Subject: Re: Passing optional parameters through a wrapper routine
Posted by bowman on Tue, 08 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

In article <MPG.130a910d141f1260989a21@news.frii.com>, davidf@dfanning.com (David Fanning) wrote:

> 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.

I admit to a certain conceptual confusion between undefined and optional parameters. If I omit an optional parameter, I have not defined it. So what's the difference between passing an undefined optional parameter and omitting it? Obviously, in practice the difference is in how you test for whether the parameter is "there" or not.

> 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

My syntax is not odd! :-) (Although it should be LT rather than EQ.)
If you use optional parameters, you need to test for each possible set of optional parameters and the existence of mandatory parameters.

IF N_Params() LT 1 THEN Message, 'Whoops, p1 required'
IF (N_PARAMS() LT 2) OR (N_ELEMENTS(p2) EQ 0)) THEN do the default thing for p2
IF (N_PARAMS() LT 3) OR (N_ELEMENTS(p3) EQ 0)) THEN do the default thing for p3

This does not add any fundamental complexity to handling optional parameters.

Your earlier solution of setting all the optional parameters in the calling requires me to figure out all of the default values for those parameters. This obviates the whole point of optional parameters (convenient default values). If optional parameters are there as a convenience, make them convenient.

On to more productive topics ... :-)

Ken

k-bowman@tamu.edu

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
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Passing optional parameters through a wrapper routine
Posted by bowman on Tue, 08 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

In article <MPG.130a39952208d999989a1f@news.frii.com>, davidf@dfanning.com (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?

Ken

Subject: Re: Passing optional parameters through a wrapper routine
Posted by Liam E. Gumley on Tue, 08 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Kenneth P. Bowman <bowman@null.edu> wrote in message news:bowman-0802001505360001@bowman.tamu.edu...
> The _EXTRA keyword handles passing keyword parameters through a wrapper
> routine, but what about optional parameters? If my wrapper for TVRD has
> one of my own (mandatory) parameters, do I have to do the following?
>
> PRO TVRD_WRAPPER, my_arg, x0, y0, nx, ny, channel, _EXTRA = tvrd_keywords
>

> CASE N_PARAMS() OF
> 1 : image = TVRD(_EXTRA = tvrd_keywords)
> 2 : image = TVRD(x0, _EXTRA = tvrd_keywords)
> 3 : image = TVRD(x0, y0, _EXTRA = tvrd_keywords)
> 4 : image = TVRD(x0, y0, nx, _EXTRA = tvrd_keywords)
> 5 : image = TVRD(x0, y0, nx, _EXTRA = tvrd_keywords)
> 6 : image = TVRD(x0, y0, nx, ny, channel, _EXTRA = tvrd_keywords)
> ELSE : MESSAGE, 'Incorrect number of arguments.'
> ENDCASE
> ...
>
> If I just try to pass undefined arguments through my wrapper, I get
> 'undefined variable' errors. (Is TVRD just counting the number of passed
> parameters and not checking to see if they are defined?)

Ken,

In cases like this I usually cross my arms, put my feet up on the desk, look
a person in the eye, and ask:

"How about telling me what the problem is?"

In other words, I'd like to know why you're writing a wrapper for TVRD. I've
done it myself for a number of reasons...

Cheers,
Liam.

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.edu) writes:

> The _EXTRA keyword handles passing keyword parameters through a wrapper
> routine, but what about optional parameters?

_Extra (and _Ref_Extra, for that matter) apply only to *KEYWORD*
parameters. Positional parameters do not count.

> If my wrapper for TVRD has
> one of my own (mandatory) parameters, do I have to do the following?
>
> PRO TVRD_WRAPPER, my_arg, x0, y0, nx, ny, channel, _EXTRA = tvrd_keywords
>
> CASE N_PARAMS() OF
> 1 : image = TVRD(_EXTRA = tvrd_keywords)

```
>     2 : image = TVRD(x0,                  _EXTRA = tvrd_keywords)
>     3 : image = TVRD(x0, y0,              _EXTRA = tvrd_keywords)
>     4 : image = TVRD(x0, y0, nx,          _EXTRA = tvrd_keywords)
>     5 : image = TVRD(x0, y0, nx,          _EXTRA = tvrd_keywords)
>     6 : image = TVRD(x0, y0, nx, ny, channel, _EXTRA = tvrd_keywords)
>     ELSE : MESSAGE, 'Incorrect number of arguments.'
>  ENDCASE
>  ...
```

That's one way to do it. :-)

Another way is to define any parameter that is NOT passed
in. That might make more sense, since this is standard
operating procedure for any optional parameter.

```
IF N_Elements(y0) EQ 0 THEN y0 = 0
IF N_Elements(nx) EQ 0 THEN nx = 100
etc.
```

> (Is TVRD just counting the number of passed
> parameters and not checking to see if they are defined?)

Hard to know what TVRD is doing, but what *you*
better be doing is making sure any variable you
plan to use is defined. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Passing optional parameters through a wrapper routine
Posted by Mark Hadfield on Wed, 09 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Kenneth P. Bowman <bowman@null.tamu.edu> wrote in message
news:bowman-0802002243470001@tl6-218-199.tca.net...
> In article <MPG.130a910d141f1260989a21@news.frii.com>, davidf@dfanning.com
> (David Fanning) wrote:
>
>> 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.

Responding to the excerpt from David's message (which I haven't seen in full yet)..

I don't see anything generally wrong with passing variables on without knowing what they are or whether they are defined. That's what a wrapper routine does -- it concerns itself with some subset of the information passed to it and let's the "wrappee" deal with the rest. RSI in their wisdom invented inheritance mechanisms to do this with keywords. For a general wrapper routine with an unknown number of positional parameters I favour the "case n_params()" syntax originally proposed by Kenneth. If I get a chance tomorrow I may illustrate this using my (almost completely) general wrapper routines that report on the execution time of the wrappee.

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield/
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand