



# Doxygen - Quick Guide

Rédigé par : Marc Chevaldonné, IUT Informatique, Université d'Auvergne Clermont1

19 septembre 2013

Version 3.0

## Sommaire

<b>Introduction</b>	<b>2</b>
Contexte	2
Documents de références	2
Table des modifications	2
<b>Qu'est-ce que Doxygen ?</b>	<b>3</b>
<b>Installation de Doxygen</b>	<b>3</b>
<b>Création d'une documentation à l'aide du Doxygen Wizard</b>	<b>4</b>
<b>Utilisation de Doxygen dans Visual Studio 2012</b>	<b>8</b>
<b>Fichiers de configuration</b>	<b>8</b>
Rappel de l'arborescence d'un projet	8
Les fichiers de documentation dans l'arborescence	9
Le fichier de configuration Doxyfile	9
Un footer pour la documentation html	10
<b>Intégration dans Visual Studio</b>	<b>10</b>
<b>Pour aller plus loin avec Doxygen et Visual Studio</b>	<b>13</b>
Ne générer la documentation qu'en mode release	13
Deux documentations différentes en une seule génération	14
<b>Annexe 1 : Exemple de Doxyfile</b>	<b>16</b>
<b>Annexe 2 : Exemple de Doxyfile publique</b>	<b>16</b>

## 1. Introduction

### 1.1. Contexte

Ce document a été écrit pour mes élèves de l'IUT d'Informatique de Clermont-Ferrand, Université d'Auvergne. Son but est de décrire succinctement les fonctionnalités de Doxygen et les outils de base pour une utilisation à travers le Doxygen Wizard d'une part et en liaison avec VisualStudio d'autre part.

Le document a été écrit dans le cadre du cours sur le C#. En conséquence, on considère que les étudiants savent déjà documenter leur code en utilisant le système de balises XML du C#. Le lecteur peut trouver plus d'informations sur la rédaction de commentaires en C# dans le document sur les règles de codage ([Coding Style Guidelines](#)) également fourni aux étudiants.

### 1.2. Documents de références

Pour plus d'informations, le lecteur peut se documenter en lisant les documentations suivantes :

Title	Version	Lien
Doxygen Manual	1.8.5	<a href="http://www.stack.nl/~dimitri/doxygen/manual.html">http://www.stack.nl/~dimitri/doxygen/manual.html</a>
Coding Style Guidelines	5.0	<a href="http://marc.chevaldonne.free.fr/ens_rech/DocumentsDivers_files/Coding_Style_Guide.pdf">http://marc.chevaldonne.free.fr/ens_rech/DocumentsDivers_files/Coding_Style_Guide.pdf</a>

### 1.3. Table des modifications

Author	Modification	Version	Date
Marc Chevaldonné	Rédaction du document	1.0	12 octobre 2009
Marc Chevaldonné	Mise à jour pour VisualStudio2010 Ajout du Doxyfile_public	2.0	06 août 2010
Marc Chevaldonné	Mise à jour pour VisualStudio2012	3.0	19 septembre 2013

## 2. Qu'est-ce que Doxygen ?

Doxygen est un système de documentation pour les langages C++, C, Java, Objective-C, Python, PHP, C#...

Doxygen vous aide à générer une documentation en ligne (html) et/ou un manuel de référence en Latex par exemple, à partir d'un ensemble de fichiers source. Doxygen autorise également la génération de documentation en RTF, PostScript PDF, pages man Unix, etc. La documentation est directement extraite du code source.

Doxygen est multi-plateforme et fonctionne sous Linux, MacOSX et Windows. Il est libre et gratuit.

## 3. Installation de Doxygen<sup>1</sup>

Vous trouverez la dernière release de Doxygen ici : <http://www.stack.nl/~dimitri/doxygen/download.html>

---

<sup>1</sup> Doxygen est déjà installé sur les machines de la salle B20

## 4. Création d'une documentation à l'aide du Doxygen Wizard

La première méthode que nous proposons ici consiste à utiliser le Doxygen Wizard pour créer une documentation de notre code. Suivez les étapes suivantes une par une pour générer la documentation de votre projet. On considère une nouvelle fois que votre code est commenté.

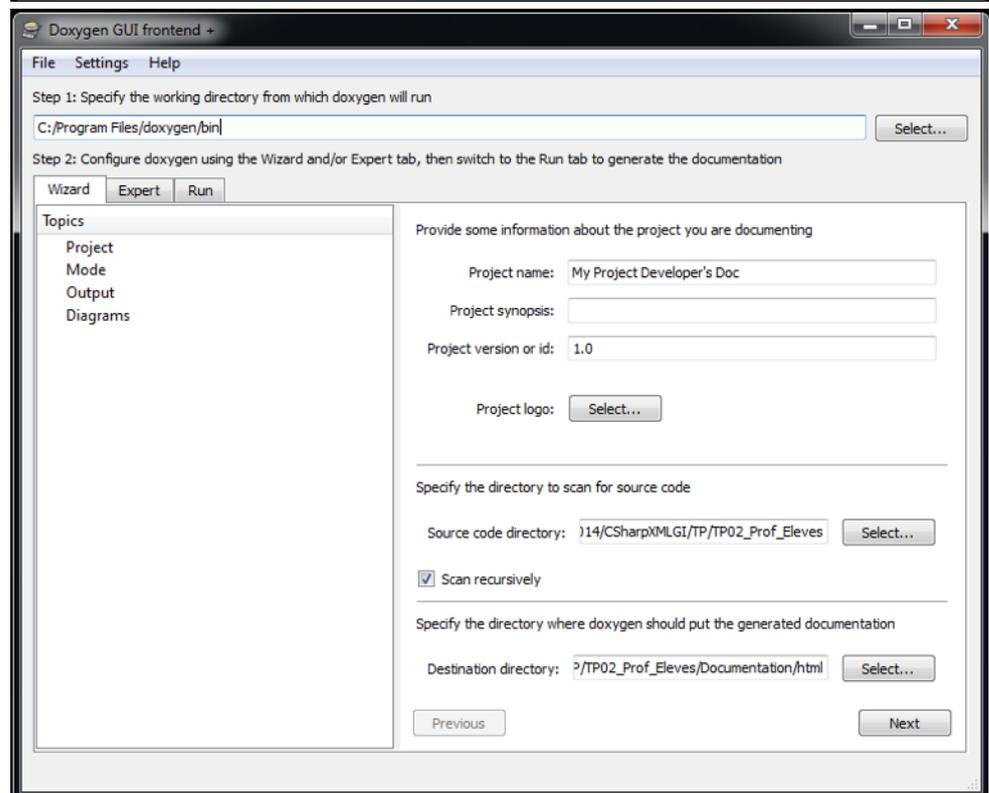
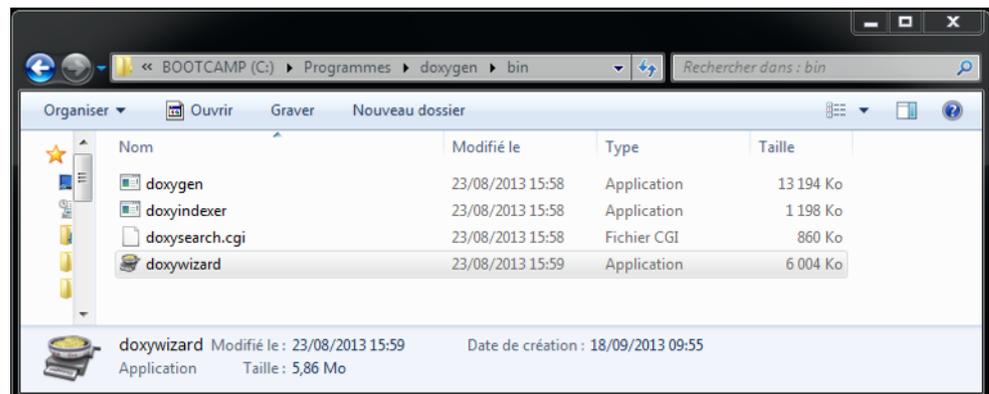
1. Rendez vous dans le dossier d'installation de Doxygen et dans le sous-dossier bin, double cliquez sur doxywizard.exe.

