

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

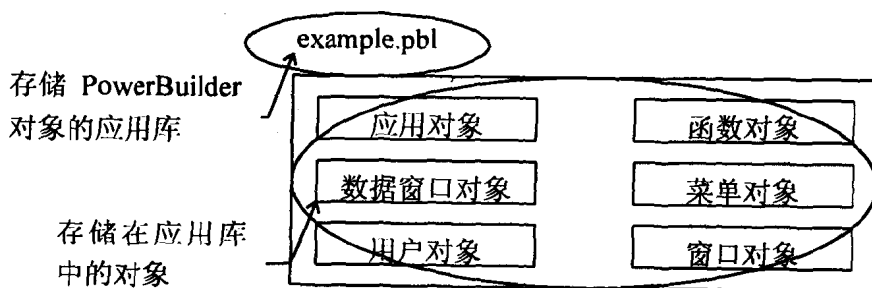
第四章 PowerBuilder 的 Application（应用）对象

4.1 概述

应用也称应用程序。一般来讲，一个应用完成一组相关的功能。在 PowerBuilder 中，应用是由一系列 PowerBuilder 对象组成。这些对象包括：PowerBuilder 的应用对象（Application Object），窗口对象（Window Object），数据窗口对象（DataWindow Object），菜单对象（Menu Object），函数对象（Function Object）以及用户对象（User Object）等。

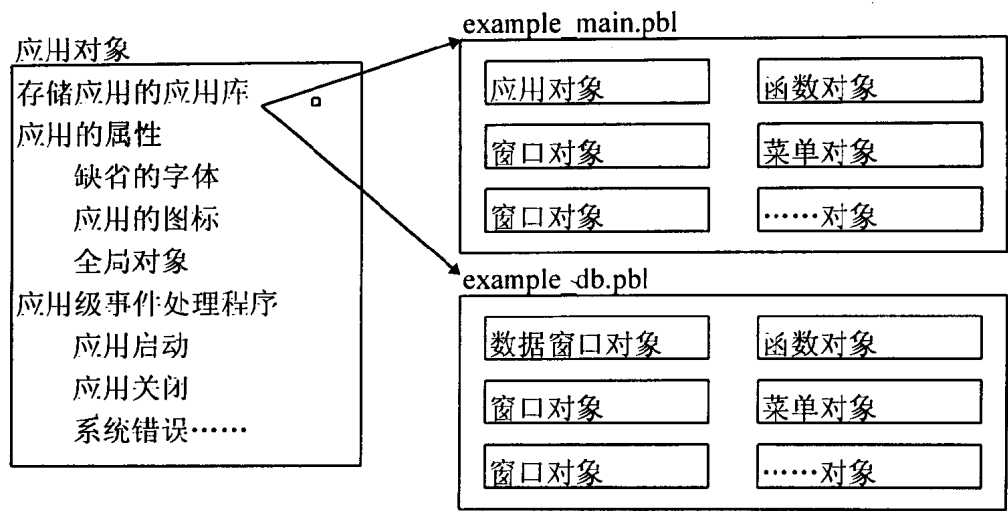
应用库是 PowerBuilder 提供的一种用于存储 PowerBuilder 应用的文件。这种文件以 .PBL 作为后缀（其含义是 PowerBuilder Library）。这种文件指示应用存放的位置，存储开发的内容（如应用对象、窗口、数据窗口、菜单、函数、用户对象等等）。这样，当用户完成了应用开发后，所有应用开发的内容就存放在这样的库文件中了。一个完整的应用既可以存放在一个库文件中，又可以存放在多个库文件中。

应用对象是用以标识应用，进入应用的入口点。它是 PowerBuilder 应用中包含的众多对象之一，同窗口对象、菜单对象、函数对象或数据窗口对象一样存储在应用库中。

