Documenting Fortran Codes with Doxygen

Best Practices Workshop, March 25-26 2019, Hampton VA

Carlos Cruz
Jules Kouatchou
Brent Smith

NASA GSFC Code 606/610 (ASTG/GMAO)
Greenbelt, Maryland 20771

Introduction

Inserting Doxygen Keywords in Source Code

Generating the Document

Why is documentation important?

- To identify the purpose of the software and its requirements
- To clarify what each component does, what is needed to maintain it, and how it can be reused elsewhere
- To provide user support
- To ensure that software is used within its region of validity

Categories of documentation

- Users guide
- Reference manual
- Readme files
- Installation guide
- Tutorials

What is Doxygen?

- An automatic documentation tool.
- Supports pretty printing, call graph generation, man page generation, and LaTeX and HTML documentation files.
- Uses a configuration file to control its behavior.
- The configuration file contains entities called tags.

Purposes I

- Learn how to include comments in your code such that Doxygen incorporates them in the documentation it generates.
- Generate documentation for source code using the Doxygen automated documentation tool.

Purposes II

Insert in the code a brief description of the following:

- Name of the module
- Purpose of the Module
- Description of the Module
- Original Author
- Modifications
- Authors who modified code with a description on why it was modified.

Doxygen Keywords/Markups

Markup	Description
@param var descr...	Document a parameter called var to a function or method.
@return descr...	Document the return value of a function
@see elem	Add a "see also" link to elem, which can be a function, class, or any other documented identifier.
@author name	Indicate the author of an element.
@verssion ver	Indicate the version of an element.
@todo desc...	Leave a note about unfinished work.
@warning descr...	Leave a warning.

Programs & Modules

```
1  !-----------------------------------------------------------------
2  !          FANCY HEADER WITH FANCY NAMES'
3  !-----------------------------------------------------------------
4  ! TITLE           : project name
5  ! PROJECT         : sub-project name
6  ! MODULE          : name of the module or program
7  ! URL             : ...
8  ! AFFILIATION     : ...
9  ! DATE            : ...
10 ! REVISION        : ... V 0.15
11 !> @author
12 !> Author name goes here
13 !
14 ! DESCRIPTION:
15 !> Module to hold the simulation class and its methods
16 !-----------------------------------------------------------------
17 module simulation_mod
18           !Some very interesting code here
19 end module simulation_mod
20
```

Routines

```
1  !-----------------------------------------------------------------
2  !> @author your name and affiliation
3  !> @brief
4  !> Simulation run method. Runs the initialized case
       main time cycle.
5  !> @param[in] casefilename
6  !-----------------------------------------------------------------
7  subroutine run(self, casefilename)
8  implicit none
9  class(simulation_class), intent(inout) :: self
10 type(string), intent(in) :: casefilename    !< case file
       name
11
12 !main time cycle
13 do while (Globals%SimTime.LT.Globals%Parameters%TimeMax
       )
14   !Do your amazing things here :D
15   Globals%SimTime = Globals%SimTime+Globals%SimDefs%dt
16 end do
17
18 end subroutine run
```

Variables

```
1  type constants_t     !< Case Constants class
2      type(vector) :: Gravity  !< Gravitational
       acceleration vector (default=(0 0 -9.81)) (m s-2)
3      real(prec)   :: Z0 = 0.0  !< Reference local sea
       level
4      real(prec)   :: Rho_ref = 1000.0    !< Reference
       density of the medium (default=1000.0) (kg m-3)
5  contains
6  procedure :: setGravity
7  procedure :: setz0
8  procedure :: setrho
9  procedure :: print => printConstants
10 end type
11
```

Configuration File

- Create a text configuration file that contains tags and their associated values.
- information in the file include:
  1. Project name
  2. Path to the source directory
  3. Type of document to generate

Doxygen Configuration FileTags

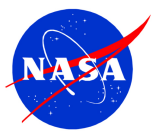
Doxygen Tag Name	Tag Setting
PROJECT_NAME	String Name e.g., "Doxygen Fortran Example"
OUTPUT_DIRECTORY	<directory path you set to place Doxygen output>e.g Doxygen_examples
STRIP_FROM_PATH	"<your home directory path>"
OPTIMIZE_FOR_FORTRAN	YES
INPUT	<directory path you set for all files Doxygen will use to produce the document¿
FILE_PATTERNS	*.txt *.f *.include
IMAGE_PATH	<your path to place images to include in the document>
GENERATE_LATEX	YES (to generate LaTEX file)
GENERATE_HTML	YES (to generate navigable pages using a browser)

Sample Configurartion File

```
PROJECT_NAME           = "Doxygen Fortran Example"
OUTPUT_DIRECTORY       = Doxygen_examples
OUTPUT_LANGUAGE        = English
OPTIMIZE_FOR_FORTRAN   = YES
EXTENSION_MAPPING      =  F90=FortranFree
```

Document Introduction

```
   /**
@mainpage Document Title

@section Introduction
 Write some introductory remarks in this section.  For example,

 This document describes the source code for the XYZ model.

 One can add diagrams to the document.  For example,
 The following diagram shows the breakdown of the entire XYZ model.

@image html exampleImage.png

*/
```



Command Line

```
doxygen <configuration file name>
```