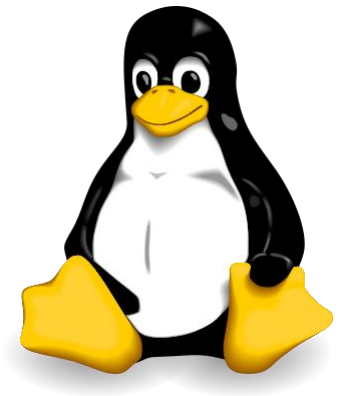


Doxygen



Facile



Normal



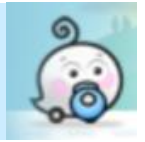
Difficile



Professionnel



Expert



- 1- Introduction
- 2 - Commandes Diverses
- 3 - Bibliographie

Doxygen est un logiciel informatique libre permettant de créer de la documentation à partir du code source d'un programme. (C++, C, Java, Objective-C, Python, IDL, Fortran, VHDL, PHP, C#.)

Pour cela, il tient compte de la grammaire du langage dans lequel est écrit le code source, ainsi que des commentaires s'ils sont écrits dans un format particulier.

Le code de Doxygen a été écrit en grande partie par Dimitri van Heesch.

<http://fr.wikipedia.org/wiki/Doxygen>

1 – Introduction : intérêt

Doxygen permet :

- l'intégration de la documentation directement dans le code sources,
- la cohérence entre la documentation et le code
- la systématisation du comportement des développeurs afin qu'ils documentent le code qu'ils produisent.

Il est également possible d'extraire de la documentation à partir d'un code source non documenté au préalable, ce qui peut faciliter la compréhension d'un programme dont le code est compliqué.

Les informations suivantes peuvent être extraites des sources :

- prototype et documentation des fonctions, qu'elles soient locales, privées ou publiques, etc.
- liste des fichiers inclus
- documentation des structures de données
- prototype et documentation des classes et leur hiérarchie
- différents types de graphes : diagrammes de classe, de collaboration, d'appels, d'inclusion, etc.
- un index de tous les identifiants
- des fichiers sources annotés et navigables

1 – Introduction : Quelques balises standards

- **\param** pour documenter un paramètre de fonction/méthode.
- **\warning** pour attirer l'attention.
- **\author** pour donner le nom de l'auteur.
- **\return** pour documenter les valeurs de retour d'une méthode/fonction.
- **\see** pour renvoyer le lecteur vers quelque chose (une fonction, une classe, un fichier...).
- **\throws** pour documenter les exceptions possiblement levées.
- **\version** pour donner le numéro de version.
- **\since** pour faire une note de version (ex : Disponible depuis ...).
- **\exception** pour documenter une exception.
- **\deprecated** pour spécifier qu'une fonction/méthode/variable... n'est plus utilisée.
- **\li** pour faire une puce.
- **\todo** pour faire un To Do (= "à faire")
- **\fixme** pour faire un Fix Me (= "Réparez-moi").

- **\struct** pour documenter une structure C.
- **\union** pour documenter un union C.
- **\enum** pour documenter un type énuméré.
- **\fn** pour documenter une fonction.
- **\var** pour documenter une variable / un typedef / un énuméré.
- **\def** pour documenter un #define.
- **\typedef** pour documenter la définition d'un type.
- **\file** pour documenter un fichier.
- **\namespace** pour documenter un namespace.
- **\package** pour documenter un package Java.
- **\interface** pour documenter une interface IDL.
- **\brief** pour donner une description courte.
- **\class** pour documenter classe
- **\date** permet l'ajout d'un date.

<http://www.stack.nl/~dimitri/doxygen/commands.html>

Doxygen, pour la génération de documentation au format HTML, LaTeX, PostScript et XML...






Graphviz, pour la génération de graphes (graphes d'appel, de dépendances, ...).