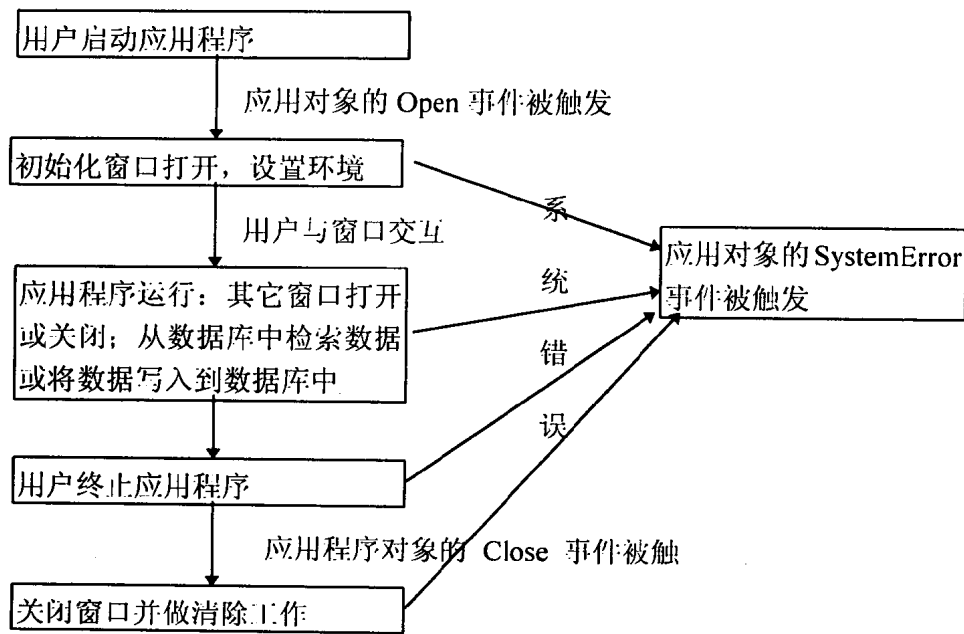


应用对象是进入应用的入口点，它定义了应用应用级的行为。它可以定义存储应用的应用库，在缺省状态下应用中文本的字体，以及应用开始、终止及出现系统错误时

应执行的操作，如下图所示：



PowerBuilder 为应用对象定义了七个可在特定情况下触发的事件，以控制应用在启动、终止及出现系统错误时应执行的操作。当用户启动应用程序时，应用对象中的 Open 事件会被触发，以打开应用的第一个窗口并对系统进行一定的设置；当应用终止时，应用对象的 Close 事件会被触发，以完成系统必需的一些清除工作，如关闭数据库或重写预定义文件；如果应用在执行期间发生系统错误，应用对象的 SystemError 事件会被触发，以向用户显示出错信息并按应用的要求跳转到特定的位置。如下图所示：



下面逐一讲述应用对象的建立,应用属性的设置和应用级事件的作用及处理程序的写法。

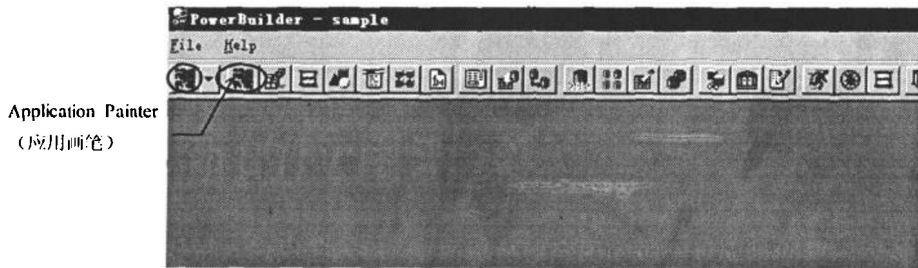
4.2 创建新的应用对象

因为应用对象是进入应用的入口点,所以创建应用的第一步应是创建应用对象,同时创建存储应用对象的应用库。

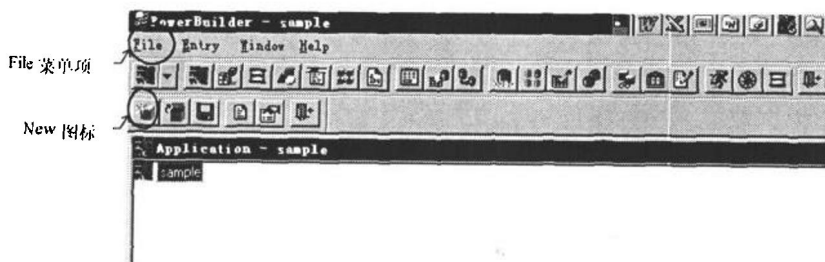
4.2.1 创建新的应用对象

创建新应用对象的具体步骤如下:

- 1) 在 PowerBuilder 主窗口中,单击 PowerBar 中 Application Painter (应用画笔) , 如下图所示:

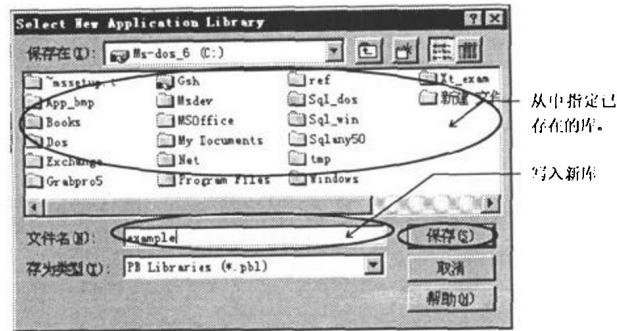


此时, 当前应用对象打开并显示在应用画笔工作区:

