

---

Subject: Re: IDL Save Object Resoration

Posted by [Michael Galloy](#) on Tue, 21 Apr 2015 12:55:57 GMT

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

---

On 4/21/15 5:17 AM, David B wrote:

> Suppose I create a variable:

>

> ----- ;Create arrays a =

> [1,2,3,4,5,6] b =[7,8,9,10] c = INDGEN(10,10)

>

> ;Save these arrays normally save, a, b, c, filename = 'file.sav'

>

> ;Run a reset ..reset

>

> ;Restore these variables, but into an object sobj=

> OBJ\_NEW('IDL\_Savefile', 'file.sav')

>

> ;Extract names names = sObj->Names()

>

> ;Print Print, names

>

> -----

>

> The problem is that I need the variables to be named something

> differently, but I cannot extract the variables into a new name. Non

> of my objects are heap variables, just standard variables.

>

> For example, following this:

>

> [http://www.exelisvis.com/docs/idl\\_savefile\\_\\_restore.html#obj ects\\_misc\\_904195448\\_1034949](http://www.exelisvis.com/docs/idl_savefile__restore.html#obj%20ects_misc_904195448_1034949)

>

> One can restore Pointers and Objects into a new object/pointer, so

> you can automatically rename your variables on the fly with the

> line:

>

> ----- ; Restore the heap variable,

> associating it with a new regular ; variable. Note that ptrName is

> (in this case) a one-element array. sObj->Restore, ptrName[0],

> /POINTER\_HEAPVAR, NEW\_HEAPVAR=myNewPtr

> -----

>

> I have an almost solid reason for doing this in my case. What I am

> therefore saying is that I want for example (but I cannot do this,

> because 'new\_variable' is not an option in the restore method!)

>

> FOR i = 0, n\_elements(names)-1, 1 DO BEGIN

>

> sobj->Restore, names[i], new\_variable = 'new\_'+names[i]

>  
> ENDFOR  
>  
> SO I end up with the following as a result:  
>  
> new\_a = a new\_b = b new\_c = c  
>  
> Where I can restore a simple variable, like an array, into a new  
> name. I may be missing the point here but I cannot think of a way to  
> do this in a strait forward way, and the 'EXECUTE' command is out of  
> the question for these objects.  
>  
> Clearly I am being stupid, but I really am stuck. Also, I know this  
> method does not work as:  
>  
> new = obj->Restore, names[i], new\_variable = 'new\_'+names[i]  
>  
> otherwise, my problem would be solved easily. Can anyone offer  
> insight into this and point out my mistake?  
>  
> Thanks much  
>  
> David  
>

I have a routine MG\_SAVE\_GETDATA that can do this:

[https://github.com/mgalloy/mglib/blob/master/src/save/mg\\_save\\_getdata.pro](https://github.com/mgalloy/mglib/blob/master/src/save/mg_save_getdata.pro)

Just call it like:

```
cow_filename = file_which('cow10.sav')  
my_name_for_polylist = mg_save_getdata(cow_filename, 'polylist')
```

Mike

--

Michael Galloy

[www.michaelgalloy.com](http://www.michaelgalloy.com)

Modern IDL: A Guide to IDL Programming (<http://modernidl.idldev.com>)

---