Outil de génération de documentation : Doxygen

Introduction

L'idée

- La pérennité d'un projet de développement dépend de la qualité de sa documentation !
- **Doxygen** :
  - extraire l'information du code source pour produire une documentation
  - créer des documentations pour C, C++, Java, ...
  - générer des différentes outputs : html, latex,, ...

Bloc de documentation

Pour langage C, deux combinaisons sont possibles pour créer des blocs de documentation.

Style C avec deux *

```
/**
*  ... documentation ...
*/
```

Style C avec un !

```
/*!
*  ... documentation ...
*/
```

Documenter un fichier

Format

```
/** \file    Nom du fichier
 *  \author  Votre nom
 *  \version Numéro de version du programme
 *
 *  \brief   Description du fichier
 *
 */
```

Documenter un fichier

Exemple

```
/**
* \file main.c
* \author Franck Dupont
* \version 0.1
* \brief Programme de test
* \date 11 septembre 2007
* \details Ceci est un petit programme de test.
*/

#include <stdio.h>
...
```

Documenter une fonction

Format

```
/** \fn       En-tête de la fonction
 *  \author   Votre nom
 *  \version  Numéro de version de la fonction
 *  \date     Date de création/modification
 *
 *  \brief    Description de la fonction
 *
 *  \param    Description du 1er parametre
 *  \param    Description du 2e parametre
 *  ....
 *  \return   Description de l'élément retourné
 *            (absent si fonction de type void)
 */
```

Documenter une fonction

Exemple

```
/**
* \fn int fact (int n)
* \brief Fonction qui renvoie la factorielle d'un entier.
* \param n l'entier en question
* \return sa factorielle, -1 si n négatif
* \details La fonction est récursive et très efficace !
*/
int fact (int n) {
...
```

Documenter une structure

Exemple

```
/**
* \struct Noeud
* \brief Maillon d'une liste chainée.
* \details Un noeud possède deux champs. Le premier
champ...
*/
typedef struct Noeud {
...
```

Documenter une énumération

Exemple

```
/**
* \enum Boolean
* \brief Simulation de l'existence du type booléen.
* \details Comme ne C il n'existe pas de type booléen,
*          nous définissons...
*/
typedef enum Boolean {
...
```

Utilisation (mode console)

- Editer le fichier de configuration *fichier_conf*
- Indiquer les paramètres de configuration :
  - INPUT = les répertoires et/ou les fichiers dont vous souhaitez éditer la documentation (séparés par un espace)
  - FILE_PATTERNS = un filtre pour les fichiers pris en compte
  - RECURSIVE = chercher aussi dans les sous-répertoires
  - ...
- Exécuter la commande `doxygen fichier_conf`

Utilisation (mode console)

Premier exemple

```
#---------------------------------
# Doxygen configuration
#---------------------------------
INPUT = main.c bib.c bib.h
```

Deuxième exemple

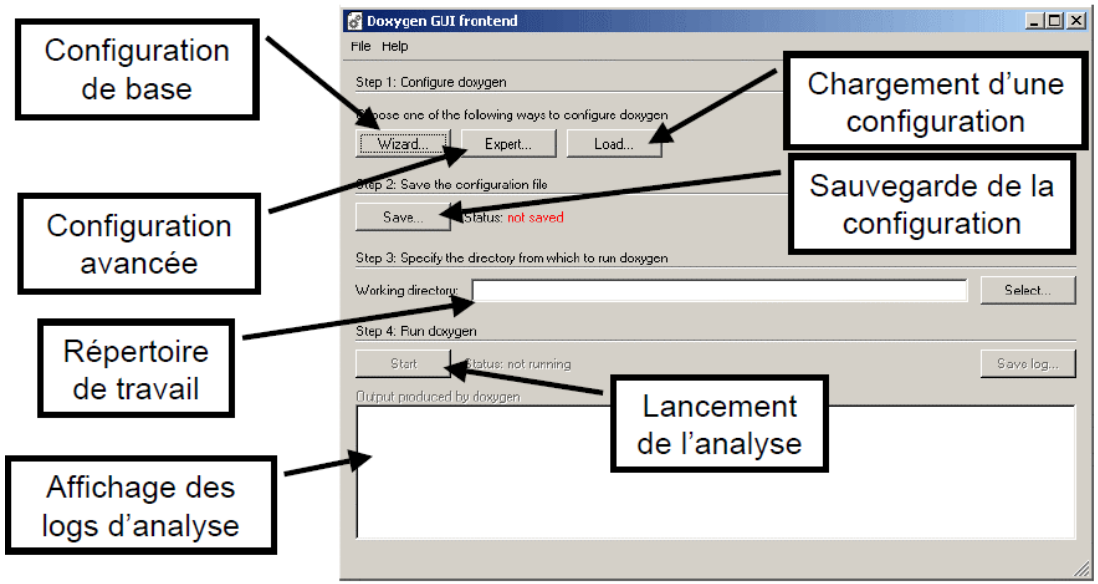
```
#---------------------------------
# Doxygen configuration
#---------------------------------
INPUT = .
FILE_PATTERNS = *.c *.h
RECURSIVE = YES
```

