

基于 Powerbuilder 开发应用的技巧与方法

张斌 陈越 范新峰 (郑州解放军电子技术学院 450004)

摘要:本文从应用实际出发,探讨了 Powerbuilder 中的表、多行编辑器、数据窗口的灵活运用,并对如何深入开发 Powerbuilder 中的打印功能提出了自己的解决方法。

关键词:Powerbuilder 表 多行编辑器 数据窗口

一、有关表的定义

表(Table)是关系数据库组织信息的方式,为应用程序创建一个新的数据库之后,接下来的工作就是创建一系列的表,在建表之前应该事先设计好所有的主键和外键以及各表之间的联系,在 Creat Table 窗口里输入表名、表的所有列名、表列的数据类型及其他属性(如列宽、小数点设置和是否允许空值等)。切记每个表只允许有一个 Long 型字段,若有多个需处理的大文本字段,则需建立多个表或增加标识(区别)字段。建立好一个表之后,若发现此表缺少一个字段,需重新打开表添加一个字段。然后编辑数据源(Edit Data Source),若此新添加的字段也被选中在 DataWindow 中出现,则可发现其 Tab 值为零,如不按要求重新赋值,则用 Tab 键移动时不能移动到新添加的字段,这时需要重新选上,也称之为加亮。要利用数据窗口和表打交道,在表定义时必须选择主键,而且在编辑数据源时也必须选上,主键可以由一个或多个非空字段组成,能起到唯一标识表中记录的作用,然而在有些情况下,不需要用户在数据窗口中看到此字段,采用的方法是让此字段对用户不可见。一种方法是用别的字段放在此字段的上面(置其 Tab 值为"0"),这样用户对数据窗口操作时感觉不到此字段的存在,用 Tab 键移动也不能移到此位置,设计者从中得到了很大的方便,可以利用程序对此字段直接操作(对用户透明),设计者和用户都满意;另一种方法是把此字段最小化,用数据窗口的 Detail、Summary、Footer 带将其覆盖,当然其 Tab 值也应为"0"。

二、多行编辑器的使用

利用 Powerbuilder 开发 MIS 系统应用时,不可避免地要和大文本字段打交道。例如某单位人员的个人简

历,遇到这种情况自然要采用多行编辑器这个对象作为与用户交互的手段。对多行编辑器采用的最常用的操作就是检索、录入、修改、存盘等操作。在窗口中放置一个多行编辑器 mle-1 之后,接下来的工作就是选定事件编制脚本(Script)。

检索脚本如下:

```
blob remark  
connect;  
select blob jl into :remark //jl 为简历字段  
from worker-table  
where 条件; //条件视情况而定  
mle-1.text = string(remark)
```

存盘脚本如下:

```
remark = blob(mle-1.text)  
update blob worker-table  
set jl = :remark  
where 条件;
```

对于录入和修改,只须在多行编辑器的 modified 事件中设置一个标志 flag,初值为 0,如果进行了录入或修改置为 1,依此标志决定是否存盘。

三、数据窗口的使用

数据窗口是 Powerbuilder 的一大特色,面向数据库的应用都是以数据为中心的,而数据窗口正是显示和操作数据的窗口,数据窗口是一种对象,数据被封装在该对象里,从而可以方便地把面向对象的各种特性(如重用、继承等)贯穿到从用户界面到数据存取的应用程序开发的全过程。利用数据窗口,可以方便地把数据从数据库中提取到界面上来,把数据窗口放置到数据窗口控件中去,可以使它同编辑框、单选按钮一样成为用户界面的一

部分。对数据窗口最常用的操作就是检索、录入、修改、删除操作,其中要注意的是在检索时,若非完全查找,一定要指明检索参数,否则系统报错。数据窗口的一个重要功能是作为动态菜单使用,这是由于数据窗口直接和数据库中指定的表打交道,通过 retrieve 函数把表中某些记录的关键字段检索出来,作为菜单选项,用户可通过鼠标进行选择。编程者通过数据窗口上的 clicked 事件,调用 dw-1.getclickedrow()、dw-1.getitemnumber()或 dw-1.getitemstring() 函数,获得用户所选的菜单项,即一个或多个关键字段的值,以便打开新的窗口。这里需要一个技巧是把点击到的菜单条加亮,其实就是把数据窗口中的某条记录加亮,以免用户出错。脚本如下:

```
script: clicked for dw-1
long rownum
int xh
rownum = dw-1.getclickedrow()
//以下三条语句完成加亮功能
scrolltorow(dw-1, rownum)
selectrow(dw-1, 0, false)
selectrow(dw-1, getrow(dw-1), true)
//得到选中记录关键字段序号 xh
if rownum <> 0 then
    xh = dw-1.getitemnumber(rownum, 1)
end if
```

