
Subject: IDLDOC question

Posted by [David Gell](#) on Mon, 29 Jun 2009 14:12:19 GMT

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

I'm starting to seriously use IDLDOC. In several program headers I have text that I want to appear in the IDLDOC page exactly as it appears in the header. For example, I have a illustration of a directory tree that I wan to include:

```
;          path/specified/by/path +---- 2003 +---
2003_180_L1A_00_00_00_1.CSV
;          |          +---
2003_180_L1A_01_00_00_1.CSV
;          |          +---
2003_181_L1A_01_00_00_1.CSV
;          |          +---
2003_181_L1A_01_00_00_1.CSV
;          +---- 2004 +---
2004_180_L1A_00_00_00_1.CSV
;          |          +---
2004_182_L1A_01_00_00_1.CSV
;          |          +---
2004_183_L1A_01_00_00_1.CSV
;          |          +---
2004_184_L1A_01_00_00_1.CSV
;          +---- 2005
```

How do I induce IDLDOC to not remove all the blank space before the start of each line of text?

And thanks Michael Galloy for a terrific tool

Dave

Subject: Re: IDLDoc Question

Posted by [Michael Galloy](#) on Thu, 11 Nov 2010 21:54:55 GMT

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

On 11/11/10 2:19 PM, David Fanning wrote:

- > I've been making an effort to use Mike Galloy's IDLDoc
- > program for all my program documentation in new programs
- > I am writing. It is going reasonably well, but I keep
- > running into things I just can't quite figure out how
- > to do. I can use some help. I am using the 'rst' markup
- > style.
- >
- > I am writing a wrapper for another routine. I am adding

- > a few of my own keywords, but most of the keywords I
- > expect will be keywords for the routine I am wrapping.
- > I can gather these up and pass them along with _EXTRA,
- > of course. But a few of the "extra" keywords I care about
- > because my wrapper handles them in slightly different
- > ways and because I want my wrapper to define some of
- > these keywords in case they *are not* used.
- >
- > I don't necessarily want to "define" these keywords.
- > The user knows what they mean and how they are used.
- > There is no point in documenting them.
- >
- > But I need to define them on the wrapper procedure
- > definition line, or I can't fish them out of the
- > "extra" structure and use them the way I want to use
- > them. Naturally, since they are defined there, these
- > "keywords" show up in my final documentation. Is there
- > a way to "hide" keywords I don't want to document?

You you can make a keyword "private", i.e.

```
; :Keywords:
;   my_secret_keyword : in, optional, private, type=boolean
;   I don't want this description to show up in the documentation
```

You can also make a keyword "hidden" in the same way: private items will show up in "developer" documentation (i.e., when you run IDLdoc without the /USER keyword set) and hidden items *never* show up.

The private/hidden status can also be set on routines, files, and directories.

Mike

--

www.michaelgalloy.com
 Research Mathematician
 Tech-X Corporation

Subject: Re: IDLDoc Question
 Posted by [David Fanning](#) on Thu, 11 Nov 2010 22:13:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Michael Galloy writes:

- > You you can make a keyword "private", i.e.
- >
- > ; :Keywords:

> ; my_secret_keyword : in, optional, private, type=boolean
> ; I don't want this description to show up in the documentation
>
> You can also make a keyword "hidden" in the same way: private items will
> show up in "developer" documentation (i.e., when you run IDLdoc without
> the /USER keyword set) and hidden items *never* show up.
>
> The private/hidden status can also be set on routines, files, and
> directories.

Thanks, Mike. Just what I was looking for. And I solved
a couple other of my problems today, too. I really like
this new version. :-)

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: IDLDoc Question
Posted by [Michael Galloy](#) on Fri, 12 Nov 2010 17:28:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 11/11/10 3:13 PM, David Fanning wrote:

> Thanks, Mike. Just what I was looking for. And I solved
> a couple other of my problems today, too. I really like
> this new version. :-)

Thanks, I'm hoping that this most recent version makes IDLdoc more
accessible for everyone to use.

Mike

--

www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: IDLDOC Question
Posted by [Michael Galloy](#) on Mon, 24 Jan 2011 23:39:37 GMT

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

--

www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: IDLDOC Question

Posted by [Paul Van Delst\[1\]](#) on Tue, 25 Jan 2011 17:27:20 GMT

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

Michael Galloy wrote:

- > 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.

Based on my own experience (not just with IDL), I think this is a good philosophy. Debugging header documentation is devilishly hard.... unless, I guess, you use a preprocessor to generate it directly from the code (like RDoc for ruby).

Anyway.... regarding David Fanning's original question.....

With my own widget code where I there is the usual bunch of private "helper" procedures/functions preceding the main routine, I have taken to either putting them in their own files (for the bigger ones) or in an include file that is included at the top of the file. I got tired of having to scroll up/down searching for the routines during editing, or having a split-pane edit window.

David, what about doing something like

```
-----%<-----  
@mywidgetapp_helper.pro  
  
; main routine documentation  
;  
;  
;  
;  
; .....
```

PRO mywidgetapp, args..., keywords.....
...just the main widget app code...
END
-----%<-----

where the "mywidgetapp_helper.pro" file contains all the associated code for the main app.

