

用 PowerBuilder 开发应用系统的技巧

万顺香 (机械部经济信息中心)

摘要: 本文从怎样使用和管理工具条的图标按钮 (Painter)、熟悉一个已开发成功的实例、充分利用联机帮助 (HELP) 功能、尽可能利用拷贝和粘贴功能和创建一个应用的流程等五个方面介绍了用 PowerBuilder 开发应用系统的技巧。

PowerBuilder 是一个不需用 C 语言, 就可以很容易开发出功能强大基于对象 Windows 图形界面应用系统的工具。它具有指指点点 (Point-and-Click) 开发环境; 良好的图形界面; 简单、灵活、易于使用; 支持跨平台的软件开发和推广; 可建立相对独立于数据库的管理系统和适用于 PC 及 Client/Server 体系结构等特点。

PowerBuilder 对于那些用逐句编程的传统方式编程, 并忙碌于多种计算机语言之间的软件人员来说, 无疑搭上了快速电车。但在编程的手法上以及观念上都要作一些改变, 那就是充分利用 PowerBuilder 提供的功能, 其主要表现形式是菜单和图标按钮。

一、怎样使用和管理工具条的图标按钮

图标 (Painter) 按钮的集合组成了工具条, 即 PowerBar。当开始 PowerBuilder 会话时, PowerBar 就显示出来。PowerBar 是建立 PowerBuilder 应用程序的主要控制点。

PowerBar 里的每个 Painter 都有一组相应的 PainterBar 按钮, 它是操作当前 Painter 组件的按钮。PowerBar 和 PainterBar 为用户提供了丰富、简便的图标操作方式。要充分利用工具条的图标, 首先必须了解它们和会管理它们。

用图标 (Painter) 的好处是: 形象直观, 使用方便, 有迅速简便的效果, 并提供许多快速生成工具。屏幕上是否显示工具条? 按钮上是否显示文本? 怎样在工具条上 (PowerBar) 添加新的按钮? 在屏幕上何处显示工具条? 改变按钮在工具条上的摆放顺序等等, 还有如何添加定制按钮? 是否使用 Power-Tips 等, 下面分几个方面说明:

1. 屏幕上是否显示工具条

屏幕上无工具条时, 怎样调出工具条? 先激活系统控制菜单, 选中菜单条 ToolBar, 单击鼠标, 菜单条 ToolBar 打上有 "SYMBOL 214 \f "Symbol" 的标记, 表示 ToolBar 菜单被选中。在 4.0 版本中将弹出窗口, 可根据需求来选择 ToolBar 的特征。如按钮上是否显示文本? 在屏幕上何处显示工具条? 是否使用 PowerTips 等。

若屏幕上有工具条, 而不想显示工具条时, 单击菜单条 ToolBar 即可。另外也可在特征菜单上, 用 PowerBar 和 PainterBar 菜单条来管理按钮。根据用户的习惯, PowerBar 和 PainterBar 可放在屏幕的顶部、底部、左边和右边。PainterBar 的按钮是随着 PowerBar 的 Painter 而产生的, PowerBar 不显示, PainterBar 也不显示。

2. 按钮上是否显示文本?

按钮上无文本时, 怎样才能显示文本? 对于使用 PowerBuilder 的新用户, 要想轻而易举地记住许多图标的功能, 并不那么容易。这时最好借助文本提示。如应用图标按钮有 Application 的文本, 数据库图标按钮有 Database 的文本, 菜单图标按钮有 Menu 的文本等提示。借助文本提示可以快速知道按钮的功能。很显然初学者喜欢使用有文本提示的按钮。怎样才能使图标按钮显示文本提示呢? 可将指针落在 PowerBar 上, 单击鼠标器的右键后菜单弹出, 在 Show Text 菜单条上点一下, 即在此菜单条上打上 "SYMBOL 214 \f "Symbol" 的标记即可。在 4.0 版本中用 ToolBar 管理 Show Text 菜单条, 作用一样。

若屏幕上的图标太多, 已超出屏幕的显示范围, 可用同样的方法, 即不选 Show Text 菜单条, 按钮上无文本提示, 使图标按钮缩小。若觉得使用时无文本提示不方便, 可使用 Power-Tips, 当鼠标指针在按钮上停留一两秒后, PowerBuilder 就显示一条与此按钮相关的简短说明。

3. 怎样添加新的按钮以及怎样改变按钮在工具条上的摆放顺序?

