
Subject: Re: Error handling and tracebacks

Posted by [David Fanning](#) on Wed, 21 Jul 2004 23:26:47 GMT

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

Michael Wallace writes:

> How do I disable tracebacks associated with the message procedure?

My goodness, you are having a tough day, today. :-)

> Before the programming gurus yell at me for potentially hiding bugs my
> code, let me explain why I want to do this. I have a library of
> procedures and in certain cases I am creating my own errors via message
> which are several calls down in the stack. When the error comes out,
> there is a long traceback, but I only want the end user to see the
> message itself and not the entire traceback since the traceback is going
> several layers down into code that the end user shouldn't even be aware
> of. I do want the user to see the error, but not all the files and line
> numbers.

I think your only hope here is to CATCH the errors
thrown by MESSAGE yourself, then prune the error traceback
(which you maybe save in a string variable, with the OUTPUT
keyword) and pass it on to the user:

```
Catch, theError
IF theError NE 0 THEN BEGIN
  Help, /Traceback, Output=helpString
  Prune, helpString
  Print, helpString
ENDIF
```

I don't have any idea what the PRUNE procedure
might look like. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

Subject: Re: Error handling and tracebacks

Posted by [Michael Wallace](#) on Thu, 22 Jul 2004 04:52:31 GMT

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

>> How do I disable tracebacks associated with the message procedure?
>
>
> My goodness, you are having a tough day, today. :-)
>

It's not very often you get two IDL questions from completely different ends of the spectrum at the same time on the same day, but it happens. Oh, well. Part of the business, I suppose. :-)

> I think your only hope here is to CATCH the errors
> thrown by MESSAGE yourself, then prune the error traceback
> (which you maybe save in a string variable, with the OUTPUT
> keyword) and pass it on to the user:
>
> Catch, theError
> IF theError NE 0 THEN BEGIN
> Help, /Traceback, Output=helpString
> Prune, helpString
> Print, helpString
> ENDIF

I played with catching the error some, but I really didn't like results too much. I just don't have enough time to play with things. Really, the problem is of no concern to me, but it seems that I'm going to have to go on an end user education project. The problem is that some people using my stuff are too quick to think that a bug has come up in my code.

When they see the traceback, 9 times out of 10, they don't even look at the message I put out at the top because they think that some crazy bug caused something screwy to happen. (Almost all of my IDL programs are on the command line, BTW) Invariably, they have simply mistyped something in the call into my code.

In short, I thought I might rid myself of this dilemma if I could somehow show them the error message with no trace. This is more *normal* to them and as such they'd actually *read* the error message like they should have in the first place.

Sigh.

> I don't have any idea what the PRUNE procedure
> might look like. :-(

Might it look like the PLUM procedure after having sat out in the sun too long?

-Mike

Subject: Re: Error handling and tracebacks
Posted by [David Fanning](#) on Thu, 22 Jul 2004 05:08:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Michael Wallace writes:

> In short, I thought I might rid myself of this dilemma if I could
> somehow show them the error message with no trace. This is more
> *normal* to them and as such they'd actually *read* the error message
> like they should have in the first place.

Well, heck, something like this will work then:

```
Catch, theError
IF theError NE 0 THEN BEGIN
  Print, !Error_State.Msg
  RETURN
ENDIF
```

Now, throw the error with Message:

```
Message, 'Whoops! Did you read the docs?'
```

You get something like this:

```
%YourProgram: Whoops! Did you read the docs?
```

That's about as simple as it gets. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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