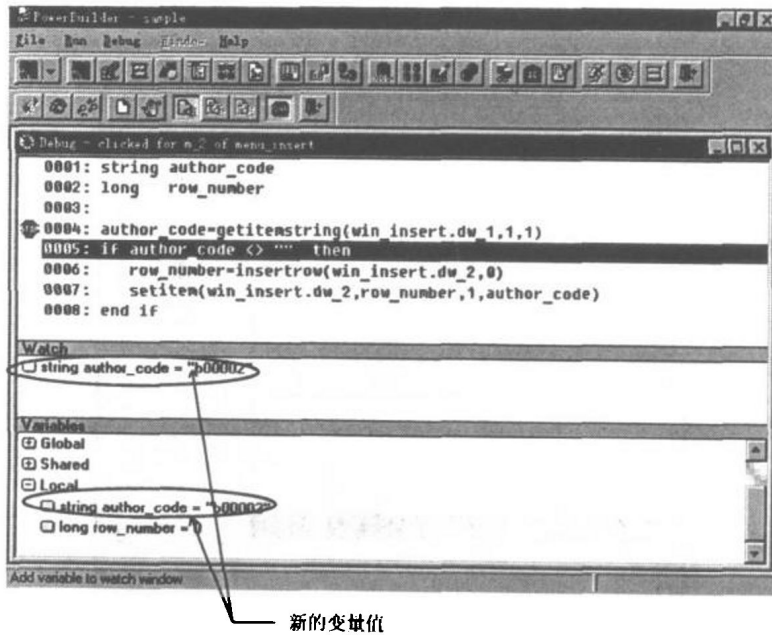
双击上图中的变量后显示如下对话框:



2) 将该变量新的值写入上图的对话框中, 如下图:



3) 单击“OK”按钮，返回调试工作区，变量已被换为新值，如下图：



在以后的跟踪执行过程中，将按该变量的新值执行。

13.1.4 PowerBuilder 应用的运行

PowerBuilder 提供了在开发环境下运行当前应用的工具—PowerBuilder 的 Run（运行）画笔。

在 PowerBar 中单击 Run（运行）画笔  即可完整地运行当前的应用。

13.2 PowerBuilder 应用的编译/发行

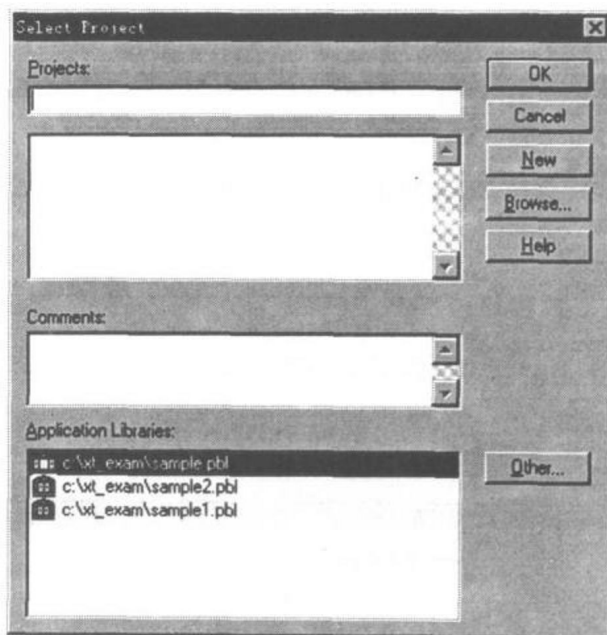
与其它软件一样，用 PowerBuilder 开发的应用软件也须经过编译才能生成可执行的应用软件。PowerBuilder 提供的编译工具可将应用软件编译成可直接在 Windows 下运行的应用软件，应用的编译是通过 Project(工程)画笔实现的。

13.2.1 PowerBuilder 的 Project(工程)画笔

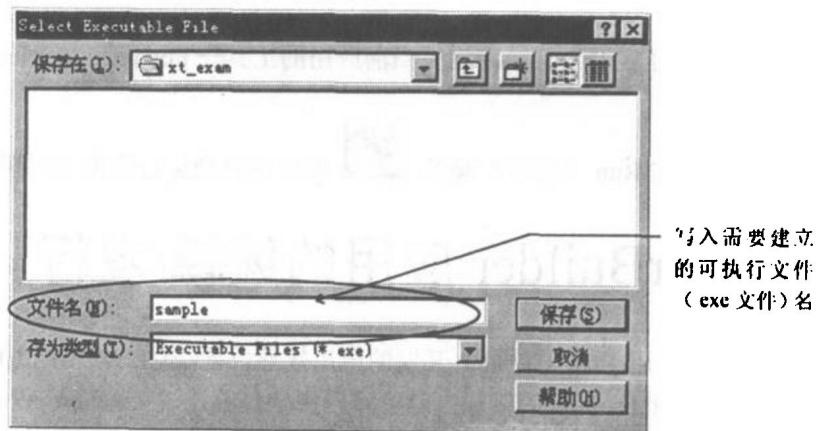
可以利用 PowerBuilder 的 Project(工程)画笔将 PowerBuilder 应用建成可直接在 Windows 下执行的应用软件。