PowerBuilder 提供了许多图标按钮, 而屏幕上只挂了一部分。怎样添加新的按钮? 首先将指针落在工具条上, 单击鼠标器右键, 选弹出菜单的菜单条选项 "Customize...", 接着弹出一个窗口, 窗口的上半部集聚了所有可用的按钮图标, 为 Select Palette 组; 而窗口的下半部是用户可以直接使用的按钮图标组, 为 Current toolbar 组。将鼠标指针落在 Select Palette 组里所选图标按钮上, 用鼠标器左键点中按住, 然后拖动鼠标把所选按钮拖到用户按钮图标组, 即 Current toolbar 框中想放的位置, 然后放开鼠标即可。

怎样改变按钮在工具条上的摆放顺序? 使常用的按钮摆放在工具条上比较靠前的位置, 这样用起来方便顺手。同样用上述类似的方法, 可改变按钮在工具条上的摆放顺序。若想删除某个按钮, 可将鼠标指针置于要删除的按钮 (在对话框的 Current toolbar 框中) 上, 按住鼠标左键, 然后拖放按钮至 Current toolbar 框外的任何地方即可。

二、熟悉一个已开发成功的实例

挑选一个已开发成功、效果好的实例, 研究和熟悉它。熟

悉的途径可分两步走:先运行此实例,研究实例的可取性。包括用户界面好的窗口,涉及到窗口的颜色搭配、尺寸大小、摆放位置;文本对象的属性,如颜色、风格、尺寸、字体及摆放位置;窗口控件对象的颜色搭配及属性设置;窗口控件对象事件驱动的 Script(即编写程序,Script 又称脚本。);数据窗口操纵等等。然后逐一打开定义应用程序的对象,如窗口、数据窗口、窗口控件对象等等的定义以及了解 Script 是怎样编写的,以此为自己在开发应用过程中提供借鉴或拷贝之用。

PowerBuilder 提供了许多种颜色,并且用户可自定义颜色。有时候颜色太丰富了,反而难于定义出颜色协调的窗口和用户控制对象,这时可采取借用的方式。其它如数据窗口的定义、窗口控件对象的定义以及编写程序(Script)等都可以借用。

三、充分利用联机帮助(HELP)功能

会使用联机帮助并充分利用联机帮助,能使用户在开发应用程序的过程中,取得事半功倍的效果。例如:我们对某条命令的书写不太清楚时,可用联机帮助;对某条命令或某个函数功能不明确时,可用联机帮助;对某条命令或某个函数用法不太清楚时,也可使用联机帮助。在有的联机帮助信息中,还提供了使用的例子。

联机帮助提供了四种方式:

- (1) 系统菜单中的 HELP
- (2) 随时用 F1
- (3) 点图形为大问号的按钮图标
- (4) 在 PowerBuilder 的 Script 中用 Shift + F1

1. 用方式(1),即系统菜单中的 HELP;点系统菜单中的 HELP,再点弹出菜单中的“Search for Help on ...”菜单条,弹出查询窗口,输入相关帮助主题后,列表窗口将出现相关主题的列表,双击选中主题,点 Go To 按钮,则显示选中主题的有关帮助信息。

2. 方式(2)和方式(3)用法一样,它们与方式(1)略有区别。选用方式(2)和方式(3)的联机帮助后,弹出窗口,窗口的顶部有一排按钮,这时可选按钮“Search”,就弹出了与用方式(1)一样的查询窗口,接下来的用法与方式(1)一样。

