

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

Subject: Re: Duplicate module names. Was: Object epiphany: ...
Posted by [Wayne Landsman](#) on Thu, 19 Apr 2001 22:45:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

> 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.
```

I've recently added a primitive FAQ to the IDL Astronomy Library (<http://idlastro.gsfc.nasa.gov/idlfaq.html>), and in question A8 I try to address the namespace conflicts with programs in other libraries that I know about. Unfortunately, once a name conflict exists, there is no easy solution to fix the problem, unless one of the procedures has much wider usage than the other one. But documenting the name conflicts can at least make it quicker to diagnose any problems.

I know that some users of the Astronomy Library will append an IAL_ to the start of every procedure to make sure they have no namespace conflicts. But I share with JD a preference for descriptive names without prefixes. (Thankfully, the DOS 8 character limits are ancient history). This does require the additional effort of trying to be aware of the names already in use in the major libraries.

Even more than "calculate.pro", my favorite uninformative (yet popular) procedure name is "convert.pro". ..

--Wayne
Landsman
landsman@mpb.gsfc.nasa.gov

P.S. The IDL Astronomy Library will be unavailable much of April 20-23 and April 27-30 due to planned power blackouts in our building.

Subject: Re: Duplicate module names. Was: Object epiphany: ...
Posted by [Kristian Kjaer](#) on Fri, 20 Apr 2001 07:29:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

JD Smith gave several examples of duplicated module names and wrote:

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

That would be very useful.

For the record, both Ray Sterner's and David Fanning's libraries have a pickcolor.pro, and I think there are a couple of clashes between the esrg library and other libraries.

Subject: Re: Duplicate module names. Was: Object epiphany: ...

Posted by [Martin Schultz](#) on Fri, 20 Apr 2001 11:54:42 GMT

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

JD Smith wrote:

>

> Maybe I'll install all the libraries I can find, do a global shadow

> listing, and post it somewhere for all to see.

>

As a start, you could take a look at my web pages

(<http://www.mpimet.mpg.de/~schultz.martin/idl/html/allalpha.html>)

where you can find routines of 11 popular libraries all listed in alphabetical order; so a name space conflict is very easy to detect ;-)

In fact, this web page of mine is one argument that I have to conceive in favor of not using a authorname prefix for routines, because ... guess what you will see when you sort all such routines by alphabet ;-). Then again, for a hierarchy of objects belonging to the same "tree", this would probably be a desired effect, so that you see all of them listed together. ... Then again: if *everyone* would use a 3-letter prefix to his/her routines, one could easily strip these off and sort by the rest of the name which should then be meaningful, of course.

Overall, there seems to be a legacy problem (as with IDL itself), and I don't see a chance that everyone (especially the maintainers of the huge libraries like JHU or Astro, or David) will go through all those routines and (a) rename them, (b) change all routine calls to the new names ... - unless someone provides a perl script to do this automatically ;-).

The *real* problems, of course, are (1) that the RSI library is insufficient (e.g. a missing colorbar routine), (2) that the RSI library can never be sufficient because different people need different routines, (3) that there are no clear rules and naming conventions for software development in IDL, (4) that routines that are made publically available are often written by "amateurs" who want to get something done, do it in a more general way and think that others may find that useful, (5) the availability of the web as such. In the old days, you just never knew about routines someone in New Zealand wrote, so you would do it yourself anyhow.

Now, with IDL objects, we may be in a somewhat more fortunate

stuff together that we're still using today. I guess it depends on the whether the magnitude of the geek factor is greater than the "how will I pay the mortgage" factor in todays New Economy (requirement #1 will be that all code be Open Source. :o) But, the consensual-ness (why does that word make me shift uncomfortably in my seat..?) of this newsgroup and the IDL community in general should make it a relatively easy thing, right? To say nothing of the fact that probably only a handful fo folks (?) understand the requirements for object programming in IDL so that the IDL Object Programming Guideline Committee should be small enough to be quite effective. :o)

paulv

--

Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
paul.vandelst@noaa.gov Alexander Pope.
