Using Doxygen Documentation in Fortran codes

December 17, 2020

Contents

1 Overview

Doxygen is a tool that creates structured documentation out of the raw source code and comments embedded in it. In order to make this possible the comments have to be marked in a certain way, which is in general programming language dependent. This text describes how to write comments in Fortran code in order to enable Doxygen to make use of them.

2 Example fortran code using Doxygen

The following section shows how to use some of the documentation tags understood by Doxygen. The code example is taken from itm_assert.f90 [1], with some more comments added.

```
module itm_assert

...

contains

!> A generic assertion, tests a given logical expression. (The short, concise description)

!> If it evaluates to .false.,
!> print the fail message and possibly stop execution. (A more detailed description)
!>
!> @param test The logical expression to test. (Explanation of individual arguments)
!> @param failmsg The message to print on fail. If omitted, a generic message is printed.
!> Can be modified with a prefix (see assertSetMsgPrefix)
!> @param doStop Controls whether to stop execution. If doStop .true.,
!> the program is stopped. If .false., only the fail message is printed
!> and bookkeeping is done for delayed stopping (see assertStopOnFailed).
!> If given, overrides the default behaviour  set by assertSetStopMode.

!> @see assertStopOnFailed (References to other routines)
!> @see assertSetStopMode
!> @see assertSetMsgPrefix

!> @author H.-J. Klingshirn (Author information)
!> @version 1.0 (Version information)

subroutine assert( test, failmsg, doStop )
```

---

[1] https://gforge6.eufus.eu/svn/itmshared/branches/grid/f90/src/itm_assert/itm_assert.f90

```
  logical, intent(in) :: test
  character(*), intent(in), optional :: failmsg
  logical, intent(in), optional :: doStop


  ...
end subroutine assert


...
end module itm_assert
```

Some comments:

- Comments relevant for Doxygen have are started with !¿.

- The documentation for a subroutine or function is placed above it. The first paragraph is used as a short description in overview tables.

- The tags used in the Doxygen documentation blocks are called "special commands". They are started either with \ or @. An exhaustive list can be found here http://www.stack.nl/˜dimitri/doxygen/commands.html. Some relevant ones are:

  - @author, @authors: Author name of a code section
  - @version: Version number
  - @warning: Some warning related to the following code
  - @param: Documentation of on or more parameters. Documentation for parameters can also be placed right next to them:

    ```
    subroutine assert( test, failmsg, doStop )
      logical, intent(in) :: test !> The logical expression to test.
      ...
    ```

  - @bug
  - @deprecated
  - @details
  - @param
  - @return Documents a result for a subroutine
  - @see Refers the reader to another entity, e.g. a subroutine
  - @todo
  - @warning

Some more comments:

- "in-body" comments (comments starting with !¿ in the body of the routine) don't seem to work for Fortran (they are falsely included in the documentation of the following routine)