3. 方式(4)的用法有所不同,是以获取上下文敏感的方式得到帮助的。在 PowerBuilder 的脚本 Script 中,用鼠标拖住所需帮助的命令或函数,再键入 Shift + F1,将弹出联机帮助的信息窗口,这时可查看相关的帮助信息。使你能很快地了解不清楚的命令或函数。

4. 因为在联机帮助信息中,有的提供了使用实例。在编写 Script 的时候,使用上下文敏感联机帮助来显示关于保留字和内部函数帮助信息。获取联机帮助信息的步骤是:打开你当前所编写的 Script,再找到相关的联机帮助信息窗口,点菜单 Ed-

it,用鼠标拖住所需拷贝的信息行,按 Copy 按钮,即可把所需的信息行拷贝到当前你所正在编写的 Script 中。

四、尽可能利用拷贝和粘贴功能

拷贝是计算机的一大特点。充分利用拷贝就能充分发挥计算机的优越性。对于窗口、数据窗口的定义都可以拷贝,然后略加修改即可,而不必费心费力地去自己一一定义。前面提到熟悉一个已开发成功的实例的目的:一方面是学习利用 PowerBuilder 开发应用程序的方法,另一方面是为了拷贝。PowerBuilder 提供了一个 Library 按钮图标,即仓库管理功能。点 Library 按钮后,在某个后缀是 .PBL 的文件上双击,则列出所有与此 .PBL 文件相关的定义文件名列列表,从列表中选择你所需的文件,用鼠标拖住,再点 Copy 按钮,在弹出的窗口中选取你自己的 .PBL 文件,拷贝到此完成,你所选中的拷贝文件已经在自己的 .PBL 文件中了。如果拷贝的结果不能满足需求,利用 Library 按钮的功能,可以很容易地对它作增、删、改操作。

与拷贝功能相似的另一功能是粘贴功能。粘贴功能也非常有用,特别是在编写程序(Script)的时候更是如此。因为许多 Script 是大同小异的;有些函数很长,编写时容易出错。这时使用粘贴的手段就可显示其优越性。使用的步骤是:打开你所需的 Script,用鼠标拖住所需的程序段,再点 Copy 图标按钮,此图标是把程序段复制到剪贴板中;然后回到你当前所编写的 Script 中,找到所要插入的行和列,点 Paste 图标按钮,即把你所需的程序段粘贴到自己的 Script 中,并可进行多次粘贴,还可到多个 Script 中。粘贴的信息来源途径如下:

- HELP 帮助信息
- 其它的 Script
- PowerBuilder 提供 PowerScript 的语句模板按钮
- PowerBuilder 提供 PowerScript 的 SQL 语句按钮
- 选择菜单 Edit 的 Paste Function 选项粘贴函数
- 选择菜单 File 的 Import 选项粘贴外部文本文件

五、创建一个应用的流程

创建一个应用的步骤大致上与(图 1)中描述的流程相似。其循环部分可以不断地增加和修改。如窗口:可创建新的窗口,也可对已创建的窗口进行修改。

首先必须定义一个应用,建立应用程序对象,包括以下方面:

- 指定名称和图标,即产生一个后缀为 .PBL 的库文件,带后缀为 .PBL 的库文件又称做应用包。用户所创建的窗口、菜单、数据窗口等都放在后缀为 .PBL 的应用包内,同时选定应用程序的图标。

