

Doxygen Manuel [dx]

Karine Zampieri, Stéphane Rivière

Unisciel  algoprogram  Version 20 mai 2018

Table des matières

1	Doxygen	3
2	Configuration Wizard	4
2.1	Onglet Project	4
2.2	Onglet Mode	5
2.3	Onglet Output	6
2.4	Onglet Diagrams	7
3	Analyse	9
4	Rendu HTML	10
4.1	Rendu HTML – code source	11
4.2	Rendu HTML – index des entités	12
4.3	Rendu HTML – diagrammes	13
5	Tags de documentation	14
5.1	Formats de documentation	14
5.2	Syntaxes des commandes	15
5.3	Quelques commandes	15
6	Configuration Expert	17
6.1	Onglet Expert>Project	17
6.2	Onglet Expert>Build	17
6.3	Onglet Expert>HTML	17
6.4	Onglet Expert>Dot	18

Introduction



Objectif

Ce document présente les principales fonctionnalités de l'outil de documentation DOXYGEN. Il est plus puissant que JAVADOC par l'export natif dans de nombreux formats et la génération de diagrammes.



Remarque

Pour plus d'informations sur les paramètres de configuration et les tags spécifiques de DOXYGEN, rendez-vous sur la documentation officielle.

1 Doxygen

Principales caractéristiques

DOXYGEN est un générateur de documentation :

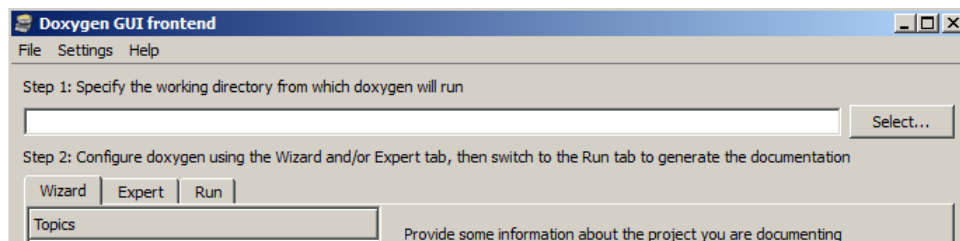
- Pour C++, C, Java, Python, VHDL, PHP, C# et autres.
- Site officiel <http://www.doxygen.org>
- Il permet des sorties HTML, Latex, RTF et XML.
- C'est un logiciel OpenSource – Licence GNU GPL.
- Il est disponible sous les plateformes Linux, Mac OS, Windows.

Doxywizard

Interface graphique (*Graphical User Interface*), elle permet de configurer DOXYGEN et de lancer l'analyse des programmes sources pour en produire une documentation.

Il existe deux modes de configuration :

- Wizard – Configuration de base
- Expert – Configuration avancée pour les experts