数据窗口的另一个重要应用就是用于制作表格,对于只有上表头的表格,一般来说只用一个数据窗口,在数据窗口的 Header 带以上制作表头,在 Header 带和 Detail 带之间放入所需字段,然后将其放入数据窗口控件,在窗口的 open 事件上编制脚本即可,若表格比较复杂,即有上表头又有左表头,且左表头形式复杂(既有竖项目,又有横项目),此时若还采用一个数据窗口,则须修改表格,即把左表头改变成全是横向项目;若想保持表格原样,只能采取两个数据窗口的方法,把左表头用一个数据窗口 dw-1 表示,上表头和表中记录用数据窗口 dw-2 表示。dw-1 中只包含一条记录,而 dw-2 中包含多条记录;若表格较大超出屏幕大小,自然要牵涉到滚动协调问题,有关脚本如下:

```
//窗口上 open 事件脚本
connect;
settransobject(dw-1, sqlca)
retrieve(dw-1)
```

```
settransobject(dw-2, sqlca)
retrieve(dw-2)
//dw-2 上 scrollvertical 事件脚本
string ls-current-position
ls-current-position = this.describe("datawindow.verticalscrollposition")
dw-1.modify("datawindow.verticalscrollposition = " +
ls-current-position)
```

四、打印功能的开发

Powerbuilder 为用户提供了一些基本的打印功能,包括 printsetup()、printopen()、printdatawindow()、printdefinedfont()、printpage()、printtext()、printclose() 等一些具体函数,其中最有一个函数就是打印数据窗口函数 printdatawindow(), 只须利用它就可迅速打印出与窗口形式一致的表格。编写脚本如下:

```
int job
job = printopen(<打印设置名>) //job 为作业号
printdatawindow(<数据窗口控件名>)
printclose(job)
```

当将一个数据窗口输出到打印机时,一般会出现一些小问题,即打印出来的数据窗口大小不合适,在屏幕上显示得很好,在纸上却显得小了许多。要解决这个问题,必须选用大一些的字体,并且最好选用 TrueType 字体或打印字体,然后特意为打印而创建数据窗口。在设计数据窗口时,可以通过设置长度的度量单位来控制打印内容的大小。

在实际应用中,窗口中不仅仅只有一个数据窗口,可能还有别的对象,打印一个数据窗口后,打印机走纸要满一页,以致再打印别的对象时不连续,而且 Powerbuilder 并没有直接提供对除数据窗口之外其他对象的打印函数,解决方法只能是自编程序打印。对于最常用的主要用于处理大文本字段的多行编辑器,对于文本的处理就很有值得推敲,笔者编了一段程序专门处理多行编辑器中的文本,具有通用性。看下面一个例子:假定窗口中包含两个对象,窗口上方为一个数据窗口,下方为一个多行编辑器,自然此时不能采用 printdatawindow() 函数先打印数据窗口,需在打印按钮上编写脚本,首先要根据定义不同的字体,然后从数据窗口中提取字段存入一个数组,定义纵、横打印坐标,调用 printtext() 函数打印数据窗口中各项。打印多行编辑器时,要处理好换行回车、空格、

半个汉字及句末标点等问题,脚本如下:

```
int job, i, x, y
```

```
string n[4] //假定数据窗口上有四个字段
```

```
int length = 64 //每行字符个数
```

```
job = printopen()
```

```
for i = 1 to 4
```

```
    n[i] = dw-1.getitemstring(1, i)
```

```
    if isnull(n[i]) then n[i] = " "
```

```
next
```

```
//用户可任选字体
```

```
PrintDefineFont(job, 1, "楷体", 200, 200, Default!, Modern!, FALSE, FALSE)
```

```
PrintDefineFont(job, 2, "黑体", 300, 300, Default!, Modern!, FALSE, FALSE)
```

```
x = 2800 //初始横纵坐标
```

```
y = 500
```

```
printtext (job, "打印示例", x, y, 2)
```

```
x = 1200
```

```
y + = 500
```

```
printtext (job, "姓名:" + n[1] + "曾用名:" + n[2], x, y, 1)
```

```
y + = 300
```

```
printtext (job, "性别:" + n[3] + "年龄:" + n[4], x, y, 1)
```

//以上处理数据窗口,处理多行编辑器的脚本略。

文中所有脚本均在联想 486/80 微机上调试通过。

参考文献

- [1] 李奇等编著, PowerBuilder 开发技术与应用, 清华大学出版社。
- [2] 蒋红等编著, PowerBuilder 语言、事件和函数, 北京晓通网络数据库研究所。

(来稿时间:1996年12月)