HOWTO
Generate documentation files from annotated Fortran/C++ sources with *Doxygen*

Laurent Korzeczek[1]

Last updated : December 11, 2013

[1]PhD, LEM, Onera, laurent.korzeczek@onera.fr

Contents

List of Figures

1 Installation files

1.1 Doxygen

Doxygen allows you to generate automatic documentation files from your fortran or C++ source codes.
You may find all the installation files you require here : Doxygen download page
If you are using Macport, simply type into a terminal :

```
port install doxygen
```

1.2 Graphviz

Graphviz is only required if you want to produce call and caller flowcharts.
You may find all the installation files you require here : Graphviz download page
If you are using Macport, simply type into a terminal :

```
port install graphviz
```

1.3 Latex

In case you want to generate pdf files, be sure you have your favorite LaTeXdistribution installed.

2 Configure Doxygen

A configuration files is already present in your /src/doc/Documentation_doxygen mM folder, named *DoxyConf_Micromegas*
However, if you want to generate a new configuration file, use the following command :

```
path/to/doxygen -g
```

It will produce a brand new doxygen configuration file named 'Doxyfile'

2.1 Some important Tags

Describe here some important tags

3 Generate documentation files

In order to generate the html and/or latex files, one shoud run the following command :

4 How to comment the code

```
path/to/doxygen DoxyConf_Micromegas
```

4.1 Fortran source codes

In order to read the comments, doxygen need the following syntax :
"!>" or "!<" starts a comment and "!!" or "!>" can be used to continuate a one line comment into a multi-line comment.

Example

```
!> The comment start here..
!! and goes on...
!! and goes on...
!! and goes on...
!! to end here!
```

4.2 C/C++ source codes

describe here how to comment in C/C++ and give examples

5 Useful doxygen command

Describe some

5.1 author

5.2 date

5.3 brief

5.4 param

5.5 todo

5.6 warning

5.7 Examples

5.7.1 Comment modules or functions

```
integer,parameter :: DP=selected_real_kind(p=14) !< Size for the variables with real type
! integer,parameter :: DPI=selected_int_kind(9) !< Size for the variables with integer type
integer,parameter :: DPI=selected_int_kind(13) !< Size for the variables with integer type
integer,parameter :: NLV_max=10 !<nombre total de lois de vitesse utilisees
integer,parameter :: NTSG_MAX=25 !<nombre maximal de systemes de glissement
integer(kind=DPI),parameter :: NSEGMAX=70000 !<dimension des tableaux
integer(kind=DPI),parameter :: GNSEGMAX=80000 !<dimension des tableaux avec images
```

```
integer, parameter   :: FACTEUR_CS=2  !< facteur d'echelle realle  ---> Base de
    vecteur
```

will produce something like this :

constantes Module Reference

Public Attributes

integer, parameter	**dp** =selected_real_kind(p=14) Size for the variables with real type. More...
integer, parameter	**dpi** =selected_int_kind(13) Size for the variables with integer type. More...
integer, parameter	**nlv_max** =10 nombre total de lois de vitesse utilisees More...
integer, parameter	**ntsg_max** =25 nombre maximal de systemes de glissement More...
integer(kind=dpi), parameter	**nsegmax** =70000 dimension des tableaux More...
integer(kind=dpi), parameter	**gnsegmax** =80000 dimension des tableaux avec images More...
integer, parameter	**facteur_cs** =2 facteur d'echelle realle -> Base de vecteur More...