进入 Project(工程)画笔工作区的步骤如下:

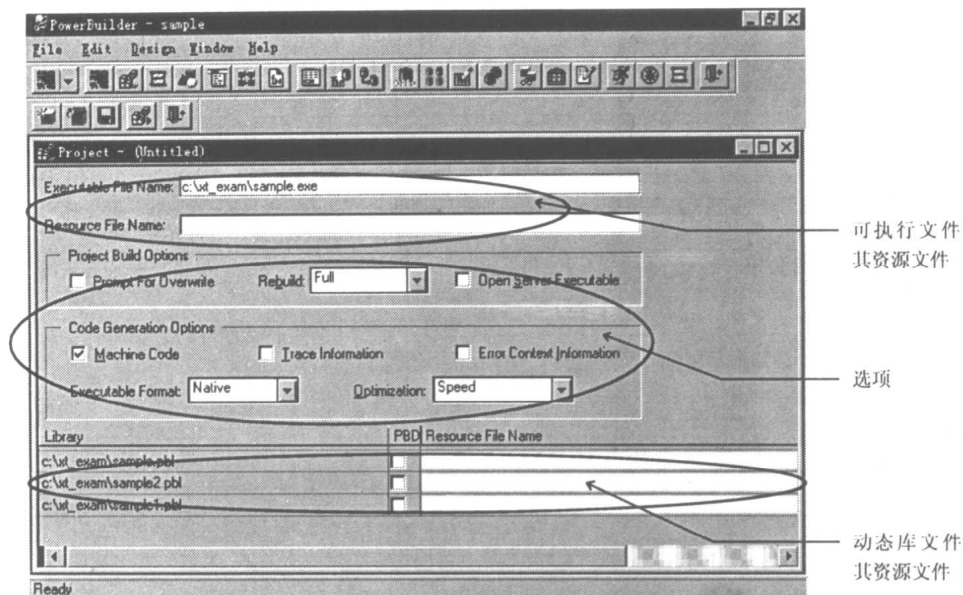
- 1) 单击 PowerBar 中的 Project (工程) 画笔  , 显示如下对话框:



- 2) 在上图的对话框中单击 “New” 按钮建立新的工程对象, 出现下面的对话框:

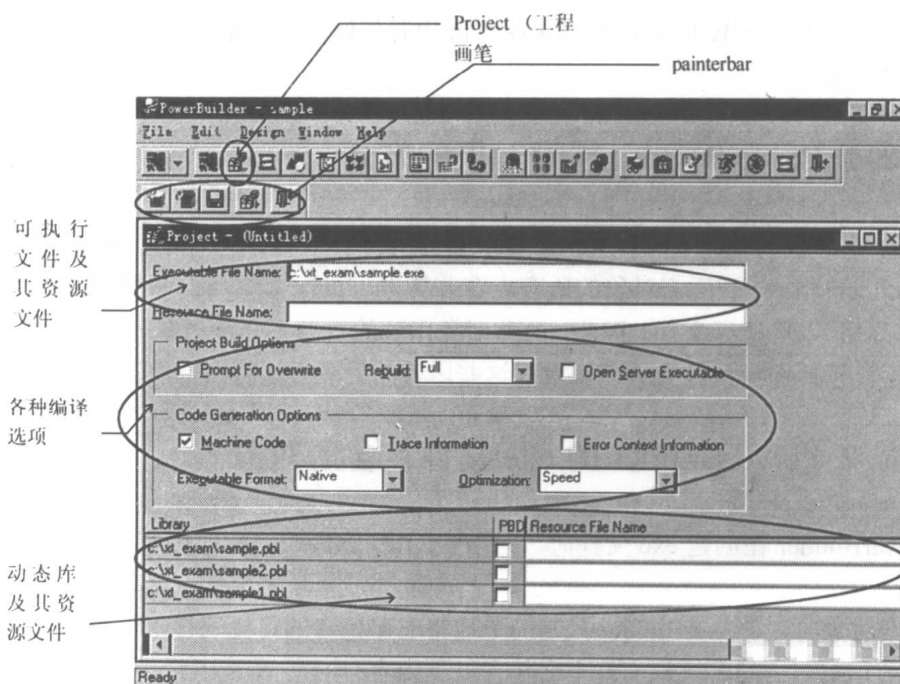


- 3) 在上图的对话框中写入需要建立的可执行文件名 (exe 文件), 然后单击 “保存” 按钮即可显示 Project (工程) 对象的画笔工作区, 如下图:



