
Subject: Transparent error handling Was: GLOBAL styles

Posted by [davidf](#) on Sun, 25 Apr 1999 07:00:00 GMT

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

Martin Schultz (mgs@io.harvard.edu) writes:

- > wouldn't it be nice to have a system variable that controls
- > the display of messages throughout the session? I believe
- > there are some people who prefer error messages displayed
- > as dialog boxes, while others prefer a "unix like" message
- > display in the log window. Currently this involves two
- > different routines DIALOG_MESSAGE or MESSAGE...

This is a good idea. I like to write code that can be executed everywhere. In particular, the code should work on my display device, in the PostScript device, and in the Z graphics buffer. But, of course, if that is to be the case I can't use an error message (i.e, DIALOG_MESSAGE) that opens a window. And yet, I prefer this kind of error message when I am working at the display.

In the absence of this proposed system variable, here is a little function, named ERROR_MESSAGE, that allows me to format the error message as normal, but uses DIALOG_MESSAGE or MESSAGE to display the message, as appropriate for the current graphics device. Setting the TRACEBACK keyword will allow traceback information to be printed to the command output window no matter how the error message is displayed for the user.

It is called like this:

```
ok = Error_Message('Whoops. Error', /Traceback)
```

Cheers,

David

--

```
FUNCTION ERROR_MESSAGE, theMessage, Traceback=traceback
```

```
; This function displays an error message to the user by  
; using DIALOG_MESSAGE if widgets are supported and MESSAGE  
; otherwise. Set the TRACEBACK keyword to get traceback  
; information from the error.
```

```
; Are widgets supported?
```

```
widgetsSupported = ((!D.Flags AND 65536L) NE 0)
IF widgetsSupported THEN answer = Dialog_Message(theMessage) ELSE BEGIN
    Message, theMessage, /Continue, /NoPrint, /NoName, /NoPrefix
    Help, Calls=callStack
    callingRoutine = (Str_Sep(StrCompress(callStack[1])," "))[0]
    Print, '%' + callingRoutine + ': ' + theMessage
    answer = 'OK'
ENDELSE
```

; Provide traceback information if requested.

```
IF Keyword_Set(traceback) THEN BEGIN
    Help, Calls=callStack
    callingRoutine = (Str_Sep(StrCompress(callStack[1])," "))[0]
    IF widgetsSupported THEN Print, '%' + callingRoutine + ': ' + theMessage
    Print, 'Execution halted at: ', callStack[1]
    numCalls = N_Elements(callStack)
    IF numCalls GT 2 THEN FOR j=2,numCalls-1 DO Print, callStack[j]
ENDIF
```

RETURN, answer

END

--

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
