Subject: Re: Using subroutines
Posted by David Fanning on Wed, 19 Jun 2013 22:07:47 GMT
View Forum Message <> Reply to Message

David writes:

> I am sure I am going through something very well known.
> I have a large code. And I wish to break it in smaller sections.
> So my hope was to do something like
>
> pro main_routine
>
> routine_reading_files
> routine_making_initializations
> routing_making_operations
> routine_writing_outputs
>
> end
>
> (I bet you are laughing)

Yes, but only because I was going to suggest you would be MUCH further along if you wrote the whole thing as an object, with these little organizing routines as methods of the object. Then, you wouldn't have to worry about communication between routines. :-)

> so it looks like "routine_making_initializations" doesn't know
> anything about what the previous routine read.
> Don't tell me that I have to make functions instead of routines.

OK, you don't have to write functions. But, you DO have to figure out a way for these routines to communicate with each other. Functions are an obvious choice if you want to return something from a routine, but other methods work as well. You could use output keywords, for example. Common blocks (yuck!). A pointer. Etc.

> As I said, I want to break my too-long code into subsections for obvious reasons. Is there any way I can accomplish this? I don't think making a function for each step would make my tasks easier.

If it is not going to make your life easier, then forget it. Breaking things into modules is BS if you don't get something out of it. Most of the time I write modular code because it makes my programs easier to maintain and extend in the future. Sometimes, at least for me, the payoff is in the future, not in the present.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")