

# Production (automatique) de documentations

F. Langrognnet



(Lm<sup>B</sup>)

laboratoire de mathématiques de besançon  
UNIVERSITÉ DE FRANCHE-COMTÉ • CNRS • UMR 6623



# PLAN

- 1 Introduction
- 2 Comment ça marche ?
  - Sur quels types de fichiers ?
  - Pour quelles sorties ?
- 3 Un exemple détaillé : Doxygen
  - Fiche d'identité
  - Flux de données
  - Comment utiliser Doxygen ?
  - Balises
  - Diagrammes et graphes
  - Personnalisation des sorties
  - Enrichir la documentation
- 4 Pour aller plus loin ...

# PLAN

## 1 Introduction

## 2 Comment ça marche ?

- Sur quels types de fichiers ?
- Pour quelles sorties ?

## 3 Un exemple détaillé : Doxygen

- Fiche d'identité
- Flux de données
- Comment utiliser Doxygen ?
- Balises
- Diagrammes et graphes
- Personnalisation des sorties
- Enrichir la documentation

## 4 Pour aller plus loin ...

# Introduction

## Objectif

Générer **automatiquement** (ou presque...) des documentations

- techniques (pour les développeurs)
- pour les utilisateurs ?

## Historique

1995-1997 : 1<sup>ers</sup> outils de génération :

- **javadoc**, ROBODoc (1995)
- **Doxygen** (1997)

# Quelles documentations ? (1)

**Documentation logicielle** (pour les développeurs)

## javadoc Java SDK

Echier Edition Affichage Ajout Signets Configuration Fenêtre Aide

Java™ 2 Platform Std. Ed. v1.4.2

Overview Package Class Use Tree Deprecated Index Help

Java™ 2 Platform Std. Ed. v1.4.2

### Java™ 2 Platform, Standard Edition, v 1.4.2 API Specification

This document is the API specification for the Java 2 Platform, Standard Edition, version 1.4.2.

See: **Description**

#### Java 2 Platform Packages

Package	Description
<a href="#">java.applet</a>	Provides the classes necessary to create applet and the classes an applet uses to communicate with its applet context.
<a href="#">java.awt</a>	Contains all of the classes for creating UI interfaces and for painting graphics and images.
<a href="#">java.awt.color</a>	Provides classes for color spaces.
<a href="#">java.awt.datatransfer</a>	Provides interfaces and classes for transferring data between and within applications.
<a href="#">java.awt.dnd</a>	Drag and Drop is a direct manipulation paradigm found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in GUI.
<a href="#">java.awt.event</a>	Provides interfaces and classes for dealing with different types of events fired by AWT components.
<a href="#">java.awt.font</a>	Provides classes and interface relating to text.
<a href="#">java.awt.geom</a>	Provides the Java 2D classes for defining, performing operations on objects related to two-dimensional geometry.
<a href="#">java.awt.im</a>	Provides classes and interfaces for the input method framework.
<a href="#">java.awt.im.spi</a>	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
<a href="#">java.awt.image</a>	Provides classes for creating and modifying images.
<a href="#">java.awt.image.renderable</a>	Provides classes and interfaces for producing rendering-independent images.
<a href="#">java.awt.print</a>	Provides classes and interfaces for a general printing API.

Echier Edition Affichage Ajout Signets Configuration Fenêtre Aide

Java™ 2 Platform Std. Ed. v1.4.2

Overview Package Class Use Tree Deprecated Index Help

Java™ 2 Platform Std. Ed. v1.4.2

### Java™ 2 Platform, Standard Edition, v 1.4.2 API Specification

This document is the API specification for the Java 2 Platform, Standard Edition, version 1.4.2.

See: **Description**

#### Java 2 Platform Packages

Package	Description
<a href="#">java.applet</a>	Provides the classes necessary to create applet and the classes an applet uses to communicate with its applet context.
<a href="#">java.awt</a>	Contains all of the classes for creating UI interfaces and for painting graphics and images.
<a href="#">java.awt.color</a>	Provides classes for color spaces.
<a href="#">java.awt.datatransfer</a>	Provides interfaces and classes for transferring data between and within applications.
<a href="#">java.awt.dnd</a>	Drag and Drop is a direct manipulation paradigm found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in GUI.
<a href="#">java.awt.event</a>	Provides interfaces and classes for dealing with different types of events fired by AWT components.
<a href="#">java.awt.font</a>	Provides classes and interface relating to text.
<a href="#">java.awt.geom</a>	Provides the Java 2D classes for defining, performing operations on objects related to two-dimensional geometry.
<a href="#">java.awt.im</a>	Provides classes and interfaces for the input method framework.
<a href="#">java.awt.im.spi</a>	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
<a href="#">java.awt.image</a>	Provides classes for creating and modifying images.
<a href="#">java.awt.image.renderable</a>	Provides classes and interfaces for producing rendering-independent images.
<a href="#">java.awt.print</a>	Provides classes and interfaces for a general printing API.

## javadoc NetBeans

Fichier Edition Affichage Aller Signets Configuration Fenêtre Aide

Overview AllClasses

Actions APIs ( javadoc Art ( javadoc ) Art-Based Project Ss, Auto Locate Service, Bootstrap ( javadoc ), Classfile Reader ( Jls ), Classpath APIs ( Javn Command Line Param Common Annotation Common Annotation Common Palette ( Jls ) Common Scanner ( Jls ) Compatibility API ( Jls ) Core - IDE ( javadoc ), Database Explorer ( J )

**NetBeans API List**

Current Development Version

26 Aug 2010

This document provides a list of *NetBeans APIs* with a short description of what they are used for, and a table describing different types of interfaces (see **What is an API?** to understand why we list DTDs, file formats, etc.) and with a stability category (stable and official, **under development**, **deprecated**, **internal or private**; see **API life-cycle** for more info). The aim is to provide as detailed a definition of *NetBeans module external interfaces* as possible and give other developers a chance to decide whether they want to depend on a particular API or not.

