
Subject: wrapper functions

Posted by [Dominic Metzger](#) on Thu, 05 Oct 2006 23:30:17 GMT

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

Hi,

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.

thanks,

dometz

Subject: Re: wrapper functions

Posted by [David Fanning](#) on Fri, 06 Oct 2006 13:34:27 GMT

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

Dometz writes:

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

We're bored. :-)

> 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"
```

You could, but the chance of someone passing an R vector without a G and B vector must be within the floating underflow error of zero. Why waste time coding for things that aren't going to happen? :-)

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

No.

If I were coding this wrapper, I would write it like this

```
FUNCTION READ_PICT_WRAPPER, filename, r, g, b, _REF_EXTRA=extra
  IF N_Elements(filename) EQ 0 THEN Message, 'Must pass a filename.'
  READ_PICT, filename, image, r, g, b, _EXTRA=extra
  RETURN, image
END
```

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 news.qwest.net on Fri, 06 Oct 2006 15:35:03 GMT

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

"David Fanning" <davidf@dfanning.com> wrote in message
news:MPG.1f90094ce6b18f6989d08@news.frii.com...

...

```
> You could, but the chance of someone passing an R
> vector without a G and B vector must be within the
> floating underflow error of zero. Why waste time
> coding for things that aren't going to happen? :-)
```

:O

Well, the answer to that question is:
If a user `_can_` break it, a user `_will_` break it.

Of course, all your other statements are right on the money.

Cheers,
bob

Subject: Re: wrapper functions
Posted by [news.qwest.net](#) on Fri, 06 Oct 2006 15:36:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

"R.G. Stockwell" <no@email.please> wrote in message
news:45267779\$0\$25778\$815e3792@news.qwest.net...
>
> "David Fanning" <davidf@dfanning.com> wrote in message
> news:MPG.1f90094ce6b18f6989d08@news.frii.com...
> ...
>> You could, but the chance of someone passing an R
>> vector without a G and B vector must be within the
>> floating underflow error of zero. Why waste time
>> coding for things that aren't going to happen? :-)
>
>
> :O
>
> Well, the answer to that question is:
> If a user `_can_` break it, a user `_will_` break it.
>
> Of course, all your other statements are right on the money.
>
> Cheers,
> bob

Hey wait, there was a smiley face at the end of it.
Please disregard my previous message.

Subject: Re: wrapper functions
Posted by [David Fanning](#) on Fri, 06 Oct 2006 15:58:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

R.G. Stockwell writes:

```
>> "David Fanning" <davidf@dfanning.com> wrote...
>>
>>> You could, but the chance of someone passing an R
>>> vector without a G and B vector must be within the
>>> floating underflow error of zero. Why waste time
>>> coding for things that aren't going to happen? :-)
>>
>> Well, the answer to that question is:
>> If a user _can_ break it, a user _will_ break it.
>
> Hey wait, there was a smiley face at the end of it.
> Please disregard my previous message.
```

When I wrote that statement I realized it was totally out of character for me (I am normally EXTREMELY anal when it comes to programming). But, honestly, who passes an R vector without a G and B? Absolutely no one.

This is probably the reason these parameters are written as optional **positional** parameters. You would be out of your mind to write them as keywords: too much work for the user. You might write a PALETTE keyword that would allow you to pass a 3-by-n array of color table vectors. That would be useful, but then the user would probably have to figure out how to create a 3 by n array of color table vectors and even I can't remember how to do this half the time. :-(

For those of you who haven't read the Dimensional Juggling Tutorial in some time, you do it like this:

```
palette = [[r],[g], [b]]
```

On the other hand, a PALETTE keyword would sure make a lot more sense here if what we are trying to get across is that sometimes you pass (or get) a color table and sometimes you don't. But this would require a LOT more programming and I would have to ask myself in the end if making the program "better" is really worth it. In this case, I think the answer is clearly "probably not".

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 news.gwest.net on Fri, 06 Oct 2006 16:16:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

"David Fanning" <davidf@dfanning.com> wrote in message
news:MPG.1f902b0dc19bcb4b989d0a@news.frii.com...

...

> When I wrote that statement I realized it was totally
> out of character for me (I am normally EXTREMELY anal
> when it comes to programming). But, honestly, who
> passes an R vector without a G and B? Absolutely
> no one.

I agree with that statement, if a user knows it is R G and B,
but a user can misuse/misunderstand/confuse with another function/
use the function in a different way/etc, and pass a different array into R.

For instance:

> READ_PICT_WRAPPER, Filename, Image, x,y
and then
> contour, Image, x,y

or

> READ_PICT_WRAPPER, Filename, Image, time
> READ_PICT_WRAPPER, Filename, Image, image_info

or as you alluded to:

> READ_PICT_WRAPPER, Filename, Image, palette

And, a user may decide that they are going to secretly encode
data into just the red component, store different
info in the green component, and ignore the blue component.

And this does not even address malicious users (if you write
an application for a company, and they have low wage workers
using it, then if they can break the computer they get a day off work.yay!)

Cheers,
bob

Subject: Re: wrapper functions
Posted by [David Fanning](#) on Fri, 06 Oct 2006 16:35:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

R.G. Stockwell writes:

- > And this does not even address malicious users (if you write
- > an application for a company, and they have low wage workers
- > using it, then if they can break the computer they get a day off work.yay!)

Yeah, I *really* worry about malicious users when I'm writing IDL programs!!! :-)

Cheers,

David

P.S. Maybe I should get a real job. Then I would have more to worry about than paying tuition bills!

--

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 19:42:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hmmm, well, my problem is that these wrappers should be transparent, meaning that all the error messages should be the same with or without the wrapper.

Is this duable?

thanks & best regards,

dometz

David Fanning wrote:

- > Dometz writes:
- >
- >> Wow, I got you guys started on something here... thanks for your input.
- >
- > We're bored. :-)
- >

