

Doxygen

Réalisation d'une documentation à partir de
sources commentées

GAUDE Emmanuel
GUILLON Benjamin

Jeudi 1er octobre 2009
Université Claude Bernard Lyon I

Plan

I. Bases

- ▶ Concept
- ▶ Installation

II. Première approche

- ▶ Syntaxe des commentaires
- ▶ Fichier de configuration
- ▶ Génération de la documentation

III. Autres possibilités

- ▶ Le GUI Doxywizard
- ▶ Structuration et mise en forme
- ▶ Fonctionnalités graphiques avancées

I a - Concept

- ▶ **Générateur de documentation** pour divers langages de programmation (C,C++,Java,PHP...)
- ▶ **Multi-plateformes** : Unix-like (dont Mac OS X), et Windows
- ▶ À l'origine, créé en partie sur une vieille version de *DOC++*
- ▶ Depuis entièrement réécrit par **Dimitri van Heesch**
- ▶ Comme pour *Javadoc*, la documentation est générée à partir de commentaires dans le code, ainsi :
 - ▶ **Maintenabilité** aisée de la documentation
 - ▶ **Renvois** de la documentation vers le code
- ▶ Doxygen est une application **gratuite et open-source** sous licence *GNU General Public License*

I b - Installation

- ▶ La dernière version stable à ce jour est la **1.6.1**.
- ▶ Pour la récupérer :
 - ▶ Sous un système Unix-like à **gestionnaire de paquets**
Doxygen est généralement déjà présent dans les dépôts
 - ❑ *aptitude install doxygen (sous Debian ou Ubuntu)*
 - ▶ Pour les autres systèmes rendez-vous à cette adresse
<http://www.stack.nl/~dimitri/doxygen/download.html#latestsrc>
 - ❑ **Mac OS X** : récupérez l'image .dmg et déployez la.
 - ❑ **Autres Unix-like** : récupérez les binaires ou compilez les sources .
 - ❑ **Windows** : utilisez l'assistant d'installation.

II a - Syntaxe - Forme des commentaires

- ▶ Commentaires Doxygen sous forme de blocs

- ▶ Façon Javadoc

- ```
/**
```

- ```
 * Commentaire de documentation
```

- ```
 */
```

- ▶ Façon QT

- ```
/*!
```

- ```
 * Commentaire de documentation
```

- ```
 */
```

II a - Syntaxe - Forme des commentaires

- ▶ Ou en utilisant les commentaires C++ sur une ligne
 - ▶ `///`
`/// Commentaire de documentation`
`///`
 - ▶ `//!`
`//! Commentaire de documentation`
`//!`

II a - Syntaxe - Forme des commentaires

► Remarques

- Les * en début de ligne, entre la première et la dernière ligne sont facultatives

/**

Commentaire de documentation

*/

- Il est possible de "décorer" un commentaire pour le rendre davantage lisible

/**
*****//**

* Commentaire de documentation

*****/

////////////////////////////////////

/// Commentaire de documentation

////////////////////////////////////

II a - Syntaxe - Position des commentaires

- ▶ Un commentaire Doxygen est généralement placé juste avant le code auquel il se rapporte.
/! **brief** Constructeur par défaut. */*
ClassA();
- ▶ Pour les membres (de classes, structures, unions...) et les paramètres il est possible de commenter immédiatement à droite.

```
enum MonEnum {  
    VAL1 = 1, /*!< Première valeur de l'énum (1) */  
    VAL2 = 10, /*!< Deuxième valeur de l'énum (10) */  
    VAL3 = 100 /*!< Troisième valeur de l'énum (100) */  
};
```


II a - Syntaxe - Balises de documentation

- ▶ Sections d'un commentaire distinguées par des balises.
- ▶ Deux notations possible :
 - ▶ **\balise** (*façon Doxygen*)
 - ▶ **@balise** (*façon JavaDoc*)
- ▶
- ▶ Chaque balise a une **signification** bien particulière et sera **exploitée** dans la documentation générée.
- ▶
- ▶ Plusieurs balises par commentaire...
- ▶ ... mais une seule balise par ligne.
- ▶ Une même balise peut-être répétée (selon le contexte).