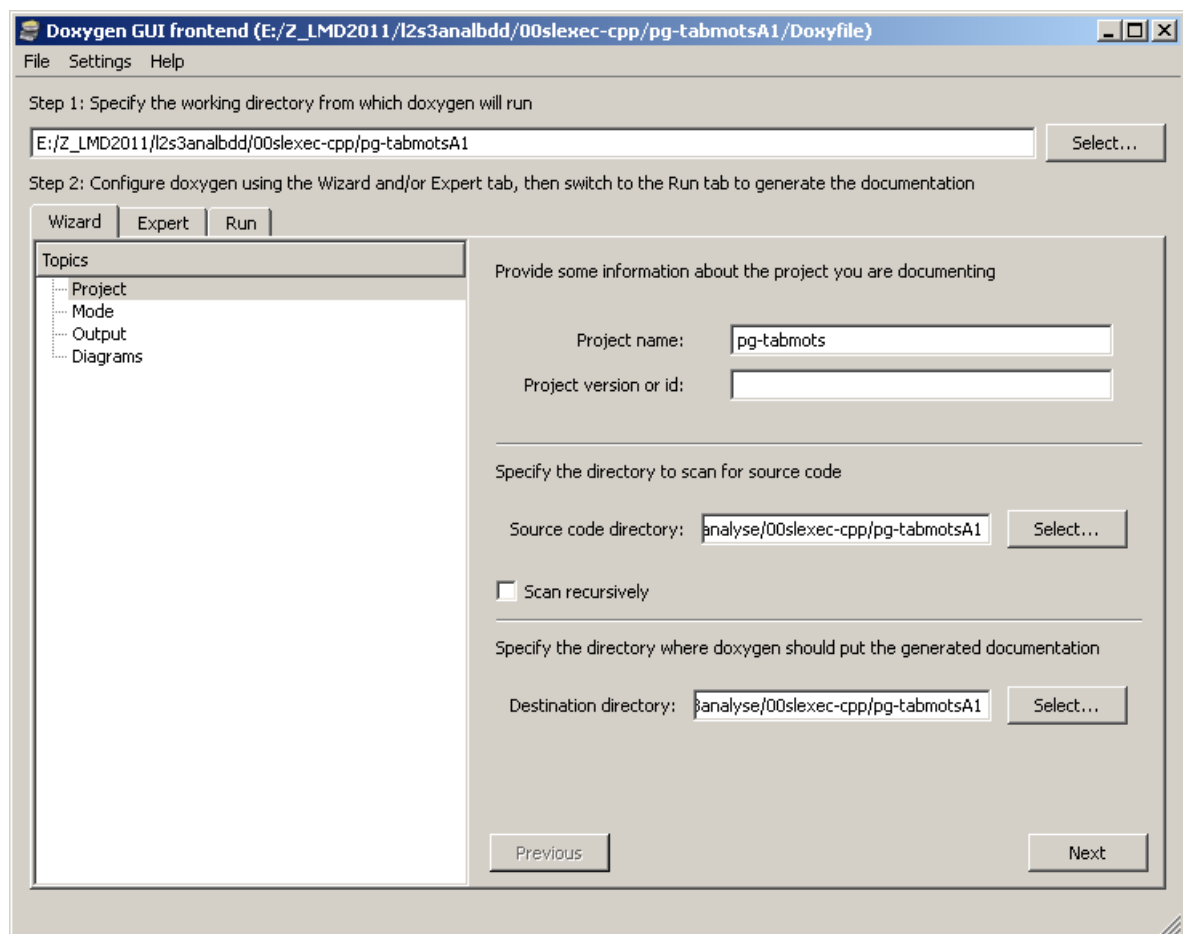


2 Configuration Wizard

La configuration de base est constituée des onglets :

- Project – Informations générales sur le projet
- Mode – Mode d'analyse
- Output – Type de documentation à générer
- Diagrams – Diagrammes à générer