Some of these APIs are part of the **NetBeans Platform** (for example lookup, loaders, utilities, nodes, explore, window systems, multiview, etc.), some of them are specific to **NetBeans IDE** (projects, javadoc, diff, etc.) and some are **not included** in the release at all and are just provided for download (usually via **update4j**). Basically when building an application based on *NetBeans* one is free to choose the set of modules and their APIs to satisfy one's needs.

This is a list of APIs for development version, if you want to see a list of APIs for a particular version, you may want to go to:

- 6.9.1 - Javadoc as released for NetBeans IDE 6.9.1
- 6.9 - Javadoc as released for NetBeans IDE 6.9
- 6.8 - Javadoc as released for NetBeans IDE 6.8
- 6.7 - Javadoc as released for NetBeans IDE 6.7
- 6.5 - Javadoc as released for NetBeans IDE 6.5
- 6.1 - Javadoc as released for NetBeans IDE 6.1
- 6.0 - Javadoc as released for NetBeans IDE 6.0
- 5.5.1 - Javadoc as released for NetBeans IDE 5.5.1
- 5.5 - Javadoc as released for NetBeans IDE 5.5
- 5.0 - Javadoc as released for NetBeans IDE 5.0

To get the API of your module listed here, see the documentation for the Javadoc building infrastructure.

Additional Sources of Information

- Changes since previous release
- How to use certain NetBeans APIs
- Index of all NetBeans API classes
- Extracted List of Layer APIs
- Extracted List of Property APIs

FAQ and Mailing List

Can't find what you're looking for? Try the FAQ and then [dev.openide.netbeans.org](http://dev.openide.netbeans.org).

Content

- Actions APIs** - Actions provides system of support and utility classes for 'actions' usage in NetBeans.
- Ant** - The Ant integration module recognizes Ant build scripts, facilitates editing them as text or structurally, lets the user run targets or create shortcuts to them, etc.
- Ant-Based Project Support** - Provides the basic infrastructure by which Ant-based projects can be created, read and write configuration parameters and properties from dist, satisfy common queries and interfaces, etc.
- Auto Update Services - org.netbeans.api.update4j.org.netbeans.api.update4j** .
- Bootstrap** - The NetBeans launcher scans the application.
- Classfile Reader** - The Classfile API is designed as a fast, low-memory-cost way to access classfile information.
- Classpath APIs** - Models basic aspects of the metadata surrounding list of source roots, such as the classpath.
- Command Line Parsing API** - Gets/sets file infrastructure to parse command line arguments with the cooperative participation of various modules.
- Common Annotations** - Provides common annotations serving as a documentation element

Fichier Edition Affichage Aller Signets Configuration Fenêtre Aide

Overview AllClasses

Actions APIs ( javadoc Art ( javadoc ) Art-Based Project Ss, Auto Locate Service, Bootstrap ( javadoc ), Classfile Reader ( Jls ), Classpath APIs ( Javn Command Line Param Common Annotation Common Annotation Common Palette ( Jls ) Common Scanner ( Jls ) Compatibility API ( Jls ) Core - IDE ( javadoc ), Database Explorer ( J )

**NetBeans API List**

Current Development Version

26 Aug 2010

This document provides a list of *NetBeans APIs* with a short description of what they are used for, and a table describing different types of interfaces (see **What is an API?** to understand why we list DTDs, file formats, etc.) and with a stability category (stable and official, **under development**, **deprecated**, **internal or private**; see **API life-cycle** for more info). The aim is to provide as detailed a definition of *NetBeans module external interfaces* as possible and give other developers a chance to decide whether they want to depend on a particular API or not.

Some of these APIs are part of the **NetBeans Platform** (for example lookup, loaders, utilities, nodes, explore, window systems, multiview, etc.), some of them are specific to **NetBeans IDE** (projects, javadoc, diff, etc.) and some are **not included** in the release at all and are just provided for download (usually via **update4j**). Basically when building an application based on *NetBeans* one is free to choose the set of modules and their APIs to satisfy one's needs.

This is a list of APIs for development version, if you want to see a list of APIs for a particular version, you may want to go to:

- 6.9.1 - Javadoc as released for NetBeans IDE 6.9.1
- 6.9 - Javadoc as released for NetBeans IDE 6.9
- 6.8 - Javadoc as released for NetBeans IDE 6.8
- 6.7 - Javadoc as released for NetBeans IDE 6.7
- 6.5 - Javadoc as released for NetBeans IDE 6.5
- 6.1 - Javadoc as released for NetBeans IDE 6.1
- 6.0 - Javadoc as released for NetBeans IDE 6.0
- 5.5.1 - Javadoc as released for NetBeans IDE 5.5.1
- 5.5 - Javadoc as released for NetBeans IDE 5.5
- 5.0 - Javadoc as released for NetBeans IDE 5.0

To get the API of your module listed here, see the documentation for the Javadoc building infrastructure.

Additional Sources of Information

- Changes since previous release
- How to use certain NetBeans APIs
- Index of all NetBeans API classes
- Extracted List of Layer APIs
- Extracted List of Property APIs

FAQ and Mailing List

Can't find what you're looking for? Try the FAQ and then [dev.openide.netbeans.org](http://dev.openide.netbeans.org).

