
Subject: Re: Duplicate module names. Was: Object epiphany: ...
Posted by [John-David T. Smith](#) on Thu, 19 Apr 2001 19:24:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Kristian Kjaer wrote:

>
> JD Smith wrote:
>> I think it only fair to let people know that I tend to shy away from
>> distributed code with people's initials in the name....
>
> Fancy that!
> I was playing with the thought of posting a request: Would all those
> generous providers of public IDL code please edit all their code so that
> no module names are duplicated from one library to the next ...
>
> Anyway, thanks for all the code!
> Kristian Kj  r, Ris   Natl. Laboratory, Denmark

Please don't confuse my comment with advocating the choice of generic or otherwise undistinguished names for distributed routines. You **must** pick good, unique names for routines you hope people will use. The shadow scanning utilities in IDLWAVE are an excellent way to explore this issue (IDLWAVE->Routine Info->Routine Shadows). Here are some examples of overlaps:

```
datatype()  
- JHU      [C--] ~/idl/jhu/datatype.pro  
- NasaLib  [C--] ~/idl/nasa/pro/misc/datatype.pro
```

So, both NASA and John's Hopkins libraries define datatype(). They both do the same thing -- return the data type of a passed variable, with a variety of possible formats. Further clues are found if you look in the DOC header of the NASA routine:

```
; REVISION HISTORY:  
;   Original Version: R. Sterner, JHU/APL, 24 October 1985.  
;   Major rewrite, add /TNAME keyword, unsigned and 64 bit datatypes  
;   W. Landsman   August 1999
```

Hmmm.. OK, so they have the same pedigree. In this case they are calling sequence and result compatible, no name change is warranted.

What about this one:

```
factor  
- JHU      [C--] ~/idl/jhu/factor.pro  
- NasaLib  [C--] ~/idl/nasa/pro/jhuapl/factor.pro
```

Another of the same routine, in two libraries. In fact everything in Nasa's jhuapl directory shadows a JHU routine (not surprisingly). Ugly, but not fatal.

And here's another:

sunpos

- JHU [C--] ~/idl/jhu/sunpos.pro
- NasaLib [C--] ~/idl/nasa/pro/astro/sunpos.pro

Uh oh, in this case they aren't compatible at all. They do similar things (calculate, surprisingly, the sun's position), but have different syntaxes and return values: one in altitude/azimuth, the other in right ascension/declination.

The problem is that people developing libraries don't want to keep remembering to type sunpos_radec, say, instead of sunpos. For code in a library, though, keeping the namespace clean is vital. I advocate using up this necessary extra discriminating text in the routine name with additional information about what the routine does, rather than author info. But in reality, both ways serve to avoid namespace collision.

Certainly, the distributed routine "JDS_calculate" is better than the routine "calculate" for this purpose, but better yet (in my estimation) would be "calculate_fried_food_calories" or some other more descriptive moniker. The way I think about it, if what your routine does cannot be substantially differentiated from pre-existing (or not-yet-existing) code by describing its function, then maybe you should just use the other code in the first place ;)

Maybe I'll install all the libraries I can find, do a global shadow listing, and post it somewhere for all to see.

JD