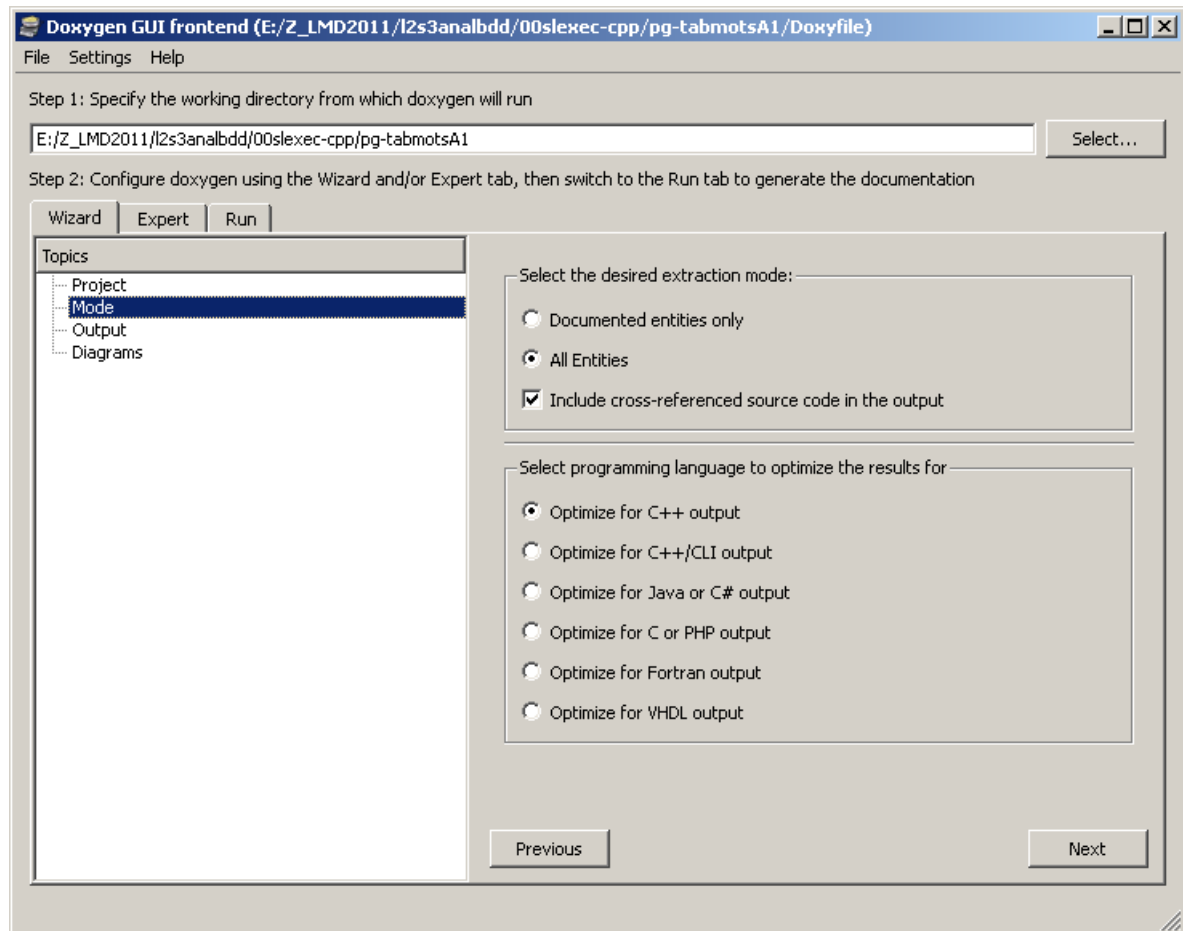
2.1 Onglet Project



Cet onglet définit le projet.

- Project Name – Nom du projet logiciel
- Project version or id – (facultatif) Numéro de version du logiciel
- Source code directory – Localisation des sources
- Scan recursively – Analyse récursive des sous-dossiers
- Destination directory – Localisation de la documentation à générer

