Subject: Re: linking programs/ procedures
Posted by John-David T. Smith on Wed, 26 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Troy Carter (tcarter@princeton.edu) writes:
>
>> You can also use:
>>
>> @/path/to/program.pro
>>
>> At the beginning of your code. This will compile program.pro before
>> running the rest of your code.
>
> Uh, no, not exactly. It will put the program.pro code *into*
> the program file as if you had actually typed it there. It
> doesn't compile anything, and it is completely unnecessary. :-(

Oh David, always making these sweeping statements to entice my refutation...

I suppose you meant that it's unnecessary in the case of running a routine from "program" elsewhere. For this you are correct. But it does have its uses. My favorite is with common blocks (whose use should, of course, be limited). As everyone knows, once a common block is defined, you can simple reference it with:

common my_common

i.e. without any of the variables listed out. This saves typing, but more importantly, leaves only one location which needs to be modified when common block variables are added or removed. But there's a catch... you have to compile the defining routine *first*... i.e. the point of entry into a suite of programs which share the common block must be a single file/routine. If you try to enter elsewhere, it will generate a compiler error, and refuse to run.. you can't simply "call" a defining routine at the beginning of each routine in the suite either, since it's a compile time, not a run-time error. So, you resort to making *all* of the routines a defining routine for the common block, but including all the variables in each common block statement. But this unfortunately means modification requires you to change each and every one (over 30 places for one of my common blocks) so you're back where you started. But enter @ to save the day. In a file, call it my_common.pro, put:

common my_common, var1, var2, var3, thefinalvar

or whatever. Then, in each routine which needs it, simply put:

@my_common

in place of "common my_common", and you get the best of both worlds.... single point of modification, yet every routine capable of defining the common block. If you need to perform a standard version lookup or some such in all your routines, you could also include it in my_common.

Just trying to defend poor little "@" from the onslaught.

JD

--

J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|