Matthew Argall writes:

> I guess my point was to make a case WITH a ton of errors to create the cascade and see how CATCH works. I could check to see if "a" and "b" are defined (one of your earlier points) but I would still have to use RETURN in CALL_ADD instead of MESSAGE.
>
> If I am 6 programs deep into the stack, I do not want to generate 6 error messages as my Catch blocks work their way back up.

I would say your concept of "error handling" just needs to grow a
little. If you are calling a routine that can fail and return a NULL
value, then part of your error handing, outside of Catch and ON_Error
and all the rest of it, is to check the return value for the NULL
condition and do something reasonable. If you NEED that value in the
rest of the program, then you HAVE to check that you have it. Otherwise,
errors *will* cascade.

Anyway, the only programs I know of that are 6 programs deep on the
stack are function graphics routines, and I think they routinely use
silent error handlers to deal with that. ;-)

Cheers,

David

P.S. Maybe it only *seems* like they use silent error handlers.
In any case, those routines are pretty much impossible for a layman to
debug.

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")