Subject: Re: Error handling by build-in IDL routines
Posted by Vapuser on Tue, 16 Mar 1999 08:00:00 GMT
View Forum Message <> Reply to Message

"R.Bauer" <R.Bauer@fz-juelich.de> writes:

```
>  Try this!
>
>  PRO t2,test
>
>    HELP,/recall,output=output
>    for_test=(STR_SEP(output[1],','))[1]
>    varsize=SIZE(routine_names(for_test,fetch=-1),/type)
>    VarValue = Routine_Names(for_test, FETCH=-1 )
>
>    IF test EQ varvalue THEN IF varsize NE 4 THEN $
>     MESSAGE,'Expression must be of type FLOAT:'+for_test,/info
>
>  END
>
>  dd='dummy'
>  t2,dd
>
>  % T2: Expression must be of type FLOAT:dd
>
>
>  R.Bauer
```

I don't think this work in a procedure.

If anyone out there in RSI land is listening...

It would be nice to have a function like the Perl package Carp.pm, which reports errors from the line number of the invocation of Carp's calling routine. So, say you have a perl routine foo which reports some error by calling carp. The linenumber given in the error message emitted by Carp is the line at which foo is called, not the line at which Carp is called. That way, you can write error handling code that doesn't have to keep track of the stack, and depend on the output from help.