Utilisation (mode graphique)

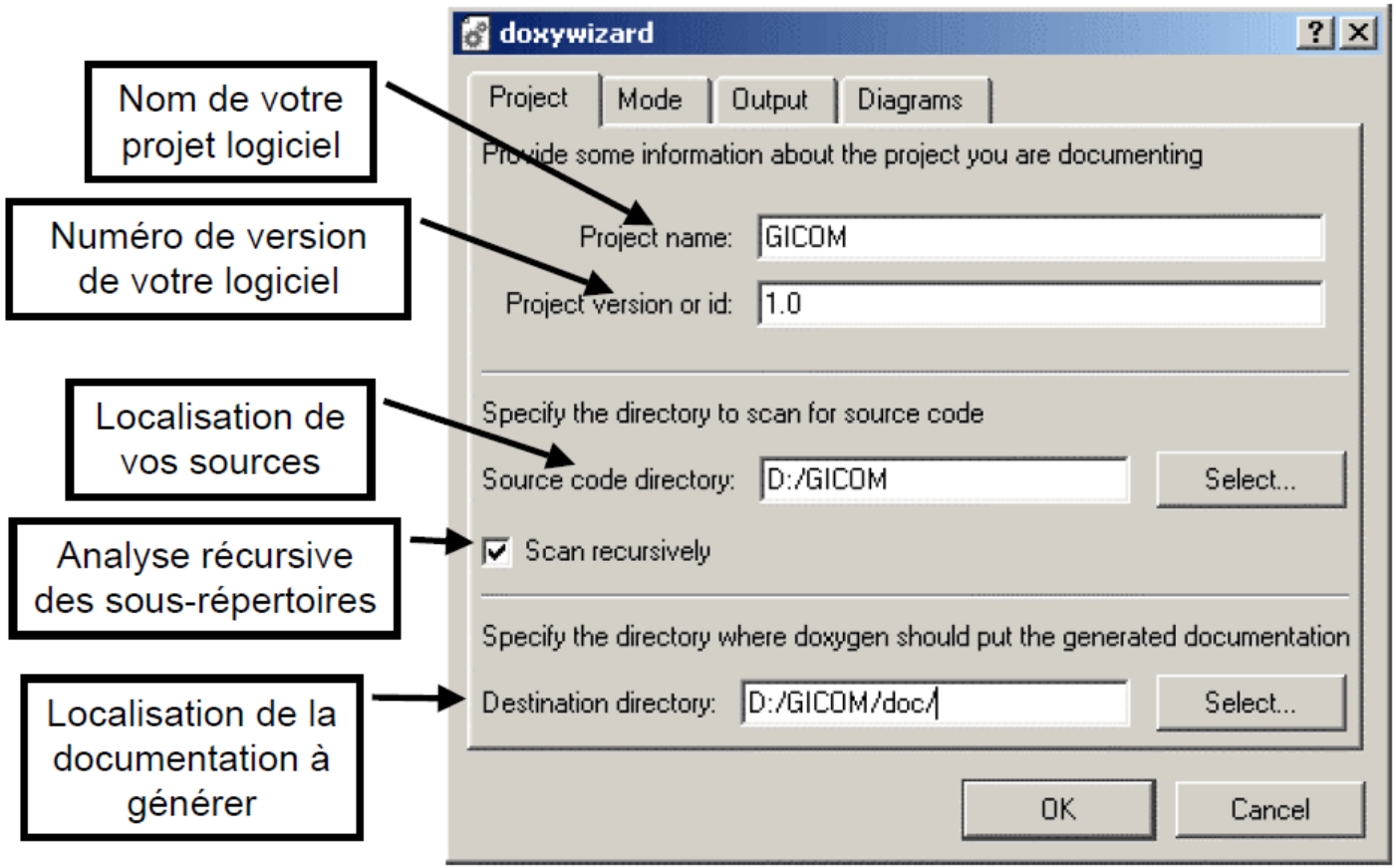
Doxygen GUI

Une interface graphique permet de configurer aisément Doxygen et de lancer l'analyse de vos sources pour en produire une documentation. Il existe deux modes de configuration :

