

---

Subject: Re: IDLDOC Question

Posted by [Michael Galloy](#) on Thu, 20 Oct 2011 19:42:21 GMT

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

---

On 10/20/11 12:48 PM, Jeremy Bailin wrote:

> On 1/24/11 6:39 PM, Michael Galloy wrote:

>> On 1/24/11 2:49 pm, David Fanning wrote:

>>> Mike, or whomever,

>>>

>>> What is the best way to document a program with IDLDOC when you  
>>> really want to document the last module in the file. I'm thinking  
>>> of FSC\_Window, but I should think any widget program would have  
>>> the same problem.

>>>

>>> I really like to have the program documentation at the top  
>>> of the file, but if I understand IDLDOC correctly, the program  
>>> documentation has to be immediately in front of the procedure  
>>> or function definition statement that the documentation applies  
>>> to. Is there any easy way around this restriction? Or, am I  
>>> misunderstanding the restriction?

>>

>> IDLdoc has both file-level and routine-level documentation headers. So  
>> for a widget application you would probably have a file-level header  
>> that describes the purpose of the application and a routine-level header  
>> on the main routine to describe the parameters/keywords. Helper  
>> routines, event handlers, cleanup routines, etc. would be marked private  
>> so that they don't show up in user-level documentation. I would probably  
>> do something like this:

>>

>> ; docformat = 'rst'

>>

>> ;+

>> ; Documentation about how to use the application, like an example of

>> ; of using it::

>> ;

>> ; IDL> mg\_application, dist(20), some\_keyword=5.

>> ;

>> ; This should produce output like:

>> ;

>> ; .. image:: screenshot.png

>> ;-

>>

>> ;+

>> ; :Private:

>> ;-

>> pro mg\_application\_eventhandler\_or\_helperoutine, event

>> end

>>

```

>> ;+
>> ; Documentation on call `MG_APPLICATION`.
>> ;
>> ;:Params:
>> ; data : in, required, type=fltarr(n)
>> ; special data to do something to
>> ;
>> ;:Keywords:
>> ; some_keyword : in, optional, type=float
>> ; specify something
>> ;-
>> pro mg_application, data, some_keyword=someKeyword
>> end
>>
>> If what you want it to do is document the parameters/keywords of the
>> main routine at the top of the file, there is no mechanism for doing
>> that. The "guiding philosophy" of IDLdoc is to place the documentation
>> for something as close as possible to the actual definition in the code
>> so that the documentation actually is updated as the code is updated. So
>> this means that documenting the params/keyword should happen immediately
>> before or after the "pro" or "function" line in the routine definition.
>>
>> Mike
>
> Is it possible to have routine-level docformats? Basically, I have a
> widget program where the main routine is documented in IDL format and I
> just want to mark all of the helper routines private.
>
> -Jeremy.

```

Ah, I think you are stuck converting to another format. The IDL format does not have a "private" tag and a file must have a single format (you can use the

```

; docformat = 'format'

```

comment on the first line of a file to set it's format).

I would suggest the rst format because it will be the best supported going forward with IDLdoc (and even a large chunk of ITT VIS' new .pro code uses it).

Also, I should have a new IDLdoc release in the next few weeks. The big new feature will be LaTeX style equations, like:

[http://docs.idldev.com/idllib/analysis/mg\\_asinh.html](http://docs.idldev.com/idllib/analysis/mg_asinh.html)

The package I'm using can produce great results on all browsers, but

requires a lot of images (200+ MB) to make it work. I've cut it down to a much smaller size, but the results are not as pretty on all browsers (though they are understandable in all the cases I've seen).

Mike

--

Michael Galloy

[www.michaelgalloy.com](http://www.michaelgalloy.com)

Modern IDL, A Guide to Learning IDL: <http://modernidl.idldev.com>

Research Mathematician

Tech-X Corporation

---