



# PB 开发应用中窗口自适应系统环境变化

辽宁省盘锦市辽河油田信息中心 杜彩玲

**摘要:** 本文首先分析了 PB 开发应用中窗口及其上的对象不能随系统环境的改变所出现的问题, 然后给出了这一问题的详细解决方法。

**关键词:** PowerBuilder 窗口 数据窗 事件

## 引言

PowerBuilder(PB)是目前最具代表性的数据库前端开发工具之一,它是一种面向对象的、可视化的开发工具。各行各业研究和开发的信息管理系统中,基于PB所开发的程序要占50%左右。在众多应用中,基于Win95或Winnt平台开发的数据库应用系统,不管使用什么开发工具,多数都存在着以下二个问题:

①窗口及其上的对象不能随着显示器分辨率的变化而改变其显示位置和大小。如基于640\*480的显示分辨率所做好的窗口,如果把显示分辨率设置成800\*600或1024\*768,那么只能利用窗口左上部分的80%或60%。

②逻辑数据窗不能随着物理数据窗的大小而自动改变其位置和大小。

这两个问题都会影响到图形界面的显示效果和屏幕的利用率,使用户界面不好,不能很好地满足用户的需求。在用PB所开发的数据库应用系统中,对这一普遍存在的问题,我们给出了一个行之有效的解决方法供大家参考。

## 窗口及其上的对象自适应系统环境的变化

### 1. 全局变量及其含义

Add\_width和Add\_height定义成Long类型的全局变量,它们的作用是存储窗口及其上对象的宽度和高度的调整值,以便在窗口的Open事件中使用,Add\_width和Add\_height的值在系统启动时得到。

### 2. 初始条件的预置

PB系统内提供了一个环境(Environment)类型,在主

应用的Open事件中,先定义一个环境变量,然后使用getEnvironment函数得到系统的环境属性,主要是机器中所安装的环境平台(Win95或Winnt)和显示分辨率。

因为系统的环境属性不包含Win或Winnt的字体情况(大或小字体),而窗口的调整值又与所选择的字体有关,这样必须先得到系统所选择的字体。所以调整窗口宽度和高度的正确值要根据环境平台(Win95或Winnt)、显示分辨率和字体三者的综合情况进行设置。主应用Open事件的Script如下:

```
string os_type,dis_type,font_type
string screen
font_type=ProfileString("c:/windows/
system.ini","boot","fonts.fon","")
Environment en_v/* 定义环境变量 en_v*/
integer i
i=getEnvironment(en_v) /* 取环境变量的值 */
choose case en_v.ostype
case macintosh!
os_type='macintosh'
case windows!
os_type='windows'
case windowsnt!
os_type='windowsnt'
end choose
dis_type=string(en_v.ScreenWidth)
dis_type=dis_type+string(en_v.ScreenHeight)
```

```
if dis_type="640480" then
  if os_type="windows" then
    add_width=0
    add_height=0
  else
    if os_type='windowsnt' then
      add_width=-369
      add_height=8
    else
      add_width=0
      add_height=0
    end if
  end if
end if
if dis_type="800600" then
  if os_type='windows' then
    add_width=730
    add_height=480
  if font_type="8514sys.fon" then
    add_width=0
    add_height=0
  end if
else
  if os_type='windowsnt' then
    add_width=271
    add_height=486
  else
    add_width=0
    add_height=0
  end if
end if
end if
:
```

3.Add\_width 和 Add\_height 正确值的设置

任何一个应用程序开发人员,在编程调试前,首先要选择好自己的开发环境(包括环境平台、显示分辨率和字体等),即默认环境,然后进行设计每一个窗口及其上的对象。本文的默认环境是: Win95 操作系统、640\*480 的显示分辨率和 Window 小字体。在这样的默认环境下,作者经过大量的和反复的实验,找出了每一种不同环境所应调整的 Add\_width 和 height 的值,列表如下:

| 项目    |           | Add_width | Add_height |
|-------|-----------|-----------|------------|
| 类型    |           |           |            |
| Win95 | 640*480   | 0         | 0          |
|       | 800*600   | 730       | 480        |
|       | 1024*768  | 1150      | 753        |
|       | 1152*864  | 2338      | 1537       |
|       | 1280*1024 | 2923      | 2180       |
|       | 1600*1200 | 4386      | 2878       |
| Winnt | 1800*1440 | 4386      | 2878       |
|       | 640*480   | -369      | 8          |
|       | 800*600   | 721       | 486        |
|       | 1024*768  | 1167      | 1153       |
|       | 1152*864  | 1480      | 1450       |
|       | 1280*1024 | 2560      | 2100       |
|       | 1600*1200 | 4480      | 2870       |
|       | 1800*1440 | 4480      | 2870       |

注: 当选择大字体时, Add\_width 和 Add\_height 的值对应上表中的分辨率依次下调一格, 即 Win95 情况下, 分辨率为 640\*480 时, Add\_height 等于 0; Add\_height 等于 0; 分辨率为 800\*600 时, Add\_width 等于 0、Add\_height 等于 0; 分辨率为 1024\*768 时, Add\_width 等于 730、Add\_height 等于 480。

4. 调整窗口及其对象的显示位置和大小

窗口及其上的对象都有自己原始的宽度和高度值, 经过第2和3节得到正确的 Add\_width 和 Add\_height 值, 在窗口被显示前, 把其原始宽度和高度分别与 Add\_width 和 Add\_height 相加, 这样就根据新的环境改变了窗口的宽度和高度值, 使其正确显示。