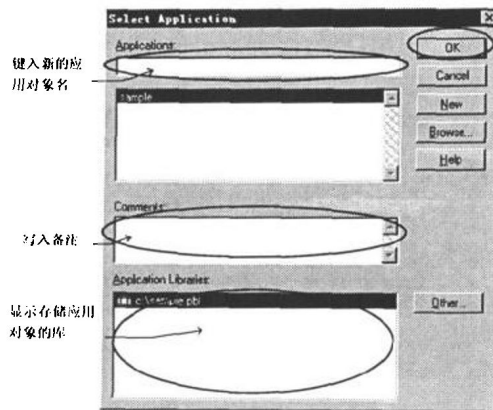


- 2) 点取上图 PainterBar 中的 New (新建)  图标或选择菜单栏的 File-->New, 此

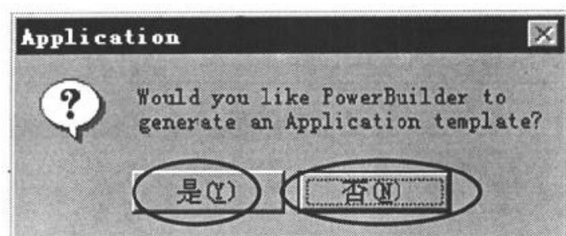
时出现下面的“Select New Application Library”（选择新的应用库）对话框，如下图所示：



- 3) 指定存储应用对象的应用库（应用本身可使用多个库，这儿指定的库是存储应用对象的应用库）。在“文件名（N）”框中，键入新应用库名。
- 4) 单击上图对话框中的“保存（S）”按钮，出现下面的“Select Application”（选择应用对象）对话框，如下图所示：

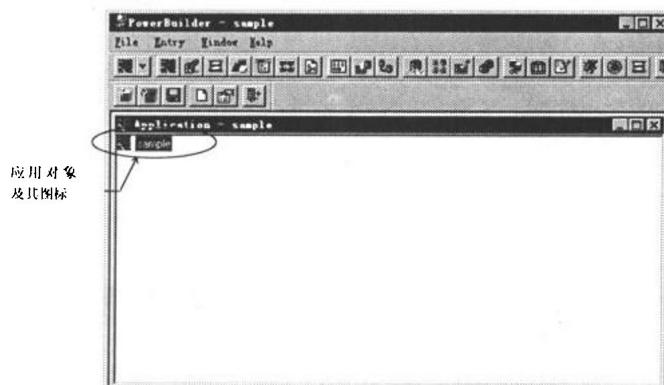


- 5) 在上图的 Application 框中写入应用对象名，可键入 1~40 个 PowerBuilder 有效字符。
- 6) （任选的）在上图的备注框里，可以写入有关应用的说明；
- 7) 单击上图的 OK 按钮，显示应用画笔工作区并显示 Application 对话框，询问是否要套用模板建立应用程序，如下图所示：



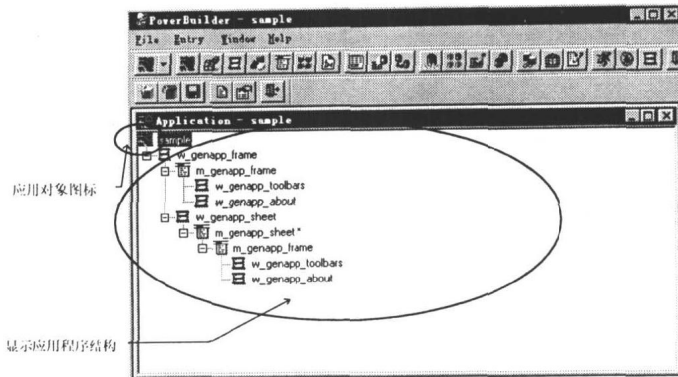
- 8) 情况一：单击框中的“否（N）”按钮，则不套用模板建立应用程序，而是从头开始创建。

新建的应用对象便显示在应用对象工作区，如下图所示：



情况二：单击框中的“是（Y）”按钮，则利用模板生成应用程序。也就是说，新创建的应用按照系统给定的模式生成，即 PowerBuilder 产生一个包含 MDI 框架（具有打开或关闭 MDI 子窗口等的完整窗口功能）、MDI 子窗口、About 对话框、菜单、工具条和事件处理程序的基本 MDI 应用的框架。