MikTeX pour la génération de documentation au format pdf avec hyperliens.(pour Windows)

HTML WorkShop, pour la génération de documentation au format HTML compressé. .(pour Windows)

1 – Introduction : installation

Ce PC > OS (C:) > Programmes > doxygen > bin

	Nom	Modifié le	Type	Taille
★	 doxygen.exe	28/01/2018 17:11	Application	14 974 Ko
★	 doxyindexer.exe	28/01/2018 17:11	Application	1 581 Ko
★	 doxysearch.cgi.exe	28/01/2018 17:11	Application	1 072 Ko
★	 doxywizard.exe	16/08/2017 11:48	Application	8 198 Ko
★	 libclang.dll	01/08/2017 08:16	Extension de l'app...	43 727 Ko

http://marc.chevaldonne.free.fr/ens_rech/DocumentsDivers_files/Quick_Guide_Doxygen.pdf

2 – Commandes diverses

2.1 - Les blocks documentaires

2.2 - `\brief`

2.3 - `\author`

2.4 - `\version`

2.5 - `\date`

2.6 - `\def`

2.7 - `\file`

2.8 - `\fn`

2.9 - `\param`

2.10 - `\return`

2.11 - `\deprecated`

2.12 - `\class`

2.13 - `\namespace`

<http://www.stack.nl/~dimitri/doxygen/commands.html>

2.1 - Les block documentaires

Diverses combinaisons sont possibles pour créer des blocs de documentation suivant le langage

Langage C

Style C avec deux *

```
/**  
 * ... Documentation ...  
 */
```

Style C avec un !

```
/*!  
 * ... Documentation ...  
 */
```

Langage C++

Style C++ avec trois /

```
///  
/// ... Documentation ...  
///
```

Style C++ avec un !

```
///  
/*! ... Documentation ...  
/*!
```

Langage Fortran

Style avec !>

```
!> ... Documentation  
!>  
!>
```

Style avec !!

```
!! ... Documentation ...  
!! ... Documentation ...
```

Style avec C>

```
C> ... Documentation ...  
C> ... Documentation ...  
C>
```

2.2 - \brief

\brief permet de donner une description courte

Langage C & C++

```
/*! \brief description succincte
* description succincte continue.
*
* description détaillée commence ici. */

/// description succincte.
/** * description détaillée . */

//! description succincte
//! description détaillée

/*! d'autres détails description détaillée */
```

Langage Fortran

```
C> @brief description succincte
C> description succincte continue.
C>
C> description détaillée commence ici. */

!> @brief description succincte
!> description succincte continue.
!>
!> description détaillée commence ici. */
```

2.3 - \author

\author permet d'ajouter un ou plusieurs nom d'auteur.

Langage C & C++

```
/*! \brief code  
* \author D.Palermo  
* \author C.Grondein  
*/
```

Langage Fortran

```
C> \brief code  
C> @author D.Palermo  
C> @author C.Grondein  
C>
```

2.4 - \version

\version permet d'ajouter un numéro de version

Langage C & C++

```
/*! \brief code  
* \author D.Palermo  
* \author C.Grondein  
*  
* \version 2.4  
*/
```

Langage Fortran

```
C> \brief code  
C> @author D.Palermo  
C> @author C.Grondein  
C>  
C> @version 2.4
```

2.4 - \date

\date permet d'ajouter une date

Langage C & C++

```
/*! \brief code  
* \author D.Palermo  
* \author C.Grondein  
*  
* \date 10 mai 2011  
* \version 2.4  
*/
```

Langage Fortran

```
C> \brief code  
C> @author D.Palermo  
C> @author C.Grondein  
C>  
C> @date 10 mai 2011  
C> @version 2.4
```

2.6 - \def

\def permet l'ajout d'un bloc de définition

Langage C & C++

```
/*! \brief code
* \author D.Palermo
* \author C.Grondein
*
* \date 10 mai 2011
* \version 2.4
*/
/*!
\def MAX(x,y)
calcul le maximum entre \a x et \a y.
*/
```

Langage Fortran

```
C> \brief code
C> @author D.Palermo
C> @author C.Grondein
C>
C> @date 10 mai 2011
C> @version 2.4
C>
```

2.7 - \file

\file permet de créer un bloc de documentation pour un fichier source ou d'en-tête.

Langage C & C++

```
/*! \file file.hh
* \brief code
* \author D.Palermo
* \author C.Grondein
*
* \date 10 mai 2011
* \version 2.4
*/
/*!
\def MAX(x,y)
calcul le maximum entre \a x et \a y.
*/
```

Langage Fortran

```
C> @file file,f77
C> @brief code
C> @author D.Palermo
C> @author C.Grondein
C>
C> @date 10 mai 2011
C> @version 2.4
C>
```

2.8 - \fn

\fn permet de documenter une fonction .

