Subject: Re: IDL 8.0 bug -- line number of errors not given
Posted by David Fanning on Wed, 13 Oct 2010 03:02:44 GMT

View Forum Message <> Reply to Message

Wayne Landsman writes:

> OK, here's my simpleton 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

You add it to the code just before the part of the code that
is going to generate the error. (The very first line in most
people's code!) I usually add it as the first piece of code after
Compile_Opt idl2.

> So what are the advantages of using CATCH?   (This is not a trick
> question -- I've had a mental block about how to use CATCH.) --Wayne

One of the advantages is that it catches all kinds of errors
(I/O, run-time, etc.) Another is that you can put catches anywhere
you anticipate an error occurring, and you can even fix errors and
continue program execution. A HUGE advantage in widget programs is
that you can keep widget programs running and alive, even when errors
occur. (This is occasionally a BAD thing, but more often than not a
GOOD thing.)

I use Catches because I can combine them with Error_Message to
get nicely formatted error results and user notification of errors.
The traceback information is accurate and complete (even in IDL 8
in the cases I've tested so far, and even when used to catch errors
coming from ON_ERROR,2 conditions in other modules).

The most compelling reason, of course, is that it will soon be
the only way to find errors in many IDL programs! ;-)

Cheers,

David

P.S. Let's just say I am beginning to miss those idyllic days
of silent error handlers in iTools!

--
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.")