II a - Syntaxe - Balises de documentation

Liste non exhaustive de balises

- ▶ Documentation d'un élément n'étant pas à proximité
 - ▶ `\struct`, `\union`, `\enum`, `\fn`, `\var`, `\def`, `\typedef`, `\file`, `\namespace`, `\package`, `\interface`

`/*!
 * iclass ClassA
 *
 * ...
 */`

`...
class ClassA {
 ...`

II a - Syntaxe - Balises de documentation

Liste non exhaustive de balises

- ▶ ***\brief*** {texte}
 - ▶ fourni un descriptif de l'élément documenté
 - ▶ première occurrence = ligne de résumé
- * ***\brief*** Classe d'exemple A
- * ***\brief*** Voici un petit bout de code qui illustre l'utilisation de
- * Doxygen au travers d'un code de classe C++.

II a - Syntaxe - Balises de documentation

Liste non exhaustive de balises

- ▶ ***\author*** {*texte*}
- ▶ nom d'un auteur du document
- ▶ ***\since*** <*date*>
- ▶ date de création
- ▶ ***\version*** <*date* / *n° version*>
- ▶ date de dernière modification ou n° de version
- ▶ ***\note*** {*texte*}
- ▶ informations diverses
- ▶ ***\see*** {*texte*}
- ▶ renvoi vers un autre document

II a - Syntaxe - Balises de documentation

Liste non exhaustive de balises

```
/*!  
* author Emmanuel GAUDE <emmanuel.gaude@etu.univ-lyon1.fr>  
* author Benjamin GUILLON <benjamin.guillon@etu.univ-lyon1.fr>  
* since 2009/09/26  
* version 2009/10/03  
*  
* see http://www.stack.nl/~dimitri/doxygen/manual.html  
* note Vous mettez ici ce que vous voulez, comme une licence.  
*/
```

II a - Syntaxe - Balises de documentation

Liste non exhaustive de balises

- ▶ ***mainpage*** {texte}
 - ▶ Le texte suivant dans le bloc sera affiché sur la première page de la documentation

```
/*! mainpage Ma page d'introduction perso  
*  
*  
* Ceci est l'introduction du document.  
*   
*/
```

II a - Syntaxe - Balises de documentation

Liste non exhaustive de balises

- ▶ ***\pre*** {*texte*}
 - ▶ Précondition
- ▶ ***\post*** {*texte*}
 - ▶ Postrelation
- ▶ ***\param***[[*in*]][[*out*]][[*in,out*]] <*nom-param*> {*description*}
 - ▶ Informations relatives à un paramètre

II a - Syntaxe - Balises de documentation

Liste non exhaustive de balises

```
/*!  
 * \pre monNom non vide.  
 * \post Construit un objet de type ClassB.  
 *  
 * \param [in] monNom Nom de l'instance.  
 */  
ClassB(const string monNom);
```


II a - Syntaxe - Balises de documentation

Liste non exhaustive de balises

- ▶ ***\return*** *{description}*
 - ▶ Informations sur le retour
- ▶ ***\retval*** *<valeur> {description}*
 - ▶ Informations quand à la valeur de retour
- ▶ ***\throw*** *<TypeException> {description}*
 - ▶ Informations quant à une exception pouvant être levée

II a - Syntaxe - Balises de documentation

Liste non exhaustive de balises

```
/*!  
* return Retourne systématiquement NULL.  
* retval NULL Valeur normale de retour.  
*  
* throw Exception Retourne une exception si appelée une seconde  
fois.  
*/  
double maMethode();
```

II a - Syntaxe - Balises de documentation

Liste non exhaustive de balises

- ▶ ***\deprecated*** {description}
 - ▶ L'élément est déprécié : présent par compatibilité mais à ne plus utiliser.
- ▶ ***\bug*** {description}
 - ▶ L'élément est bugué.
- ▶ ***\todo*** {description}
 - ▶ L'élément n'est pas encore (entièrement) réalisé.
- ▶ ***\warning*** {description}
 - ▶ Avertissement quant à l'utilisation de cet élément.