Ce wizard va nous aider à configurer les fichiers à documenter et la forme de la documentation.

2. Dans la fenêtre qui s'ouvre, remplissez le champ «Step 1» avec le dossier bin de Doxygen. Pour cela, vous pouvez cliquer sur le bouton «Select» à droite de cette ligne et parcourir l'explorer Windows pour le sélectionner.

3. Dans l'étape 2 «Step 2», cliquez sur l'onglet Wizard, puis le sujet «Project» comme sur l'image ci-dessous. Vous pouvez alors donner un nom à votre projet et un numéro de version. Vous devez ensuite spécifier un chemin pour votre code. En cochant la case «Scan recursively», vous

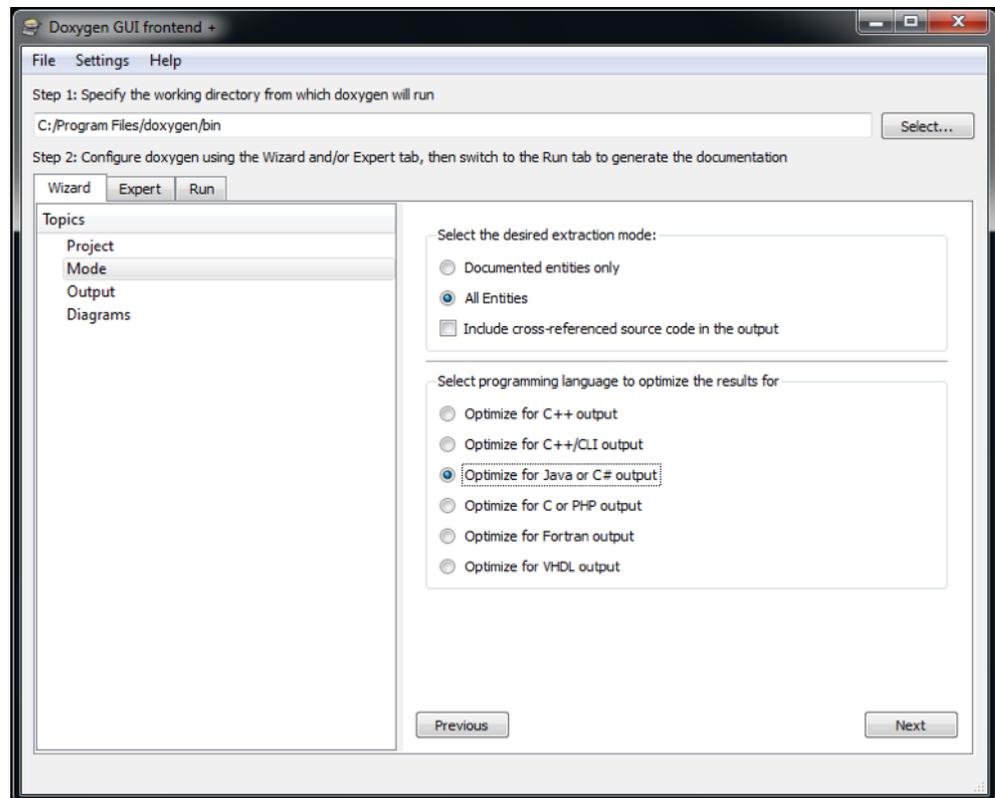
indiquerez à Doxygen de chercher dans les dossiers fils, les fichiers sources à documenter. Enfin, vous devez indiquer le chemin de sortie de votre documentation, par exemple Documentation -> html<sup>2</sup>.



<sup>2</sup> Voir la [partie](#) sur les conseils sur l'arborescence de votre projet

