

Power Builder 事件触发时机的掌握及函数的使用

韩瑞新 朱桂林 齐先义 (天津 海军后勤学院 300450)

摘要:提出了一种运用单行编辑器和相关函数,对事件的触发时机及触发时的状态进行测试、比较的方法,并以数据窗口控件的四个重要事件为例进行分析。

关键字:PowerBuilder、事件、函数、事件触发时机。

PowerBuilder 事件和函数的合理使用,是正确实现、优化应用程序的前提。要合理使用事件,必须搞清各事件的触发时机;要使函数正确获取数据,必须正确选择事件。数据窗口(DataWindow)是 PowerBuilder 最有特色的对象之一,数据窗口控件(DataWindow Control)的 Clicked、EditChanged、ItemChanged 和 ItemFocusChanged 四个事件,触发时机相似、容易混淆,本文以这四个事件为例子进行分析。

1. 事件触发时机及触发优先级的比较

数据窗口的定义由属性和事件组成,属性定义窗口的风格(外观)及行为,事件定义窗口中所发生的操作。对数据库中的数据进行处理,必须灵活运用各种事件(DataWindow Control Events)和函数(DataWindow Control Functions),通过事件触发优先级所决定的对象与行

间消息的传送关系,实现消息的接收、处理和发送。

表 1 几个数据窗口事件的触发时机及触发原因

事件	触发时机	触发原因
Clicked	单击数据窗口中的任一点	鼠标单击
EditChanged	用户在数据窗口任一编辑区输入字符	键盘输入、下拉列表框、下拉数据窗口及剪切板等输入数据
ItemChanged	数据窗口某一列已被修改并失去焦点	回车、按 TAB 键、按光标键、鼠标在另一列单击
ItemFocusChanged	数据窗口当前项已改变	回车、按 TAB 键、按光标键、鼠标单双击等

执行 Power Builder 的应用 (Application), 可对事件的触发情况进行观察, 对事件的触发时机有个大体了解。但对那些触发时机相仿, 运行结果不明显、不便于观察的事件, 必须在事件的脚本 (Script) 中设计事件触发时的显示信息, 以动态了解事件触发的时机, 比较事件触发的优先级。表 1 列出了几个数据窗口事件的触发时要及触发原因。

由表 1 可知, Clicked、ItemChanged 和 ItemFocusChanged 这三个事件都可以由鼠标单击触发, 那么这三个事件是同时触发还是按一定顺序触发呢?

我们首先在窗口上为这些事件定做单行编辑器 (SingleLineEdit), 再在这些事件的脚本里向对应的单行编辑器写入信息, 或选用一些函数如 `getrow()`、`getcolumn()` 和 `getitemstring()` 等, 将函数的值送入单行编辑器。应用执行时, 通过分析、比较单行编辑器中的信息和结果, 可得出这三个事件触发的先后顺序为 Clicked, ItemChanged, ItemFocusChanged。

2. 事件触发时的环境状态

表 2 鼠标单击触发事件时环境状态

事件	当前行、列变化情况	能否取到当前项显示的数据
Clicked	以鼠标所击的行和列为当前行、列	可以
EditChanged	无变化	只能取到键盘输入的字符。由下拉列表框、下拉数据窗口及剪切板等得到的数据不能取出。
ItemChanged	焦点已失去, 但当前行、列未改变	只能取到当前项的旧数据
ItemFocusChanged	焦点已失去, 当前行、列已改变	保存好当前行、列变化前的行、列号, 可取出失去焦点的项的数据

了解事件触发优先级后, 就可实现不同事件间消息的接收、处理和发送。数据窗口是选择和操作数据的窗口, 它是用户和数据库的中间媒介, 用户可通过 PB 应用程序文本 Script 实现插入、修改、更新、删除、显示和打印数据库中的数据。Script 文本由 PowerScript 命令、语句和函数组成, 包括 Script 语句和嵌入式 SQL 语句。其中 PowerScript 函数的应用至关重要, 但函数能否正确取得期望的数据, 或向数据窗口写入数据, 与事件触发时的环境状态

直接相关。在事件脚本中使用函数来获得数据窗口在事件触发时的一些信息, 仍将获得的信息送相应的单行编辑器, 通过对单行编辑器中数据的分析, 可得出以下几个事件在鼠标单击触发时的环境状态。

3. 事件应用举例

掌握事件的触发时机及触发优先级, 了解事件触发时环境状态, 可使程序设计得简洁明了, 短小精悍, 使程序效率得到最优化。例 1, 针对实现“数据窗口中根据适当条件自动添加空白记录”的功能, 比较了用 clicked 事件和 itemchanged 事件, 及用 editchanged 事件三种方法实现的优劣。例 2, 用 editchanged 事件和 ItemFocusChanged 事件, 实现“DataWindow 中输入某一列数据后, 自动检测该数据是否与该行其他数据重复”的功能。这两个例子在 PowerBuilder 4.0 调试通过。