??

cheers,

paulv

p.s. I use the same approach when I generate "syntactic sugar" routines for objects - the include file is at the top of my "XXXXX__define.pro" file.

Subject: Re: IDLDOC Question

Posted by [David Fanning](#) on Tue, 25 Jan 2011 18:33:30 GMT

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

Paul van Delst writes:

```
> David, what about doing something like  
>  
> -----%<-----  
> @mywidgetapp_helper.pro  
>  
> ; main routine documentation  
> ;  
> ;  
> ;  
> ; .....
```

> ;
> ;
> ;
> PRO mywidgetapp, args..., keywords.....

> ...just the main widget app code...
> END
> -----%<-----
>
>
> where the "mywidgetapp_helper.pro" file contains all the associated code for the main app.

This is a pretty good idea, but I guess the main reason I don't care for it is that it creates too many "parts."

I would say something approaching 70% of the e-mail I get on problems with Coyote Library routines involves having library routines that are out of date. Or, rather, failure to update the Library and test the routine one more time before they contact me. I could easily pull out the two objects that make up the FSC_Window application, for example, and just document the small FSC_Window program. But how many people would just update the FSC_Window program and not the guts of the application!? This is a nightmare scenario in my eyes.

And, personally, I like having all the helper routines in the same file. I find it easier to work with them then when they are separate. Different strokes, I guess.

I think my main problem is that I am betwixt and between the old IDL documentation template for older routines and the new IDLDOC method for newer routines. I'm going to need to bite the bullet someday when its very slow and all I feel like doing is drinking a beer. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: IDLDOC Question
Posted by [Jeremy Bailin](#) on Thu, 20 Oct 2011 18:48:01 GMT

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.

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

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

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

Research Mathematician

Tech-X Corporation

Subject: Re: IDLDOC question

Posted by [David Fanning](#) on Wed, 07 Dec 2011 21:13:28 GMT

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

David Fanning writes:

> Have I don't something wrong?

I really have GOT to stop getting up at 5AM. Or, I have to make Coyote hold off on serving the margaritas until at least after noon. :-(

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: IDLDOC question
Posted by [Brian Wolven](#) on Wed, 07 Dec 2011 21:32:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Have you perhaps enabled autocorrect in your browser settings?

Subject: Re: IDLDOC question
Posted by manodeep@gmail.com on Wed, 07 Dec 2011 21:56:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Dec 7, 3:10 pm, David Fanning <n...@dfanning.com> wrote:

> Folks,
>
> I have an IDLDOC question. I have created a directory
> of program documentation for my Coyote Graphics routines.
> I can send a user to the documentation by forming a link
> with this URL:
>
> <http://www.idlcoyote.com/idldoc/cg/index.html>
>
> But, if I send the user to a particular page of documentation,
> say like this:
>
> <http://www.idlcoyote.com/idldoc/cg/cgimage.html>
>
> Then, the user can't find the rest of the programs!
> Have I don't something wrong? Is there a way to correct
> this?
>

If I click on the "Use frames" link on the cgimage.html page, then I can see all the programs, i.e., it takes me to ../index.html. Hope that helps.

Cheers,
Manodeep

Subject: Re: IDLDOC question
Posted by [David Fanning](#) on Wed, 07 Dec 2011 22:07:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Brian Wolven writes:

> Have you perhaps enabled autocorrect in your browser settings?

Probably. I do this reflexively. :-)

The problem really seems to be that the Overview tab means two different things, depending upon whether I start with index.html or cgimage.html. I am wondering if this is a bug or something I configured incorrectly as I was building the documentation directory.

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: IDLDOC question
Posted by [David Fanning](#) on Wed, 07 Dec 2011 22:32:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Manodeep Sinha writes:

> If I click on the "Use frames" link on the cgimage.html page, then I
> can see all the programs, i.e., it takes me to ../index.html. Hope
> that helps.

Ah, OK. Can I turn that on by default?

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: IDLDOC question

Posted by [Michael Galloy](#) on Thu, 08 Dec 2011 03:03:35 GMT

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

On 12/7/11 2:07 PM, David Fanning wrote:

> Brian Wolven writes:

>

>> Have you perhaps enabled autocorrect in your browser settings?

>

> Probably. I do this reflexively. :-)

>

> The problem really seems to be that the Overview tab

> means two different things, depending upon whether

> I start with index.html or cgimage.html. I am wondering

> if this is a bug or something I configured incorrectly

> as I as was building the documentation directory.

I don't understand: the "Overview" link takes you to the Overview page

-- that is independent of whether you are using frames or not. The

"single page"/"use frames" links takes you between the single page view

or the frames; the only oddity is that "use frames" always takes you

back to the overview page in frames (I had to do that so I wouldn't

break the back button, i.e., one click on "use frames" would require two

hits of the back button to back out of).

Mike

--

Michael Galloy

www.michaelgalloy.com

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

Research Mathematician

Tech-X Corporation