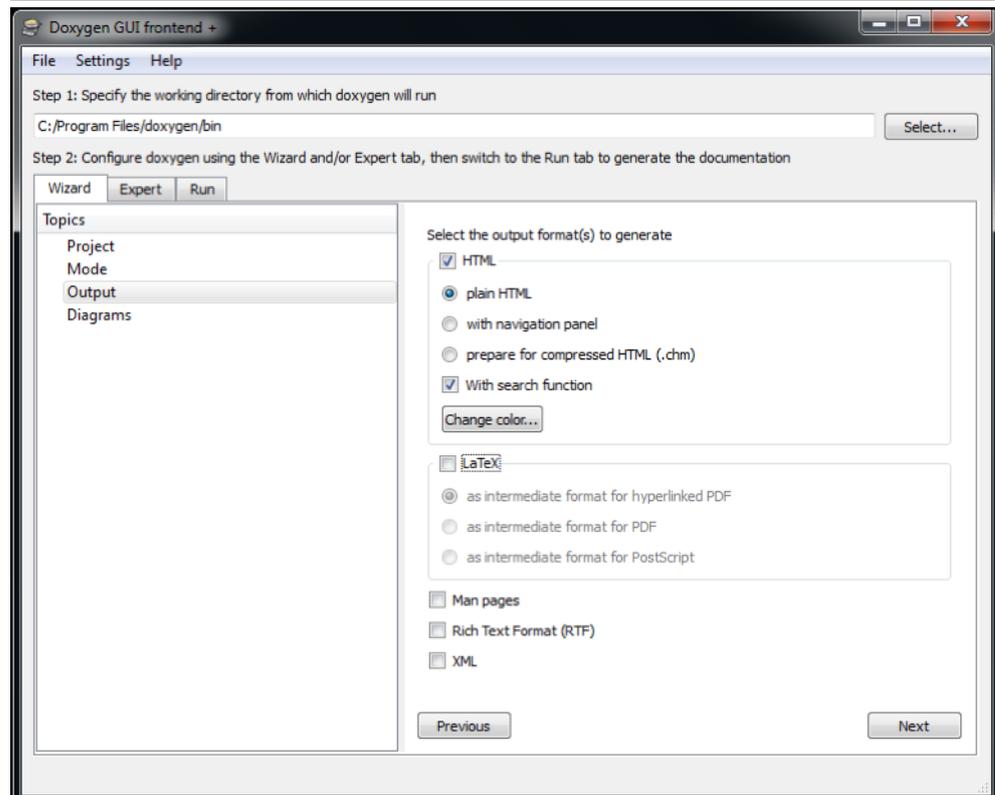
4. Cliquez ensuite sur le sujet Mode (toujours dans l'onglet Wizard de l'étape 2), ou sur le bouton «Next» en bas à droite.

Choisissez un mode d'extraction «All Entities» : ceci vous permettra rapidement d'identifier les morceaux de code non documentés. Puis choisissez «Optimize for Java or C# output» puisque nous nous intéressons ici au cours de C#.



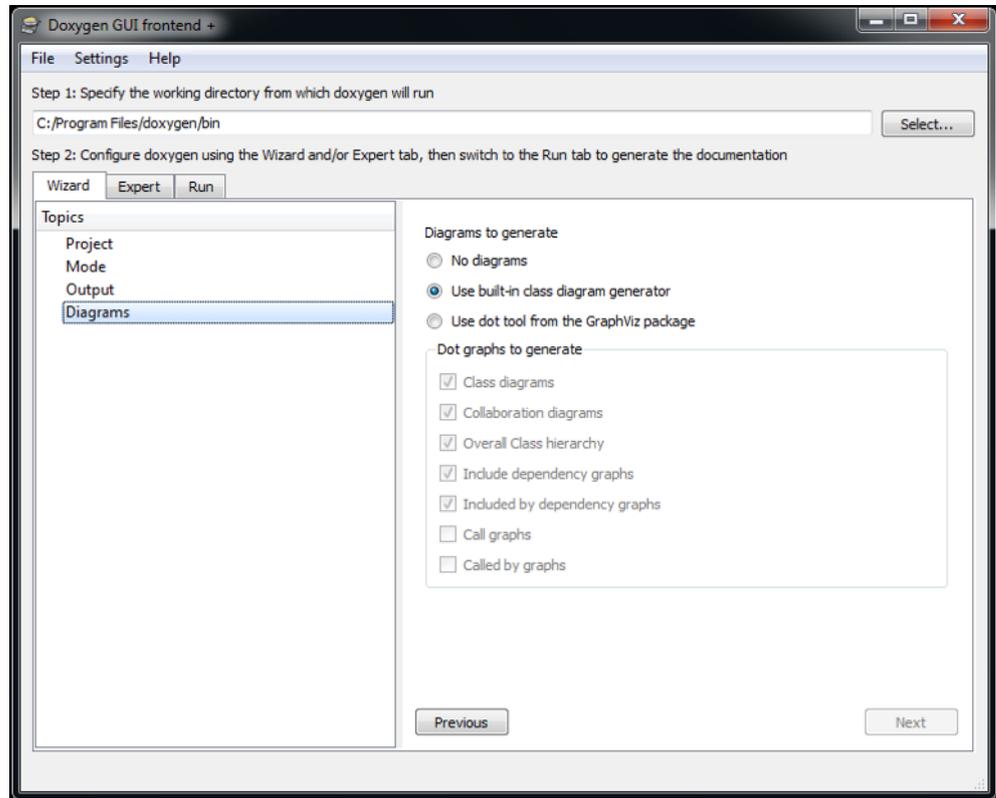
5. Cliquez ensuite sur le sujet Output (toujours dans l'onglet Wizard de l'étape 2), ou sur le bouton «Next» en bas à droite.

Cochez «HTML» puis «plain HTML» et décochez «LaTeX» (ici nous ne chercherons qu'à faire une documentation HTML).



6. Cliquez ensuite sur le sujet Diagrams (toujours dans l'onglet Wizard de l'étape 2), ou sur le bouton «Next» en bas à droite.

