

---

Subject: IDLgrLegend broken

Posted by [Pavel A. Romashkin](#) on Wed, 06 Dec 2000 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

I found out that if an instance of IDLgrLegend object is saved to a .sav file and then restored, the IDLgrLegend class definition is not restored correctly (unless IDLgrLegend is already instanced in the current IDL session). Moreover, attempts to use IDLgrLegend in the same IDL session fail if an instance of IDLgeLegend was first restored in that session. Here is a reproducible example (IDL 5.3 PPC), for those who want to try it, step by step.

```
temp = junk()
```

```
; A window should pop up with some meaningless plots,  
; and a Legend in the middle, with lines and diamonds, red and green.  
; Save all contents of that window by typing
```

```
save, temp, filename='temp.sav'
```

```
; Kill the window created by junk.pro  
; You should see variable TEMP lose its object content.  
; Now, restore the saved data and re-plot it.
```

```
restore, 'temp.sav'  
tmp = junk(temp)
```

```
; You should see the same exact plots with a legend.  
; Now, exit IDL and start it again.  
; Execute,
```

```
restore, 'temp.sav'  
tmp=junk(temp)
```

```
; The window that pops up has large diamonds that belonged to Legend,  
; and is not like the one made by JUNK originally.  
; Now, IDLgrLegend is not functioning anymore.
```

```
tmp = obj_new('IDLgrLegend')  
tmp -> setProperty, gap=1.0  
;% Keyword GAP not allowed in call to: IDLGRMODEL::SETPROPERTY  
;% Execution halted at: $MAIN$
```

```
; Neither will you be able to run "temp = junk()" anymore.  
; If you restart IDL, create IDLgrLegend and restore temp.sav,  
; it will re-display just fine.
```

; Restart IDL, and type

```
tmp = obj_new('IDLgrLegend')
restore, 'temp.sav'
tmp=junk(tmp)
```

; Produces what's expected.

Is this issue known? I run 5.3 on a Mac.

Cheers,

Pavel

```
. *****
;
```

```
pro junk_cleanup, id
widget_control, id, get_uvalue=all_objs
obj_destroy, all_objs
end
```

```
. *****
;
```

```
function junk, all_objs
```

```
if n_params() ne 0 then begin
```

```
top = widget_base()
```

```
draw = widget_draw(top, graphics_level=2, xsize=400, ysize=300, retain=2)
```

```
widget_control, top, /realize
```

```
widget_control, draw, get_value=obj_win
```

```
xmanager, 'junk', top, /no_block, cleanup='junk_cleanup'
```

```
view = all_objs -> get(position=0)
```

```
obj_win -> draw, view
```

```
widget_control, top, set_uvalue=all_objs
```

```
return, 0
```

```
endif
```

```
top = widget_base()
```

```
draw = widget_draw(top, graphics_level=2, xsize=400, ysize=300, retain=2)
```

```
widget_control, top, /realize
```

```
widget_control, draw, get_value=obj_win
```

```
xmanager, 'junk', top, /no_block, cleanup='junk_cleanup'
```

```
all_objs = obj_new('IDL_container')
```

```
view = obj_new('IDLgrView')
```

```
model = obj_new('IDLgrModel')
```

```
view -> add, model
```

```
symbol_1 = obj_new('IDLgrSymbol', 4, color=[200,0,0])
```

```
symbol_2 = obj_new('IDLgrSymbol', 4, color=[0,200,10])
```

```
graph_1 = obj_new('IDLgrPlot', findgen(10), symbol=symbol_1)
```

```
graph_2 = obj_new('IDLgrPlot', findgen(10)+1, symbol=symbol_2, linestyle=6)
```

```
legend = obj_new('IDLgrLegend', ['Graph 1', 'Graph 2'], gap=0.5, glyph_width=3., $
```

```
item_object=[symbol_1, symbol_2], $
```

```
item_linestyle=[0, 6])
```

```
model -> add, graph_1
model -> add, graph_2
model -> add, legend
```

```
obj_win -> draw, view
```

```
all_objs -> add, view
all_objs -> add, model
all_objs -> add, obj_win
all_objs -> add, symbol_1
all_objs -> add, symbol_2
all_objs -> add, graph_1
all_objs -> add, graph_2
```

```
widget_control, top, set_uvalue=all_objs
return, all_objs
```

```
end
. *****
;
```

---

## File Attachments

1) [junk.pro](#), downloaded 143 times

---