

---

Subject: Re: IDL 8.0 bug -- line number of errors not given  
Posted by [Paul Van Delst\[1\]](#) on Wed, 13 Oct 2010 14:50:37 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

wlandsman wrote:

```
> On Oct 12, 6:16 pm, Paul van Delst <paul.vande...@noaa.gov> wrote:
>
>> Maybe people should start switching to using CATCH instead for their day-to-day error
handling?
>
> OK, here's my singleton reason for not using CATCH.  If I use
> ON_ERROR I just add 1 line of code at the beginning of each procedure
>
>   On_error, 2
>
> But if I use CATCH I need to add a whole paragraph somewhere (where?)
> in the code
>
>   Catch, theError
>   IF theError NE 0 THEN BEGIN
>     Catch, /Cancel
>     void = Error_Message()
>     RETURN
>   END
```

Oh yes, I agree - using CATCH is a more verbose method. Applying the DRY[\*] principle, what I do for a particular application is to create include files for functions and procedures (say "func\_err\_handler.pro" and "pro\_err\_handler.pro") and then simply include them right at the top of functions/procedures. E.g:

```
function testfunc, a, b, c
  @func_err_handler
  ....
end
```

```
pro testpro, a, b, c
  @pro_err_handler
  ....
end
```

I don't think it's an ideal solution (it's still a bit "wet") but I have isolated everything CATCH-y in two files. Easy to change if I need to.

With a bit more thought, one could probably come up with content for these include files that are applicable throughout ones entire library of routines.

> So what are the advantages of using CATCH?

It's not "ON\_ERROR, 2"? :o)

> (This is not a trick

> question -- I've had a mental block about how to use CATCH.) --Wayne

cheers,

paulv

[\*] <http://c2.com/cgi/wiki?DontRepeatYourself>

---