
Subject: Re: Locating IDL source code file
Posted by [Mark Hadfield](#) on Tue, 14 Jun 2005 23:30:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

R.G. Stockwell wrote:

> Mark Hadfield wrote in message
>> My second piece of advice: give each routine its own source file.
>
> Again I agree. If you think of what one is doing when they have
> multiple routines in a file, it is an attempt at hiding code from other
> modules. What you really want to be doing is object programming,
> and making methods/routines (i.e. Tom) that are hidden from everything
> except
> the Fred class.

I don't agree with that. Hiding routines from those who don't need to know about them should not have to be synonymous with object-orientation. It's just that in IDL classes do have a mechanism for this hiding and routines don't. Look at Python's modules to see how this should be done.

(I recommend that any IDL programmer who is interesting in language design should look at Python and Numeric Python. But be aware in doing so that some of the choices made in Python have made it *way* slower than IDL in some situations.)

Back to IDL now. If we have a routine "fred" and we want it to call another routine that has no other reason to exist, then the conventional way of achieving this is to give it a name that reflects its relation to fred *and* its function, then put it in fred.pro *before* the main routine:

```
*** file fred.pro ***  
  
function fred_add, x, y  
  
    return, x+y  
  
end  
  
pro fred  
  
    if fred_add(1, 1) ne 2 then message, 'Huh?'  
  
end  
  
*** end file fred.pro ***
```

This is a kludge, because it's not really private. If file bill.pro

defines a function `fred__add` then the clash will break things (and will be very hard to find, to boot). But as long as all the private routines in `bill.pro` start with `"bill_"`, there shouldn't be a problem.

Yes, you could avoid this by rewriting `fred` as a class, but I think in most cases this would be a perversion of object-oriented programming.

You're still left with the problem that there might be two `"fred"`s on the path. This is a problem with IDL's flat name space for routines. Again, something like Python modules would alleviate the problem, but it ain't gonna happen any time soon.

--

Mark Hadfield "Kei puwaha te tai nei, Hoesa tahi tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)