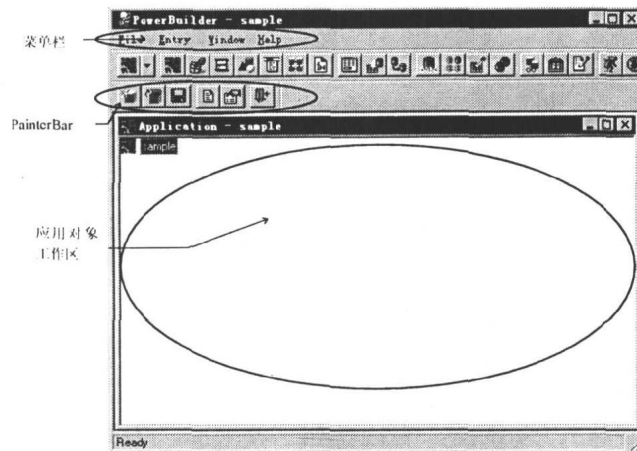
对于熟练的开发人员来说，在此框架基础上，只要根据需要对其进行恰当的修改便可使用，而不必从头开始创建。但对于初学者来说，修改模板有较多的困难，不如从头开始创建。下图是利用模板生成的应用，并在应用对象工作区显示该应用的结构：



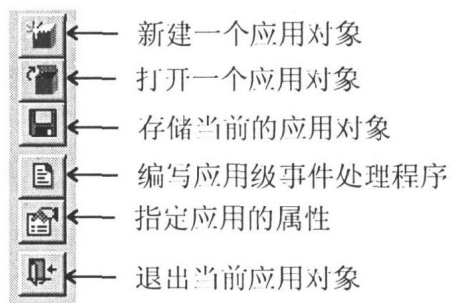
查看应用程序的结构、指定应用的属性及编写应用级处理程序都在应用画笔中实现的，下面介绍应用画笔。

4.2.2 关于应用画笔

应用画笔是由菜单栏、PainterBar 以及应用对象工作区组成，如下图所示：



下面主要介绍 PainterBar 的功能：



下面介绍如何在 Application Painter（应用画笔）工作区中查看应用的结构。

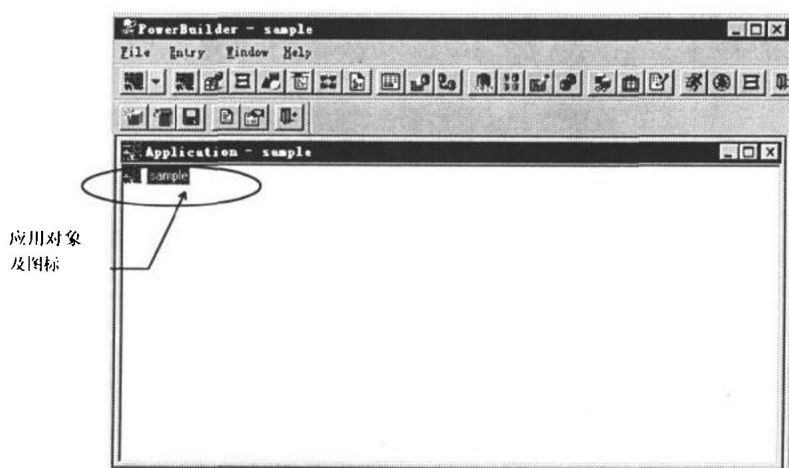
4.2.3 查看应用程序的结构

应用程序结构是指应用中各个对象之间的互相调用关系。在 Application Painter（应用画笔）工作区中可查看应用程序结构。显示应用程序结构的步骤如下：

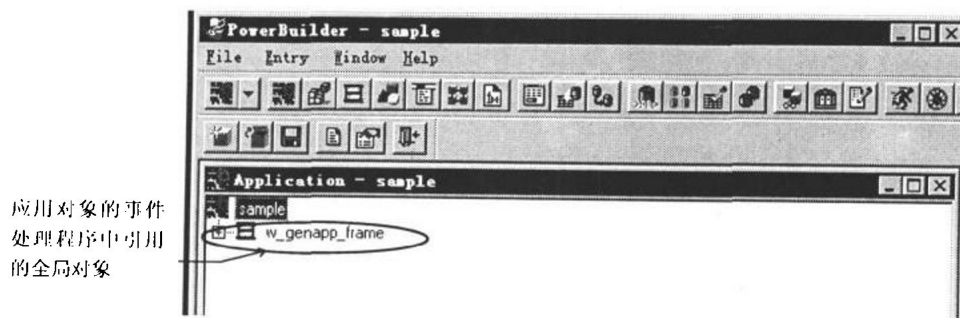
- 1) 在 PowerBuilder 主窗口中，单击 PowerBar 中 Application Painter（应用画笔）