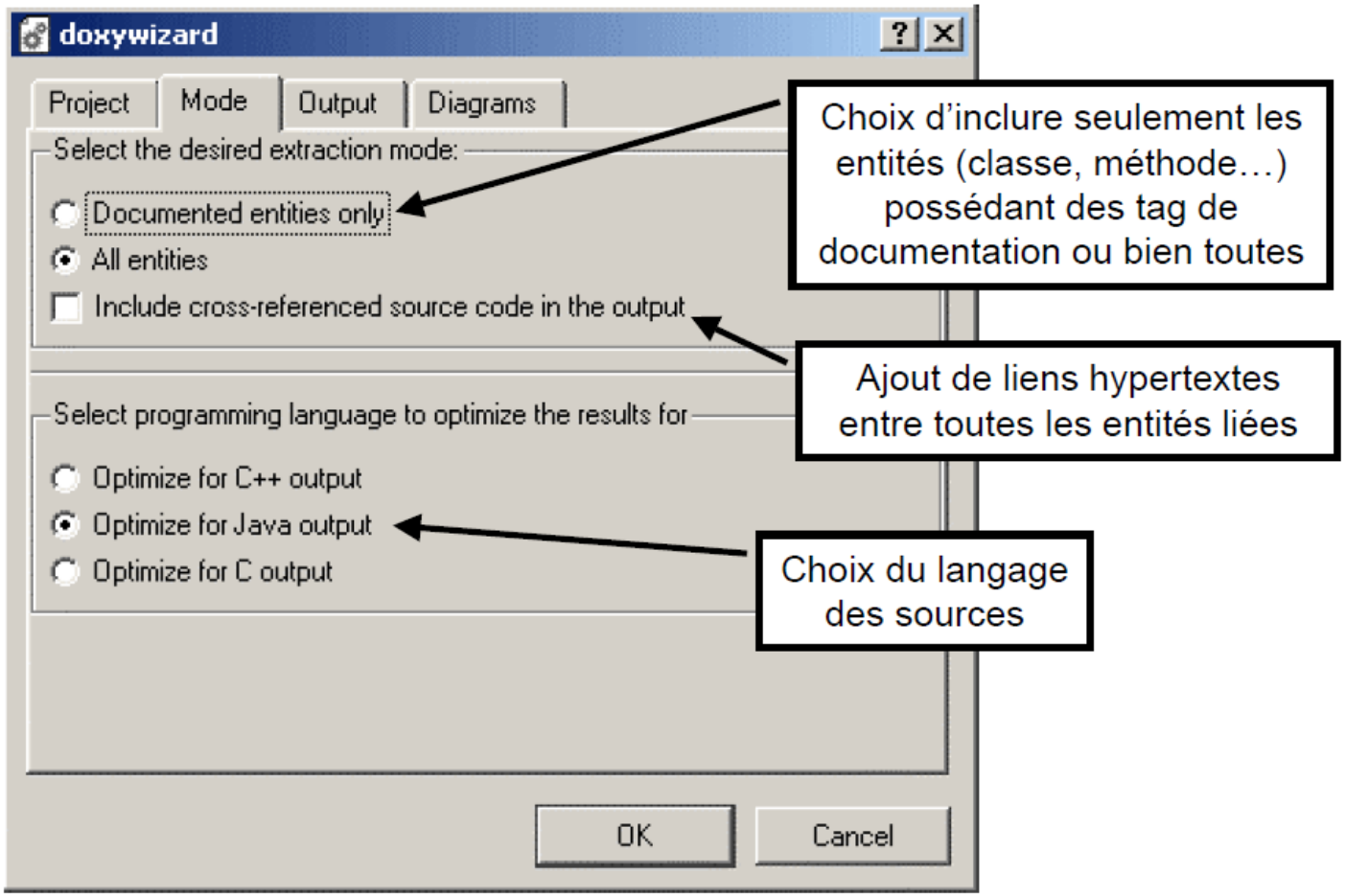
- Wizard : configuration de base (doxywizard)
- Expert : configuration avancée pour les experts