Content

- Actions APIs** - Actions provides system of support and utility classes for 'actions' usage in NetBeans.
- Ant** - The Ant integration module recognizes Ant build scripts, facilitates editing them as text or structurally, lets the user run targets or create shortcuts to them, etc.
- Ant-Based Project Support** - Provides the basic infrastructure by which Ant-based projects can be created, read and write configuration parameters and properties from dist, satisfy common queries and interfaces, etc.
- Auto Update Services - org.netbeans.api.update4j.org.netbeans.api.update4j** .
- Bootstrap** - The NetBeans launcher scans the application.
- Classfile Reader** - The Classfile API is designed as a fast, low-memory-cost way to access classfile information.
- Classpath APIs** - Models basic aspects of the metadata surrounding list of source roots, such as the classpath.
- Command Line Parsing API** - Gets/sets file infrastructure to parse command line arguments with the cooperative participation of various modules.
- Common Annotations** - Provides common annotations serving as a documentation element

Doxygen  
KDE

Etcher Edition Affichage Ajout Signets Configuration Fenêtre Aide

KDE 3.5 API Reference

KDE API Reference API Reference Sitemap Contact Us

## KMainWindowInterface Class Reference

DCOP interface to KMainWindow. More...

```
#include <kmainwindowinterface.h>
```

Inheritance diagram for KMainWindowInterface:

```

graph TD
    DCOPObj[DCOP Object] --> KMainWindowInterface[KMainWindowInterface]
    KMainWindowInterface --- Legend[Legend]
  
```

List of all members.

### Public Member Functions

```

QStringList functionDynamic ()
QStringList KMainWindowInterface(KMainWindow *mainWindow)
bool processDynamic(const QString &fun, const QByteArray &data, QStringList &argv, QStringList &argvData)
~KMainWindowInterface ()
  
```

### Detailed Description

DCOP interface to KMainWindow.

This is the main interface to the KMainWindow. This will provide a consistent DCOP interface to all KDE applications that use it.

**Author:**  
Ian Reinhart Geiser <geiser@yahoo.com>

Definition at line 40 of file kmainwindowinterface.h.

### Constructor & Destructor Documentation

```

KMainWindowInterface(KMainWindowInterface *KMainWindow *mainWindow)
  
```

Construct a new interface object.

**Parameters:**  
mainWindow - The parent KMainWindow object that will provide us with the KAction objects.

Definition at line 31 of file kmainwindowinterface.cpp.

```

KMainWindowInterface::~KMainWindowInterface()
  
```

Destroy the interface object.

Etcher Edition Affichage Ajout Signets Configuration Fenêtre Aide

KDE 3.5 API Reference

KDE API Reference API Reference Sitemap Contact Us

## KHistoryCombo Class Reference

A combobox for offering a history and completion. More...

```
#include <kcombobox.h>
```

Inheritance diagram for KHistoryCombo:

```

graph TD
    QComboBox[QComboBox] --> KHistoryCombo[KHistoryCombo]
    KCompletionBase[KCompletionBase] --> KHistoryCombo
    KHistoryCombo --- Legend[Legend]
  
```

List of all members.

### Public Slots

```

void addHistory(const QString &item)
void clearHistory ()
  
```

### Signals

```

void cleared ()
  
```

### Public Member Functions

```

QStringList historyItems () const
KHistoryCombo(const QWidget *parent=0L, const char *name=0L)
KHistoryCombo(QWidget *parent=0L, const char *name=0L)
KHistoryCombo(QWidget *parent=0L, const char *name=0L, QStringList &items)
KHistoryCombo(QWidget *parent=0L, const char *name=0L, QStringList &items, bool setCompletionList)
void addHistory(const QString &item)
void clearHistory ()
void removeFromHistory(const QString &item)
void reset ()
void setHistoryItems(const QStringList &items, bool setCompletionList)
void setHistoryItems(const QStringList &items)
void setItemProvider(KItemProvider *prov)
void setItemProvider(KItemProvider *prov)
~KHistoryCombo ()
  
```

### Protected Member Functions

```

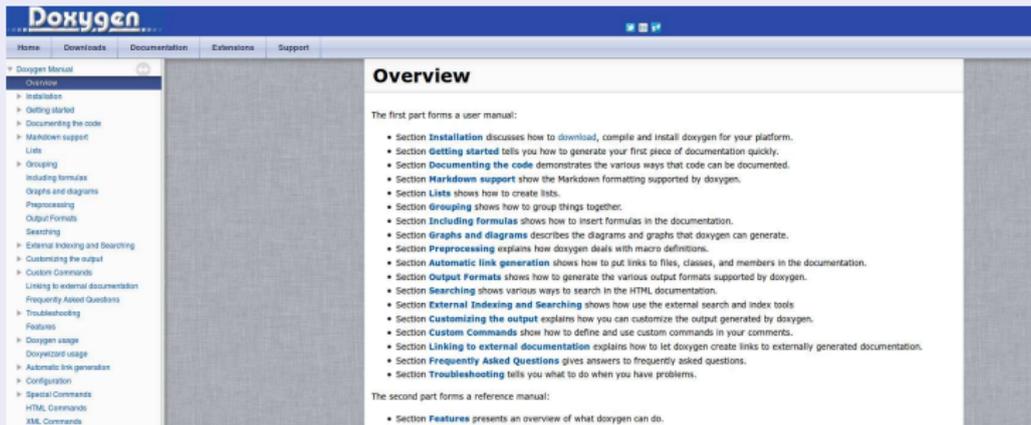
void itemItems(const QStringList &items)
virtual void keyPressEvent(QKeyEvent *)
bool useCompletion () const
  
```

# Quelles documentations ? (2)

## Documentation pour l'utilisateur final

Quelques motivations :

- Documentation intégrée dans le processus de développement  
Le système de gestion de version peut/doit servir aussi pour les documentations !
- Profiter des fonctionnalités de l'outil de production de documentation
  - ▶ Formats de sortie
  - ▶ Mise en page
  - ▶ Description des algorithmes, des formules



The screenshot shows the Doxygen website interface. The top navigation bar includes 'Home', 'Download', 'Documentation', 'Extensions', and 'Support'. The left sidebar contains a tree view of the 'Doxygen Manual' with 'Overview' selected. The main content area is titled 'Overview' and contains the following text:

