Subject: Namespaces (redux)
Posted by JD Smith on Tue, 24 Sep 2002 23:42:58 GMT
View Forum Message <> Reply to Message

I wrote this almost two weeks ago in another thread, but the newfeed here seems to have taken a stance resolutely against free expression. They tell me the problem is fixed now. I figure it's a good test because it's a controversial enough topic that I'll get immediate responses, if it makes it out:

On Wed, 11 Sep 2002 23:42:22 -0700, Reimar Bauer wrote:

> David Fanning wrote:
>> Wayne Landsman (landsman@mpb.gsfc.nasa.gov) writes:
>>
>>
>>> I have heard a rumor that there may be a standardized way of counting
>>> the number of lines in a file in the next release of IDL ;-)
>>
>>
>> And I hear it even has a name very similar to Riemer's little File_Line
>> program, which I think is too bad. Something like COUNT_ROWS really
>> makes more sense to me. :-)
>>
>> Cheers,
>>
>> David
>
>
> I don't know at the moment if I should be happy or not. It's fine to see
> that's good routines would be implemented into the idl binary but always
> this is done I got the problem that's all of our sources using these
> routines need changes. This happens last time by file_search. We have
> had nearly the same functionality in our routine but not the same
> parameters or keywords. Internal routines are first called sources with
> the same name are ignored.
>
> I believe they like to start with FILE_ because of the other file
> handling routines. I would prefer FILE_COUNT_ROWS if possible. This
> gives more sense as the word I have choosen in the past.

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:

1. Use Objects. Object methods don't count against the namespace, since they're encapsulated beneath the class name. Use unique class names. "Box" is probably not a good name for a class. "tvRubberBandBox" might be better.

2. If not using objects, imitate their namespace parsimony by consistently prepending a class-like prefix to all related routines... e.g. tvRubberBandBoxDisplay.pro, or tvRBandBox_display.

3. Resist the urge to give routines short catchy names, especially if they will ever be distributed (even if just emailed to a colleague). To save time for interactive use, consider using local "abbreviation" routines which will never be distributed, and which call longer-winded programs, e.g.

```
pro tvrbd, args, _EXTRA=e
        tvRubberBandBoxDisplay, args, _EXTRA=e
    end
```

   Never call these interactive abbreviation routines directly in any code.

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.

5. Look before you leap. An excellent (if increasingly outdated) online-browser for almost all widely distributed libraries of IDL code is available at:

   http://www.astro.washington.edu/deutsch/idl/htmlhelp/

   Use it to help you pick unique names, but don't imitate the spendthrift ways of our forerunners: we've inherited the problem from them, and must act to counter it.

6. If you are internalizing and distributing individual routines from libraries, with no plan to maintain compatibility with future versions of that code, change the name. If your code calls a routine with an established name (e.g., readfits), it should endeavor to work with the latest version of that routine. Otherwise, the name should be changed (e.g. projx_readfits). At the very least, document what version of the code it requires.

Other suggestions or additions?

JD