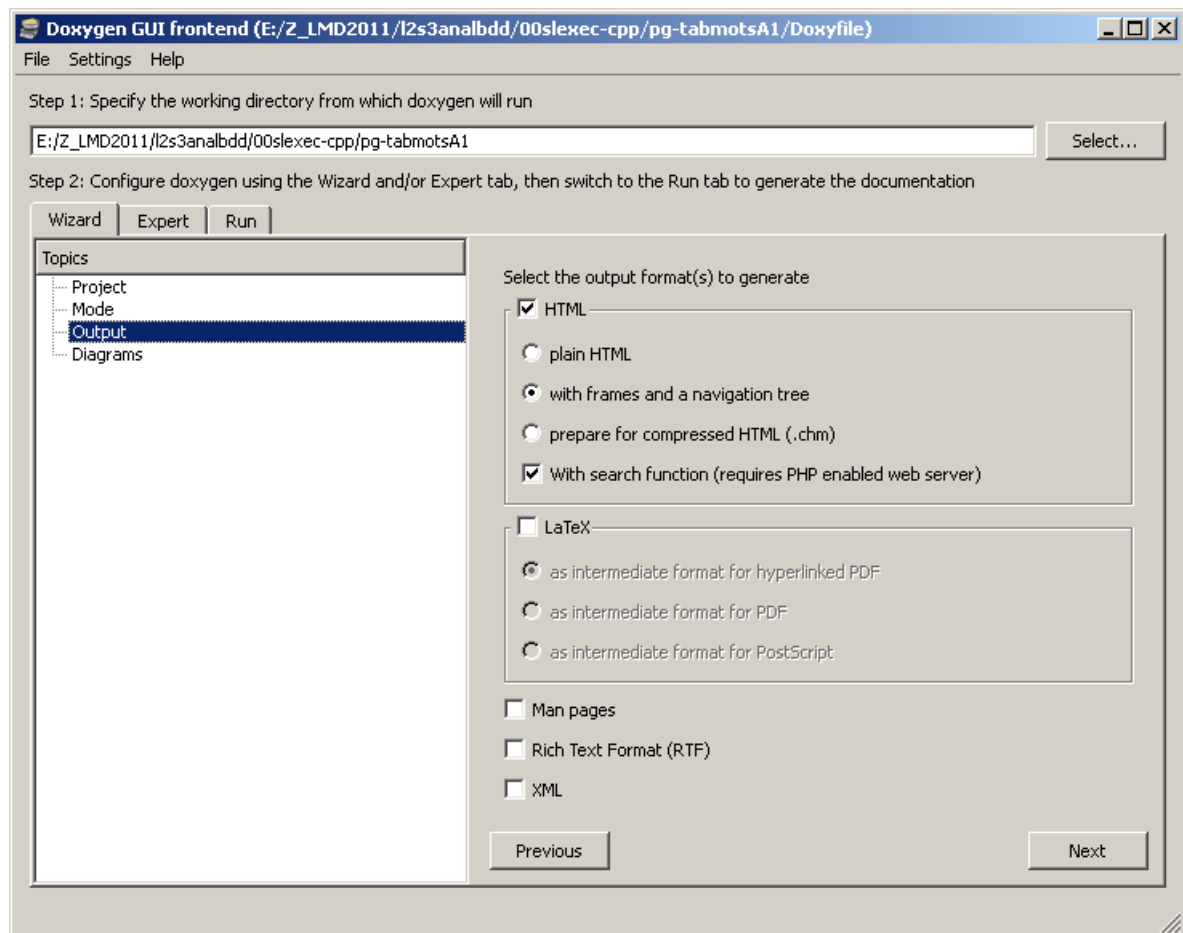
2.2 Onglet Mode



Ce deuxième onglet détermine le détail du contenu.

- **Documented entities only** – Choix d’inclure seulement les entités (classe, méthode, fonction...) possédant des parties de documentation
- **All entities** – Choix de référencer tout le code, y compris les parties `private`, `static`, etc.
- **Include cross-referenced...** – Ajout de liens hypertextes entre toutes les entités liées
- **Select programming language...** – Choix du langage des sources

2.3 Onglet Output



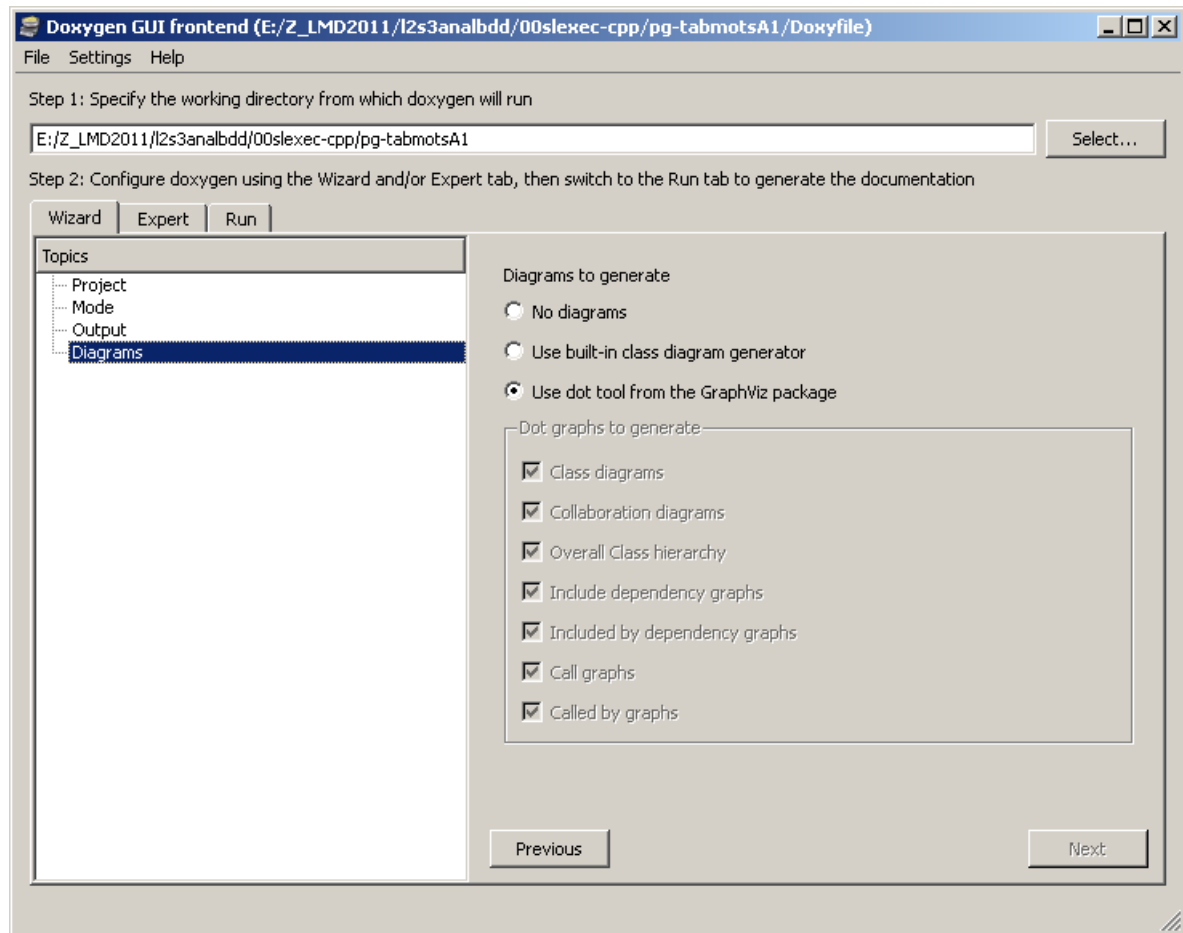
Ce troisième onglet permet le choix du type de sortie de la documentation. Plusieurs formats sont supportés : HTML, LaTeX, Man pages (pages de manuels, Linux), RTF (*Rich Text Format*) ou XML.

