Subject: Re: Saving / Restoring Objects
Posted by mfeldt on Tue, 13 May 2003 10:20:20 GMT
View Forum Message <> Reply to Message

JD Smith <jdsmith@as.arizona.edu> wrote in message
news:<pan.2003.05.09.19.33.29.826287.15802@as.arizona.edu>...
> On Fri, 09 May 2003 06:33:06 -0700, David Fanning wrote:
>
>> mfeldt (mfeldt@mpia.de) writes:
>>
>>> trying myself and surfing the web I find a lot of remarks how useful
>>> it is to save and restore idl objects. However, for me it doesn't
>>> seem to work - am I doing something wrong here? Basically what I do is
>>> this:
>>>
>>> IDL> a=obj_new('fancy_object')
>>> % Compiled module: FANCY_OBJECT__DEFINE. ;; ;; this compiled my object
>>> with lots of functions etc.. ;; IDL> a-> set,'Debug',1 ; internal
>>> variable access... ;; ;; many more of those may go here ;; ;; then:
>>> IDL> save,a,file='test.sav'
>>>
>>> IDL> exit
>>>
>>> ;; now a new session
>>>
>>> IDL> restore, file='test.sav'
>>> IDL> help,a,/obj
>>> ** Object class FANCY_OBJECT, 0 direct superclasses, 0 known methods
>>>
>>> ;; i.e. the object seems to be there, but all information on it and
>>> all methods are lost... I also tried compiling the related code first,
>>> but no use...
>>>
>>> Is there any way to make this work??
>>
>> I don't think the methods are "lost", they just haven't yet made
>> themselves known to IDL. (Although I would have thought compiling the
>> routines should have worked and would be essential--you use
>> Resolve_Routine, right?--to restore the object properly) In any case, I
>> would feel free to use your restored object as normal and see what
>> happens.
>>
>> I guess whether the methods can be "found" will depend to some extent on
>> how you are naming the methods and the files that contain them. But this
>> SAVE/RESTORE method certainly works for objects.
>>
>>
> If all your methods are in the fancy_object__define file, then you'll have

> to explicitly compile this file. This is because the saved object
> implicitly contains the class definition (but no methods). Hence, IDL
> never feels the need to run the class definition procedure at the end of
> fancy_object__define.pro, and your methods defined there remain hidden.
> This problem has been discussed in great detail over the years, and David
> even has a topic on his site outlining a workaround.
>
> One alternative easy option is to make a separate file for each method,
> e.g. fancy_object__init.pro. This gets fairly distracting fairly
> quickly, so most people prefer the resolve_obj method.
>
> JD

O.K., thanks folks .. that's it - quite easy once you know it. The
trick was actually to use ".compile" instead of ".r". After using IDL
quite extensively for almost 10 years now, I have hardly ever used
this ccommand before...

BTW: resolve_obj ... from which version on does this exist??

Thanks

Markus

--
Markus Feldt Voice: +49 6221 528 262
Max-Planck-Institut Fax: +49 6221 528 246
fAï¿½r Astronomie mailto:mfeldt@mpia.de
KAï¿½nigstuhl 17 http://www.mfeldt.de
D-69117 Heidelberg, Germany Si, !asi es la vida!