
Subject: Re: IDLDOC Question

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

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

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