- 4) 在上图工作区中指定可执行文件（exe 文件）、动态库及其资源文件（pbr 文件）以及各编译选项（有关可执行文件、动态库及其资源文件将在后面详细介绍）。

下图是 Project(工程)画笔工作区：



在 Project（工程）画笔工作区中的 PainterBar 把工程画笔菜单中的常用功能设成图标，排成一排以便快速选择常用的操作命令，其功能如下：



New 建立一个新的工程对象。



Open 打开一个已有的工程对象。



Save 将建立的工程对象存盘。



Build 开始建立可执行应用。



Close 关闭 Project（工程）画笔工作区

建成的可执行文件（exe 文件）可在 windows 环境下直接执行。Project（工程）画笔的具体使用方法将在后面详细介绍。这里先介绍一下 PowerBuilder 可执行应用的构成及其建立方法。

13.2.2 PowerBuilder 可执行应用的建立

下面介绍 PowerBuilder 在建立可执行应用时如何搜索应用库中的对象和如何创建可执行的应用。

13.2.2.1 PowerBuilder 搜索对象的方式

为创建可执行文件，PowerBuilder 将搜索应用库中的对象。搜索对象时，PowerBuilder 无须找到所需对象的所有内容并拷入可执行文件，而只是简单地将对象包含的目标码移入可执行文件。因此，一些对象用到的资源不会包括在可执行文件中（哪些内容不会包括在可执行文件中以及对这些内容的处理方法将在下面详细介绍）。

13.2.2.2 PowerBuilder 的 exe 文件包含的对象

PowerBuilder 在创建 exe 文件时，将搜索库中的对象。

搜索对象时，PowerBuilder 只是简单地将对象包含的目标码移入可执行文件），因此，PowerBuilder 的可执行文件不会包含以下内容：

- 在程序中引用某一对象时,如果只是通过字符串变量而不是直接通过该对象的名字来引用,则该对象不会被写入可执行文件。例如:程序中用下列程序调用窗口对象 win_insert :

```

window win
string winname="win_insert"
open(win,winname)

```

因为窗口对象 win_insert 在程序中是通过字符串变量引用而不是直写其名的,所以不能被写入可执行文件。在这种情况下,即使建立的只是一个很小的应用程序,也必须建立一个动态库,并将窗口对象 win_insert 放入动态库中。

又如,程序中用下列语句引用数据窗口对象 dw_auths :

```

auths_datastore.DataObject="dw_auths"

```

因为数据窗口对象 dw_auths 以字符串形式引用,因此数据窗口对象 dw_auths 不能被写入可执行文件。在程序执行时无法找到数据窗口对象 dw_auths。在这种情况下,应为数据窗口对象 dw_auths 建立一个资源文件或将其放入动态库中。

- 图标、图象文件也不被写入可执行文件。例如:定义在菜单工具栏中的图标。在这种情况下,须为这些图标、图象等资源定义资源文件或将它们与可执行文件一起发行。

PowerBuilder 的可执行文件可以包含以下内容:

- 直接引用的对象(全局函数、窗口、菜单等)。例如,程序中用下列程序引用窗口对象 win_insert :

```

open(win_insert)

```

因为窗口对象 win_insert 在程序中是直写其名的,所以可以被写入可执行文件。

- 在画笔中引用的对象。例如,由于菜单是在窗口画笔中与窗口相连的,所以菜单将与相连窗口一起拷入可执行文件。

把 PowerBuilder 应用的所有内容都放入一个 exe 文件的方法只适用于较小的应用

系统，大型的 PowerBuilder 应用系统除可执行文件外往往还包括动态库（ pbd 文件）和一些外部资源。大型的 PowerBuilder 应用系统一般由三部分构成：

- 1. 可执行文件（ exe 文件）；
- 2. 动态库（ pbd 或 dll ）文件。这是 PowerBuilder 提供的一种动态调用的库文件，类似于 windows 中的动态连接库（ DLL 库）；
- 3. 外部资源文件。主要有图象、图标等资源文件。

这样，不被写入可执行文件的对象可采用下表列出的方法进行处理：

对象	与可执行文件一起发行（装入可搜索路径）	列入资源文件，编译时连入可执行文件或动态库文件中（ pbd 或 dll 文件）	直接放入动态库文件（ pbd 或 dll 文件）
资源（图形、图标等）	X	X	
数据窗口对象		X	X
其它对象（如通过字符串变量引用的窗口等）			X

由上表可以看出，对于不同类型的不能被写入可执行文件的对象，应采取不同的处理方法：

- 1. 外部资源：对于图形、图标等资源，可以选择下列两种方法之一：
 - 加入可搜索路径单独分发。采用这种方式在发行可执行应用时必须将图形、图标等文件一起发行，并将这些文件加入可搜索路径。
 - 在资源文件中列出它们。在建立可执行应用时， PowerBuilder 会将它们连入并包含在可执行应用中，在发行应用时不必再发行这些图形、图标等文件。
- 2. datawindow （数据窗口）对象的处理：对于通过字符串变量引用的数据窗口对象，可以采用下列两种方法之一进行处理：
 - 在资源文件中列出它们，在建立可执行应用时， PowerBuilder 会将它们连入可执行应用中。