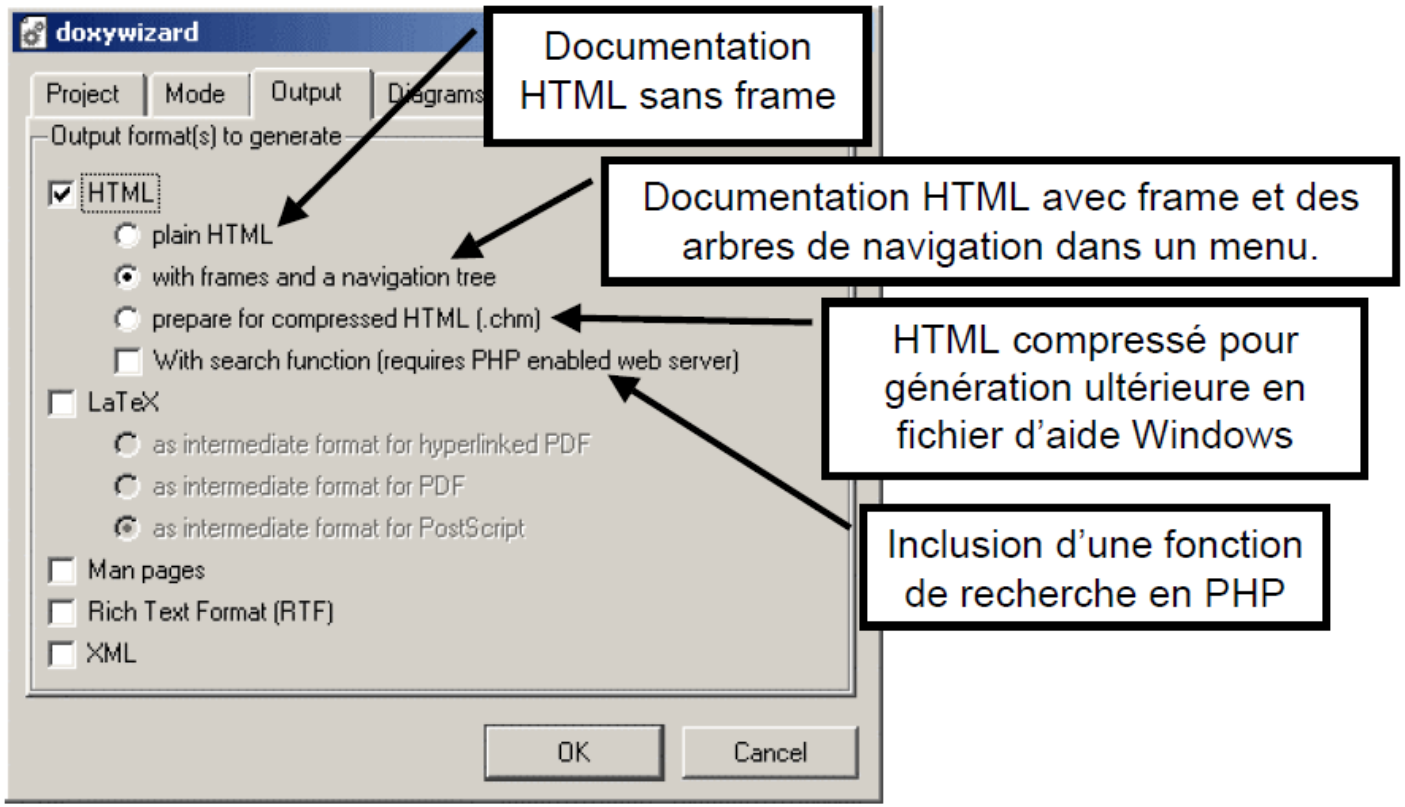
Doxywizard


Nom de votre projet logiciel
Numéro de version de votre logiciel
Localisation de vos sources
Analyse récursive des sous-répertoires
Localisation de la documentation à générer
doxywizard
Project
Mode
Output
Diagrams
Provide some information about the project you are documenting
Project name:
GICOM
Project version or id:
1.0
Specify the directory to scan for source code
Source code directory:
D:/GICOM
Select...
Scan recursively
Specify the directory where doxygen should put the generated documentation
Destination directory:
D:/GICOM/doc/
Select...
OK
Cancel

Doxywizard


doxywizard
Project
Mode
Output
Diagrams
Select the desired extraction mode:
Documented entities only
All entities
Include cross-referenced source code in the output
Select programming language to optimize the results for
Optimize for C++ output
Optimize for Java output
Optimize for C output
OK
Cancel
Choix d'inclure seulement les entités (classe, méthode…) possédant des tag de documentation ou bien toutes
Ajout de liens hypertextes entre toutes les entités liées
Choix du langage des sources

Doxywizard


doxywizard
Project
Mode
Output
Diagrams
Output format(s) to generate
HTML
plain HTML
with frames and a navigation tree
prepare for compressed HTML (.chm)
With search function (requires PHP enabled web server)
LaTeX
as intermediate format for hyperlinked PDF
as intermediate format for PDF
as intermediate format for PostScript
Man pages
Rich Text Format (RTF)
XML
OK
Cancel
Documentation HTML sans frame
Documentation HTML avec frame et des arbres de navigation dans un menu.
HTML compressé pour génération ultérieure en fichier d'aide Windows
Inclusion d'une fonction de recherche en PHP