
Subject: Re: Destroying objects

Posted by [David Fanning](#) on Mon, 16 Feb 2004 22:06:35 GMT

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

JD Smith writes:

> I wonder if you can expand on this error handling technique a bit.
> One feature I like to have is the ability for errors to "just work",
> regardless of if the object is being used on the command-line, or via
> its GUI. In the latter case, errors should be trapped and displayed
> in a pop-up. In the former case, they should call MESSAGE to print to
> the command line and exit. I've managed it so far by having a
> super-class which implements Error, Warning, and other methods, which
> checks to see if you are in a command-line invocation or not, and uses
> message or dialog_message as appropriate. Suitable use of CATCH
> allows deep errors in other routines further down the stack (ones you
> didn't write, for instance) to be handled in the same way, but that
> requires some care to "set the trap" when necessary. This is somewhat
> similar to the way XManager can catch and disregard errors (as it does
> by default), but with reporting, and valid even when XManager isn't
> running.

I would have thought you knew me well enough by now, JD,
to realize that if I don't immediately provide the IDL
code to solve a problem I'm either trying to get rich (ha!)
or the solution is so simpleminded I prefer to not risk
certain embarrassment. Alas, it is the latter in this case. :-(

We don't do anything fancy. All our methods contain standard
error handler code in them, added with a @error_handler
sort of call. (We originally had one code fragment because I figured
out a really slick way to tell if I was in a procedure or function
and I could either return something or not, but the method
used an EXECUTE function. When we wanted to run on the VM,
we had to go to two error handlers: one for procedures
and one for functions. What a shame!) The error handler
simply sets up a CATCH for the method. If an error is caught,
it is dispatched to our ERROR method, which is attached to
the "atom" object all objects inherit.

When we get an error we can handle in our error handler,
we add a "handled" to the front of the error message.
(I told you this was simpleminded.) As the error propagates
back up the object chain, if the error handler sees a "handled"
in front of the message it returns immediately.

I can't think of a way to tell if the object is running
at the IDL command line or not (I've shown you mine, you

should show me yours), but our error handler can write to a log file (if there is one) or to the display. Typically, we issue a pop-up dialog (DIALOG_MESSAGE) with a short error message, and write a better formatted traceback of the error in the command output log of the display.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting

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