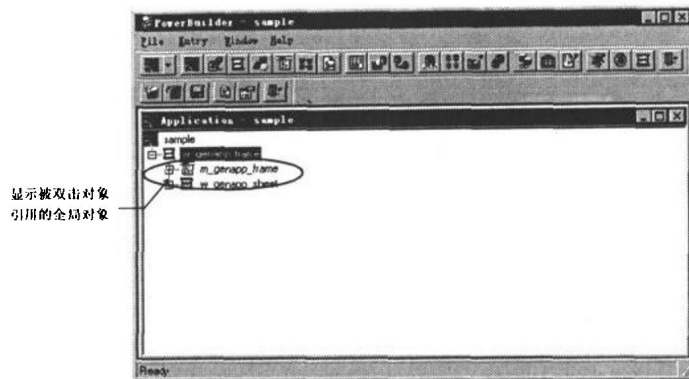
，应用对象作为一个图标显示在工作区，如下图所示：



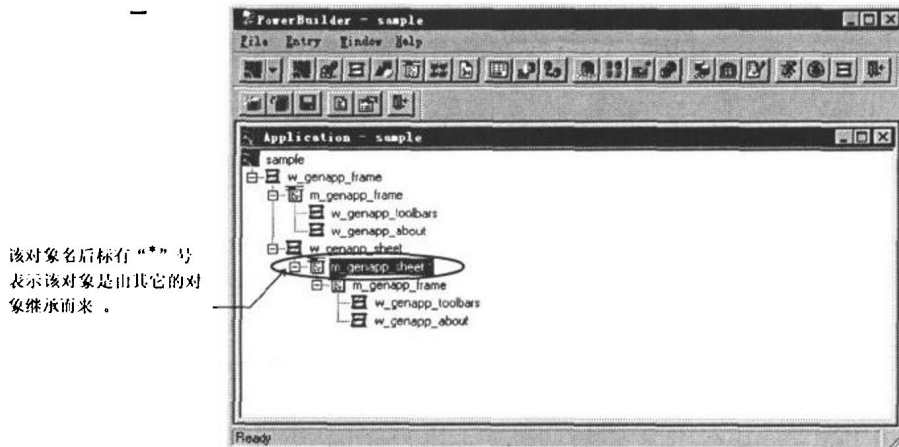
- 2) 双击上图的应用对象图标，应用对象工作区显示在应用对象的事件处理程序中引用的全局对象（仅指用 PowerBuilder 画笔创建的全局对象），如下图所示：



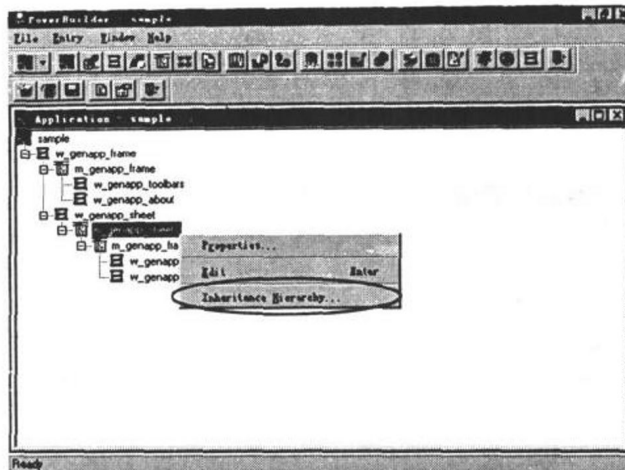
- 3) 如果要查看某一个全局对象所引用的对象，可双击该全局对象名或图标，或点“+”号，如下图所示：