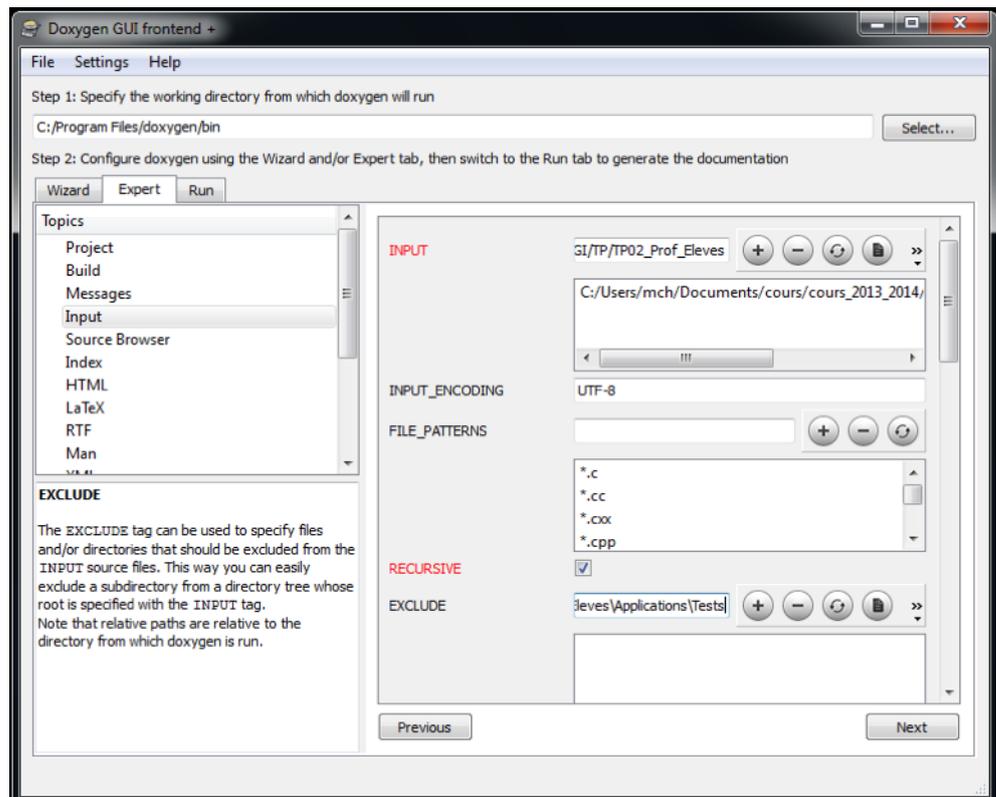
Sélectionnez «Use built-in class diagram generator».



7. Cliquez ensuite sur l'onglet «Expert» (toujours dans l'étape 2).

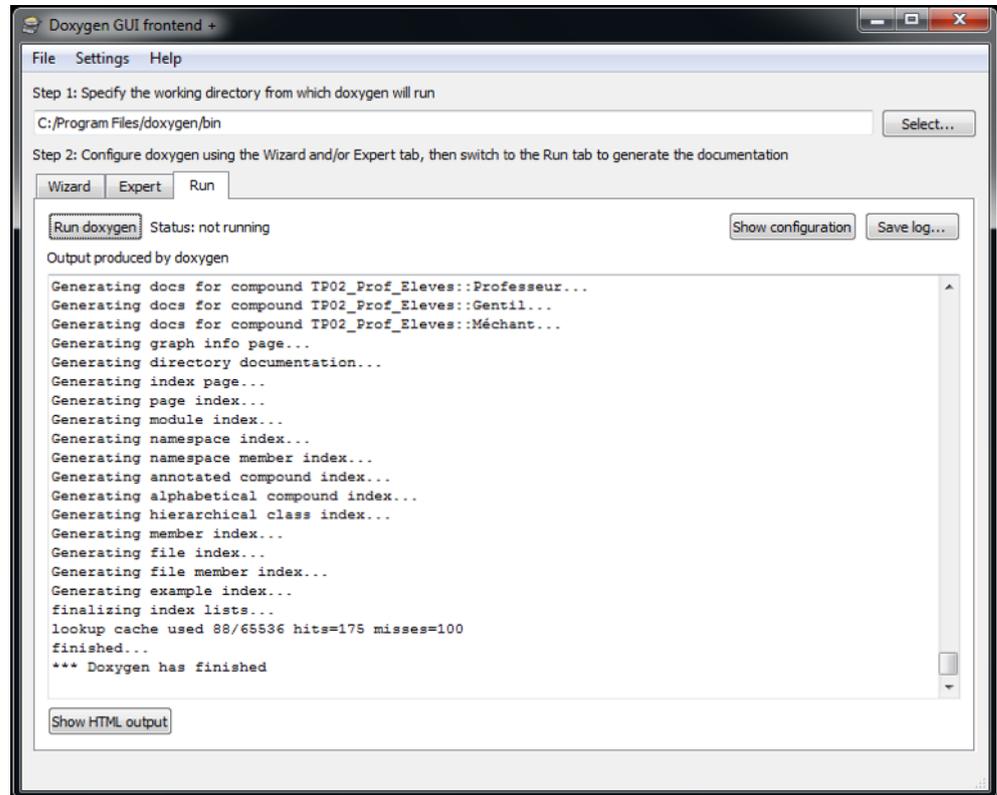
Ici, vous avez accès à de nombreux sujets qui vous permettront de modifier le contenu et la forme de votre documentation. Certains se comprennent facilement d'autres sont un peu plus tordus.

Placez-vous dans le sujet «Input» et modifiez l'entrée EXCLUDE en cliquant à droite sur le double flèche pour sélectionner le dossier. Allez chercher le chemin «Solutions/Applications/Tests» de votre projet. De cette manière, les codes source de tests ne seront pas documentés.



8. Enfin, placez-vous dans le dernier onglet «Run» de l'étape 2. Lancez doxygen en cliquant sur «Run doxygen». Votre documentation est construite. Cliquez sur «Show HTML output» pour la visualiser. Si elle vous semble un peu vide, allez vite commenter votre code.

Si vous voulez fournir votre documentation à votre client, recopiez l'ensemble du dossier html de sortie. Pour observer la documentation depuis ce dossier, double cliquez sur le fichier index.html.



## 5. Utilisation de Doxygen dans Visual Studio 2012

Il est également possible de générer votre documentation à l'aide de Doxygen et à partir de Visual Studio 2012 à chaque compilation. Ceci peut s'avérer très pratique et très agréable car :

- votre documentation est ainsi toujours à jour,
- la documentation que vous générez est liée au projet ouvert,
- vous pouvez sauvegarder vos options de documentation directement dans votre projet.

Dans un premier temps, nous allons proposer une arborescence pour la gestion de la documentation, puis nous présenterons comment utiliser Doxygen dans VisualStudio2012.