Format HTML

Il permet plusieurs types de présentations :

- Plain HTML – Documentation HTML sans frame
- With frames... – HTML avec frame et arbres de navigation dans un menu
- Prepare for compresses HTML – HTML compressé pour génération ultérieure en fichier d'aide Windows
- With search function – Inclusion d'une zone de recherche (nécessite PHP)

2.4 Onglet Diagrams



Ce dernier onglet permet le choix des graphiques.

- No diagrams – Pas de diagramme
- Use build-in class... – Diagrammes de classe générés par DOXYGEN
- Use dot tool from... – Multitude de diagrammes (de classe, de collaboration, de dépendance, d'appel...) générés par le programme externe GRAPHVIZ (voir la documentation DOXYGEN pour l'installation)

Options GraphViz

Le générateur crée un graphique :

- **Class diagrams** – Pour les classes (et structures en C)
- **Collaboration diagrams** – Les relations directes et indirectes de classes
- **Include dependency graphs** – Dépendances entre les fichiers
- **Included by ...** – Dépendances entre chaque fichier et fichiers inclus
- **Overall Class hierarchy** – Hiérarchie d'une classe
- **Call graphs** – Pour chaque fonction/procédure (appel direct ou indirect) (*).

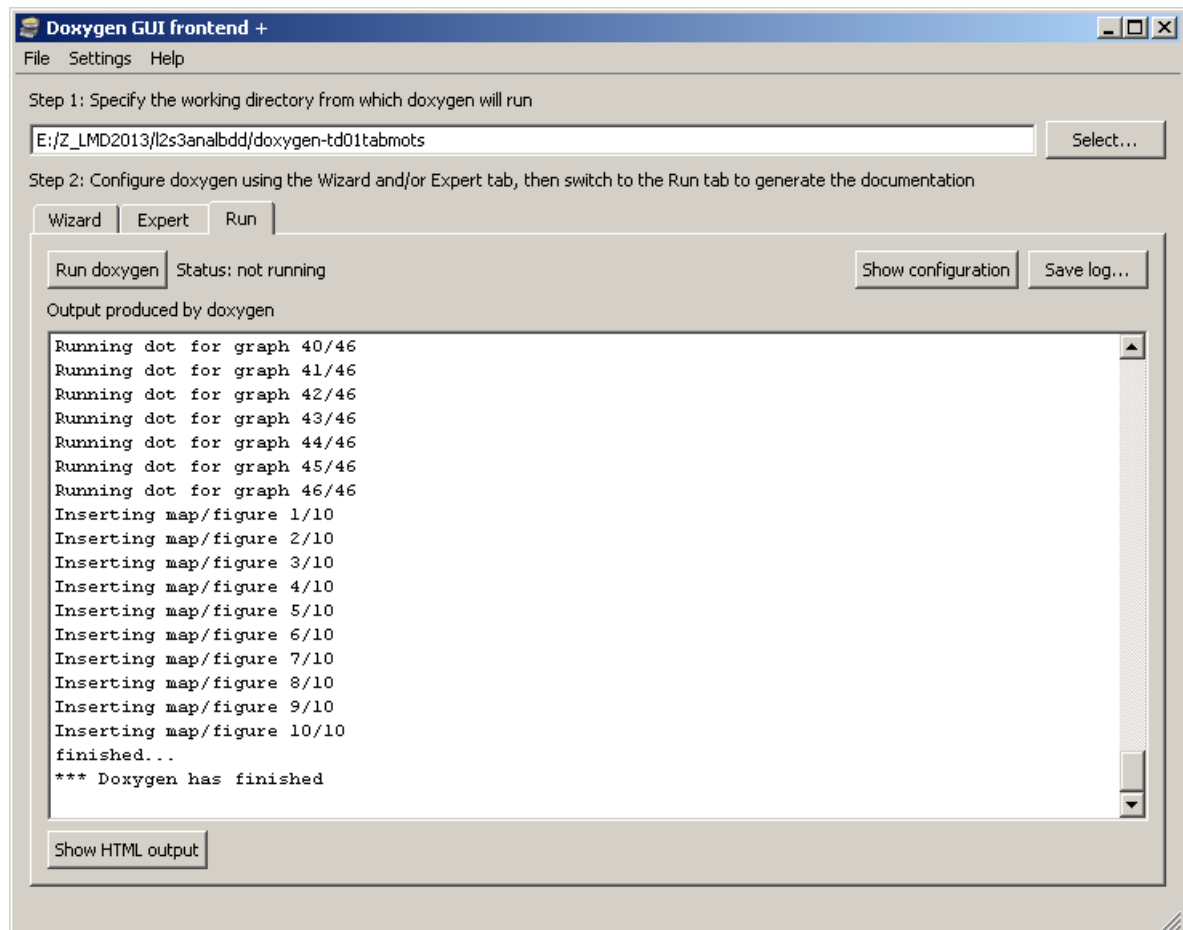
(*) Cette option n'est utilisable que si vous cochez la case **Include cross-reference...** dans le deuxième onglet.