The first part forms a user manual:

- Section **Installation** discusses how to [download](#), compile and install doxygen for your platform.
- Section **Getting started** tells you how to generate your first piece of documentation quickly.
- Section **Documenting the code** demonstrates the various ways that code can be documented.
- Section **Markdown support** show the Markdown formatting supported by doxygen.
- Section **Lists** shows how to create lists.
- Section **Grouping** shows how to group things together.
- Section **Including formulas** shows how to insert formulas in the documentation.
- Section **Graphs and diagrams** describes the diagrams and graphs that doxygen can generate.
- Section **Preprocessing** explains how doxygen deals with macro definitions.
- Section **Automatic link generation** shows how to put links to files, classes, and members in the documentation.
- Section **Output Formats** shows how to generate the various output formats supported by doxygen.
- Section **Searching** shows various ways to search in the HTML documentation.
- Section **External Indexing and Searching** shows how use the external search and index tools
- Section **Customizing the output** explains how you can customize the output generated by doxygen.
- Section **Custom Commands** show how to define and use custom commands in your comments.
- Section **Linking to external documentation** explains how to let doxygen create links to externally generated documentation.
- Section **Frequently Asked Questions** gives answers to frequently asked questions.
- Section **Troubleshooting** tells you what to do when you have problems.

The second part forms a reference manual:

- Section **Features** presents an overview of what doxygen can do.

## Caractéristiques

### Formats d'entrée

- Code source (texte)
- Binaire

### Formats de sortie

- HTML
- PDF
- Latex
- ps
- XML
- man pages
- RTF
- DocBook

### Fonctionnalités

- Description des classes, des fonctions
- Diagrammes (classes, appels, ...)
- Possibilité d'étendre à d'autres langages
- Personnalisation des sorties
- Mise en page, contenu :
  - ▶ Possibilité d'insérer des balises HTML
  - ▶ Possibilité d'insérer du code Latex (formules mathématiques, ...)
  - ▶ Facilité pour la mise en page

# PLAN

- 1 Introduction
- 2 Comment ça marche ?
  - Sur quels types de fichiers ?
  - Pour quelles sorties ?
- 3 Un exemple détaillé : Doxygen
  - Fiche d'identité
  - Flux de données
  - Comment utiliser Doxygen ?
  - Balises
  - Diagrammes et graphes
  - Personnalisation des sorties
  - Enrichir la documentation
- 4 Pour aller plus loin ...

# PLAN

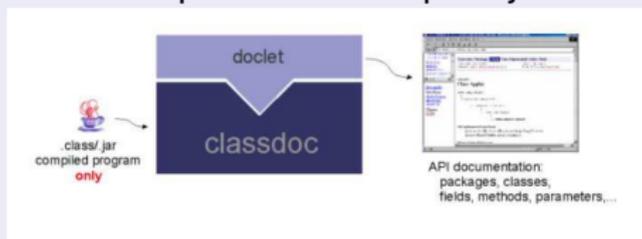
- 1 Introduction
- 2 **Comment ça marche ?**
  - Sur quels types de fichiers ?
  - Pour quelles sorties ?
- 3 Un exemple détaillé : Doxygen
  - Fiche d'identité
  - Flux de données
  - Comment utiliser Doxygen ?
  - Balises
  - Diagrammes et graphes
  - Personnalisation des sorties
  - Enrichir la documentation
- 4 Pour aller plus loin ...

# Sur quels types de fichiers ?

## Fichiers binaires

Documentation générée à partir des **binaires seulement**

Exemple : **classDoc** pour java



## Fichiers sources

Documentation générée **à partir du code source** en utilisant

- La grammaire, les mots clés du langage  
**On peut donc générer des documentations sans travail spécifique**
- Des commentaires, des balises spécifiques  
**Pour enrichir la documentation**

# Documentation produite sans commentaire spécifique ni balise

```
Date.h
Fichier Édition Affichage Aller Signets Sessions Outils Configuration Aide
Nouveau Ouvrir Précédent Suivant Enregistrer
class Date {
private:
    int _jour;
    int _mois;
    int _annee;
public:
    //-----
    // Constructeurs/deconstructeurs
    //-----
    Date();
    Date(int iJour, int iMois, int iAnnee);
    Date(const Date & iDate);
    ~Date();

    //-----
    // Selecteurs
    //-----
    int getJour() const;
    int getMois() const;
    int getAnnee() const;
    void setJour(int iJour);
    void setMois(int iMois);
    void setAnnee(int iAnnee);

    //-----
    // Methodes
    //-----
    void saisir();
    void afficher() const;

    //-----
    // Surcharges d'operateurs
    //-----
    bool operator < (const Date& d) const;
    bool operator > (const Date& d) const;
    bool operator <= (const Date& d) const;
    bool operator >= (const Date& d) const;
    bool operator == (const Date& d) const;
    bool operator != (const Date& d) const;

    friend istream& operator >> (istream& is, Date& d);
    friend ostream& operator << (ostream& os, const Date& d);
};
```

## Documentation générée (html)

Fichier Édition Affichage Aller Signets Configuration Fenêtre Aide

Main Page Classes Files Search

### Date Class Reference

List of all members.