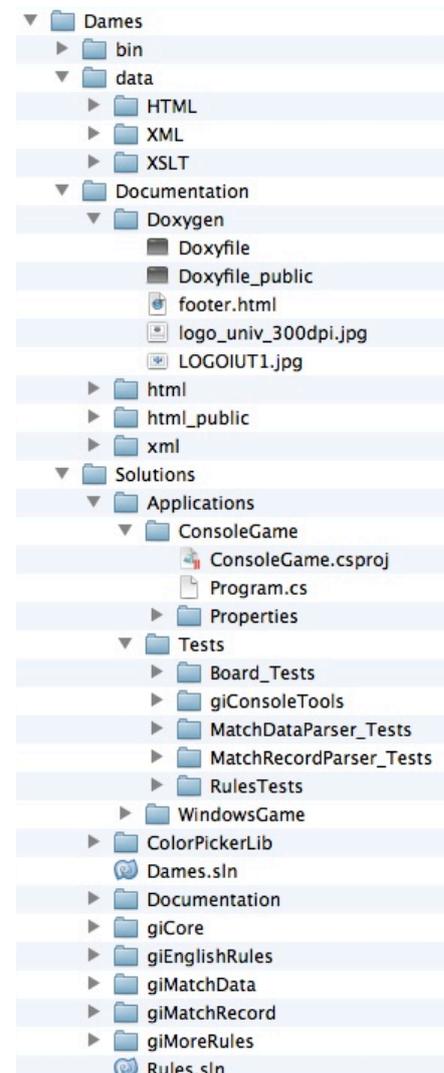
### 5.1. Fichiers de configuration

#### 5.1.1. Rappel de l'arborescence d'un projet

Voici un petit rappel sur l'arborescence conseillée pour la gestion de projets en C#<sup>3</sup>. Celle-ci est constituée d'un dossier Solutions contenant :

- la ou les solutions du projet (par exemple Rules.sln, Dames.sln et Dames\_all.sln où la 2ème contient également les tests).
- un sous-dossier par bibliothèque de classes (par exemple giCore, giEnglishRules) contenant le projet correspondant \*.csproj et les fichiers sources.
- un sous-dossier Applications contenant les projets exécutables comme ConsoleGame (les interfaces graphiques notamment comme WindowsGame) et un sous-dossier Tests contenant un sous-dossier par projet de test.
- un sous-dossier data avec les fichiers HTML, XML, XSLT, mais aussi les éventuelles images ou autres fichiers de ressources.
- un dossier Documentation dont nous parlerons par la suite.

Enfin, elle contient également un dossier bin contenant deux sous-dossiers Debug et Release correspondant aux chemins de sortie des projets précédents.



<sup>3</sup> cf . [Coding Style Guidelines v3.0](#)

### 5.1.2. Les fichiers de documentation dans l'arborescence

L'arborescence donnée en cours propose de séparer les fichiers de configuration de VisualStudio2012 du projet. Cela peut s'avérer utile au cas où les utilisateurs souhaiteraient générer la documentation sans passer par Visual Studio.

Nous proposons donc de rajouter un dossier Documentation (contenant un sous-dossier Doxygen) au même niveau que les dossiers Solutions et bin. Puis, nous rajouterons un sous-dossier Documentation à Solutions contenant le projet MakeFile (ceci sera fait automatiquement par VisualStudio2012).

### 5.1.3. Le fichier de configuration Doxyfile

Dans le dossier Documentation->Doxygen, ajoutez un nouveau fichier «Doxyfile» sans extension. Ce fichier va contenir les mêmes informations que celles créées par le Doxywizard dans la [section précédente](#) (en particulier le [numéro 7](#)). Vous avez plusieurs façons de le créer, en voilà deux :

- ✓ dans l'onglet «Run» du Doxywizard, cliquez sur le bouton «Show configuration» et recopiez le contenu du fichier de configuration qui apparaît dans la fenêtre.
- ✓ recopiez le contenu de l'[annexe 1](#) et adaptez-le à votre besoin.

Ce fichier contient de nombreuses options, celles que vous avez le plus de probabilités de modifier sont :

- ▶ **PROJECT\_NAME** : le nom de votre projet
- ▶ **OUTPUT\_DIRECTORY** : le chemin de sortie de votre projet. Si vous respectez l'arborescence conseillée, celui-ci sera «../»
- ▶ **EXTRACT\_PRIVATE** : si cette option vaut **YES**, les membres privés des classes sont inclus dans la documentation
- ▶ **INPUT** : les dossiers à prendre en compte dans la recherche de fichiers source. Vous pouvez en indiquer plusieurs à l'aide du symbole \. Si vous respectez l'arborescence conseillée, vous utiliserez certainement la valeur suivante : ../..
- ▶ **FILE\_PATTERNS** : l'extension des fichiers à documenter. Dans notre cas, il s'agit donc de \*.cs. Toutefois, si vous ne mettez rien comme valeur, Doxygen considérera tous les fichiers avec les extensions suivantes : \*.c \*.cc \*.cxx \*.cpp \*.c++ \*.java \*.ii \*.ixx \*.ipp \*.i++ \*.inl \*.h \*.hh \*.hxx \*.hpp \*.h++ \*.idl \*.odl \*.cs \*.php \*.php3 \*.inc \*.m \*.mm \*.py \*.f90
- ▶ **RECURSIVE** : si la valeur est **YES** (conseillée), alors les sous-dossiers de **INPUT** sont eux aussi parsés pour la recherche de fichiers source.
- ▶ **EXCLUDE** : permet d'indiquer les dossiers à exclure de la documentation. Dans notre cas, nous ne souhaitons pas documentés les fichiers source des applications de tests. Nous pouvons donc exclure : ../..Solutions/Applications/Tests
- ▶ **GENERATE\_HTML** : **YES** si vous voulez générer une sortie en html
- ▶ **HTML\_OUTPUT** : nom du dossier qui sera ajouté au chemin **OUTPUT\_DIRECTORY**, (html généralement).
- ▶ **HTML\_HEADER** , **HTML\_FOOTER** : si vous voulez rajouter un header ou un footer à vos pages html générées, vous pouvez les inclure ici.

#### 5.1.4. Un footer pour la documentation html

Doxygen permet d'ajouter très facilement un footer ou un header à votre documentation html : c'est joli et c'est pratique pour garder les liens vers votre projet.

Téléchargez d'abord les images (pour faire joli) de votre entreprise ou laboratoire. Dans notre cas, nous pouvons marquer notre documentation avec les images de l'IUT et de l'Université (les logos sont sur Hina). Placez-le dans le dossier Documentation->Doxygen.

