
Subject: Re: Objects, File Names, and the Save command.

Posted by [davidf](#) on Thu, 23 Jul 1998 07:00:00 GMT

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

J.D. Smith (jdsmith@astrosun.tn.cornell.edu) writes:

> I am exploring a very promising use of the save/restore commands in
> conjunction with objects. Given some complex object which contains a
> host of different types of data (with pointers, etc.), as part of a
> class method, one adds:
>
> save, self, FILENAME=fname
>
> to register on disk an accurate snapshot of the object. To restore,
> later, use:
>
> restore,pname,RESTORED_OBJECTS=obj,/RELAXED_STRUCTURE_ASSIGNMENT
>
> and the object is in obj, but also brought back as the local variable
> *self*.

> [cut...]

> This is all very convenient but leads to the strange situation of a
> loaded object in memory which exists there *before* any of the class
> methods, and/or the __define procedure for that object class are
> compiled. Therefore, the usual paradigm of putting all class methods in
> the __define procedure file before this procedure (suggested by RSI
> itself in the manual) fails. How can the method be found if the
> __define doesn't have to be compiled and isn't in it's own file? I
> would like to come up with a solution which doesn't involve a separate
> class__method.pro file for each method. Any ideas?

How about something like this:

```
thisClass = Obj_Class(self)
Resolve_Routine, thisClass + '_define'
```

I haven't tested this, but don't see any reason it wouldn't work.
Resolve_Routine is the way IDL procedures and functions can
be compiled from *within* other procedures and functions.

Cheers,

David

David Fanning, Ph.D.

Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