Figure 1: Example of generated documentation regarding variables

5.7.2 Comment modules or functions

```
!> \brief Algorithme (vectoriel) de recherche de la distance entre l'origine
!! d'un segment et le point d'intersection de ce segment avec une
!! surface qu'il transperce.
!!\todo Has to be translated in english
!> \author Benoit DEVINCRE
!> \date 04/01/1995
!> \param IDEB  Please describe parameter
!> \param IFIN  Please describe parameter
!> \param INCRE Please describe parameter
!> \param NMAX  Please describe parameter
!> \param IA    Please describe parameter
!> \param VEC   Please describe parameter
```

will produce something like this :

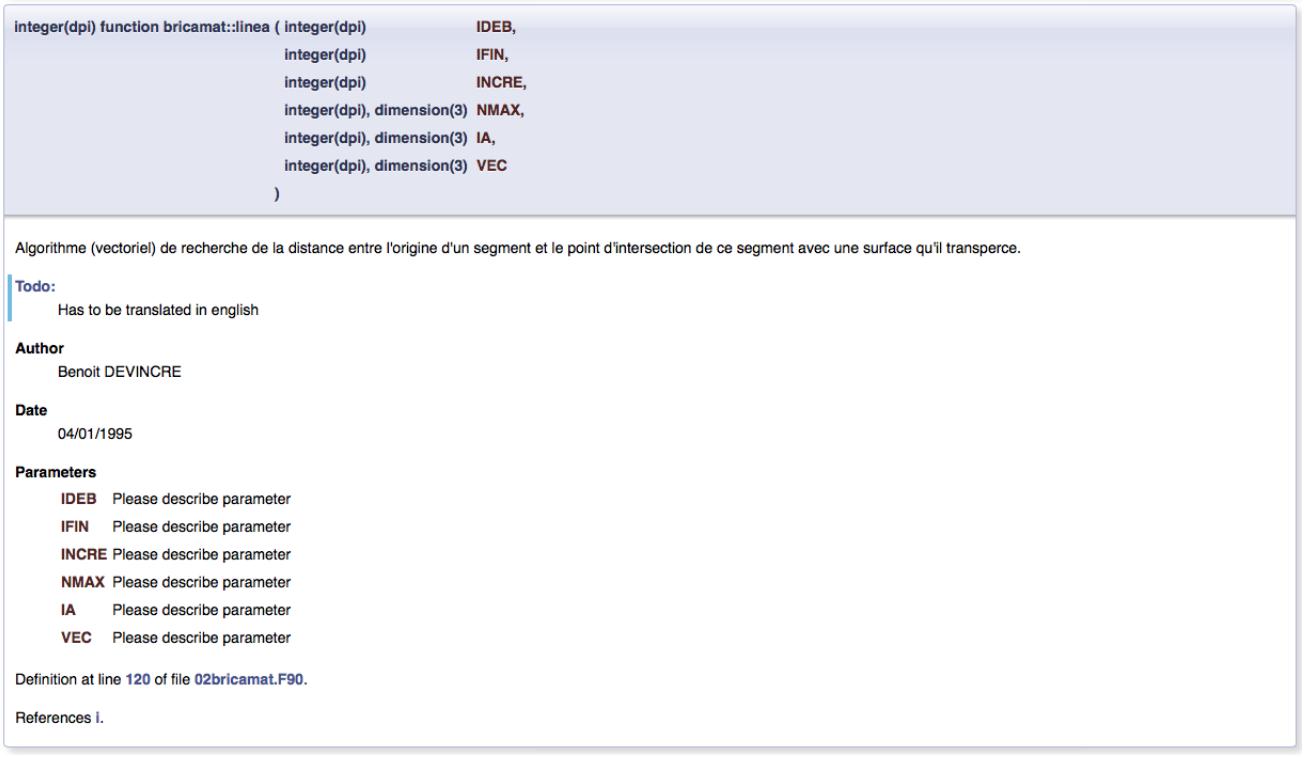

Figure 2: Example of generated documentation regarding function or modules

6 Good behaviors while coding for Micromegas

As a minimum, the user SHOULD :

1. Add an author, date, brief description for each module, function
2. Define Define AND comment each parameters

7 Generate LaTeXdocument

In DoxyDoc_Micromegas/latex/ use the following command

```
make
```

It will produce the pdf file 'refman.pdf'