Subject: Re: main or procedure
Posted by Marshall Perrin on Wed, 20 Jun 2007 14:39:14 GMT
View Forum Message <> Reply to Message

David Fanning <news@dfanning.com> wrote:
> sujian@gmail.com writes:
>
>> When I .compile a procedure, it says,
>> You compiled a main program while inside a procedure. Returning.
>>
>> And all the information before the compilation was lost.
>>
>> How to fix it?
>
> You need to do two things: (1) Add more error handling
> to your code, so that when a program crashes it goes
> back to the main program level (ON_ERROR, 1), and (2)
> learn to type RETALL *immediately* when your program
> crashes, and certainly before you do any more compiling.

It feels odd to disagree with the great Dr. Fanning, but in this case I want to offer a counter-suggestion to the above advice. I much prefer for my programs to stop at the point of failure when they crash, because that way I have a chance to examine the variables and program state and try to figure out what went wrong. I *hate* it when programs return back to the main level, since then it's usually much harder to understand the situation that caused the crash.

Besides, often when the error is in a subroutine, it can be fixed and the program execution resumed from that point. This can be a big timesaver when running some computationally intensive code that you don't want to have to restart from the very beginning:

```
dxfinish                ; set breakpoint after broken subroutine
return                  ; or return,0 if you're in a function
.comp fixed_subroutine.pro   ; after you've fixed the error
fixed_subroutine, /arguments ; paste in the relevant line to re-run the
      ; routine, which will hopefully now work
```

I think the difference between David's approach and mine probably comes down to how we structure our main programs. His are probably mostly Widget programs, so if you retall to the main level, hopefully all your state is retained and you can resume your work without losing all your information. I on the other hand usually just run long computations or data analysis routines in regular procedures, so a retall in the middle of that would indeed lose all my state. Hence my deep aversion to "on_error, 1". I'd argue that these are equally good, if different, approaches, but maybe other folks here will care

to chime in with their opinions.

For the original poster: The reason it returned automatically to the top level in your case is that you compiles a main routine (i.e. a file with IDL code but no "PRO whatever" or "FUNCTION whatever") while inside a procedure or function. That, you can't do. You can compile a PRO or FUNCTION while inside another function (but not while inside itself), but main-level routines can only be compiles from the main level. Which means if you want to have a batch of commands that you can execute interactively from the command line while at a breakpoint, you either need to package it up inside a procedure, or you need to run it with the @FILENAME syntax (with all the hassles about loops and punctuation that entails.)

- Marshall

---