Subject: Re: wrapper functions
Posted by David Fanning on Thu, 05 Oct 2006 23:39:36 GMT
View Forum Message <> Reply to Message

Dometz writes:

> So, I am trying to create wrapper functions with the use of _EXTRA,
> _REF_EXTRA. I am not sure what I am doing wrong or if _REF_EXTRA can
> only be used for named arguments.
>
> Why does the following give me the attached error?
>
> LOG_READ_JPEG, './testall/read_jpeg.jpeg', a
>
>
> PRO LOG_READ_JPEG, _REF_EXTRA=e
> COMPILE_OPT HIDDEN
> print, "foobar"
> READ_JPEG, _EXTRA=e
> END
>
>
> Error message:
> READ_JPEG, _EXTRA=e
> ^
> % READ_JPEG: Incorrect number of arguments.

The _EXTRA mechanism is called *keyword* inheritance,
since only keyword parameters, not *positional* parameters
can be collected and passed with this approach. You
will have to define the positional parameters for READ_JPEG
in your wrapper routine.

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.")

Subject: Re: wrapper functions
Posted by Dominic Metzger on Thu, 05 Oct 2006 23:51:42 GMT
View Forum Message <> Reply to Message

Hah, I see... is there a way that I wont have to define the positional

parameters since otherwise I will have to check again which ones were
passed in?

Example
READ_PICT, Filename, Image [, R, G, B]

so, here I would have to check if R was passed in, if G was passed in,
if B was passed in to see if I need to include them in the call inside
of the wrapper function.

best regards,

dometz

David Fanning wrote:
> Dometz writes:
>
>> So, I am trying to create wrapper functions with the use of _EXTRA,
>> _REF_EXTRA. I am not sure what I am doing wrong or if _REF_EXTRA can
>> only be used for named arguments.
>>
>> Why does the following give me the attached error?
>>
>> LOG_READ_JPEG, './testall/read_jpeg.jpeg', a
>>
>>
>> PRO LOG_READ_JPEG, _REF_EXTRA=e
>> COMPILE_OPT HIDDEN
>> print, "foobar"
>> READ_JPEG, _EXTRA=e
>> END
>>
>>
>> Error message:
>> READ_JPEG, _EXTRA=e
>> ^
>> % READ_JPEG: Incorrect number of arguments.
>
> The _EXTRA mechanism is called *keyword* inheritance,
> since only keyword parameters, not *positional* parameters
> can be collected and passed with this approach. You
> will have to define the positional parameters for READ_JPEG
> in your wrapper routine.
>
> 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.")

Subject: Re: wrapper functions
Posted by Robbie on Fri, 06 Oct 2006 00:13:27 GMT
View Forum Message <> Reply to Message

Spot on,

I tend to use something like

```
pro READ_PICT_WRAPPER, Filename, Image, R, G, B
case (N_PARAMS()) of
5: READ_PICT, Filename, Image, R, G, B
4: READ_PICT, Filename, Image, R, G
3: READ_PICT, Filename, Image, R
2: READ_PICT, Filename, Image
else: message, "READ_PICT must have at least two parameters"
endcase
end
```

Robbie

Subject: Re: wrapper functions
Posted by David Fanning on Fri, 06 Oct 2006 00:52:05 GMT
View Forum Message <> Reply to Message

Dometz writes:

> Hah, I see... is there a way that I wont have to define the positional
> parameters since otherwise I will have to check again which ones were
> passed in?
>
> Example
> READ_PICT, Filename, Image [, R, G, B]
>
> so, here I would have to check if R was passed in, if G was passed in,
> if B was passed in to see if I need to include them in the call inside
> of the wrapper function.

If you don't want to rely on the error handling of the
routine you are writing the wrapper for, you need to
check the required positional parameters (filename,
and image in this case). I would define the optional
positional parameters, but I wouldn't bother checking
them, since the routine you are calling will do that.
(At least if it is written half-way decently. :-)

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.")

Subject: Re: wrapper functions
Posted by Robbie on Fri, 06 Oct 2006 01:14:39 GMT
View Forum Message <> Reply to Message

<language_lawyer>

> (At least if it is written half-way decently. :-)
I disagree, procedures should not use N_ELEMENTS() to check for
UNDEFINED positional parameters. It should be presumed that the
positional parameters are defined according to the documentation,
anything else should throw an error.

