
Subject: Re: IDL 8.0 bug -- line number of errors not given
Posted by [Paul Van Delst\[1\]](#) on Tue, 12 Oct 2010 22:16:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paulo Penteado wrote:

> On Oct 12, 4:45 pm, Chris Torrence <gorth...@gmail.com> wrote:
>> This is by design. When developing the new graphics, we noticed that
>> many of the error messages had overly-long stack traces, because
>> on_error,2 always dumped out the stack trace from where the error
>> message was triggered. We changed it in IDL 8.0, so that now it only
>> prints out the stack trace from where IDL actually stops execution.
>> For on_error,2 this is the "caller of the program unit that called
>> ON_ERROR".

>
> I was wondering if this was the case, since I agree this was
> desirable. But because I saw no mention to it in the help, and I
> happened to misread the test case above, I thought something could be
> wrong.

>
> Not showing the full trace is consistent with returning to the caller.

<disclaimer>

I do not use ON_ERROR for error handling. I always use CATCH.

</disclaimer>

Not sure if I agree with the consistency you mention. Since the error handler prints out a message it should at least provide the name of the procedure/function in which the error occurred. Without it, there is zero context for the error.

But, one thing I don't think is good programming practice in general is the advice provided in the IDL help for ON_ERROR:

<quote>

This form of error recovery makes debugging a routine difficult because the routine is exited as soon as an error occurs; therefore, [ON_ERROR,2] should be added once the code is completely tested.

</quote>

which, to me, translates to:

Completely test your code. Then change it.

I'm a notorious fat-finger-er when it comes to typing so once I've completely tested something, I'm rather loathe to change it. Particularly when the part being changed is the error handling....which may or may not be able to be fully tested? I'm somewhat confused (and probably paranoid about the subject :o).

Maybe people should start switching to using CATCH instead for their day-to-day error handling?

cheers,

paulv

> That is, the code below behaves as I expected, and agree that should
> be the case:

>

> function test1

> On_error,2

> a = bindgen(32)

> c = long(a,30,1)

> return,1

> end

>

> pro test

> print,test1()

> return

> end

>

> IDL> test

> % Compiled module: TEST.

> % Specified offset to array is out of range: A.

> % Execution halted at: TEST 9 /home/penteado/idl/

> test.pro

> % \$MAIN\$

>

> As the trace shows where the place where the routine with 'on_error,2'

> was called.
