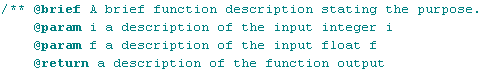
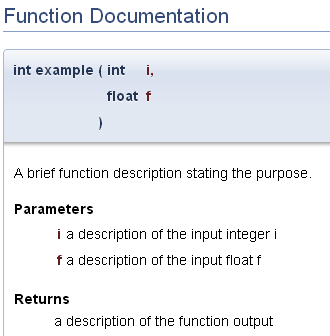
Doxygen-style commenting in OpenPET

Overview

Doxygen is a great tool for automatically generating HTML documentation of source code based on comments written within the text. If the comments are structured in a certain way, the Doxygen executable can parse out the necessary information and present it in a user-friendly way.

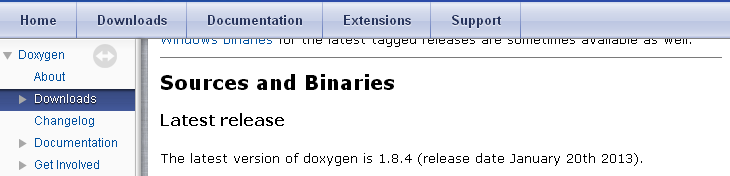
For example, a comment structured as shown generates the following output.



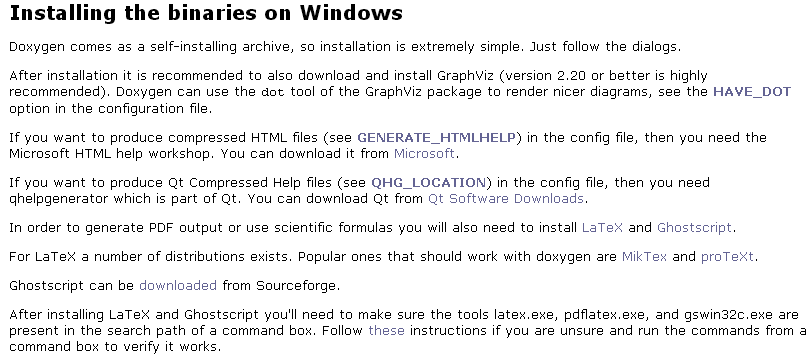


# Installation

Downloading and installing Doxygen is very simple thanks to an easy installer provided by Doxygen. It can be found under the heading “Sources and Binaries” of the “Downloads” page on the Doxygen website (<http://www.stack.nl/~dimitri/doxygen/download.html>) for Windows, Mac OS, and Linux.



The basic package will suffice for OpenPET development. Doxygen is also capable of generating nicer plots, compressed HTML, LATEX and PDF documents. To do so requires downloading extra third party software and is outlined in the installation instructions included with the Doxygen documentation included with installation.



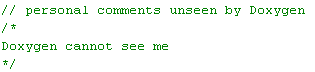
# Software Development with Doxygen in mind

The best starting place and resource for any questions is the Doxygen website (<http://www.stack.nl/~dimitri/doxygen/index.html>) and particularly the page on “Documenting the code” (<http://www.stack.nl/~dimitri/doxygen/manual/docblocks.html>).

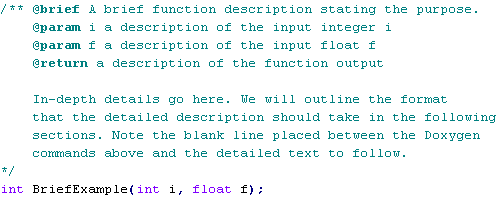
There are many different commenting formats that are acceptable within Doxygen. To maintain consistency within the OpenPET project however, we ask developers to follow the coding standards outlined here. They are very similar to the common Javadoc commenting standards. We have tried to establish a standard that is simple and does not create extra busy-work for developers.

## General

Normal C-style comments will not be recognized by Doxygen and can be used by developers to include personal notes wherever necessary.



Now, let’s take a look at the format of a complete Doxygen comment block.



