

Utilisation de doxygen

Doxygen permet de générer une documentation très efficace de vos codes sources aux formats HTML, LaTeX, man, rtf ou XML. Associé avec le logiciel graphviz, Doxygen va aussi permettre de générer de façon graphique les liens entre les classes.

Installation et configuration

1. Installer Graphviz puis Doxygen (dans cet ordre)
2. Configuration
Lancez Doxywizard

➤ **Onglet Wizard**

Topic Project : Définir le nom du projet, sa version, l'emplacement des fichiers source et l'emplacement de la documentation à générer. Cochez scan recursively si des sous-dossiers existent au niveau de votre répertoire source.

Topic Mode : Si vous voulez que la doc inclut aussi le code source, sélectionnez "all entities" et cochez "include cross reference source code in the output". Choisissez aussi "optimize for C++ output".

Topic Output : Choisissez HTML

Topic Diagrams : sélectionnez "use dot tools from the GraphViz package to generate".

➤ **Onglet Expert**

Topic Project : Sélectionnez Output_Language : French

Topic Build : Extract all

Topic Input :

INPUT_ENCODING : ISO-8859-1 ➔ pour avoir les accents

IMAGE_PATH : "Répertoire de vos images"

Topic Dot : Cochez Class Diagram et UML_LOOK et indiquez le chemin de GraphViz dans la zone DOT_PATH (chez moi "C:\Program Files\Graphviz\bin")

- Sauvez ensuite votre fichier de configuration (**bouton Save**)

- Lancez doxygen (**bouton Start**)

*A ce stade, une documentation est déjà générée par Doxygen.
Elle sera d'autant plus riche que vous y aurez inclus des balises spécifiques*

Principe général

Commenter son code source avec doxygen consiste à ajouter des commentaires spéciaux qui vont être interprétés par Doxygen.

Plusieurs solutions sont possibles pour commenter le code "façon" Doxygen

```
/**  
 blabla  
 blabla  
 */
```

```
/*!  
 blabla  
 Personnellement, j'utilise cette solution  
 */
```

Pour les commentaires sur une seule ligne, on peut écrire :

```
///  
 blabla pour commenter ce qui est écrit après ce commentaire
```

```
///  
 blabla pour commenter ce qui est écrit avant ce commentaire
```

C'est quoi ce blabla ?

Nous allons ajouter des balises (\NomDeLaBalise) pour commenter différents aspects du code :

- des informations générales sur l'auteur, la date, les versions, les bugs détectés, ...
- des informations pour détailler la structure des classes, les paramètres des méthodes, leurs types de retours, etc.

Le document généré peut aussi être l'occasion d'ajouter des précisions, des images, des schémas, etc. Comme c'est une documentation de code source, le but est bien sûr d'assurer la maintenabilité maximale du projet pour que d'autres personnes que vous puissent comprendre et reprendre aisément votre programme.

[La documentation Doxygen](#), très riche, vous permettra de trouver votre bonheur !

Bon, on y va ?

Informations pour la page principale de notre projet à commenter

Nous pouvons ajouter ces informations en haut de notre programme principal

```

/! \mainpage Modélisation de la location de voitures
\author Super développeur
\version 1.0
\date 2019
\image html voiture1.jpg
\warning N'oubliez pas de laver votre voiture avant de la rendre :- )
\bug Aucun bug détecté à ce jour
\section intro_sec Introduction
Modélisation de la location de voiture
\section install_sec Informations pour la mise en oeuvre
\subsection software_sec Informations logicielles
Le logiciel développé fonctionne dans l'environnement Windows et devrait pouvoir avec
des modifications minimales fonctionner sous UNIX
*/

```

Tout ce qui suit est un mot clé Doxygen :

`intro_sec` est le nom que je donne à ma section (j'aurais pu l'appeler autrement).

Commentons les classes

Nous allons travailler dans un premier temps sur la classe Personne

Voici quelques informations à ajouter dans le fichier de déclaration.

```

#pragma once
#include <string>
using namespace std;

class Voiture; // je déclare l'existence de la classe Voiture

/! \class Personne
\brief Classe permettant de modéliser une personne
\details Classe élémentaire modélisant le loueur de la voiture
\image html cht1.jpg
*/
class Personne
{
private:
    string nom, prenom;

public:
    Personne(string nom , string prenom);
    void louer(Voiture *tasDeBoue); // louer une voiture

    // Ajoutez ici le lien avec la classe Voiture
    Voiture *laVoiture; //!< Association
};

```

On peut inclure une image

Quand le commentaire concerne quelque chose qui précède, on utilise la balise `//!<`

Voici maintenant d'autres informations pour le fichier de définition

```
#include "Personne.h"

/*!
Constructeur
\param nom          Le nom de la personne
\param prenom       Le prénom de la personne
*/
Personne::Personne(string nom, string prenom)
{
    this->nom = nom;
    this->prenom = prenom;
}

/*!
\param tasDeBoue     La voiture louée
*/
void Personne::louer(Voiture *tasDeBoue)
{
    this->laVoiture = tasDeBoue;
}
```

- \param pour préciser les paramètres d'une fonction ou méthode.
- \return pour préciser ce que retourne la fonction
- On peut aussi placer des balises HTML (par ex)

La bonne démarche

Maintenant, quand vous coderez, pensez à tout de suite commenter votre code "à la sauce Doxygen". Ainsi, à la fin, vous aurez très peu de choses à ajouter pour générer une documentation de code source quasi parfaite !