

---

Subject: Re: Passing optional parameters through a wrapper routine  
Posted by [davidf](#) on Tue, 08 Feb 2000 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Kenneth P. Bowman (bowman@null.tamu.edu) writes:

```
> (David Fanning) wrote:
>
>> Hard to know what TVRD is doing
>
> If a program with optional parameters did something like this:
>
> PRO OPT_PARAM, p1, p2, p3
>
> IF (N_PARAMS() EQ 2) OR (N_ELEMENTS(p3) EQ 0)) THEN
>   ... do the default thing for p3
> ENDIF
>
> then it would handle "wrapping", no?
```

I'm not sure I understand this question, Ken. In general, with most built-in IDL routines, you will get an error by passing an undefined variable. You could argue (quite convincingly, I imagine) that built-in IDL routines should handle this situation more gracefully, but that is the situation we find ourselves with. You can certainly make your own IDL routines more graceful when you write *\*them\**, but there is little you can do about what is already there.

Having said that, it absolutely behooves you to check each and every variable you plan to use in your program to make sure you have a defined variable at the time you use it. This is normally done with the `N_Elements` function, since this function returns a 0 if its argument is undefined.

You would probably NOT use the odd syntax above for the simple reason that it would fail if the number of parameters was 0 or 1. What you probably *would* do if p1 and p2 are required parameters and p3 is optional, is something like this:

```
IF N_Params() LT 2 THEN Message, 'Whoops, two parameters required'
IF N_Elements(p3) EQ 0 THEN p3 = p3DefaultValue
```

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438 E-Mail: [davidf@dfanning.com](mailto:davidf@dfanning.com)

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

Toll-Free IDL Book Orders: 1-888-461-0155

---