
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.