II a - Syntaxe - Balises de documentation

Liste non exhaustive de balises

```
/*!  
 * \todo Ecrire un code fonctionnel.  
 * \warning La méthode de la classe A n'est pas la seule de ce nom  
 finalement !  
 */  
double maMethode();
```

```
/*!  
 * \deprecated Ne pas l'utiliser sinon plantage assuré...  
 * \bug BSOD  
 */  
MaStruct * maMethode();
```

II a - Syntaxe - Balises de documentation

Liste complète des balises

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

II b - Fichier Doxyfile

- ▶ Fichier de configuration d'un projet Doxygen.
- ▶ Donne l'ensemble des paramètres nécessaires à la génération.
- ▶ Possibilité de générer un fichier de base via Doxygen ou d'en générer un entièrement configuré par Doxywizard

doxygen -g [fichier-configuration]

- ▶ Le fichier obtenu informe sur l'utilisation de chaque paramètre via des commentaires.

II b - Fichier Doxyfile

Syntaxe

- ▶ # en début de ligne correspond à un commentaire
- ▶ Le format d'une instruction est
PARAMETRE = valeur [valeur, ...]
- ▶ Pour les listes supportant l'ajout, format possible :
PARAMETRE += value [value, ...]
- ▶ Valeurs contenant des espaces entre double quotes ("")

II b - Fichier Doxyfile

Liste non exhaustive de paramètres utiles

- ▶ DOXYFILE_ENCODING = <encodage fichier config>
- ▶ PROJECT_NAME = "Nom du projet"
- ▶ PROJECT_NUMBER = <N° de révision>
- ▶ OUTPUT_DIRECTORY = <Répertoire de sortie de la doc>
- ▶ OUTPUT_LANGUAGE = <langage de la doc (ex: French)>

II b - Fichier Doxyfile

Liste non exhaustive de paramètres utiles

- ▶ Options permettant de faire figurer/masquer du contenu non documenté ou absent par défaut.
 - ▶ `EXTRACT_ALL` = <YES | NO>
 - ▶ `EXTRACT_PRIVATE` = <YES | NO>
 - ▶ `EXTRACT_STATIC` = <YES | NO>
 - ▶ `EXTRACT_LOCAL_CLASSES` = <YES | NO>

 - ▶ `HIDE_UNDOC_MEMBERS` = <YES | NO>
 - ▶ `HIDE_UNDOC_CLASSES` = <YES | NO>

II b - Fichier Doxyfile

Liste non exhaustive de paramètres utiles

- ▶ INPUT = [dossier des sources]
- ▶ INPUT_ENCODING = <encodage des sources>
- ▶ RECURSIVE = <YES | NO> (Présence de sous-répertoires ?)

II b - Fichier Doxyfile

Liste non exhaustive de paramètres utiles

- ▶ Type de documents à générer en sortie
 - ▶ `GENERATE_HTML` = <YES | NO>
 - ▶ `GENERATE_LATEX` = <YES | NO>
 - ▶ `GENERATE_MAN` = <YES | NO>
 - ▶ ...
- ▶ + de très nombreux autres paramètres à régler selon la sortie.

II b - Fichier Doxyfile

Liste non exhaustive de paramètres utiles

- ▶ Traitement et évaluation du préprocesseur
 - ▶ `ENABLE_PREPROCESSING` = <YES | NO>
Evaluation du préprocesseur avant documentation
 - ▶ `MACRO_EXPANSION` = <YES | NO>
Evaluation de toutes les macro sans tenir compte de la compilation conditionnelle

II b - Fichier Doxyfile

Liste complète des paramètres

Voir Doxyfile généré par défaut

II c - Génération de la documentation

- ▶ Une fois les sources documentées et le fichier de configuration réalisé il ne reste plus qu'à générer la documentation avec :

doxygen [fichier-configuration]