3 Analyse

L'analyse des codes sources :

- Se lance en cliquant sur le bouton Run doxygen (onglet Run).
- Le détail des opérations effectuées par DOXYGEN apparaît dans la zone de texte.
- Lorsque la génération de la documentation a été réalisée, DOXYGEN affiche :

```
*** Doxygen has finished
```

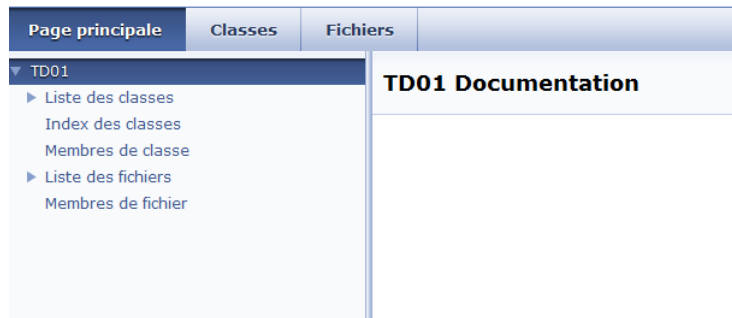


4 Rendu HTML

La documentation HTML produite par DOXYGEN se trouve dans le sous-répertoire **html/** du répertoire de sortie spécifié dans la configuration.

Il existe deux menus de navigation :

- Un à gauche dans une frame (option `With frames...`).
- Et un en haut.



La documentation HTML des projets comporte les chapitres suivants :

- Main Page – Page principale du projet avec son nom et le numéro de version
- Packages – Liste des packages avec une courte description de chacun d’eux
- Class Hierarchy – Hiérarchie des classes
- Class List – Liste alphabétique des classes et structures et interfaces avec une courte description
- File List – Liste des fichiers avec une courte description et un lien vers leur code source colorisé et lignes numérotées
- Class Members – Liste alphabétique des méthodes et variables du projet en correspondance avec leur classe

4.1 Rendu HTML – code source

Le code source des classes documentées est intégré à documentation. Des liens hypertextes sur chaque méthode, classe et attribut documenté pointe vers leur documentation respective. Les lignes de code sont numérotées et DOXYGEN intègre la coloration syntaxique.

Exemple : Classe et structure

tabmotsA1/tabmotsA1a1.hpp

Aller à la documentation de ce fichier.

```

00001 #ifndef TABMOTSA1A1_HPP
00002 #define TABMOTSA1A1_HPP
00003
00004 #include <string>
00005 using namespace std;
00006
00008 const int NMAX = 100;
00009 struct SMots
00010 {
00011     string mots[NMAX];
00012     int nligs;
00013     int ncols;
00014 };
00015
00016 #endif

```

Exemple : Méthodes et fonctions

Fonctions

```
bool chargerSMots (SMots &t, int nmax, const string &fn)
```

Documentation des fonctions

```
bool chargerSMots ( SMots & t,
                    int nmax,
                    const string & fn
                    )
```

Charge un tableau 2D de mots depuis un flux d'entrée.