Dans un fichier footer.html que vous pouvez créer dans le dossier trunk->Documentation->Doxygen, ajoutez les lignes suivantes :

```
<html>
  <body>
    <p>
      <hr size="1"/>
      <address style="text-align: right;">
        <small>Generated on $datetime at &nbsp;
          <a href="http://iutweb.u-clermont1.fr/">
            
          </a>
          <a href="http://www.u-clermont1.fr/">
            
          </a> by Doxygen version $doxygenversion
        </small>
      </address>
    </p>
  </body>
</html>
```

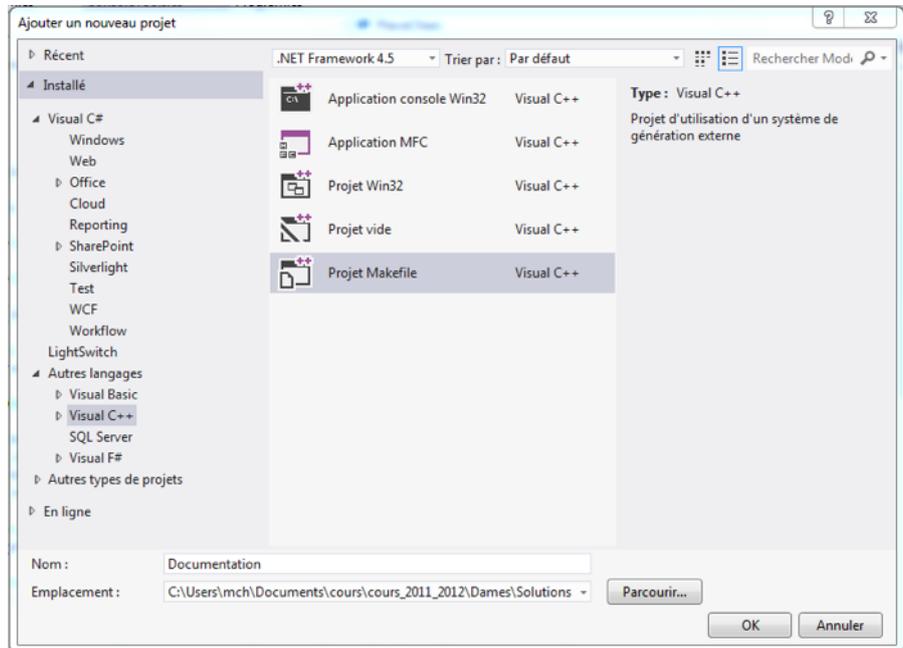
Les étiquettes `$datetime` et `$doxygenversion` seront remplacées par la date et l'heure de la génération de la documentation et le numéro de version de Doxygen utilisé lors de la génération.

## 5.2. Intégration dans Visual Studio

Dans cette section, nous allons intégrer Doxygen dans la solution de notre projet afin que la documentation se génère automatiquement à chaque compilation.

1. Ouvrez votre solution (Dames.sln par exemple) et ajoutez-lui un nouveau projet. Choisissez un projet de type «Makefile Project» dans l'onglet de type de projets «Visual C++» à gauche dans la fenêtre d'ajout d'un projet<sup>4</sup>.

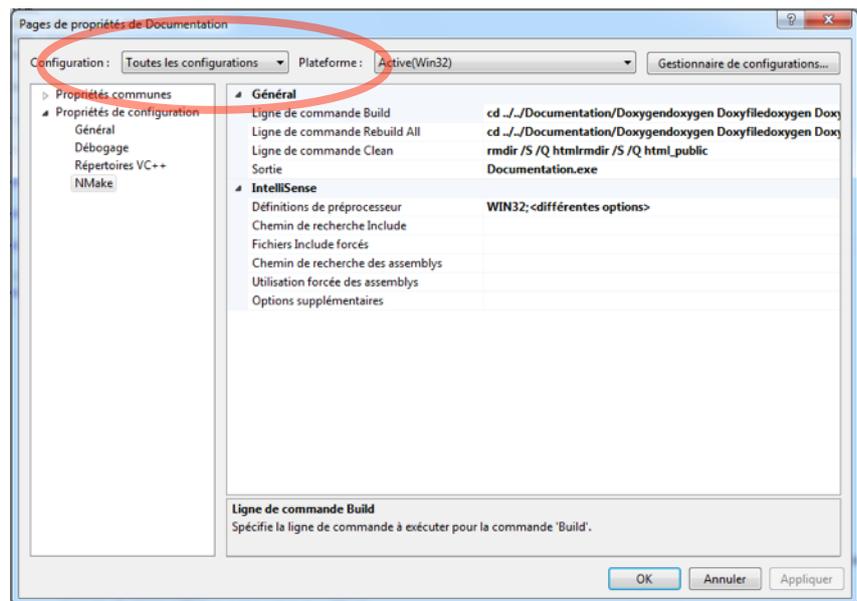
Nommez ce projet «Documentation» et placez-le dans le dossier trunk- >VisualStudio2010. Cliquez sur OK.



Dans la nouvelle fenêtre qui s'ouvre pour configurer votre projet Makefile, cliquez sur «Terminer» («Finish»).

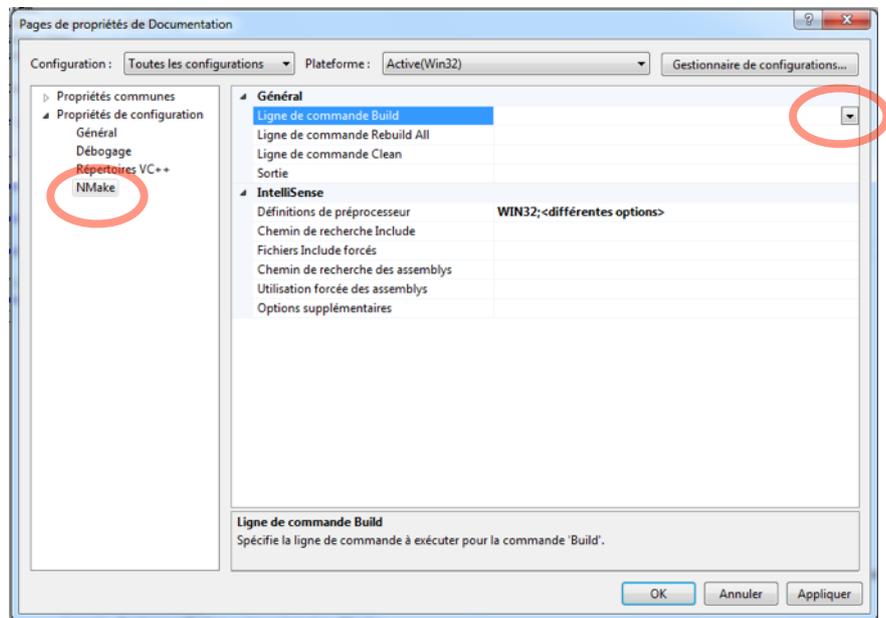
2. Dans VisualStudio, faites un clic droit sur le projet «Documentation» et cliquez sur «Propriétés» («Properties»).