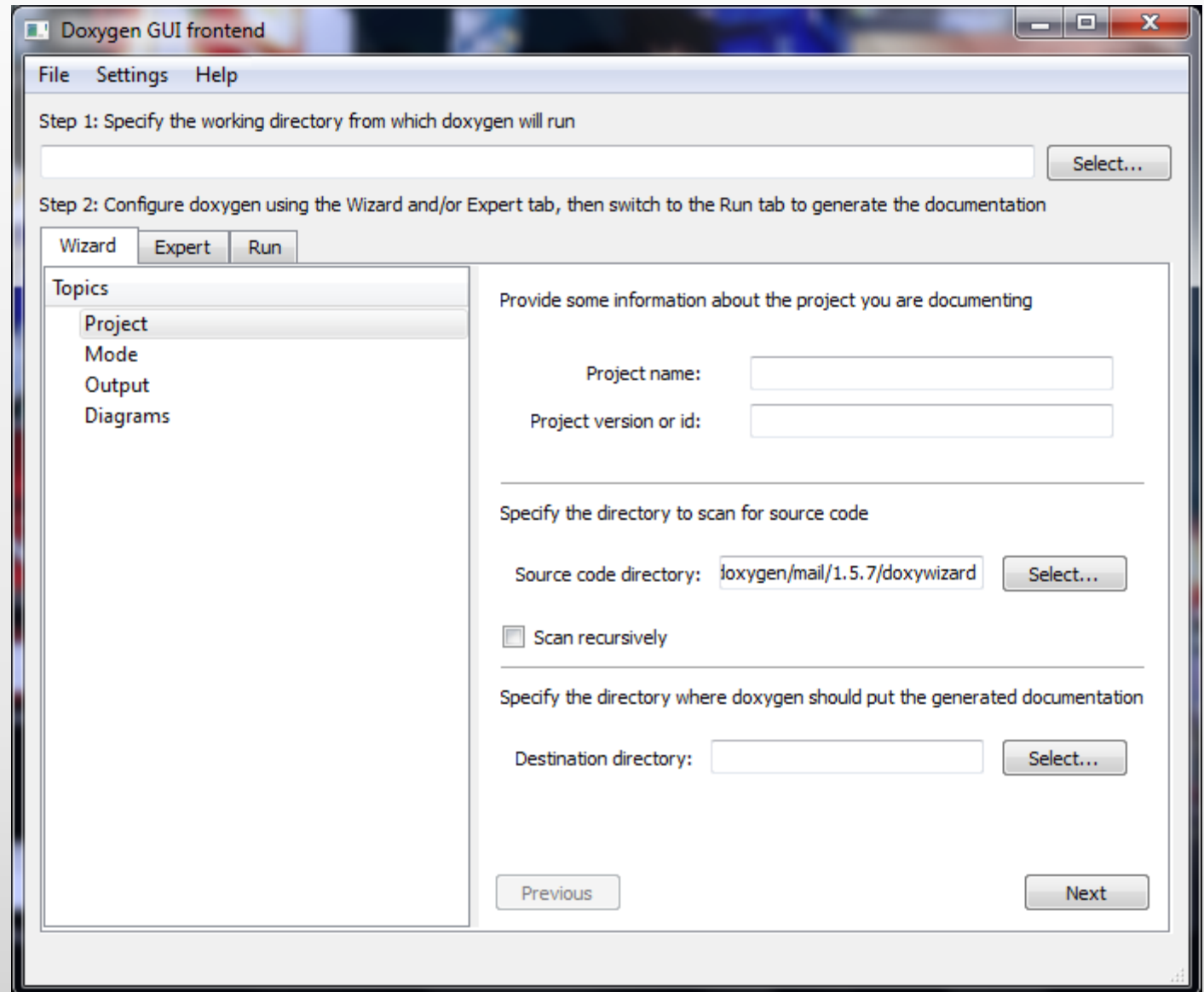
- ▶ Sachant que :
 - ▶ La commande doxygen se trouve dans le PATH
 - ▶ Fichier-configuration par défaut = Doxyfile
 - ▶ Dossier des sources non précisé dans le fichier = dossier courant
- ▶ On obtient alors les différentes documentations attendues (HTML, LaTeX, pages man, rtf, xml...)

III a - Doxywizard

- ▶ Assistant graphique pour la réalisation du fichier Doxyfile et la génération.
- ▶ Présent par défaut avec l'installateur Windows,
- ▶ Il peut être nécessaire de le récupérer séparément sous les autres plateformes.
- ▶ Première approche + conviviale.
- ▶ Plus rapide/simple à utiliser sur les systèmes orientés GUI ?

