Subject: Re: Many procedures, what to do?
Posted by Alex Schuster on Mon, 15 Jan 2001 17:45:14 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:

> 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.

That's what I do for large projects, too.

> 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.

I tell all users that they have to call the big routine once before
doing anything else, when they want to use it. Well, maybe short after
DEVICE, DECOMPOSED=0 and such.

> 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...

What about a startup file (the one that gets executed when IDL is
started) containing all the FORWARD_FUNCTIONs. It could as well compile
your CMSVLIB, but that would take some time, probably too long.
That's another disadvantage of the many files, whenever IDL is started,
it scans the whole $IDL_PATH for files. This is fast, but well
noticeable here (2000 files).

Alex

--
Alex Schuster   Wonko@weird.cologne.de   PGP Key available
alex@pet.mpin-koeln.mpg.de