

Doxygen

Outil de génération de
documentation

Doxygen : description 1/2

- Génération de documentation d'un programme
 - Pour programmeurs ou utilisateurs finaux
- Permet de générer la documentation :
 - à partir des commentaires insérés dans le code source
 - à partir de la structure du code lui même. La documentation générée sera dans ce cas minimale.

Doxygen : description 2/2

- Doxygen est la contraction de « *dox* » (« *docs* ») et de « *gen* » (« *generator* »), « créateur de documentation ».
- Flux de génération :
 - Entrée : code source : C, C++, Obj.C, Java...
 - Sortie : Html, Latex, PostScript, Unix Man, PDF...

Doxygen : avantages 1/2

- Avantages d'une documentation ?
 - Synthétiser l'architecture globale du code
 - Prise de connaissance plus rapide des sources
 - Mieux évaluer la portée d'une modification du code
 - Permet de mieux cibler où intervenir dans le code
 - Regroupe les informations rajoutées sur des contraintes de codage

Doxygen : avantages 2/2

- Les informations suivantes peuvent être extraites des sources :
 - prototype et documentation des fonctions, qu'elles soient locales, privées ou publiques
 - 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

Doxygen : fonctionnement 1/2

- 1^{ère} étape : remplir un fichier de configuration
 - Création : `doxygen -g <nom du fichier>`
 - édition (options simples et explicites)
- 2^e étape : lancer la génération
 - Ligne de commande
 - `doxygen nom du fichier` (s'il n'y a pas de nom Doxyfile est le nom par défaut)

Doxygen : fonctionnement 2/2

- Comment est généré la doc ?
 - Exploration du code source : Doxygen a un analyseur sémantique
 - Exporte ces infos. Ex: hiérarchie des classes navigable en html.
 - Associe à chaque classe/fonctions/attributs... des commentaires placés entre balises
 - Liens hypertextes possible ?

Doxygen : fichier de configuration 1/2

- Configuration:
 - PROJECT_NAME : nom du projet
 - PROJECT_NUMBER : numéro de version du projet
 - OUTPUT_DIRECTORY : emplacement de la doc générée
 - OUTPUT_LANGUAGE : langue dans laquelle la doc est écrite
 - OPTIMIZE_FOR_XX : s'il est à YES, optimise la tolérance sur les tags de XX. XX étant le nom du langage.

Doxygen : fichier de configuration 2/2

- `INPUT` : spécifie l'emplacement du répertoire dans lequel se trouve les sources, ou le nom du fichier dont on veut générer la documentation.
 - `FILE_PATTERNS` : spécifie l'extensions des sources si `INPUT` définit un répertoire.
 - `EXCLUDE` : exclue les fichiers non concernés par la génération de la documentation.
-
- Exemple à venir...

Doxygen : licence & extensions

- Licence GPL pour le programme, aucune pour les fichiers générés.
 - Docs utilisables dans les applications commerciales
 - Support par une large communauté
- Possibilité de développer un module pour un langage non supporté (/!\ GPL)

Doxygen : Exemple pour C++

```
/**
```

```
* @brief Description du comportement de  
* la fonction  
* @param nom du paramètre et sa description  
* @param (autant de fois que nécessaire)  
* @return (s'il y a un retour)  
* @author nom et prénom de l'auteur de la méthode  
*/
```

- Remarque: le tag @brief n'existe pas en langage java

Doxygen : Démonstration

Démonstration d'une génération de documentation à partir d'un fichier c++