Dans la fenêtre qui s'ouvre, en haut à gauche dans «Configuration», choisissez «Toutes les configurations» («All Configurations»).

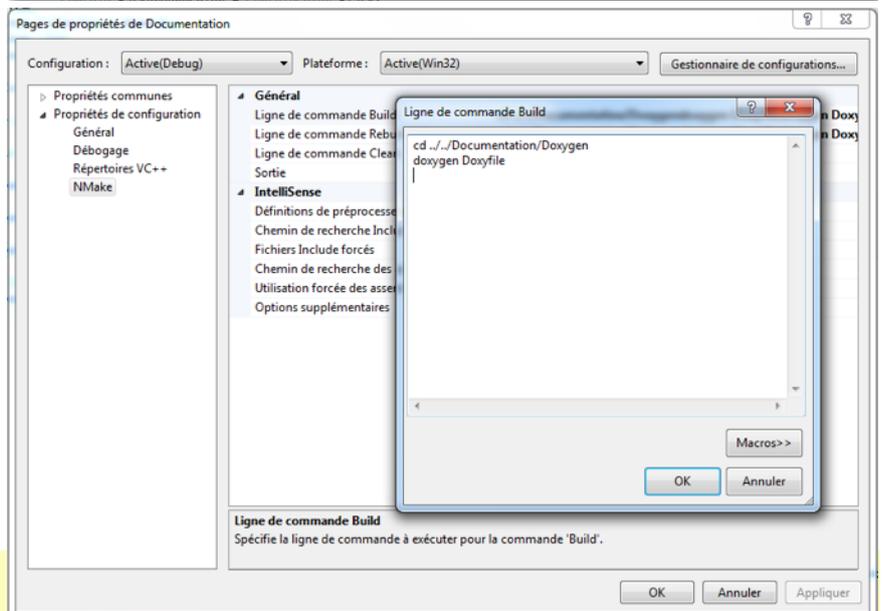


<sup>4</sup> Si vous n'avez installé que Visual C# Express, il est possible que vous n'ayez pas accès aux projets de type Visual C++. Il vous faut donc également installer Visual C++ Express.

Cliquez ensuite sur «NMake» dans «Propriétés de configuration» («Configuration Properties») sur la gauche de la fenêtre. Cliquez ensuite sur le bouton «▼» à droite de la ligne «Ligne de commande Build» («Build Command Line»).



Dans la nouvelle fenêtre, tapez les deux lignes suivantes (si vous n'utilisez pas les [conseils d'arborescence](#) présentés précédemment, vous devrez certainement changer la première ligne) :



```
cd .././
Documentation/
Doxygen
doxygen Doxyfile
```

Répétez la même opération pour la ligne «Ligne de commande Rebuild All» («Rebuild All Command Line»).

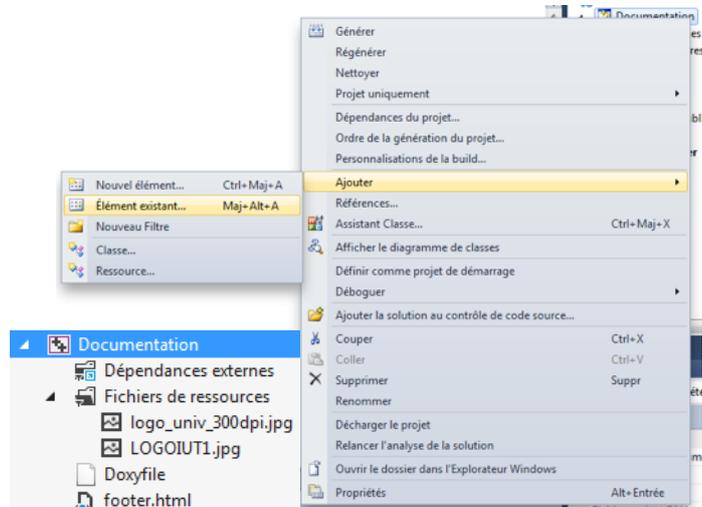
Pour la ligne «Ligne de commande Clean» («Clean Command Line»), tapez la ligne suivante (où html correspond au dossier fourni dans [HTML\\_OUTPUT](#) du fichier Doxyfile) :

```
rmdir /S /Q html
```

Fermez la fenêtre en cliquant sur OK.

- Faites un clic droit sur votre projet, puis cliquez sur «Ajouter» («Add»), puis «Élément existant...» («Existing Item...»).

Parcourez l'explorer Windows pour vous placer dans le dossier Documentation->Doxygen et choisissez les fichiers Doxyfile, footer.html ainsi que les images utilisées par ce footer. Vous devriez obtenir l'arborescence suivante pour votre projet Documentation (remarquez qu'on peut supprimer les dossiers Header Files et Source Files ainsi que le fichier readme.txt, inutiles pour notre projet).



- Maintenant, vous pouvez générer votre documentation en générant le projet «Documentation.vcproj». Faites un clic droit sur ce projet et cliquez sur «Générer» («Build»). Vous devriez maintenant avoir un dossier «html» dans votre dossier Documentation->Doxygen. Sous VisualStudio, faites un clic droit sur le projet Documentation, cliquez sur «Ajouter» («Add») puis «Élément Existant...» («Existing Item...») et allez chercher le fichier «index.html» dans ce dossier «html». Désormais, à chaque fois que vous générerez la documentation, il vous suffira de faire un clic droit sur index.html sous VisualStudio pour l'afficher dans votre browser par défaut, sans avoir à aller le chercher dans le dossier html.

**Remarque importante** : si vous utilisez Subversion ou Git ou n'importe quel autre système de contrôle de version pour votre projet, ne versionnez aucun des fichiers du dossier html ! Ceux-ci sont générés automatiquement lors de la génération du projet Documentation. Il n'est donc pas utile de les versionner.

### 5.3. Pour aller plus loin avec Doxygen et Visual Studio

Dans cette partie, nous présentons deux améliorations à apporter à votre projet lors de l'utilisation conjointe de Visual Studio et Doxygen.