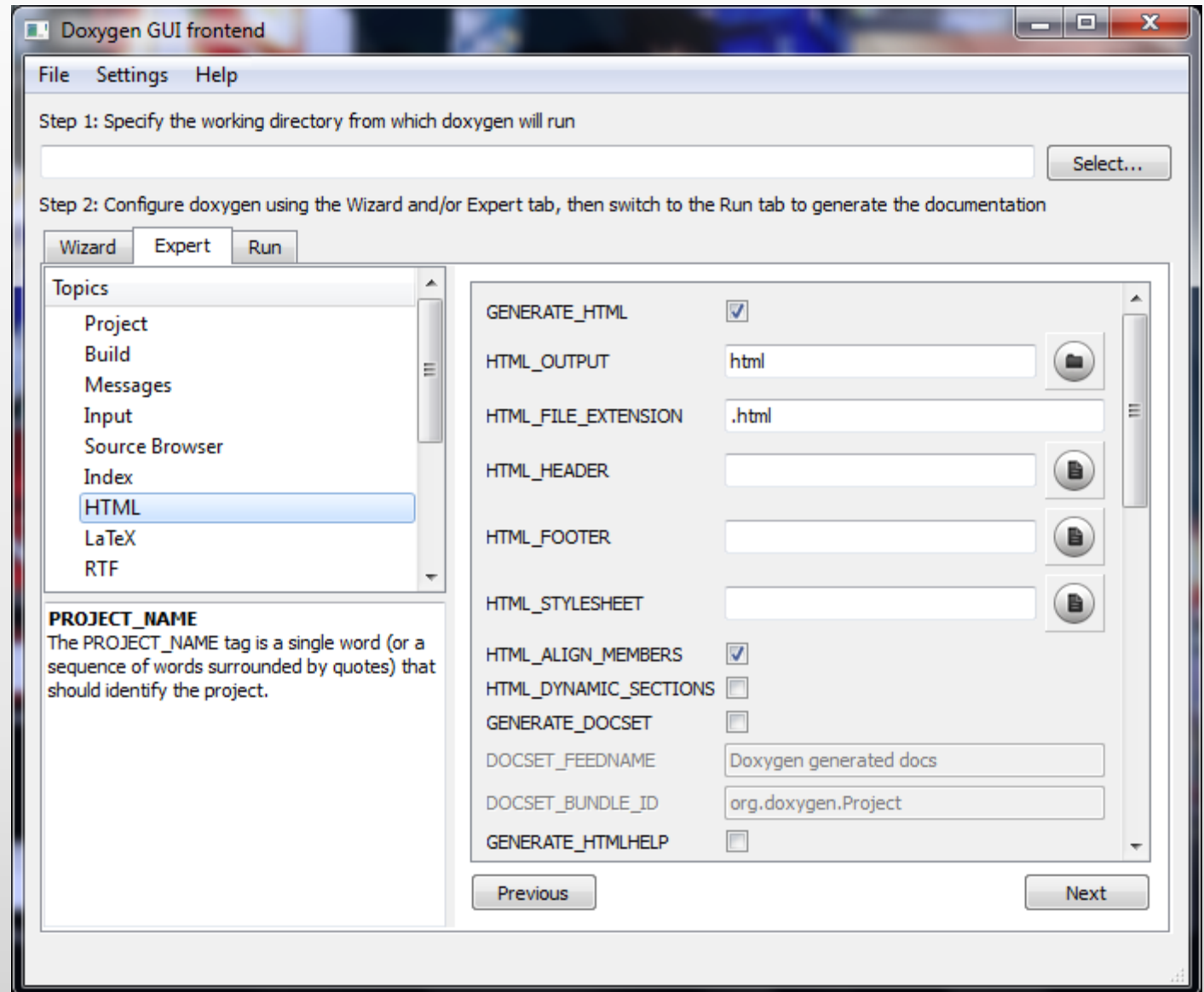
III a - Doxywizard

Mode Assistant



III a - Doxywizard

Mode Expert



III b – Structuration et mise en forme

- ▶ Création de listes
 - ▶ Possibilité de créer des listes via des – et #, ou de code HTML.
- ▶ Groupements
 - ▶ Possibilité de grouper plusieurs éléments de documentation comme des « modules ».
- ▶ Commandes HTML et XML
 - ▶ Possibilité de mettre en forme certaines parties de la documentation dans le cadre d'une sortie en HTML
 - ▶ Possibilité de structurer les commentaires via XML comme en C#

III c – Fonctionnalités graphiques avancées

- ▶ Utilisation de formules mathématiques
 - ▶ Si *latex*, *dvips* et *gs* sont installés, il est possible d'écrire des formules mathématiques mises en forme via LaTeX dans la documentation.
- ▶ Génération de graphiques
 - ▶ Si l'outil *dot* de GraphViz est installé, Doxygen est capable de générer des diagrammes présentant les dépendances entre fichiers, classes...

