
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.
