








## **PG\_FreeWindowForm FAT (1.0)**

### **Content of the package:**

This package contains the following parts.

	PG_FreeWindowControl
	PG_FreeWindowForm (FAT)
	PG_FreeWindowMover
	PG_FreeWindowRoutines
	PG_WDEF

### **Installation:**

Drop the plugin **PG\_FreeWindowForm (FAT)** into the plugin folder of REALbasic  
The plugin needs REALbasic 2.0 or higher.

### **Description:**

This plugin gives you the possibility to create your own window form. Leave the standard interfaces behind you. Create your own fantastic interfaces.

To create your own window form, you can use bitmap PICT's from the resource fork of your application. Look into the DEMO-Application which you can find in a folder called „DEMO“. In this application you can find the following bitmap PICT inside the resource fork.



### **NOTE:**

If you have some bug reports or ideas for more RB-Plugin's then let us know.  
You can contact us via e-mail, the address is mentioned at the end of this documentation.

©1998 PERGAMON interactive  
Heerstraße 50  
D-60488 Frankfurt

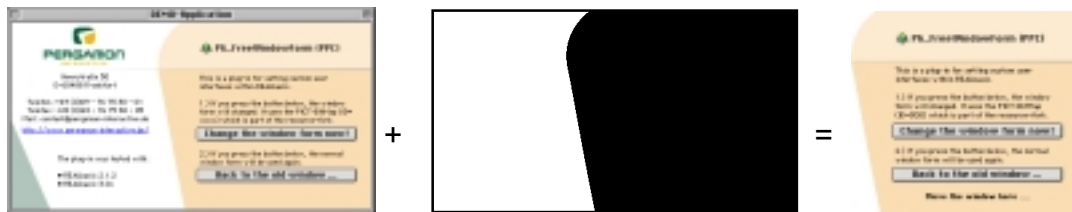
[contact@pergamon-interactive.de](mailto:contact@pergamon-interactive.de)  
<http://www.pergamon-interactive.de>





## PG\_FreeWindowForm (1.0)

This mask and the normal dialog gives the following result:



Look inside the REALbasic code to find out how it works in detail.

### The PG\_WDEF:

Sometimes it is useful to have a window without any standard elements. If you want such a window, then copy the PG\_WDEF from the resource into the resource of your application. After you have done that, you can change the MacProcID of the window to 16000 (16\*resource ID) and it will be shown without any mover or sizer. You can handle the window by your own.

Use the DEMO-Application for testing this. Set the MacProcID of the „test window“ to 16000 and start the application again. To make the window visible, for the first time it will be shown, set code into the activate-handler of the window to a routine which will define the window form.

### The Tools:

If you want to control a window by yourself, you must do the window moving for your own. Therefore we have made a new canvas class, which will do the work for you.

### PG\_FreeWindowMover:

Place a Canvas in your window and let it inherit from *PG\_FreeWindowMover*. After you have done this put the following call into the open handler of the canvas:

```
me.setPicture(self,0)
```

This will set the window for the mover and a specific picture ID from the applications

©1999 PERGAMON interactive  
HeerstraÙe 50  
D-60488 Frankfurt

[kontakt@pergamon-software.com](mailto:kontakt@pergamon-software.com)  
<http://www.pergamon-software.com/>





## **PG\_FreeWindowForm (1.0)**

resourcefork. If you pass a 0 for the picture ID, no picture will be shown and you can place the canvas as a invisible hot area inside your window.

### PG\_FreeWindowControl:

For making your own controls you can use the PG\_FreeWindowControl class. It is based on a canvas object. You can use it to define a control out of PICT's coming from the resourcefork of your application.

*me.setstates(3003,3004,3003,-20877)*

You can define three pictures for the following states (normal,selected,mouse over). It is also possible to define a mousecursor for the rollover effect. If you pass a 0 for the mouse cursor, no mouse change will take place.

### **Plugin Routines:**

*SetForm(w as window,maskID as integer,x as integer,y as integer) as integer*

This Routine will set the structure region and contend region of the window at the same time. Pass the window and the ID of the mask PICT from the resourcefork. Also pass a offset point to the routine. This offset is normally the w.left and w.top coordinate.

*SetStructureForm(w as window,maskID as integer,x as integer,y as integer) as integer*

This Routine will set the structure region. Pass the window and the ID of the mask PICT from the resourcefork. Also pass a offset point to the routine. This offset is normally the w.left and w.top coordinate.

*SetContendForm(w as window,maskID as integer,x as integer,y as integer) as integer*

This Routine will set the contend region. Pass the window and the ID of the mask PICT from the resourcefork. Also pass a offset point to the routine. This offset is normally the w.left and w.top coordinate.



## **PG\_FreeWindowForm (1.0)**

---

*SetStructureFormRect(w as window, top as integer, left as integer, height as integer, width as integer) as integer*

This Routine will set the structure region to a rect. Pass the window and the rect coordinates.

*SetContendFormRect(w as window, top as integer, left as integer, height as integer, width as integer) as integer*

This Routine will set the contend region to a rect. Pass the window and the rect coordinates.