Langage C & C++

```
/*! \fn const char *Test::member(char c,int
n)
* \brief fonction membre de la classe
Test.
* \param c un caractère.
* \param n in entier.
* \exception std::out_of_range
parameter is out of range.
* \return un pointeur sur caractere.
*/

const char *Test::member(char c,int n)
{
    .....
}
```

Langage Fortran

```
C> @file file,f77
C> @brief code
C> @author D.Palermo
C> @author C.Grondein
C>
C> @date 10 mai 2011
C> @version 2.4
C>
!> @fn IntRestBuild(aggr,ghost)
!! method.
!! @param aggr information about the aggregates
!! @param ghost information about the aggregates

subroutine IntRestBuild(aggr, ghost)
```


2.9 - \param

\param permet d'ajouter un paramètre dans un bloc de documentation d'une fonction ou d'une macro..

Langage C & C++

```
/*! \fn const char *Test::member(char c,int
n)
* \brief fonction membre de la classe
Test.
* \param[in] c un caractère.
* \param[out] n in entier.
* \exception std::out_of_range
parameter is out of range.
* \return un pointeur sur caractere.
*/

const char *Test::member(char c,int n)
{
    .....
}
```

Langage Fortran

```
C> @file file,f77
C> @brief code
C> @author D.Palermo
C> @author C.Grondein
C>
C> @date 10 mai 2011
C> @version 2.4
C>
!> @fn IntRestBuild(aggr,ghost)
!! method.
!! @param aggr information about the aggregates
!! @param ghost information about the aggregates

subroutine IntRestBuild(aggr, ghost)
```

2.10 - \return

\return permet de décrire la fonction de retour..

Langage C & C++

```
/*! \fn const char *Test::member(char c,int
n)
* \brief fonction membre de la classe
Test.
* \param[in] c un caractère.
* \param[out] n in entier.
* \exception std::out_of_range
parameter is out of range.
* \return un pointeur sur caractere.
*/

const char *Test::member(char c,int n)
{
    .....
}
```

Langage Fortran

```
C> @file file,f77
C> @brief code
C> @author D.Palermo
C> @author C.Grondein
C>
C> @date 10 mai 2011
C> @version 2.4
C>
!> @fn IntRestBuild(aggr,ghost)
!! method.
!! @param aggr information about the aggregates
!! @param ghost information about the aggregates
!! @return integer
function IntRestBuild(aggr, ghost)
```

2.11 - \deprecated

\deprecated permet d'ajouter un paragraphe précisant que la fonction, macro, etc... est dépréciée.

Langage C & C++

```
/*! \fn const char *Test::member(char c,int
n)
* \brief fonction membre de la classe
Test.
* \param[in] c un caractère.
* \param[out] n in entier.
* \exception std::out_of_range
parameter is out of range.
* \return un pointeur sur caractère.
* \deprecated Fonction dépréciée, ne
plus utiliser !
*/
const char *Test::member(char c,int n)
{
    .....
}
```

Langage Fortran

```
C> @file file,f77
C> @brief code
C> @author D.Palermo
C> @author C.Grondein
C>
C> @date 10 mai 2011
C> @version 2.4
C>
!> @fn IntRestBuild(aggr,ghost)
!! method.
!! @deprecated Fonction dépréciée, ne plus utiliser !
!! @param aggr information about the aggregates
!! @param ghostinformation about the aggregates
!! @return integer
function IntRestBuild(aggr, ghost)
```

2.12 - \class

\class permet de créer un bloc de documentation d'une classe (C++) pour le C on utiliser la balise **\struct**

Langage C++

```
/* A dummy class */  
  
class Test  
{  
};  
  
/*! \class Test class.h "inc/class.h"  
* \brief This is a test class.  
*  
* Some details about the Test class  
*/
```

2.11 - \namespace

\namespace permet de créer un bloc de documentation pour un espace de nom (C++).

Langage C++

```
/*! \namespace PLEIADES
 *
 * espace de nommage regroupant les outils composants de PLEIADES
 */

namespace PLEIADES
{
}
```

- Le site de Doxygen
 - <http://www.doxygen.nl/>
- Tutorial :
 - <http://franckh.developpez.com/tutoriels/outils/doxygen/>
 - <http://fr.wikipedia.org/wiki/Doxygen>
 - <http://www.softeng.rl.ac.uk/blog/2010/jan/30/callgraph-fortran-doxygen/>