例 1 Data Window Control: dw-2 中第一列值 (timestamp 型) 由程序写入, 当用户写入第二列 (char(2)) 数据后, 如果当前行是最后一行, 则自动添加一空白行。

实现: 通过选用不同事件, 可由三种方法实现

//dw-2 初始化 Script:

dw-2. settransobject(sqlca)

dw-2. insertrow(0)

dw-2. setrowfocusindicator(hand!)

dw-2. setitem(1, 1, datetime(date-20, time("00:00:00")))

方法一 用 clicked 事件和 itemchanged 事件实现

说明: 由于认为 clicked 事件在 itemchanged 事件之前触发, 当前行、列已改变, 因此设置标志变量 teamnum-20 和 click-20, 来记录事件触发之前当前编辑的行列是否为最后一行、第二列。itemchanged 事件根据两个标志变量决定是否增加一空白行。

//global Variables:int teamnum-20 //若当前编辑的行为最后一行、第二列, 置 1

int click-20 //若当前编辑的列为第二列, 置 1

//global Variables 初始化(可写入 dw-2 初始化 Script 中)

teamnum-20 = 0 ; click-20 = 0

//dw-2 的 clicked 事件 Script:

if dw-2. getclickedrow() = dw-2. rowcount() and &

dw-2. getclickedcolumn() = 2 then

teamnum-20 = 1

end if

```

if dw-2.getclickedrow() = dw-2.rowcount() then
    click-20 = 1
else
    click-20 = 0
end if //dw + 2 的 itemchanged 事件 Script:
if click-20 = 1 and teamnum-20 = 1 then
    dw-2.insertrow(0)
    dw-2.setitem(dw-2.gettror(), 1, datetime(date-20,
time("00:00:00")))
end if

```

teamnum-20 = 0; click-20 = 0

方法二 用 itemchanged 事件实现

认识到 itemchanged 事件触发时,当前行、列尚未改变,直接判断当前编辑的行列是否为最后一行、第二列来决定是否增加一空白行。

//Script of itemchanged event

```

if getrow() = rowcount() and getcolumn() = 2 then
    dw-2.insertrow(0)
    dw-2.setitem(dw-2.getrow() + 1, 1, datetime(date-20, time("00:00:00")))
end if

```

方法三 用 editchanged 事件实现

说明:当前编辑项的内容只要获得键盘、下拉数据窗口及剪切板等输入的数据,即触发 editchanged 事件,可直接判断当前编辑的行列是否为最后一行、第二列来决定是否增加一空白行。脚本同方法二。

三种方法比较:第一种方法由于不了解事件触发时的环境状态,脚本设计繁杂,第二种和第三种方法,脚本都比较简洁,程序执行效率高,但由于第二种方法需要在编辑项修改完后由回车、按光标键、鼠标单击等触发,执行效果不如第三种方法理想。

例2 DataWindow Control: dw-2 中第一列值由程序写入,当用户写入第二列数据后,自动检测该数据是否与该列其它数据重复。如重复,提示“重输数据”,并将该数据置为空。

实现:用 editchanged 事件和 ItemFocusChanged 实现

//Instance Variables:

Long teamrow-20 //保存最近一次所输入、修改第二列数据的行

int itemfocuschang-flag // 若最近一次输入、修改数据为第二列,置 1, 否则置 0

//Script of editchanged event

itemfocuschang-flag = 0

if getcolumn() = 2 then

itemfocuschang-flag = 1

iteamrow-20 = getrow()

end if

//Script of ItemFocusChanged event

if rowcount() > 1 and itemfocuschang-flag = 1 then

longi = 1, i0, j, k = 1; string a, b

i0 = rowcount(); j = teamrow-20; a = getitemstring(j,

2)

do while <= i0

b = getitemstring(i, 2)

if i < > j and a = b then

messagebox("警告", "班次不允许重复, 请重新输入!")

dw-2.setitem(j, 2, "")

dw-2.setrow(j); dw-2.setcolumn(2)

exit

end if

i = i + 1

loop

end if

说明:ItemFocusChanged 触发时,虽然最近输入、修改的项值已能取出,但当前行、列已改变。可通过 editchanged 事件来记录最近输入、修改的项所在行和列。

参考文献

[1] 侯志平、蒋红等, PowerBuilder 应用开发方法和技巧, 北京晓通网络数据研究所
 [2] 李奇、常方, PowerBuilder 开发技术与应用, 清华大学出版社
 [3] PowerBuilder 4.0 随机资料