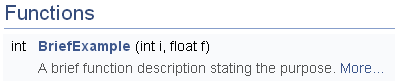
For our purposes, all Doxygen style comments will begin with /\*\*. This will tell the Doxygen executable that this information is pertinent for final documentation.

Following the /\*\* is a list of Doxygen commands. These are Doxygen keywords followed by the @ symbol. These will tell the Doxygen executable where to place specific information. A complete list of commands is available (<http://www.stack.nl/~dimitri/doxygen/manual/commands.html>). @sa or “see also” is particularly useful for linking together related functions.

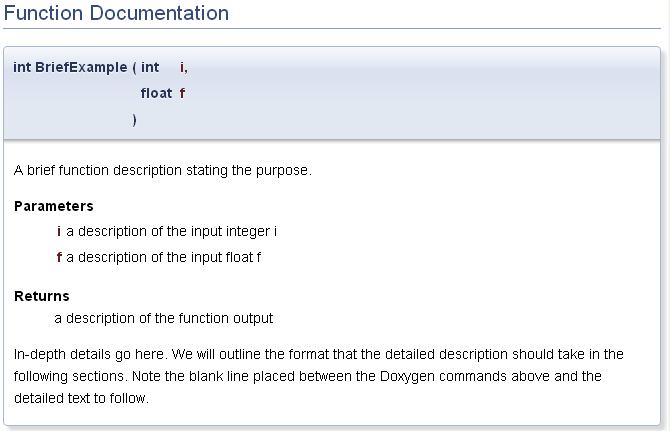
All text following these commands will be included as the detailed description of the file or function. A blank line should be used to separate these commands from the detailed description.

Note that the placement of the comment block is also very important. It must be placed directly before the function prototype. The comment block contains no mention of the name of the function itself. For Doxygen to know that this is a description of the function BriefExample(), the comment block should immediately proceed the function prototype.

From this simple comment block and function prototype, the following documentation will be created. The documentation will show a list of all functions defined with the brief description.



Clicking on the “More…” link will send the user to the detailed documentation shown below.

