Subject: Re: Namespaces (redux)
Posted by JD Smith on Wed, 25 Sep 2002 17:13:46 GMT
View Forum Message <> Reply to Message

On Wed, 25 Sep 2002 08:06:00 -0700, Craig Markwardt wrote:
> JD Smith <jdsmith@as.arizona.edu> writes:
>> This brings up an excellent point that probably needs reiterating: the
>> IDL namespace is a rapidly dwindling resource, as libraries grow and
>> disseminate. While in the past most IDL users might have worked
>> primarily with their own routines, with a few bits and pieces of
>> outside code cobbled together from various sources, that era seems to
>> be passing, given the improving quality and availability of large
>> libraries, and the new RSI initiative to facilitate code sharing among
>> IDL users. What can we do to preserve this vanishing resource? A few
>> guidelines come to mind:
>
> JD, I think you are being a bit too alarmist. The IDL namespace is not
> rapidly dwindling the same way that (pick favorite species) are
> declining. For a typical 10-character function name, there are about a
> million billion unique combinations. Of course, if we restrict to
> English-sounding names, then the unique combinations is less, but still
> huge.

The alarmism was a clever ploy to engage critical thinkers such as yourself ;). By no means am I equating the IDL namespace to our rainforests or endangered species (after all, we could always all just switch to Python: the red panda doesn't have that option).

Yes, if IDL programmers would choose randomly even from the 8^30 or so potential 8.3 DOS-friendly names, this would never be a problem. And even if they stuck to the 50,000 or so english words with 8 or fewer letters, trouble would be averted. This, as you know, is not the case. Instead, they understandably tend to cluster around a very limited set of key words like "plot", "func", etc. For instance, in my not-overly-large collection of libraries, I found 131 of 1991 total routines had the word "plot" in their name. Thirty of these were object methods, so those can be forgiven, but the extremely small portion of the potential namespace we stick to is evident: it's *much* smaller than the domain of English-sounding names.

>> 4. Make qualifying prefixes descriptive and unique. Using initials
>>   as qualifiers is of course an option, but wastes precious filename
>>   space on information of little real value. If you do use initials,
>>   don't skimp on additional information in the rest of the name.

> I'm not sure it's of little real value. For example, from the names
> REPLICATE and CMREPLICATE, you can probably guess that both do something
> similar.

>
> Or, the names of functions MPFIT, MPFITFUN, MPFITEXPR, MPFITPEAK, and so
> on, indicate that they all belong to the same library, obey the same
> calling conventions, and that is valuable knowledge. [And pays homage
> to the MINPACK heritage.]
>
> I'm not a huge fan of typing routine names like
> tvRubberBandBoxUpdatePanelWithMarchingAnts().

Yes, you have to weigh conciseness and clearness against potential for namespace collision. But clearness and small namespace footprint are not congruent: I could also guess that "plotxy" does something similar to "plot", which makes it a descriptive name, but simultaneously gives it a high likelihood for collision. I also don't like overly long routine names: the point is to be descriptive and unique, not pedagogical. In the case of your MINPACK functions, that's a perfect example of using a clear, well-motivated prefix qualifier.

>> 5. Look before you leap. An excellent (if increasingly outdated)
>> online-browser for almost all widely distributed libraries of IDL
>> code is available at:
>
>
> Now here is the crux of the matter. The reason that we have name
> clashes is that everybody and their brother has written *slightly
> different* routines to count lines in files, called it the same thing,
> and published it. However, most people are respectful. If you search
> the IDL Astronomy library or the U. Washington index for terms like FIT
> or LEGENDRE, you will find lots of routines, but they are for the most
> part unique. Rule: If it's an *obvious* name, then attach a *non*
> obvious prefix or suffix to the name. End of story.

An excellent synopsis of the recommendation. Unfortunately, unless you've been in the business a long time, it might not be clear what an *obvious* name is. Hence the guidelines. On another note, Eric Deutsch is no longer maintaining the IDL library browser. I'm sure he'd be pleased to pass it along to another willing maintainer.

> It is notable that in almost every case, the offending entity for these
> types of namespace clashes is none other than Research Systems
> Incorporated. We should be preaching to them.

On that point I fully agree. And I have. I'm sure RSI feels a distinct and righteous ownership of the IDL namespace and believes that routines in !DIR are free to trample all over it. Here's an interesting random bit of output from IDLWAVE's shadow detection:

PNT_LINE()

```
  - SystemLib   [C--] /usr/local/rsi/idl_5.5/lib/pnt_line.pro
  - SystemLib   [C--] /usr/local/rsi/idl_5.5/lib/efont.pro
read_pict_item
  - SystemLib   [C--] /usr/local/rsi/idl_5.5/lib/read_pict.pro
  - SystemLib   [C--] /usr/local/rsi/idl_5.5/lib/query_pict.pro
READ_PPM_NEXT_LINE()
  - SystemLib   [C--] /usr/local/rsi/idl_5.5/lib/read_ppm.pro
  - SystemLib   [C--] /usr/local/rsi/idl_5.5/lib/query_ppm.pro
READ_PPM_NEXT_TOKEN()
  - SystemLib   [C--] /usr/local/rsi/idl_5.5/lib/read_ppm.pro
  - SystemLib   [C--] /usr/local/rsi/idl_5.5/lib/query_ppm.pro
```

In any case, it's too late to undo the sins of the parent, but we can strive to avoid repeating them ourselves.

> My addition: I attempt to always keep the main routine name unique
> within the first 8 characters from all other routines I know about.

A fine suggestion, if only for those few remaining fossil systems which enforce such a filename length limit.

Thanks for the comments.

JD