#### Public Member Functions

Date (int iJour, int iMois, int iAnnee)
Date (const Date &iDate)
int getJour () const
int getMois () const
int getAnnee () const
void setJour (int iJour)
void setMois (int iMois)
void setAnnee (int iAnnee)
void saisir ()
void afficher () const
bool operator< (const Date &d) const
bool operator> (const Date &d) const
bool operator<= (const Date &d) const
bool operator>= (const Date &d) const
bool operator== (const Date &d) const
bool operator!= (const Date &d) const

#### Friends

istream & operator>> (istream &is, Date &d)
ostream & operator<< (ostream &os, const Date &d)

The documentation for this class was generated from the following file:

- [date.h](#)

Generated on Thu Aug 26 15:29:20 2010 by doxygen 1.6.3

# PLAN

- 1 Introduction
- 2 **Comment ça marche ?**
  - Sur quels types de fichiers ?
  - **Pour quelles sorties ?**
- 3 Un exemple détaillé : Doxygen
  - Fiche d'identité
  - Flux de données
  - Comment utiliser Doxygen ?
  - Balises
  - Diagrammes et graphes
  - Personnalisation des sorties
  - Enrichir la documentation
- 4 Pour aller plus loin ...

# Pour quelles sorties ?

## Quelles informations ?

- Description des fonctions (paramètres, type de retour, ...)
- Description des classes
- Description des fichiers
- Graphes d'appels
- Diagrammes (de classes, de collaboration, ...)
- Liens vers les fichiers sources
- ...

## Format des fichiers de sortie

- Différents formats (en fonction des possibilités de l'outil) :  
html, pdf, latex, ps, XML, ...
- Personnalisation possible des sorties

# PLAN

## 1 Introduction

## 2 Comment ça marche ?

- Sur quels types de fichiers ?
- Pour quelles sorties ?

## 3 Un exemple détaillé : Doxygen

- Fiche d'identité
- Flux de données
- Comment utiliser Doxygen ?
- Balises
- Diagrammes et graphes
- Personnalisation des sorties
- Enrichir la documentation

## 4 Pour aller plus loin ...

# PLAN

- 1 Introduction
- 2 Comment ça marche ?
  - Sur quels types de fichiers ?
  - Pour quelles sorties ?
- 3 **Un exemple détaillé : Doxygen**
  - **Fiche d'identité**
  - Flux de données
  - Comment utiliser Doxygen ?
  - Balises
  - Diagrammes et graphes
  - Personnalisation des sorties
  - Enrichir la documentation
- 4 Pour aller plus loin ...

# Doxygen - Fiche d'identité

- Auteur : Dimitri Van Heesch
- licence : GNU GPL
- 1<sup>re</sup> version : 1997
- OS : BSD, Linux, Mac OS, Unix, Windows
- Entrée : **code source**
- Sorties : HTML, LATEX, Man Pages, RTF, XML, Qt Help Project, PDF, PS, ...
- Nombreuses possibilités de personnalisation

## Documentation KDE

Doxygen-generated KDE 3.5 API Reference for **KMainWindow Class Reference**. The page includes a class hierarchy diagram showing KMainWindow inheriting from QMainWindow. It lists public member functions such as `setMainWindowTitle()` and `setMainWindowIcon()`. A detailed description explains that this is the main interface for the KMainWindow class. The page also includes a list of all members and constructor/destructor documentation.

Doxygen-generated KDE 3.5 API Reference for **KHistoryCombo Class Reference**. The page includes a class hierarchy diagram showing KHistoryCombo inheriting from KComboBox. It lists public member functions such as `addEntry()` and `removeEntry()`. A detailed description explains that this is a combination for adding a history and completion. The page also includes a list of all members and public slots.

Doxygen-generated KDE 3.5 API Reference for **KParts::ReadOnlyPart Class Reference**. The page includes a class hierarchy diagram showing KParts::ReadOnlyPart inheriting from KParts::Part. It lists public member functions such as `setReadOnly()` and `isReadOnly()`. A detailed description explains that this is the main interface for the KParts::ReadOnlyPart class. The page also includes a list of all members and public slots.

# Pour quels langages ?

## Langages



+ Possibilités d'extension pour d'autres langages *non natifs*

# PLAN

## 1 Introduction

## 2 Comment ça marche ?

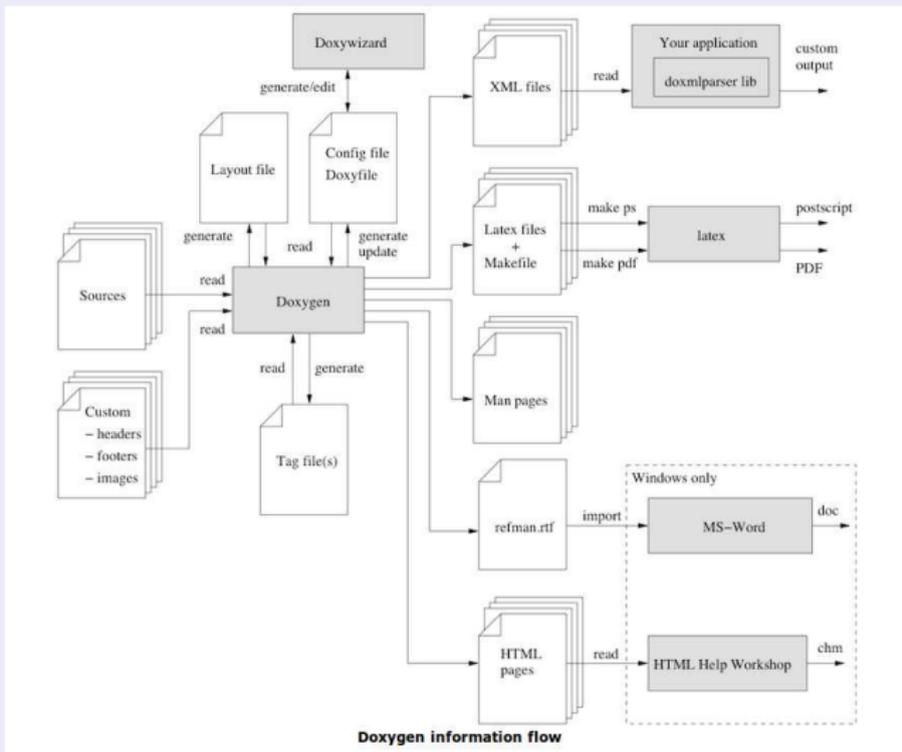
- Sur quels types de fichiers ?
- Pour quelles sorties ?

