
Subject: Catch Error Handling Traceback

Posted by [davidf](#) on Fri, 30 Apr 1999 07:00:00 GMT

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

Folks,

One of the main things that makes it impossible for me to give up IDL is that I learn something new about it almost every week. Sometimes what I learn is subtle and like a fine distinction. Sometimes it's more like a Mac truck. But I'm always humbled by how much I don't know about it and how much more there is to learn.

Occasionally, what I learn surprises me very much. That is the case with what I report today, which could--for all I know--be common knowledge for all of you, since it appears to be a feature that has been in IDL for some time. (I confess that I have even read the appropriate documentation for this feature several times without understanding the significance of what I read. In fact, I'm quite sure I could have read the documentation several thousand times without understanding the significance of what I read, but that probably has more to do with me than with the documentation.)

In any case, what I have wanted for some time is the ability to write a CATCH error handler that could provide an error trace. In particular, I want to indicate to the user exactly what the error was and which line in the code caused the error. For the life of me, I could not figure out how to get the line number. My error handlers could provide a traceback, but the line number was always pointing to the line in the CATCH error handling code and not to the line that actually caused the error.

But thanks to Pavel Romashkin (who sensibly ignored my learned advice in this matter and proceeded to find out how things *really* worked) and the technical support people at RSI I now know that IDL has had the ability to provide the information I wanted all the time. It is accessed through the LAST_MESSAGE keyword to the HELP command.

Here is a little test program that shows how this is done:

PRO TEST

```

Catch, theError
IF theError NE 0 THEN BEGIN
    Catch, /Cancel
    Help, /Last_Message, Output=theErrorMessage
    FOR j=0,N_Elements(theErrorMessage)-1 DO BEGIN
        Print, theErrorMessage[j]
    ENDFOR
    RETURN
ENDIF

Print, xxx
END

```

Running the program results in this output:

```

IDL> test
% PRINT: Variable is undefined: XXX.
% Execution halted at: TEST 13 C:\RSI\IDL52\DAVID\test.pro
%                $MAIN$

```

This was the last piece of information I needed to make my little device-independent ERROR_MESSAGE program I posted here the other day complete. I've updated that program and made it available on my web page for those who are interested. You can find it at:

http://www.dfanning.com/programs/error_message.pro

Happy Programming!

David

--

David Fanning, Ph.D.
 Fanning Software Consulting
 Phone: 970-221-0438 E-Mail: davidf@dfanning.com
 Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
 Toll-Free IDL Book Orders: 1-888-461-0155