Paramètres:

[out] *t* - référence au tableau 2D de mots chargé
[in] *nmax* - taille maximale du tableau linéaire
[in] *fn* - nom du fichier existant à ouvrir

Renvoie:

vrai si les données ont été chargées, faux sinon

4.2 Rendu HTML – index des entités

Le chapitre *Class Members* liste toutes les méthodes et attributs du projet par ordre alphabétique et par catégorie qui pointent vers leurs classes respectives.

Exemple : Membres de fichier

Page principale | Classes | **Fichiers**

Liste des fichiers | **Membres de fichier**

Tout | Fonctions | Variables | Énumérations | Valeurs énumérées

a c f i l m n t

▼ TD01

- ▶ Liste des classes
 - Index des classes
 - Membres de classe
- ▶ Liste des fichiers
 - Membres de fichier**

Liste de tous les membres de fichier avec des liens vers les fichiers auxquels ils appartient :

- a -

- afficherSMots() : [fonction.cpp](#) , [fonction.h](#)

- c -

- chargerSMots() : [fonction.cpp](#) , [fonction.h](#)

- f -

- fixerSMots() : [fonction.cpp](#) , [fonction.h](#)
- FMT_ALIGN_DROITE : [fonction.h](#)
- FMT_ALIGN_GAUCHE : [fonction.h](#)
- FMT_TRONCATURE : [fonction.h](#)
- Format : [fonction.h](#)

- i -

Exemple : Toutes les entités

Tout | Fonctions | Variables | Énumérations | Valeurs énumérées

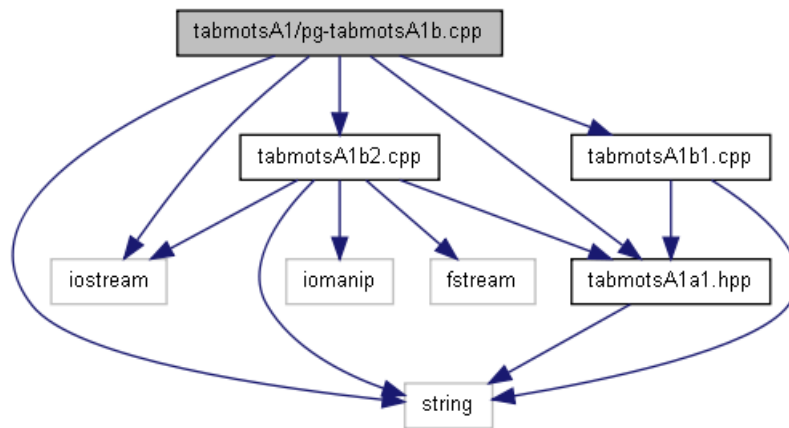
Liste de tous les membres de fichier avec liens vers les fichiers auxquels ils appartient :

- afficherSMots() : [tabmotsA1b1.cpp](#)
- chargerSMots() : [tabmotsA1b2.cpp](#)
- fixerSMots() : [tabmotsA1b1.cpp](#)
- FMT_ALIGN_DROITE : [tabmotsA1a2.hpp](#)
- FMT_ALIGN_GAUCHE : [tabmotsA1a2.hpp](#)
- FMT_TRONCATURE : [tabmotsA1a2.hpp](#)
- indexTab2D() : [tabmotsA1b1.cpp](#)
- LGMAX : [tabmotsA1a2.hpp](#)
- lireSMots() : [tabmotsA1b1.cpp](#)
- longueurPlusLongLigne() : [tabmotsA1c.cpp](#)
- longueurPlusLongMot() : [tabmotsA1c.cpp](#)
- main() : [pg-tabmotsA1b.cpp](#) , [pg-tabmotsA1d.cpp](#) , [pg-tabmotsA1c.cpp](#)
- NMAX : [tabmotsA1a1.hpp](#)
- nreplicatats() : [tabmotsA1d1.cpp](#)
- test_affichage() : [pg-tabmotsA1d.cpp](#)
- test_chargement() : [pg-tabmotsA1b.cpp](#)
- test_pluslongmot() : [pg-tabmotsA1c.cpp](#)
- TFormat : [tabmotsA1a2.hpp](#)
- tracerBandeau() : [tabmotsA1d1.cpp](#)
- tracerLigne() : [tabmotsA1d2.cpp](#)
- tracerSMots() : [tabmotsA1d2.cpp](#)

4.3 Rendu HTML – diagrammes

La documentation de chaque classe est accompagnée d'un diagramme d'héritage et d'interface. Chaque classe du diagramme comprend un lien hypertexte vers sa documentation.