If you want to write a procedure which has more relaxed conditions then
I would recommend using keywords.

Don't most IDL procedures use N_PARAMS() to check for positional
parameters?
This means that there is a difference between calling
READ_PICT, Filename, Image, R, G, B
and
READ_PICT, Filename, Image
regardless of whether R,G,B are defined or not

</language_lawyer>

Subject: Re: wrapper functions
Posted by David Fanning on Fri, 06 Oct 2006 01:28:09 GMT

Robbie writes:

> I disagree, procedures should not use N_ELEMENTS() to check for
> UNDEFINED positional parameters.

Say what!? I would argue that procedures should ALWAYS
use N_ELEMENTS to check for undefined parameters. You
should use N_ELEMENTS *everywhere*!

> It should be presumed that the
> positional parameters are defined according to the documentation,
> anything else should throw an error.

Even I'm not that anal, Robbie. Do you really do this?
Seriously? It's a LOT of work for almost no payoff. Do
you check data type, too?

> If you want to write a procedure which has more relaxed conditions then
> I would recommend using keywords.
>
> Don't most IDL procedures use N_PARAMS() to check for positional
> parameters?
> This means that there is a difference between calling
> READ_PICT, Filename, Image, R, G, B
> and
> READ_PICT, Filename, Image
> regardless of whether R,G,B are defined or not

I'm not a big fan of optional positional parameters, either.
My rule of thumb is that required parameters are positional
parameters and optional parameters are keyword parameters,
unless you have a VERY good reason for doing something else.
And then, of course, if you write a lot of objects you soon
learn that life can be made a LOT easier if you just do away
with positional parameters entirely.

But I never go ANYWHERE without my good ol' N_ELEMENTS by
my side! :-)

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.")

Subject: Re: wrapper functions
Posted by David Fanning on Fri, 06 Oct 2006 01:35:24 GMT
View Forum Message <> Reply to Message

Robbie writes:

> Don't most IDL procedures use N_PARAMS() to check for positional
> parameters?

I think I've used N_PARAMS() about a dozen times in my IDL
lifetime. :-)

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.")

Subject: Re: wrapper functions
Posted by Robbie on Fri, 06 Oct 2006 02:01:26 GMT
View Forum Message <> Reply to Message

> Even I'm not that anal, Robbie. Do you really do this?
> Seriously? It's a LOT of work for almost no payoff. Do
> you check data type, too?

Sorry, I didn't mean "throw an error" as in error checking every
parameter. I just meant that setting a positional parameter to
UNDEFINED usually results in an error on the first line where the
parameter is used an arithmetic.

I just did a search for N_PARAMS in <IDL_DIR>. There are 164 (15.9%)
library routines which use N_PARAMS() and plenty of them are sitting in
the iTools directory.

Mate, I'd get on board. The trains leaving soon :-)

Subject: Re: wrapper functions
Posted by David Fanning on Fri, 06 Oct 2006 02:11:21 GMT
View Forum Message <> Reply to Message

Robbie writes:

> Mate, I'd get on board. The trains leaving soon :-)

Mate, you start writing code like you find in the IDL
directories and a train wreck in the least of your worries! :-)

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.")

Subject: Re: wrapper functions
Posted by Kenneth P. Bowman on Fri, 06 Oct 2006 02:28:38 GMT
View Forum Message <> Reply to Message

In article <MPG.1f8f60c52b1f6f4b989d05@news.frii.com>,
David Fanning <davidf@dfanning.com> wrote:

> Robbie writes:
>
>> Don't most IDL procedures use N_PARAMS() to check for positional
>> parameters?
>
> I think I've used N_PARAMS() about a dozen times in my IDL
> lifetime. :-)

I find that N_PARAMS() with SWITCH is very handy for setting defaults in
procedures where optional positional parameters make sense.

```
PRO BLAH, A, B, C

SWITCH N_PARAMS() OF
  0 : A = ...
  1 : B = ...
  2 : C = ...
ENDSWITCH
```

Ken Bowman

Subject: Re: wrapper functions
Posted by David Fanning on Fri, 06 Oct 2006 03:04:47 GMT

Kenneth P. Bowman writes:

> I find that N_PARAMS() with SWITCH is very handy for setting defaults in
> procedures where optional positional parameters make sense.
>
> PRO BLAH, A, B, C
>
> SWITCH N_PARAMS() OF
> 0 : A = ...
> 1 : B = ...
> 2 : C = ...
> ENDSWITCH

In thinking about it, I guess I use N_ELEMENTS for
all my parameters because I can throw better errors
since I know *exactly* which parameter is missing:

```
  IF N_Elements(foo) EQ 0 THEN $
     Message, 'Argument FOO must be a 2D array.'
```

I can also trap the error of the user calling my
routine with an undefined variable, rather than with
a real variable. This happens more often than you would
think in IDL programming courses. N_PARAMS, of course,
can't tell you much of anything about the parameter.

