
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 101 times

Subject: Re: IDLgrLegend broken

Posted by [John-David T. Smith](#) on Mon, 11 Dec 2000 19:28:58 GMT

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

"Pavel A. Romashkin" wrote:

```
>
> Mark Hadfield wrote:
>>
>> You are right .... snip
>> ---
>> Mark Hadfield
>
> Oh, no. Mark, I never meant to make it sound like that. I apologize if
> you feel I insisted *you* were wrong. I only wanted to see that one can
> really use explicit naming to avoid *all* confusion. Who will follow
> this path, anyway, with dozens of methods for every object :-(
```

I have a single file containing 50 methods. So, I don't think I'll be opting for the one method, one file format anytime. Or was that one method, one vote? You do the (fuzzy) math.

In any case the process of precompiling all the methods to avoid the chicken and egg scenario mentioned is not terribly cumbersome,

especially since you have to predefine the class anyway, if you want your updates to the class__define to be present (instead of the version of the class structure in the file). Since this is a necessary feature of object restoration, i.e. since compiling the methods associated with the object's class is a side-effect of defining that class, I'd say this reduces it to a non-issue.

In simpler terms, just use:

```
resolve_obj, CLASS='ThisCoolClass'  
restore, self.saveFile,  
RESTORED_OBJECTS=obj, /RELAXED_STRUCTURE_ASSIGNMENT
```

if you know, or think you ever might, update ThisCoolClass. All these problems will then be circumvented. And yes, it should probably be named resolve_class, but originally it was primarily called like:

```
resolve_obj, obj
```

ex post facto (which you can still do if you don't fear the evolving class boogeyman).

JD

P.S. Craig thinks he may be able to undo this whole process by reading in part of the save file himself, by hand. You could then extract the classes defined within it, resolve_obj those, and then restore the file in entirety. More as available.

Subject: Re: IDLgrLegend broken
Posted by [Mark Hadfield](#) on Mon, 11 Dec 2000 21:22:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Pavel A. Romashkin" <pavel.romashkin@noaa.gov> wrote in message news:3A351026.2C587812@noaa.gov...

>> You are right [and I was wrong]

>

> Oh, no. Mark, I never meant to make it sound like that. I apologize if
> you feel I insisted *you* were wrong.

Please don't apologise. You didn't insist I was wrong. I was wrong all by myself! I've been wrong about this subject (order of searching for method definitions) for a couple of years now and it's good to have someone actually check things out and correct me.

Perhaps my posting (quoted above in its full) sounded disgruntled...or tragic? I thought it was just succinct. Somehow people misunderstand me when

I am succinct. I used to have a message on my answerphone that said

"Mark here. Please leave a message."

I thought that said it all, but I had to alter it because of the complaints. (I got one of the females in my household to record a more human-friendly message. You know the sort of thing: "You have reached the house of No-one can come to the phone right now but we'd love to hear from you so please leave a message after the beep." Me, I find it slightly insulting to be told stuff I already know (like I don't know to leave the message after the beep?) but I accept that the rest of the human race (well, the female half anyway) likes to use redundant communication for personal affirmation, sort of a form of verbal grooming. (I read about it in "Men are from Mars. Women are from Venus".)

But I digress...

- > ...use explicit naming to avoid *all* confusion. Who will follow
- > this path, anyway, with dozens of methods for every object.

I don't think that it's out of the question to have each method definition in a separate file. This wouldn't cause excessive clutter if you put each class definition in its own directory. I'm not sure what the effect on performance would be: more directory searching would be required but time would be saved in not compiling methods until they're used (if ever).

The only reason cited in the IDL documentation for bundling class definitions into a single file is this one:

"Note - If you are working in an environment where the length of filenames is limited, you may want to consider defining all object methods in the same .pro file you use to define the class structure. This practice avoids any problems caused by the need to prepend the classname and the two underscore characters to the method name. If you must use different .pro files, make sure that all class (and superclass) definition filenames are unique in the first eight characters.

But who uses IDL on "environments where the length of file names is limited" these days?

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield/>
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand

Subject: Re: IDLgrLegend broken
Posted by [davidf](#) on Mon, 11 Dec 2000 21:45:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mark Hadfield (m.hadfield@niwa.cri.nz) writes:

> But who uses IDL on "environments where the length of file names is limited"
> these days?

Actually, I've gone back to Window 3.1 in an attempt
to get *something* working on my portable computer. :-(

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: IDLgrLegend broken
Posted by [Richard French](#) on Tue, 12 Dec 2000 00:39:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Actually, I've gone back to Window 3.1 in an attempt
> to get *something* working on my portable computer. :-(
>

I heard that the International Space Station astronauts spent several
hours updating their laptop from Windows-95 to Windows-98 so that
they could run a new piece of software. They were surprised at how
hard it was. I guess they have not been reading this newstroup.

I'm sure they were just trying to get to the bottom of one of
David's new object methods..... :) (only kidding - just ordered
the 2d edition from Amazon, David, so I can be a wizard, too.)

Dick

Subject: Re: IDLgrLegend broken
Posted by [davidf](#) on Tue, 12 Dec 2000 03:38:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Richard G. French (rfrench@wellesley.edu) writes:

> I'm sure they were just trying to get to the bottom of one of
> David's new object methods..... :) (only kidding - just ordered
> the 2d edition from Amazon, David, so I can be a wizard, too.)

Actually, I was kidding about Windows 3.1. But I did go back to Windows 98 for the moment. I love a computer that can install software you are only *thinking* about installing. :-)

Let me know about this Amazon order, Richard. I've jumped way up on the Amazon hit charts lately, but they haven't been buying too many books. That makes me *awfully* nervous. Either they are starting to take that 4-6 week delivery stuff seriously, or they have another supplier, which *REALLY* makes me nervous. :-(

Just so you know, to get Amazon to say that the book ships in 2-3 days, the publisher is required to give Amazon a 45% discount and free shipping. I guess that makes sense when you are selling 40,000 copies of a book, but it doesn't make too much sense to me, when I'm selling 10s of copies. So I give Amazon the standard "technical book discount" that is common in the industry. (And I usually throw in free shipping because I like them, and they do sell far and away more books than any other bookstore for me.)

Anyway, Amazon usually buys books a *bit* ahead of demand. Most people who order from them receive a book within a week or so. But I don't know if it is the Christmas rush or what, but I'm pretty sure they are sitting on more orders now than they have books to fill the orders. So I'm curious to know how long it takes to receive your book. With information, I can sometimes "bargin" better service for you out of it.

Of course, if it is service you want, I ship 95% of the book orders I receive the same day I receive the order. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com

