
Subject: Re: eval

Posted by [David Fanning](#) on Wed, 04 Oct 2006 21:21:48 GMT

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

Dometz writes:

> So for example for OPENU, I want to create a wrapper function that will
> do some logging and then call the original OPENU:
>
> example:
> PRO LOG_OPENU, UNIT, FILE, APPEND=APPEND, COMPRESS=COMPRESS,
> BUFSIZE=BUFSIZE, DELETE=DELETE, ERROR=ERROR,
> F77_UNFORMATTED=F77_UNFORMATTED, GET_LUN=GET_LUN, MORE=MORE,
> NOEXPAND_PATH=NOEXPAND_PATH, STDIO=STDIO, SWAP_ENDIAN=SWAP_ENDIAN,
> \$
> SWAP_IF_BIG_ENDIAN=SWAP_IF_BIG_ENDIAN,
> SWAP_IF_LITTLE_ENDIAN=SWAP_IF_LITTLE_ENDIAN, VAX_FLOAT=VAX_FLOAT,
> WIDTH=WIDTH, XDR=XDR, RAWIO=RAWIO
> print, FILE
> print, SWAP_IF_BIG_ENDIAN
> ;print etc...
> ;Now here would like to call the original method but I have to call
> it only with the named arguments that are set
> OPENU, ...?
>
> So, either I could have a LOOOONG list of if or case statements or I
> maybe I could use _EXTRA?

This is exactly what _EXTRA is used for. Note that if you want to use output keywords and get information *back* from your LOG_OPENU procedure, you should use _REF_EXTRA instead of _EXTRA. And if you want to be sure that ONLY keywords for OPENU are passed in, you should use _STRICT_EXTRA. I just noticed there is no _STRICT_REF_EXTRA, but I guess I've never really needed it, so probably that's why. :-)

> Or:
> I would need to build the method call dynamically:
> 1) Is there a hack around this so that I could use call_function,
> call_procedure or call_method dynamically?

There is no reason to use CALL_PROCEDURE here, since you know *exactly* what procedure you want to call: OPENU. You might use CALL_PROCEDURE if you didn't know until run-time whether you wanted to call OPENU, or OPENR, or OPENW, for example.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

Sepore ma de ni thui. ("Perhaps thou speakest truth.")
