$1#2K3+4 *Window Sizer Help*

Written by: Brendan Delumpa
Benson Software Systems, Inc. (415)433-200
CompuServe ID 72410,520

1$aMain
2#Topic0001
3KMainIndex
4+Topic0001:0000

$5#6K7+8Introduction

This Windows help file was created using a help authoring tool called Help Edit. With Help Edit, assembling the file took about an eighth of the time it normally took me because I didn't have to write my own RTF file or HPJ file. For any one who needs to write help systems in Windows, this program comes highly recommended.

If you're familiar with WinSize, then you're already familiar with what getPos does with regards to fetching window size coordinates. However, I wrote getPos long before WinSize was available in response to the lack of a sizing utility in Paradox. But in addition to fetching window size values, I wanted to be able to dynamically size windows from getPos.

Included are all the files I used to create getPos. I also included all the files used to create this help file so you can see what it takes. If you have any questions, please contact me at the number above, or send me mail via CIS.

Brendan

5$Introduction
6#Topic0002
7KIntroduction
8+Topic0001:0001

$9#10K11+12# Definitions

Click on one of the selections to bring up a list of defined items on the Window Sizer Form.

Key Combinations
Field Object Definitions
pushButtons

9$DefinitionMain
10#Topic0003
11KDefinitions
12+Topic0001:0002

Key Combinations

I've included some Alt-<key> combinations with getPos to make it easier to use the form. You'll notice that the key combinations are non-standard, or perhaps, even illogical key assignments. This is because getPos is NOT MODAL, which means that everything in Paradox is available to the designer, and that includes menus, which can also be accessed with Alt-<key> combinations. Below is the list of valid key combinations in getPos:

Alt-O Open Selected Form
Alt-I File Information
Alt-P Clipboard Copy
Alt-D Reset Coordinates
Alt-C Close getPos

```
13$KeyCombos
14#Topic0004
15KKeys KeyCombos
16+Topic0001:0003
```

$17#18K19+20Field Object Definitions

The following contain specific descriptions on purpose and usage of each object within the form. While no ObjectPAL example are present, the concept is presented.

Form Name
Coordinate and Size : X, Y, Width, Height

17$FieldDefinitions
18#Topic0005
19KFields
20+Topic0001:0004

$21#22K23+24Form Name Field

The Form Name Field is a list box object that contains the valid form names for the current directory. The values are inserted into the list box at form open. The open method calls a custom method called getFileList() which performs an enumFileList("*.fsl",dForms). dForms is a dynamic array that's defined in the Var method of the form. The dynArray values are then passed into the list box values through a FOR loop.

21$FldFormName
22#Topic0007
23KFormName
24+Topic0001:0005

$25#26K27+28Coordinate and Size Fields

The coordinate and size fields (denoted by X, Y, Width, and Height) define the position and size of the selected form when it's opened. These fields may be left blank when opening a form. When left blank, once the called form is closed, its default position and size are returned to Window Sizer and Positioner and the corresponding values are returned to the fields. This helps in gauging the amount needed to be increased or decreased to current values.

25$FldCoordinateReset
26#Topic0008
27KCoordinate
28+Topic0001:0006

$29#30K31+32PushButtons and Features

The following are a list of features available in the utility:

Windows Help Facility
Open Selected Form
File Information
Copy Coordinates to Clipboard
Coordinate Reset to Blank
Close

29$pushButtons
30#Topic0009
31KPushButton
32+Topic0001:0007

$33#34K35+36Help File

This file explains the features and functions of the various objects contained in the utility. Short definitions are made with references to ObjectPAL to give people an idea of what is going on behind the scenes in the form.

33$btnHelp
34#Topic0011
35KHelp
36+Topic0009:0001

$37#38K39+40Open Selected Form

The Open button opens up the selected form from the picklist, and places it on the workspace. In addition to opening the form, it grabs the current getPosition() parameters and inserts them into the appropriate fields. This gives the developer a way of gauging how much a window must be sized or moved.

37$btnOpen
38#Topic0012
39KOpen
40+Topic0009:0002

$41#42K43+44DOS File Information

This simply brings up a form that contains DOS file information. The form's table is loaded using the second form of enumFileList() which writes the file information to a user-defined table.

41$btnFileInfo
42#Topic0013
43KFileInfo
44+Topic0009:0003

$45#46K47+48Copy Coordinates to Clipboard

Pressing this button will copy the currently display coordinates to the clipboard.

Hint: If you open up the form again immediately after copying, you go into Design Mode and enter in the setPosition method in the Open method of the form.

45$btnCoordCopy
46#Topic0014
47KCoordinates
48+Topic0009:0004

$49#50K51+52Reset Coordinates and Sizes

This clears all the coordinate and size parameters. It's much easier than deleting the values field-by-field.

49$btnResetCoord
50#Topic0015
51KReset
52+Topic0009:0005

$53#54K55+56Close
This will close the getPosition Form.

53$btnClose
54#Topic0016
55KClose
56+Topic0009:0005