Exemple



5.2 Syntaxes des commandes

Le bloc de documentation se met au début du programme ou avant une procédure ou fonction. DOXYGEN reconnaît le nom de la routine et génère sa documentation (nom, liens, etc.)



Format Javadoc

```
/**  
  @commande  
*/  
... code source ...
```



Format Latex

```
/**  
  \commande  
*/  
... code source ...
```

5.3 Quelques commandes



Commandes de programme

```
@file...    // fichier source ou d'en-tête  
@author...  // les auteurs  
@date...    // la date  
@version... // la version  
@todo...    // Les "A faire"
```



Ordre des balises

Il n'a que très peu d'importance mais cela a un impact sur l'ordre de génération du paragraphe et donc de l'affichage.



Commandes de fonctions et procédures

```

description...           // description du bloc qui suit
@brief...                // courte description
@param[in,out] nom - ... // description du paramètre
@return...               // résultat d'une fonction
@pre...                  // préconditions
@post...                 // postconditions
@exception...           // synonyme de throws
@throws                  // exception
@test...                 // descriptif du test
@attention               // les attentions
@note                   // une note (= remarque)
@bug                     // bug(s) – description locale!
@internal                // pour documentation interne
@see                     // voir
@deprecated              // déprécié

```



Commandes de types

```

@struct nom [header-file] // description d'une structure
@enum nom                  // énumération de constantes
@union nom [header-file]  // description d'une union
@class nom [header-file]  // description d'une classe
@namespace nom             // description d'un espace de nom

```



Remarque

Pour une procédure/fonction/méthode, il faut fournir une balise correcte @param car une vérification syntaxique de celle-ci est faite par Doxygen.

6 Configuration Expert

Cette section détaille quelques options pour peaufiner les réglages et la présentation générale de la documentation.

6.1 Onglet Expert>Project

OUTPUT_LANGUAGE Langue de génération (Par défaut l'Anglais).

USE_WINDOWS_ENCODING Permet de forcer l'encodage des caractères par celui utilisé par Windows si la case est cochée, sinon c'est un encodage de type Unix/Linux qui est utilisé.

BRIEF_MEMBER_DESC Permet d'afficher ou non la description courte dans la liste des fonctions/énumérations/structures/classes/etc. liste qui se trouve en haut de page donc avant la partie détaillée de la documentation.

REPEAT_BRIEF Permet d'afficher ou non la description courte dans la partie détaillée de la documentation. Cette option et la précédente sont conjointement liées car si vous désactivez ces deux options, la description courte sera quand même affichée dans la section détaillée de la documentation.

DETAILS_AT_TOP Cochée, cette option affiche la description détaillée du fichier en haut de la page.

6.2 Onglet Expert>Build

EXTRACT_PRIVATE (C++ et autres langages orientés objet) Permet de faire afficher la documentation des fonctions/méthodes et autres membres ayant le qualificatif `private`.

EXTRACT_STATIC Permet de faire afficher la documentation des fonctions/structures/etc. ayant le qualificatif `static`.

SORT_MEMBER_DOCS Cochée, cette option permet de trier les descriptions détaillées par ordre alphabétique. Sinon l'ordre est celui de leur déclaration dans le code source.

EXTRACT_ALL Même effet que d'activer l'option `All entities` dans les options Wizard.

6.3 Onglet Expert>HTML

HTML_HEADER Personnalise l'en-tête des documentations.

HTML_FOOTER Personnalise le pied des pages des documentations.

HTML_STYLESHEET Précise le fichier CSS (le *design*).

HTML_OUTPUT Redéfinit le nom du dossier de générations (par défaut `html/`).

HTML_FILE_EXTENSION Type de l'extension.

DISABLE_INDEX Permet de ne pas afficher l'index en haut de chaque page HTML de la documentation.

6.4 Onglet Expert > Dot

UML_LOOK Permet de générer des diagrammes dans le style UML.

DOT_IMAGE_FORMAT Permet de choisir le format d'images (png, jpg, gif).

DOT_PATH Chemin du logiciel GRAPHVIZ
(par défaut, référencé dans la variable PATH).

MAX_DOT_GRAPH_WIDTH Permet de changer la largeur/régler maximale d'un graphe.

MAX_DOT_GRAPH_HEIGHT La hauteur maximale d'un graphe.

MAX_DOT_GRAPH_DEPTH La profondeur maximale d'un graphe.

DOT_TRANSPARENT Permet de rendre le fond des images transparent (utile lorsque l'on personnalise l'aspect général de la documentation).