## 3 Un exemple détaillé : Doxygen

- Fiche d'identité
- **Flux de données**
- Comment utiliser Doxygen ?
- Balises
- Diagrammes et graphes
- Personnalisation des sorties
- Enrichir la documentation

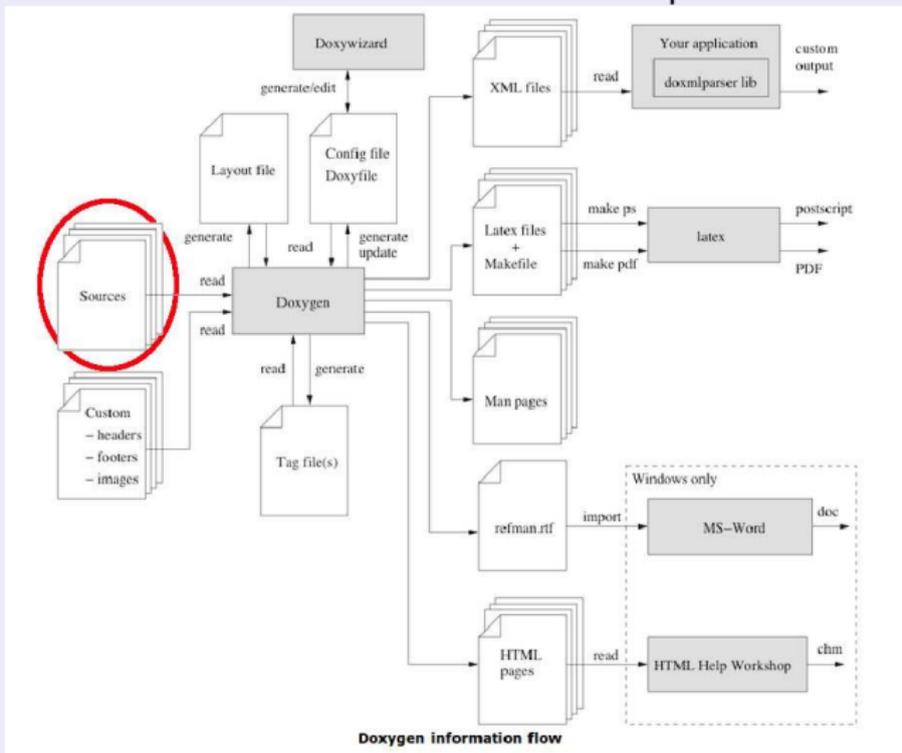
## 4 Pour aller plus loin ...