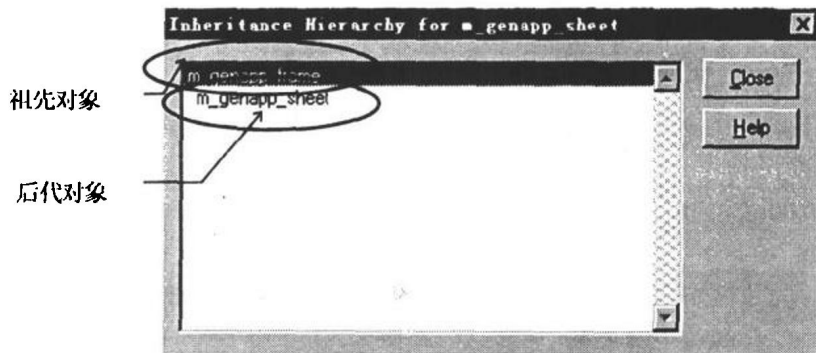
- 4) 如果要隐藏全局对象引用的对象，则双击该对象名或图标，或单击“-”号，则隐藏该全局对象引用的对象。
- 5) 若对象名后标有“*”号，表示此对象是从另一个对象继承而来的，如下图所示：



- 6) 如果要查看其继承结构，则将鼠标放在该对象上，单击鼠标右键，显示弹出菜单，从中选择 Inheritance Hierarchy（继承结构），如下图所示：



此时，PowerBuilder 弹出一个对话框来显示被选对象的继承结构，如下图所示：



以上讲述了在应用画笔工作区中如何查看应用中各个对象的调用情况。但是并非应用中引用的所有对象都可显示出来。下面详细讲述在应用画笔工作区的系统结构图中哪些对象是可显示出来而哪些对象是不可显示出来的。

4.2.3.1 可显示的对象

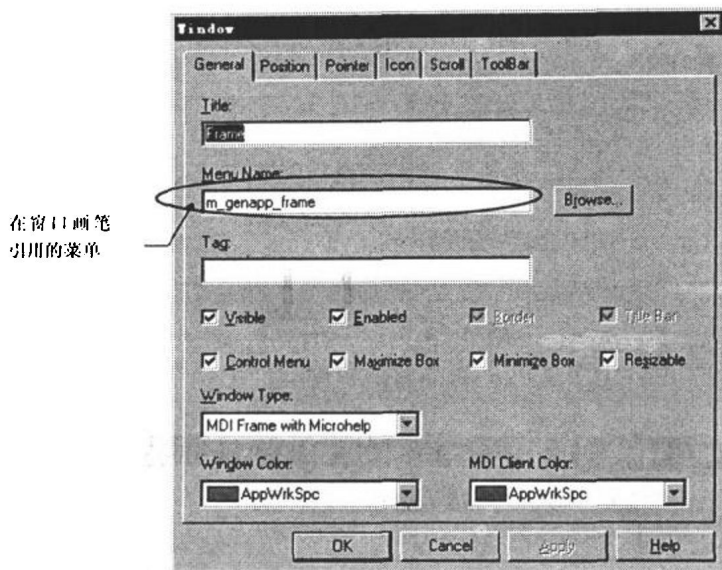
在应用画笔工作区的系统结构图中可以显示的对象有：

- 画笔中引用的对象
- 程序中直接引用的对象

下面举例介绍在应用对象工作区中可以显示的对象。

1、画笔中引用的对象

例1：如果在窗口画笔中把菜单与窗口相关联，则展开窗口时，菜单将出现。例如，如果在窗口画笔中将一个名为 `w_genapp_frame` 的窗口与一个定义好的名为 `m_genapp_frame` 的菜单相关联，如下图所示：



则当展开窗口 `w_genapp_frame` 时，菜单 `m_genapp_frame` 将出现。

例2：如果在窗口画笔中的 `DataWindow` 对象与 `DataWindow` 控制相关联，展开窗口时，则 `DataWindow` 对象出现。

例3：如果窗口包含定制用户对象，该定制用户对象又包含其它用户对象，展开窗口时定制用户对象出现，并且展开定制用户对象时其它用户对象出现。

2、程序中直接引用的对象

例1：如果某个窗口的事件处理程序中包含下面的语句：

```
Open(w_genapp_frame)
```

则当该窗口展开时，`w_genapp_frame` 会显示在应用对象工作区。

例2：如果某菜单项的事件处理程序中引用全局函数 `f_calc`：

```
f_calc(EnteredValue)
```

则当菜单展开时 `f_calc` 会显示在应用对象工作区。

例3：如果窗口通过以下语句使用弹出菜单：

```
m_new      mymenu
mymenu = create    m_new
mymenu.m_file.PopMenu(PointerX(),PointerY())
```

则当窗口展开时， `m_new` 将会显示在应用对象工作区。

4.2.3.2 不可显示的对象

在应用画笔工作区的系统结构图中不可显示的对象有：

- 只通过实例变量或属性引用的对象
- 通过字符串变量被动态引用的对象

下面举例介绍在应用对象工作区中不可以显示的对象

1、只通过实例变量或属性被引用的对象

例1：如果 `w_genapp_frame` 的事件处理程序中有如下语句(并且没有其它语句引用 `w_genapp_toolbars`)：

```
w_genapp_toolbars.title = “工具栏窗口 ”
```

