
Subject: Re: idl syntax validation

Posted by [Michael Galloy](#) on Sat, 14 Oct 2006 01:16:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

The below routines have to be in separate files, named
validate_syntax.pro and validate_syntax_helper.pro. Then

```
IDL> print, validate_syntax('foo')
```

```
0
```

```
IDL> print, validate_syntax('validate_syntax')
```

```
1
```

My "normal" trick of using EXECUTE(..., 1) didn't work, so I had to up
the ante to the IDL_IDLBridge.

Mike

--

www.michaelgalloy.com

```
function validate_syntax, routineName
  compile_opt strictarr, hidden

  catch, error
  if (error ne 0) then return, 0

  o = obj_new('IDL_IDLBridge')
  cmd = 'result = validate_syntax_helper(' + routineName + ')"
  o->execute, cmd
  result = o->getVar('result')
  obj_destroy, o
  return, result
end
```

```
function validate_syntax_helper, routineName
  compile_opt strictarr, hidden

  catch, error
  if (error ne 0) then return, 0

  resolve_routine, routineName, /either, /compile_full_file
  return, 1
end
```

Dometz wrote:

> Hi JD,

```

>
> Thanks you for your help... it works pretty well, but I still get some
> errors that I am not able to catch... well, I am able to catch them but
> it prints an error message. How can I get rid of that error message /
> meaning that it wont print any error message?
>
> IDL> validate_routine, ["valid"]
>      1
> IDL> validate_routine, ["foo"]
>
> read_png()
> ^
> % Syntax error.
> At: /home/dometz/test_cases/idl/foo.pro, Line 8
>      0
> IDL> exit
>
>
> function validate_syntax, ROUTINE
>     COMPILE_OPT HIDDEN
>     CATCH, ERROR
>     IF (ERROR NE 0) THEN RETURN, 0
>     RESOLVE_ROUTINE, ROUTINE, /EITHER, /COMPILE_FULL_FILE
>     RETURN, 1
> END
>
> pro FOO
>     read_png()
> END
>
>
> any ideas?
>
> dometz
>
>
>
> JD Smith wrote:
>> On Wed, 11 Oct 2006 15:11:05 -0700, Dometz wrote:
>>
>>> I guess I wasnt clear enough... I mean something that I can run in a
>>> bash script to test if a .pro file has valid syntax.
>>>
>>> something like:
>>> echo '.compile foo.pro' | idl
>>> and then checking if there are any syntax error messages and if there
>>> are act accordingly.
>>>

```

```
>>> But is there a better way than using .compile for this?
>> I guess I didn't do all your homework for you ;). You could use this
>> idea to write a function called check_syntax, ala:
>>
>> function check_syntax,routine
>>   catch,err
>>   if err ne 0 then return,err
>>   resolve_routine,routine,/EITHER,/COMPILE_FULL_FILE
>>   return,0
>> end
>>
>> Then a batch file which uses this function, like do_check.pro:
>>
>> ;=====
>> status=check_syntax((command_line_args())[0])
>> if status ne 0 then print,!ERROR_STATE.MSG
>> exit,STATUS=status
>> ;=====
>>
>> then call, from your shell script:
>>
>> idl -arg routine do_check.pro
>>
>> and check the exit status. You might need to monkey IDL_PATH to ensure
>> it attempts to compile the correct version of the routine, like:
>>
>> idl -IDL_PATH "$(pwd):+/other/libs:<IDL_DEFAULT>" -arg routine do_check.pro
>>
>> JD
>
```
