

---

Subject: Re: IDL 8.0 bug -- line number of errors not given  
Posted by [David Fanning](#) on Tue, 12 Oct 2010 22:51:42 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Chris Torrence writes:

> 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".

This turns out not be as big a problem for me as I thought  
it might be. The method I use to catch most errors in my  
programs:

```
Catch, theError
IF theError NE 0 THEN BEGIN
    Catch, /Cancel
    void = Error_Message()
    RETURN
END
```

still seems to work normally with the "new" ON\_ERROR  
behavior. Error\_Message prints out the proper trace to the  
error.

> The general philosophy is that on\_error,2 should be used for "library"  
> routines, where the caller should not need to care about the internal  
> workings of the library. If you are writing your own routines, or are  
> debugging an existing routine, then I would recommend that you disable  
> the on\_error,2 command until you have completed your routine and are  
> ready to "release" it. I think the IDL help for ON\_ERROR mentions  
> this.

I'm not sure who's "general philosophy" we are talking about here,  
but I would say my "general philosophy" is not to change the way  
software works unless there is some extremely compelling reason  
for it. Maybe scaring the bejesus out of customers with long error  
messages from over-complicated software is a compelling reason.  
I couldn't say.

But, general philosophy be damned, a lot of people rely on error  
handling to remain the same from one version of IDL  
to the next. I would say that is a reasonable expectation.

It's one thing to be the 500-pound gorilla and stomp all over established IDL code when you decide to name your programs. It's something else to make changes that break a lot of established code. Can anyone *\*really\** think this is good for business? Even *\*new\** business?

I still think this change is a mistake.

Maybe this should be implemented as a new keyword:

```
On_Error, 2, /POLYANNA_MESSAGE  
On_Error, 2, /BRUTAL_TRUTH
```

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

---