Subject: Re: Help needed on calling an IDL procedure from the Linux command line
Posted by peter.albert@gmx.de on Thu, 25 Aug 2005 08:49:26 GMT
View Forum Message <> Reply to Message

Hi Sanjay,

I have nothing additional to say about passing parameters, but I'd like to shed some light onto your reported "Programs can't be compiled from a single statement mode" error. Calling IDL from the shell like

$ idl test.pro

is just the same as if you started IDL and then typed one line of your test.pro file after the other into the IDL command line.

Thus, if you do the follwoing:

$ idl

IDL> pro test.pro

you get the same error message.

The problem disappears of course if you use one of the wrapper solutions provided by Ken (and the others in the cited discussion) as all those wrappers first (automaically) compile test.pro and then call it in a way similar to

IDL> test, variable

Just another comment: if for one reason or the other you will find yourself in a situation where you have an IDL program which will run without input variables, and you still want to use the

$ idl test.pro

approach, then (almost) everything is fine if you just don't make the file test.pro a routine, i.e. no "pro" or "function" command on top. Just a list of IDL commands. However, there are two traps: If you don't want to get stuck in the IDL session after completion of test.pro, the last line should be

exit

in fact, if there is an "end" command in the last line, IDL will stop compiling with a syntax error (well, you should not type "end" in the IDL command line neither).

The next trap is a bit more tricky, and it is about for-loops or
if-clauses. Say, your test.pro file could look like this:

```
for i = 0, 10 do begin
  print, i
  print, i^2
endfor
exit
```

```
$ idl test.pro

results in
...

      11
     121

endfor
 ^
% Syntax error.
  At: /home/palbert/test.pro, Line 4
```

The problem is that IDL works on each line after the other, and
similarily

```
IDL> for i = 0, 10 do begin
IDL> print, i
```

just produces the 11, the value of i after it went through the for-loop
11 times, trying to execute "begin". A follwoing call like

```
IDL> endfor
```

just produces the same syntax error.

The solution is that you have to make the for-loop a one-line statement
using "&". On the command line, something like

```
IDL> for i = 0, 10 do begin & print, i & endfor
```

just works fine, so the test.pro must look like

```
for i = 0, 10 do begin &$
  print, i &$
  print, i^2 &$
```

endfor
exit

(You need the "$" in order to tell IDL to continue in the next line).

Looks ugly but works. The same is true for if clauses.

Best regards,

Peter