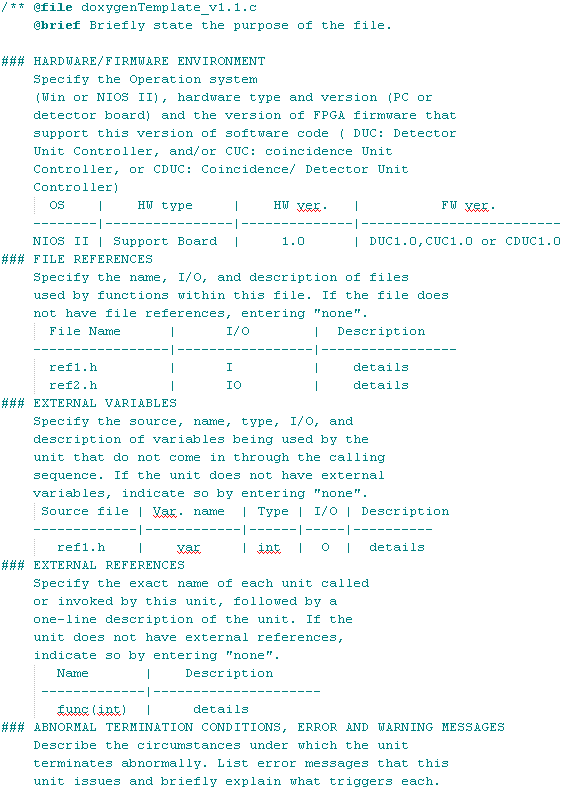


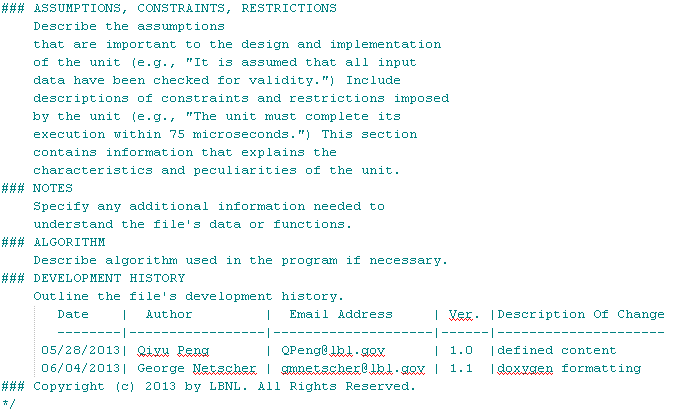
## Files

In Doxygen, comment blocks should always be placed for the thing that they describe. Since it is impossible to put a comment block before the beginning of a file, a special command must be used to tell Doxygen that this is file documentation - @file. This command should be followed by the file name. The comment block describing the file should always be placed at the very beginning of the file.



Just like functions, files require a detailed description. In order to maintain organization with the OpenPET project, we ask you to use the following format.





If certain subheadings are not applicable to a given file, we ask that you still include them in the file description with the value “none”. For example, if there are no external variables, the following will suffice.



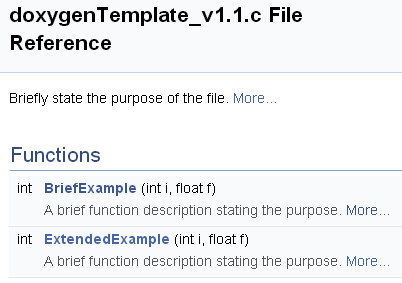
Beginning a line with ### defines it as a subheading within the documentation.

Creating plain text tables by using ----- to separate the heading from the data rows and using | to separate columns. Note that ----- is not used to separate all rows, but only the heading row from the following rows.

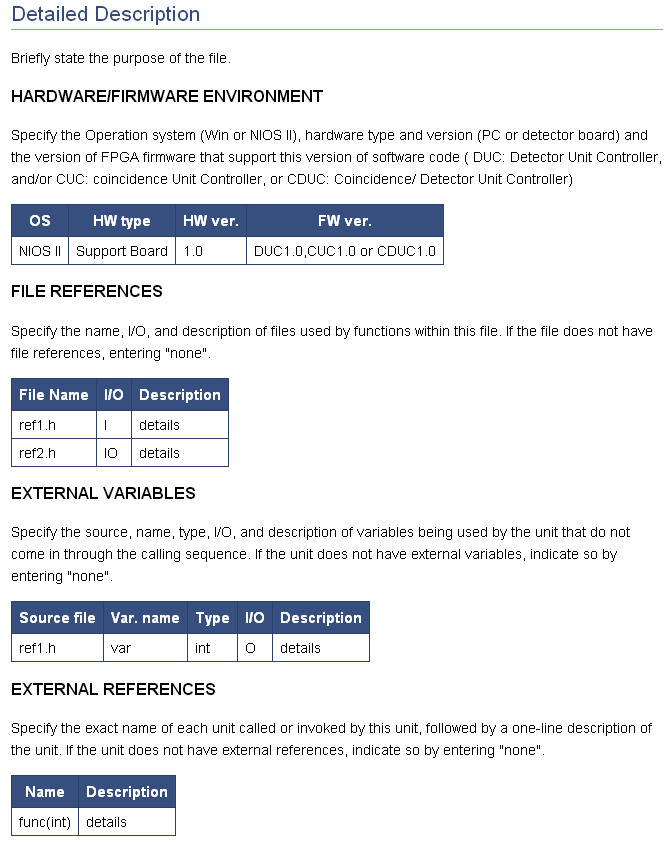
Using this plain text formatting will allow Doxygen to create a file list like the one below.