Cheers,

David

P.S. And don't even get me started on the mostly
mis-used KEYWORD_SET. I see that function misused
almost constantly in IDL programs to perform the
function of N_ELEMENTS. Folks, it DOESN'T!! :-)

--
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.")

Subject: Re: wrapper functions
Posted by Dominic Metzger on Fri, 06 Oct 2006 05:32:14 GMT

Wow, I got you guys started on something here... thanks for your input.

So assuming that READ_PICT would also have some keyword parameters,
then I could use REF_EXTRA in the following way... correct?

```
pro READ_PICT_WRAPPER, Filename, Image, R, G, B, _REF_EXTRA=E
case (N_PARAMS()) of
5: READ_PICT, Filename, Image, R, G, B, _EXTRA=E
4: READ_PICT, Filename, Image, R, G, _EXTRA=E
3: READ_PICT, Filename, Image, R, _EXTRA=E
2: READ_PICT, Filename, Image, _EXTRA=E
else: message, "READ_PICT must have at least two parameters"
```

```
pro READ_PICT_WRAPPER, Filename, Image, R, G, B, _REF_EXTRA=E
IF N_Elements(B) EQ 0 THEN $
READ_PICT, Filename, Image, R, G, B, _EXTRA=E
ELSE...
```

For the second option: Could I build an array for { Filename, Image, R,
G, B} and pass it in?

best regards,

dominic

David Fanning wrote:

```
>  Kenneth P. Bowman writes:
>
>>  I find that N_PARAMS() with SWITCH is very handy for setting defaults in
>>  procedures where optional positional parameters make sense.
>>
>>  PRO BLAH, A, B, C
>>
>>  SWITCH N_PARAMS() OF
>>     0 : A = ...
>>     1 : B = ...
>>     2 : C = ...
>>  ENDSWITCH
>
>  In thinking about it, I guess I use N_ELEMENTS for
>  all my parameters because I can throw better errors
>  since I know *exactly* which parameter is missing:
>
>    IF N_Elements(foo) EQ 0 THEN $
>       Message, 'Argument FOO must be a 2D array.'
>
>  I can also trap the error of the user calling my
>  routine with an undefined variable, rather than with
```

> a real variable. This happens more often than you would
> think in IDL programming courses. N_PARAMS, of course,
> can't tell you much of anything about the parameter.
>
> Cheers,
>
> David
>
> P.S. And don't even get me started on the mostly
> mis-used KEYWORD_SET. I see that function misused
> almost constantly in IDL programs to perform the
> function of N_ELEMENTS. Folks, it DOESN'T!! :-)
>
> --
> 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.")

Subject: Re: wrapper functions
Posted by K. Bowman on Fri, 06 Oct 2006 13:35:00 GMT
View Forum Message <> Reply to Message

In article <MPG.1f8f75b654670c76989d07@news.frii.com>,
David Fanning <davidf@dfanning.com> wrote:

> Kenneth P. Bowman writes:
>
>> I find that N_PARAMS() with SWITCH is very handy for setting defaults in
>> procedures where optional positional parameters make sense.
>>
>> PRO BLAH, A, B, C
>>
>> SWITCH N_PARAMS() OF
>> 0 : A = ...
>> 1 : B = ...
>> 2 : C = ...
>> ENDSWITCH
>
> In thinking about it, I guess I use N_ELEMENTS for
> all my parameters because I can throw better errors
> since I know *exactly* which parameter is missing:

As I said, this is only useful in cases where optional positional parameters
makes sense. That is, where some positional parameters are optional AND they
have a logical ordering such that you can pass A, or A and B, or A and B and C,
but not A and C or B and C. In that case, keyword parameters make more sense.

And, of course, KEYWORD_SET should only be used with binary keyword parameters, that is, parameters that are set like this: /KEYWORD.

Ken