则展开 `w_genapp_frame` 时，窗口 `w_genapp_toolbars` 不显示。

2、通过字符串变量被动态引用的对象

例1：如果窗口的事件处理程序有以下语句：

```
window mywin
string winname = “w_genapp_toolbars”
open(mywin,winname)
```

则展开该窗口时不显示窗口 `w_genapp_toolbars`，窗口只在字符串中被引用。

例2：如果 `DataWindow` 对象 `d_emp` 动态地通过以下的语句与 `DataWindow` 控制 `dw_info` 相关联：

```
dw_info.DataObject = "d_emp"
```

则当展开包含 `DataWindow` 控制 `dw_info` 的窗口时不显示 `d_emp`。

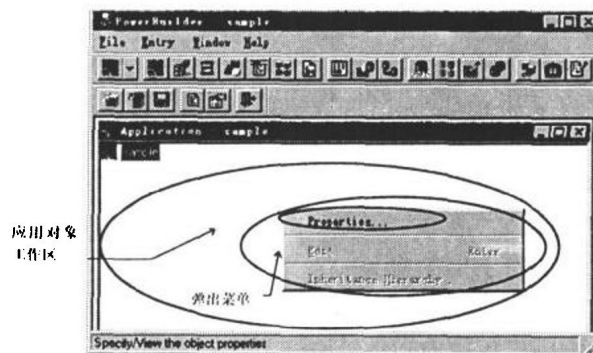
4.3 指定应用的属性

PowerBuilder 通过使用应用对象的属性设置对话框来指定应用的属性。可指定的属性如下：

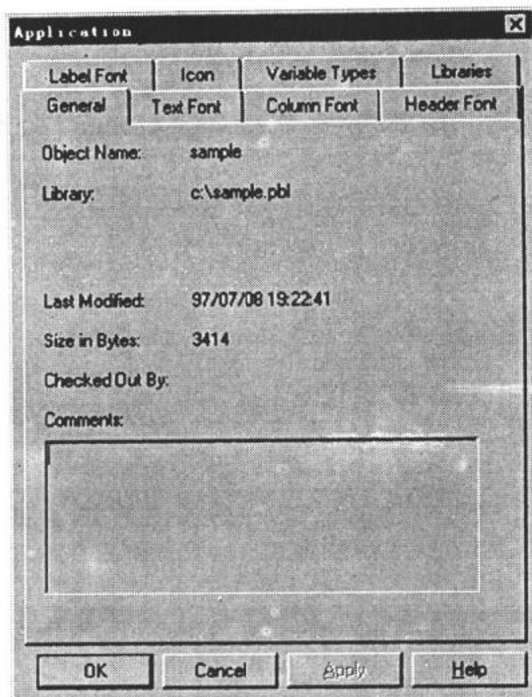
- 默认的文本属性（字体，风格，大小和颜色）
- 应用的图标
- 默认的全局对象
- 应用的库查找路径

指定应用属性值的具体操作步骤如下：

- 1) 在应用对象工作区单击鼠标右键，此时出现一弹出菜单：



- 2) 从弹出菜单中选择 **Properties**，显示应用对象的属性设置对话框，如下图所示：



下面介绍应用对象属性设置对话框中的每个属性页的功能：

- 默认的文本属性

在应用对象属性设置对话框中有四个属性页用来定义默认的文本属性，如下表所示：

属性页	功能
Text Font	指定显示在窗口对象、用户对象、数据窗口对象上的静态文本默认字体。
Column Font	指定数据窗口上被录入或显示数据的默认字体。
Header Font	指定表状格式和网状格式的数据窗口的标题带中的默认字体。
Label Font	指定自由格式数据窗口列标签的默认字体。