```

>> 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"
>
> You could, but the chance of someone passing an R
> vector without a G and B vector must be within the
> floating underflow error of zero. Why waste time
> coding for things that aren't going to happen? :-)
>
>> For the second option: Could I build an array for { Filename, Image, R,
>> G, B} and pass it in?
>
> No.
>
> If I were coding this wrapper, I would write it like this
>
> FUNCTION READ_PICT_WRAPPER, filename, r, g, b, _REF_EXTRA=extra
>   IF N_Elements(filename) EQ 0 THEN Message, 'Must pass a filename.'
>   READ_PICT, filename, image, r, g, b, _EXTRA=extra
>   RETURN, image
> END
>
> 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 19:47:31 GMT

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

Dometz writes:

```

> Hmmm, well, my problem is that these wrappers should be transparent,
> meaning that all the error messages should be the same with or without
> the wrapper.

```

>
> Is this duable?

Are you one of those guys who writes all his code in theory?

This is a five line program. How do you think you
might test it? :-)

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 Fri, 06 Oct 2006 20:35:20 GMT

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

Well, at least I would like to do the best I can for the cases I
already know of. Of course there is always the chance that I will miss
some cases.

dometz

David Fanning wrote:

> Dometz writes:

>

>> Hmmm, well, my problem is that these wrappers should be transparent,
>> meaning that all the error messages should be the same with or without
>> the wrapper.

>>

>> Is this duable?

>

> Are you one of those guys who writes all his code in theory?

>

> This is a five line program. How do you think you
> might test it? :-)

>

> 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 20:42:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dometz writes:

> Well, at least I would like to do the best I can for the cases I
> already know of. Of course there is always the chance that I will miss
> some cases.

You may have to make some assumptions. Thank goodness
for the command buffer and the arrow keys, huh?

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 Fri, 06 Oct 2006 21:02:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yeah, I dont know if I will test all of the cases... I will just write
the code so that it is impossible that anything can ever go wrong which
is an easy task with my enormous knowledge about IDL. :-O

Thank you for all your help.... I just started another thread... on
another possible approach... heheheh... hope to get your input on that
one too.

dometz

David Fanning wrote:

> Dometz writes:
>

>> Well, at least I would like to do the best I can for the cases I
>> already know of. Of course there is always the chance that I will miss
>> some cases.
>
> You may have to make some assumptions. Thank goodness
> for the command buffer and the arrow keys, huh?
>
> 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 Fri, 06 Oct 2006 21:06:41 GMT

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

Hey Ken,

Yeah, I am still trying to find a solution that will allow my to write more or less completely transparent wrappers. Not quite there yet... or in other words havent figured out to what extend this is possible or not.

thanks for your input!

dometz

Kenneth Bowman wrote:

> 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
```

Subject: Re: wrapper functions
Posted by [JD Smith](#) on Mon, 09 Oct 2006 18:03:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Fri, 06 Oct 2006 14:06:41 -0700, Dometz wrote:

```
> Hey Ken,
>
> Yeah, I am still trying to find a solution that will allow my to write
> more or less completely transparent wrappers. Not quite there yet... or
> in other words havent figured out to what extend this is possible or
> not.
```

If you're simply passing through a known number of positional parameters, you don't need to check anything:

```
pro wrapper, a,b,c,_REF_EXTRA=e
  routine,a,b,c,_EXTRA=e
end
```

should do it. The a,b,c vars will either be defined if passed, or undefined if they weren't used.

There is a nice story in the archives about whether you can tell the difference between an undefined variable having been passed as a positional argument, and no argument having been passed at all (related to the sound of one hand clapping). The answer is yes, but since no code actually does this (certainly no RSI code), this simple wrapper method works fine.

A more complicated case arises when you want a dynamic number of

positional arguments, for example with `call_procedure`. Then you are forced to build large switch statements, etc. Often you're better off building your own `_EXTRA` structures by hand in this case.

JD

P.S. See http://www.dfanning.com/tips/keyword_check.html
