Subject: Re: Many procedures, what to do?
Posted by Theo Brauers on Mon, 15 Jan 2001 14:52:37 GMT

Hi Craig:

I had a similar problem some time ago. My solution was to have a
number of different source files with routines and functions grouped
together:

easy.pro (main)
easy_define.pro (10+ functions)
easy_fit_proc.pro (3 procedures)
....
(about 20 source files with up to 20(?) singe funcs)

Secondly , I have a procedure to make the savfile (The procedure also
writes some status, user and date information but I cut that) :

easy_make_sav.pro

```
 PRO easy_make_sav, SHORT=short

 ; compile the procedures
 RESOLVE_ROUTINE, 'easy'
 RESOLVE_ROUTINE, 'easy_utilities'
 RESOLVE_ROUTINE, 'easy_operate'
 ....
 ; compile all functions
 RESOLVE_ROUTINE, 'easy_diagnostic_plot', /IS_FUNCTION
 RESOLVE_ROUTINE, 'easy_error', /IS_FUNCTION
 .....
 RESOLVE_ROUTINE, 'easy_model2userfile', /IS_FUNCTION

 IF NOT KEYWORD_SET(short) THEN BEGIN
     RESOLVE_ALL, SKIP_ROUT=excluded
 ENDIF

 SAVE, /ROUTINE,  FILENAME=savfile, COMPRESS=compress ; , /VERBOSE

 RETURN
 END
```

Finally I wrote a routine easy_startup which loads the sav file for correct
version of IDL and does some talking to the user if it was not found.

The concept works fine since then and most of the users in our group

do just now that there is easy_startup which loads the whole thing if
necesarry and starts the easy environment. I normally distribute
easy_startup.pro and easy_54.sav (easy_53.sav etc.) only.

If you have questions or if you want to have a closer look drop me a note.

Best regards Theo

Craig Markwardt wrote:
>
> Greetings all!
>
> I have about twenty five functions that go into a single larger
> library. They are all pretty much inextricably linked together. This
> is the SAVE file library I have mentioned once or twice, with the
> ability to read and write and interogate save files yourself.
>
> Anyway, my question is what to do with all these files? I would
> imagine that for most people, including myself, this many files is a
> pain in the neck. Also they share a lot of the same parameters so it
> would be difficult to keep the documentation up to date.
>
> My other option is to merge them into a single file called, say,
> CMSVLIB. There are a couple of problems with that.
>
> First, how to get them compiled. That's easy, I just require every
> program which calls the library to invoke CMSVLIB first. As long as
> there is actually a procedure called CMSVLIB at the end of the file,
> this should force all the other routines in the file to be compiled.
>
> The other problem is more subtle. Since none of the individual files
> are compiled when the invoking procedure is compiled, IDL won't know
> about the functions. It will see the round parenthesis of
> "cmsv_rlong(block, pointer)" and think it's an array subscript.
> Arghh.
>
> Okay, that can be solved by forcing everybody to declare the functions
> they use with FORWARD_FUNCTION. Now it's starting to get annoying
> again. I guess I could rewrite everything to be procedures...
>
> Does anybody else have suggestions, or experiences with something like
> this?
>
> Craig
>
> --

--

Dr. Theo Brauers
Institut fuer Atmosphaerische Chemie (ICG-3)
Forschungszentrum Juelich
52425 JUELICH, Germany
Tel. +49-2461-61-6646   Fax. +49-2461-61-5346
 http://www.kfa-juelich.de/icg/icg3/MITARBEITER/th.brauers.ht ml