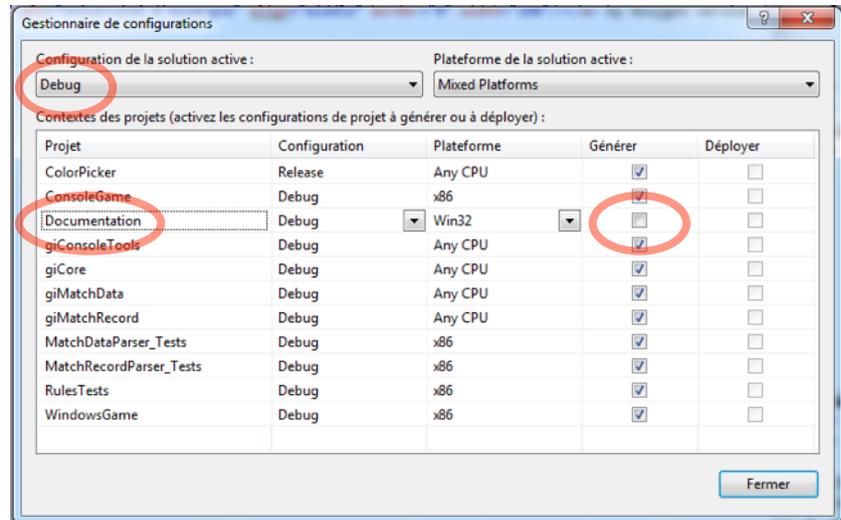
#### 5.3.1. Ne générer la documentation qu'en mode release

Lors du développement d'un projet, la compilation de chaque projet Visual prend un certain temps. En mode développement, il n'est donc pas nécessaire de générer à chaque fois la documentation avec Doxygen. En général, on développe en mode Debug et on compile la version client en mode Release. Une bonne solution consiste donc à n'activer la génération qu'en mode Release.

Pour cela, faites un clic droit sur la solution de votre projet (par exemple Dames.sln) sous Visual Studio, puis cliquez sur «Gestionnaire de configurations» («Configuration Manager...»).

Choisissez le mode Debug dans «Configuration de la solution active» («Active solution configuration»), puis dans la ligne correspondant à votre projet «Documentation», décochez la case «Générer» («Build»).

Désormais, lorsque vous compilerez votre solution, le projet Documentation.vcproj ne sera compilé qu'en mode release (vous pouvez quand même le compiler à la main (clic droit puis «Générer» («Build»)) en mode debug si vous le souhaitez).



### 5.3.2. Deux documentations différentes en une seule génération

Il peut être parfois utile de générer différentes documentations : une pour les développeurs du projet et une autre pour les personnes extérieures qui voudraient ajouter des plugins. La première documentation serait donc complète. En revanche, la deuxième ne contiendrait que certaines parties publiques (un SDK). Dans le cadre du développement d'un jeu de Dames, on peut donc imaginer une documentation pour les développeurs du jeu ; on peut laisser la possibilité aux joueurs de développer leur propre joueur avec Intelligence Artificielle (en leur permettant de faire un plugin), et pour cela, on peut leur fournir une autre documentation pour les développeurs d'IA, moins complète.

Pour cela, il suffit juste d'utiliser deux Doxyfile différents et deux chemins de sortie différents.

On peut donc rajouter au projet Documentation.vcproj un nouveau fichier Doxyfile que nous renommerons «Doxyfile\_public»<sup>5</sup>. Dans ce fichier, recopiez le contenu du Doxyfile précédent, et modifiez quelques options, par exemple :

- ▶ **PROJECT\_NAME** = Dames Software Development Kit : le nom de votre projet pour les personnes extérieures
- ▶ **EXTRACT\_PRIVATE** = NO : les membres privés des classes ne seront pas inclus dans la documentation
- ▶ **HTML\_OUTPUT** = html\_public : nom du dossier qui sera ajouté au chemin **OUTPUT\_DIRECTORY**, ainsi, la documentation n'écrase pas la documentation complète placée dans html par le Doxyfile précédent.
- ▶ **FULL\_PATH\_NAMES** = NO : il n'est pas nécessaire de connaître le schéma complet dans cette Documentation.

<sup>5</sup> Un Doxyfile\_public est proposé en [annexe 2](#).

- ▶ **HIDE\_UNDOC\_MEMBERS = YES** : si un membre n'est pas documenté, il n'a pas de raison d'apparaître dans cette Documentation.
- ▶ **HIDE\_UNDOC\_CLASSES = YES** : idem.
- ▶ **SHOW\_FILES = NO** : enlève l'onglet Files de la Documentation.
- ▶ **SOURCE\_BROWSER = YES** : on ne veut surtout pas voir le code source dans cette Documentation.

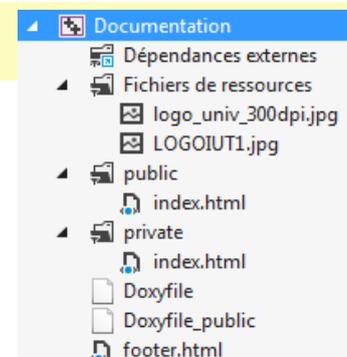
Il faut ensuite modifier les options du projet Documentation.vcproj. Recommencez les opérations de la [partie sur les Propriétés](#) du projet Documentation de la section sur [l'intégration dans Visual Studio](#). Dans les lignes «Build Command Line» et «Rebuild Command Line», écrivez les lignes suivantes<sup>6</sup> :

```
cd ../../Documentation/Doxygen
doxygen Doxyfile
doxygen Doxyfile_public
```

Dans la ligne «Clean Command Line» écrivez les lignes suivantes<sup>7</sup> :

```
rmdir /S /Q html
rmdir /S /Q html_public
```

Vous pouvez pour conclure, organiser vos fichiers comme sur l'image ci-contre sous VisualStudio2012.



<sup>6</sup> si vous n'avez pas choisi la même arborescence que celle conseillée ou les mêmes noms de fichiers Doxyfile, vous devrez adapter ces lignes en conséquence.

<sup>7</sup> si vous n'avez pas choisi la même arborescence que celle conseillée ou les mêmes noms de dossiers de sortie, vous devrez adapter ces lignes en conséquence.

## **Annexe 1 : Exemple de Doxyfile**

Vous trouverez un exemple de Doxyfile en suivant ce [lien](#).

## **Annexe 2 : Exemple de Doxyfile publique**

Vous trouverez un exemple de Doxyfile publique en suivant ce [lien](#).