- 应用的图标

Icon 属性页：指定应用的图标。

- 全局变量

Variable Type 属性页：指定应用的全局变量。

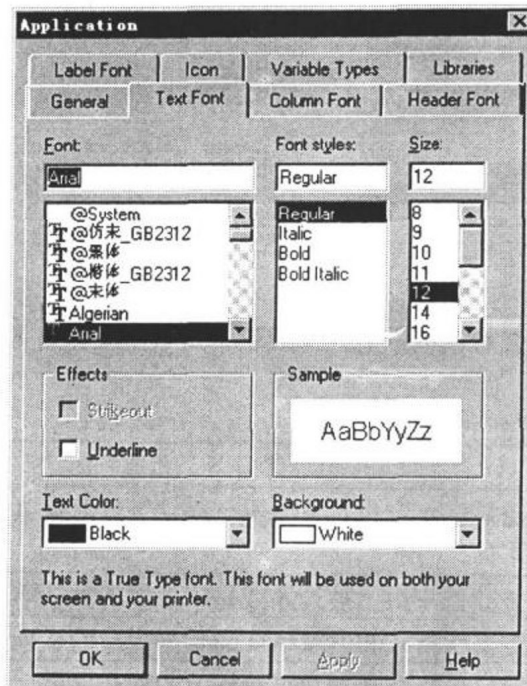
- 应用库及其查找路径

Libraries 属性页：指定应用包含的库及其查找路径。

4.3.1 指定默认文本属性

下面以指定 Text Font 属性页中的属性为例，讲解默认文本属性的指定方法。操作步骤如下：

1) 选择属性设置对话框中的 Text Font 属性页，显示如下：



下面介绍该属性页中各个属性的功能，如下表所示：

属性	功能
Font	指定文本字体，该方框下面的列表框是系统给定的字体，从里面选择其一，则该应用的窗口对象、用户对象和数据窗口对象上静态文本的默认字体为指定的字体；

Font Style	指定字体风格，从该方框下面的列表框中选择其一：Regular（标准）、Italic（斜体）、Bold（粗体）、Bold Italic（粗斜体），默认字体风格为 Regular；
Size	指定字的大小，从该方框下面的列表框中选择其一，文本按照指定的大小显示；
Text Color	指定文本颜色，从下拉列表框中选择文本的颜色，默认的文本颜色为黑色；
Background	指定背景颜色，从下拉列表框中选择文本的背景色，默认的背景色为白色；

- 2) 按照上表所示，根据需要选择文本的字体、风格、大小和颜色。定义完毕后，单击框中的 OK 按钮。

按照同样步骤，从应用对象属性设置对话框中选择 Column Font 属性页、Header Font 属性页、Label Font 属性页，定义相应文本的字体、风格、大小和颜色。

4.3.2 指定应用的库查找路径

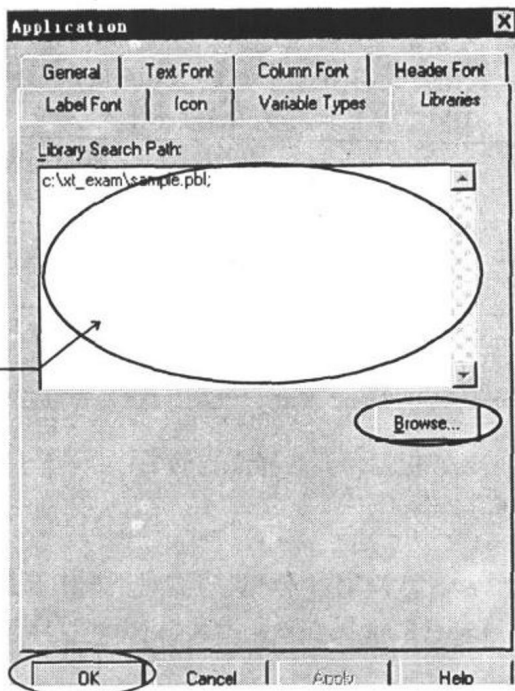
在各种画笔中建立的对象都存储在 PowerBuilder 库（pbl 文件）里，PowerBuilder 应用可存放在一个或多个库文件中，所有的库文件都应定义在库的查找路径中。

PowerBuilder 在运行期间使用查找路径来查找引用的对象。当引用一个新的对象时，PowerBuilder 按照库查找路径中指定的顺序查找所有的库，直到找到该对象为止。

定义库查找路径的具体步骤如下：

- 1) 选择属性设置对话框中的 Libraries 属性页，如下图所示：

显示当前库查找路
下所有的应用库。



- 2) 在 Library Search Path 列表框中键入库名，或使用 Browse 按钮将其它路径下的库包含到此路径下。
- 3) 如果要从 Library Search Path 列表框中删除库，操作很简单，只要从 Library Search Path 列表框中选中想要删除的库并按下 DELETE 键即可；
- 4) 单击框中的 OK 按钮（上图标出），应用库的查找路径定义完成。

4.3.3 指定应用图标

如果在 Application Painter（应用画笔）中指定了图标，则当用户在执行过程中最小化应用程序时，会出现这个指定的图标。指定应用图标的具体操作步骤如下：

- 1) 选择应用对象属性设置对话框中的 Icon 属性页，如下图所示：