程序中的具体做法是在每一个窗口的 Open 事件内编写调整程序, 实例如下:

```
this.resize(this.width+add_width,this.hieght+add_height) /* 窗口按显示屏幕最大化 */
st_1.x+add_width*0.5/* 适当地调整st_1的位置和大小 */
st_1.y=st_1.y+add_height*0.5
st_1.width=st_1.width+add_width*0.1
st_1.height=st_1.height+add_height*0.1
dw_1.width=dw_1.width+add_width /*数据窗按最大比例调整 */
dw_1.height=dw_1.height+add_height
```

## 逻辑数据窗口自适应物理数据窗的大小

数据窗口(Data Window)是PB的一个最重要的对象,它是一种高效的数据访问方式,在PB内使用最为广泛,物理数据窗直接与数据库表相关联,其内隐含着SQL语句。物理数据窗的类型又十分丰富,但以 Tabular 和 Grid 二种类型使用最多,这样的物理窗宽度要随着相关表的列数而变,表列的宽度和数目直接影响物理窗的宽度。物理窗的使用都必须与逻辑窗相关联,其关联方法有以下二种:

①在逻辑窗的属性中直接写入物理窗的名字,这种方法很死板,已很少使用。

②不把物理窗的名字直接写入逻辑窗的属性中,而在程序中动态地给逻辑窗赋值,可以使一个逻辑窗动态地与多个物理窗相关联。这种方法应用灵活,可减少开发的工作量,易于维护。正由于此好处,才引出如下的问题。

因为每一个逻辑窗的位置和大小在窗口设计时都已固定好,虽然它可以按上节中所述的规律而变化,但它不能随着物理窗的宽度而变化。当物理窗的宽度比窗口的宽度大时,逻辑窗的宽度和位置不需调整;当物理窗的宽度比窗口的宽度小时,逻辑窗的宽度和位置就要重新调整,使逻辑窗的宽度等于物理窗的宽度,显示位置在窗口上左右居中。具体的过程如下:

①给全局变量g\_lastcol和g\_scroll赋值,g\_lastcol的含义是物理窗的最后一列的名字,g\_scroll的含义是物理窗左右分割的位置。

②根据g\_lastcol计算物理窗的宽度dw\_width。

③把dw\_width与窗口的宽度workspacewidth相比较。

④如果dw\_width小于workspacewidth,则重新调整;否则,不调整。

上述过程已经调试为公共软件包,只要在PB的Script合适位置引用该软件包即可实现逻辑数据窗自适应物理数据窗的大小。软件包的内容如下:

```
long row_cou /* 数据窗内检索出的记录数 */  
string last_row /* 数据窗页面内的最大记录号 */  
string c_wid,c_x /* 数据窗内最后 Column 或  
ComputeField 的宽度和 X 位置 */  
string c_name_w,c_name_x,modstr  
string ii  
string unit /* 物理数据窗所使用的度量单元 */  
long dw_x,dw_width /* 数据窗的X位置和宽度 */
```

```
string crosstab_cols  
ros_cou=dw_par.rowcount() /* dw_par是数据窗类  
型的参数 */  
last_row=dw_par.Describe("Data Window.Las-  
tRowOnPage")  
unit=dw_par.Describe("DataWindow.Units")  
c_name_w=g_lastcol+".Width"  
c_name_x=g_lastcol+".X"  
c_wid=dw_par.Describe(c_name_w)  
c_x=dw_par.Describe(c_name_x)  
dw_x=dw_par.x  
dw_width=win_par.workspacewidth()-23 /* win-  
par 是窗口类型参数、23 是微量 */  
crosstab_cols=dw_par.Describe("Data Windows.  
Crosstab.Columns")  
if crosstab_cols="" then  
if unit="3" then /* unit='3' is Centimeter point;  
unit='0' is PowerBuilder point */  
c_wid=string(double(c_wid) / 5.785)  
c_x=string(double(c_x) / 5.785)  
end if  
if double(c_wid)+double(c_x)<win_par.  
workspacewidth()-23 then  
if row_cou>double(last_row) then /* 判断从数  
据库中检索出来的记录数是否大于逻辑窗一页内可显示的  
记录数 */  
dw_par.width=double(c_wid)+double(c_x)+  
100 /* 100 是数据库垂直滚动杆的宽度 */  
dw_par.X =(dw_width-double(c_wid)-double  
(c_x)-100)/2  
else  
dw_par.width=double(c_wid)+double(c_x)+23  
dw_par.X =(dw_width-double(c_wid)-double  
(c_x)+23)/2  
end if  
if font_type="8514sys.fon" then  
dw_pqr.width=dw_par.width-70  
end if  
g_scroll=0  
else  
(下转 12 页)
```

(上接 36 页)

```
dw_par.x=1
```

```
dw_par.width=win_par.workspacewidth()-23
```

```
modstr="datawindow.HorizontalScrollSplit="+string
```

```
(g_scroll)
```

```
dw+par.modify(modstr)/*按照g_scroll把逻辑窗
```

```
分割为左右二部分，以便达到最佳查看效果*/"
```

```
end if
```

```
end if
```

## 结束语

本文的这一技术,在辽河油田的“统计”、“开发”、“科技”等大型信息管理系统中已广泛使用,并收到良好的应用效果。尽管如此,随着系统环境的不断变化,可能会出现一些新的问题,所以此项技术也需继续完善,以适应更高的要求。■

## 参考文献

1 PowerBuilder5.0 随机资料

2 侯志平、蒋红 等, PowerBuilder5.0 高级应用, 北京晓通网络技术研究所