## Schéma général



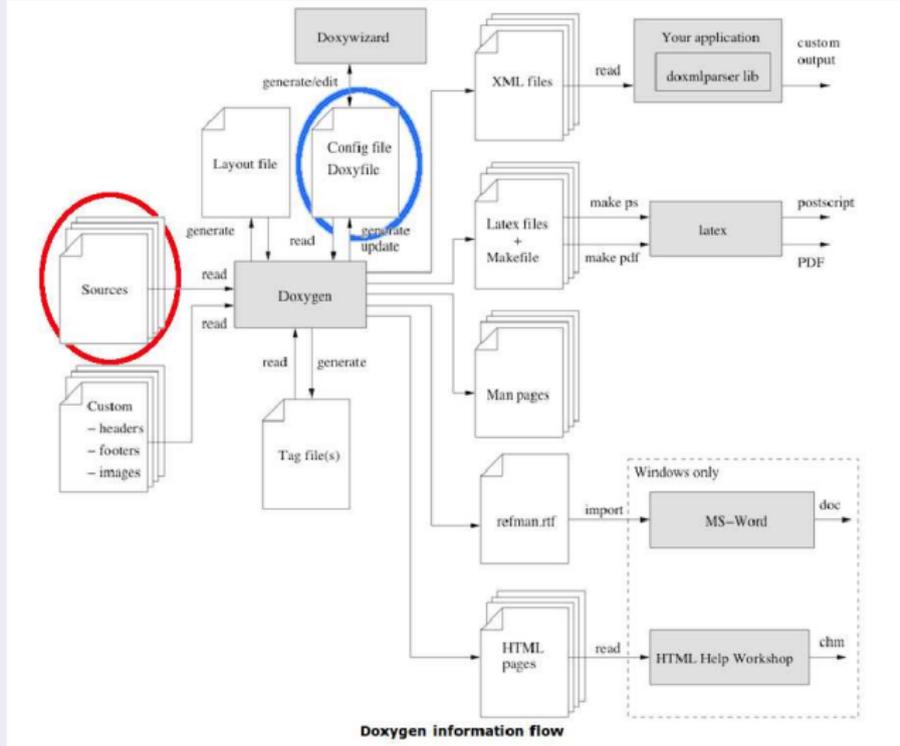
# Fichiers sources

## Seuls les fichiers sources sont indispensables



# Fichier de config

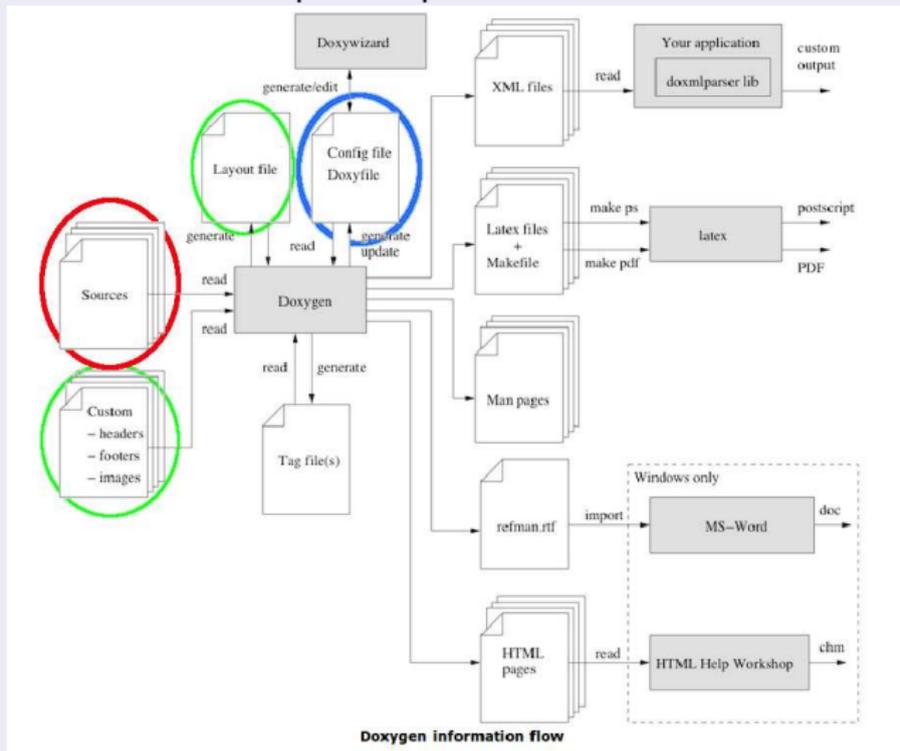
Pour définir/conservier les préférences, les options, ...



**DoxyWizard** permet de gérer ce fichier de configuration

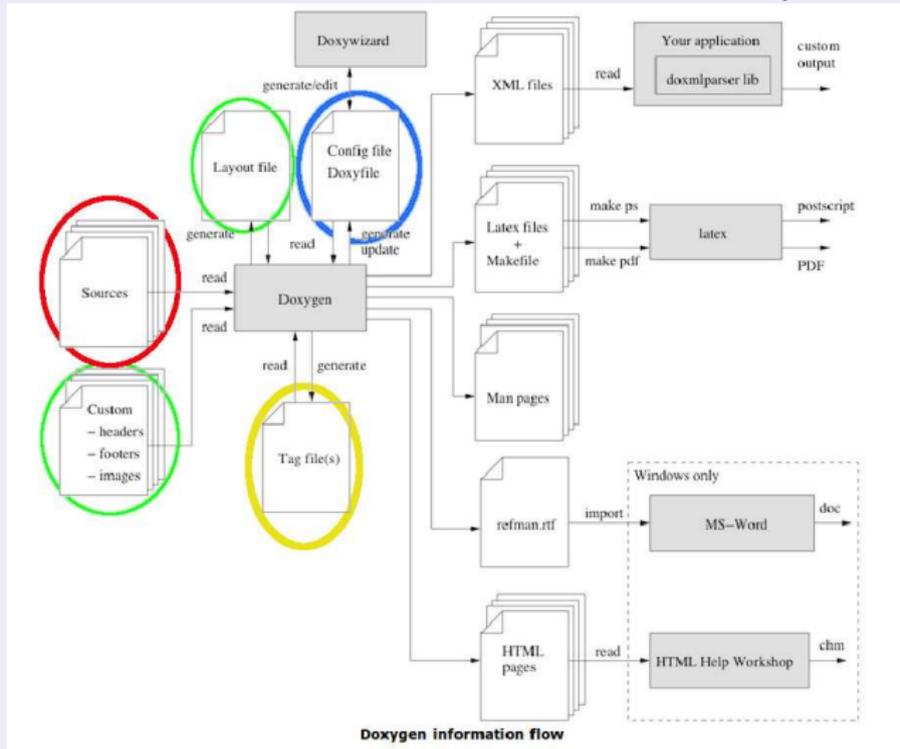
# Personnalisation des sorties

## Fichiers utilisés pour la personnalisation des sorties



# Documentations tierces

Pour insérer des documentations d'autres composants



# PLAN

## 1 Introduction

## 2 Comment ça marche ?

- Sur quels types de fichiers ?
- Pour quelles sorties ?

## 3 Un exemple détaillé : Doxygen

- Fiche d'identité
- Flux de données
- **Comment utiliser Doxygen ?**
- Balises
- Diagrammes et graphes
- Personnalisation des sorties
- Enrichir la documentation

## 4 Pour aller plus loin ...

# Comment utilise t-on Doxygen ?

## 1. En ligne de commande

### Uniquement avec les fichiers sources

```
user$ doxygen *.h
```

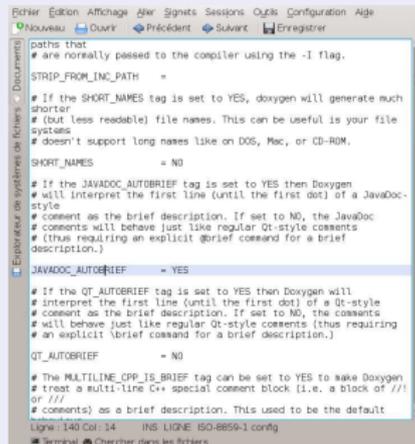
En utilisant la **configuration** et des **options de sortie** par défaut

### En utilisant un fichier de configuration

- Créer un fichier de configuration  

```
user$ doxygen -g my_config.txt
```
- Utiliser un fichier de configuration  

```
user$ doxygen my_config.txt
```

