
Subject: Re: Error Handling Advice

Posted by [David Fanning](#) on Fri, 11 Jan 2013 22:51:30 GMT

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

Matthew Argall writes:

> I have been trying to pick up good programming practices by copying what other people do. What I have not been able to figure out, though, is error handling.

>

> Right now, I tend to check all of the input parameters and keywords to make sure they are suitable for the program that follows. But then again, there is the whole RTFM argument. When do you check a specific input parameter and when do you just throw/catch an error and return to main?

Here is what I do.

1. I assume no one can read. At least there is no evidence for it.
2. I include a CATCH error handler in every program module I write.
3. The only exception to 2 is for utility functions, for which I use an On_Error,2 condition to return to the caller of the function, which presumable DOES have a CATCH error handler.
4. I NEVER return to the main IDL level. I ALWAYS return to the caller of my program, hoping whoever wrote the damn thing that's calling my program knows enough to include error handling it it.
5. I make sure I check EVERY SINGLE ONE of my parameters and keywords to see that they are defined as *something*. If I intend to use them, I intend to have them defined.
6. The only exception to 5 is if I know that the program I'm calling checks the parameter or keyword and has decent error handling (i.e, I wrote the darn thing), then I sometimes pass the parameter or keyword along, knowing it will eventually get checked before it is used.
7. My practice is to make positional parameters required arguments, and keywords parameters optional arguments. I sometimes make a positional parameter an optional argument, if there is a compelling reason to do so, but I NEVER make a keyword parameter a required argument. Seeing the latter in other people's code makes me crazy.
8. I try to come up with values for optional parameters that "make sense". For example, one of my pet peeves for IDL 8 graphics is that when you save a graphics window as a PNG file, the PNG file is the size of my office wall. I don't think anyone wants a file that large, so I think the default value for the RESOLUTION keyword is poorly chosen. At

least I notice that on every program I've seen from ExelisVis lately this keyword is used to get a PNG file that is a smaller size. That, to me, makes the RESOLUTION keyword a required keyword, no matter how the code is written. :-)

Cheers,

David

--

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.")