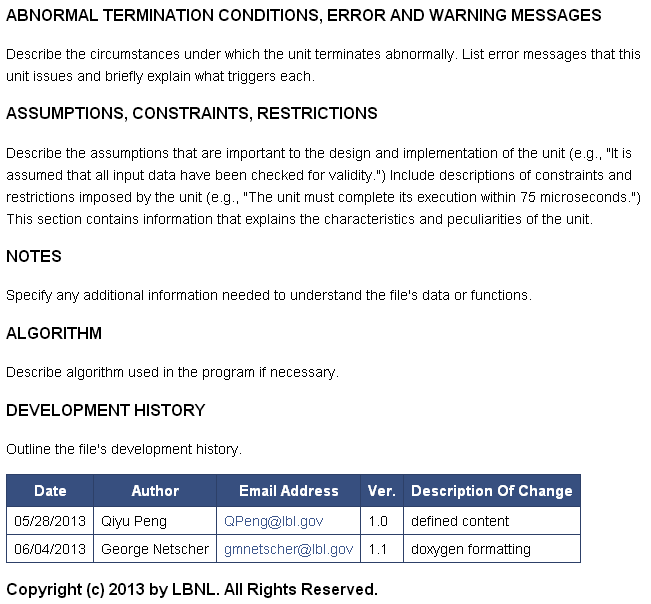


Clicking on the file will provide all information about the file including the descriptions of any functions defined within the file.



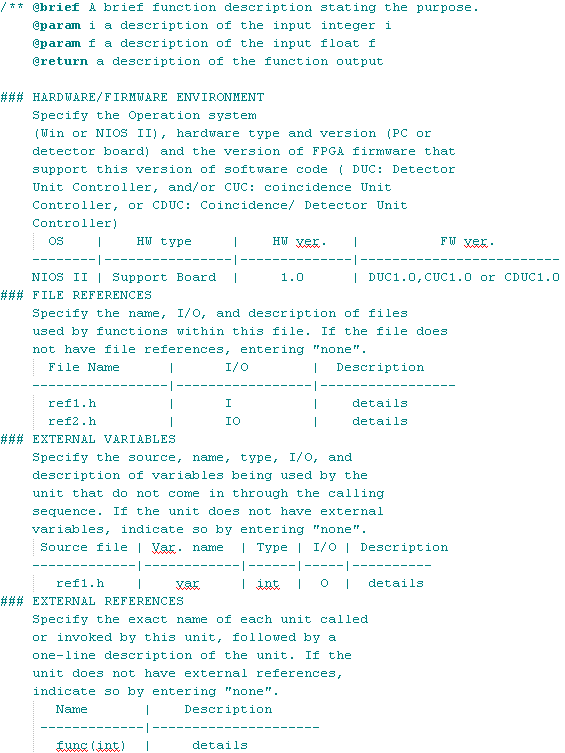
By clicking on the “More…” link following the brief file description. The detailed file description created previously will be shown.

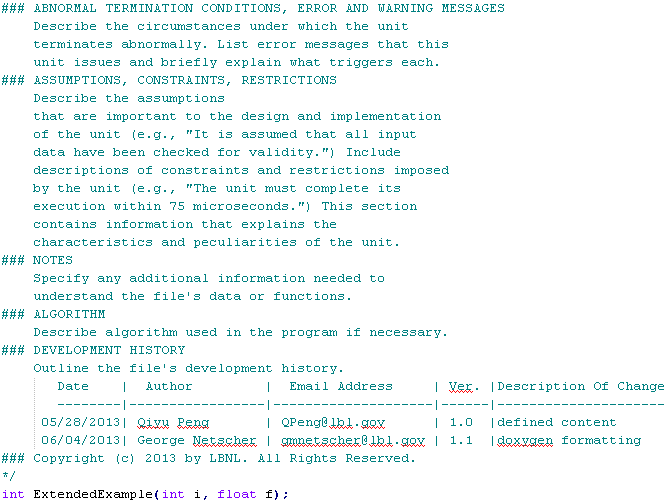




## Functions

You already know almost all the information needed to properly document a function for the OpenPET project. To maintain consistency within the OpenPET project, we ask that you document your functions in the following manner.





Since functions are much more common than files, we do not ask that you include every subheading. Only include those necessary.

All of the example code provided within this document can be found at <https://github.com/openpet-developer/HostPC> under

# Using the Doxygen Executable

# Summary

For an example Doxygen template, see \_\_insert doc location\_\_.