- 为应用程序建立缺省的文本属性,如字体、风格、尺寸和颜色等。
- 应用程序库的搜索路径,即指定你所建立的后缀为 .PBL

的库文件的路径。

·建立应用程序级的脚本(Script),此时应指定缺省全局对象。

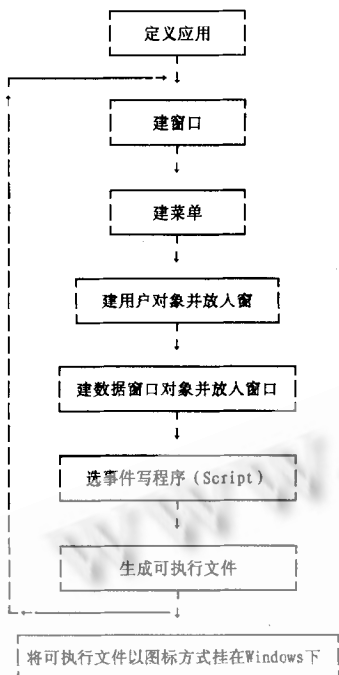


图1 创建一个应用流程图

PowerBuilder 提供了五种在所有应用程序中预定义的内部全局对象:

- 如 SQLCA Transaction 对象,用来与数据库通讯
- SQLDA DynamicDescriptionArea,用于动态 SQL
- SQLSA DynamicStagingArea,用于动态 SQL
- Error 在执行过程中报告错误
- Message 用于处理非 PowerBuilder 定义的事件的信息和在窗口间传递参数

建窗口是一步不可缺少的流程,菜单、用户对象、数据窗口、控件和脚本(Script)等都附属于窗口。

窗口分下面几类:

- 主窗口
- 弹出式窗口
- 子窗口
- 响应式窗口
- 多文档界面(MDI)框架窗口
- 带 MicroHelp 的 MDI 框架窗口

建菜单的过程可放在这一步,也可等所有的窗口都已建好后再做。

用户对象即用户自定义对象(User Object Painter),一旦定义了某个用户对象,就可以根据需要多次使用它,而不必重新定义它。使用用户对象编程是优化应用程序结构的途径。

用户对象分可视(Visual)和类(Class)两大类。

·可视(Visual)分四小类:

Standard -- 标准可视用户对象

标准可视用户对象从有一个标准的 PowerBuilder 控件中继承其定义,修改定义使控件专用于某个应用程序。

Custom -- 定制可视用户对象

定制可视用户对象具有几个控件的用户对象,它们作为一个整体发挥作用。

External -- 外部可视用户对象

外部可视用户对象包含的控件来自 PowerBuilder 系统之外创建的基本窗口系统的对象。

VBX -- VBX 可视用户对象

VBX 可视用户对象与 Visual Basic Version 1.0 兼容。可创建或购买定义 VBX 控件的文件并且在 PowerBuilder 的应用程序中使用它们。

·类(Class)分二小类:

Standard -- 标准类用户对象

类用户对象为不可视组件的处理,标准类用户对象从内部的不可视的 PowerBuilder 对象继承其定义,如事务对象或 Error 对象。可修改定义使对象专用于应用程序中。也可增加实例变量和函数来增强内部对象。

Custom -- 定制类用户对象

定制类用户对象是用户自己设计的对象,封装了属性和函数。

上面曾多次提到控件,控件是一个不可缺少的对象,这里对控件略加说明。控件分两类:带有事件驱动的控件和不带事件驱动的控件。用户运行应用程序时主要是通过与放置在窗口中的控件进行交互的。

带有事件驱动的控件如下:

- | | |
|-----------------|----------------|
| CommandButton | SingleLineEdit |
| PictureButton | MultiLineEdit |
| RadioButton | StaticText |
| CheckBox | Picture |
| Listbox | User Object |
| DropDownListBox | EditMask |
| DataWindow | Graph |
| HScrollBar | VScrollBar |

OLE 2.0 不带事件驱动的控件如下:

- | | |
|-----------|-----------------|
| Line | Oval |
| Rectangle | RoundRectangle |
| GroupBox | Drawing objects |

数据窗口对象很重要,对数据库的增、删、改和显示库里数据都是通过数据窗口来实现的。

PowerBuilder 应用程序是以事件来驱动的,通过编写程序(Script)可以指定事件发生时应当进行的处理过程。应用程序对象、窗口及控件、菜单和用户对象的事件都可以编写程序。

通过调试后生成可执行文件和将可执行文件以图标方式挂在 Windows 下两步比较简单,这里不作详细的介绍。