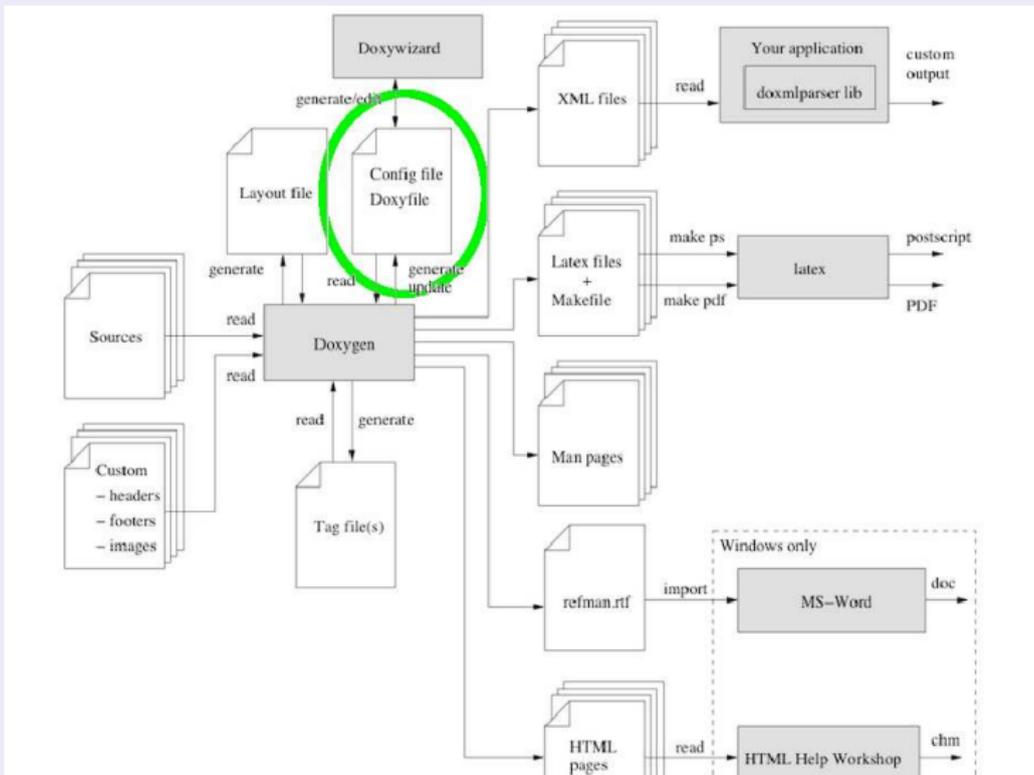


```
paths that
# are normally passed to the compiler using the -I flag.
STRIP_FROM_INC_PATH =
# If the SHORT_NAMES tag is set to YES, doxygen will generate much
shorter
# (but less readable) file names. This can be useful if your file
system
# doesn't support long names like on DOS, Mac, or CD-ROM.
SHORT_NAMES = NO
# If the JAVADOC_AUTOBRIEF tag is set to YES then Doxygen
# will interpret the first line (until the first dot) of a JavaDoc-
style
# comment as the brief description. If set to NO, the JavaDoc
# comments will behave just like regular Qt-style comments
# (thus requiring an explicit @brief command for a brief
description.)
JAVADOC_AUTOBRIEF = YES
# If the QT_AUTOBRIEF tag is set to YES then Doxygen will
# interpret the first line (until the first dot) of a Qt-style
# comment as the brief description. If set to NO, the comments
# will behave just like regular Qt-style comments (thus requiring
# an explicit @brief command for a brief description.)
QT_AUTOBRIEF = NO
# The MULTILINE_CPP_IS_BRIEF tag can be set to YES to make Doxygen
# treat a multi-line C++ special comment block (i.e. a block of ///
or ///
# comments) as a brief description. This used to be the default
Ligne : 140 Col : 14 INS LIGNE ISO-8859-1 config
Terminal Chercher dans les fichiers
```

# Comment utilise t-on Doxygen ?

## 2. Avec un outil graphique (1)

### DoxyWizard



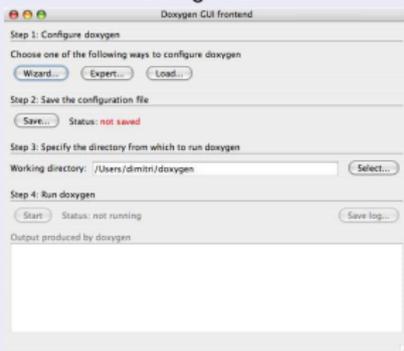
# Comment utilise t-on Doxygen ?

## 2. Avec un outil graphique (2)

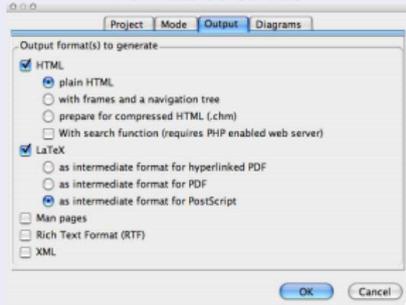
### DoxyWizard

Pour **configurer** et **lancer** doxygen

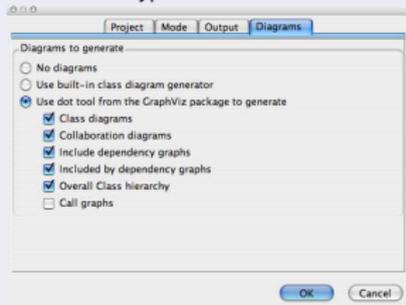
#### Configuration



#### Formats de sorties



#### Types de sorties



# PLAN

- 1 Introduction
- 2 Comment ça marche ?
  - Sur quels types de fichiers ?
  - Pour quelles sorties ?
- 3 Un exemple détaillé : Doxygen
  - Fiche d'identité
  - Flux de données
  - Comment utiliser Doxygen ?
  - **Balises**
  - Diagrammes et graphes
  - Personnalisation des sorties
  - Enrichir la documentation
- 4 Pour aller plus loin ...

# Rappel ...

## Documentation produite sans commentaire spécifique ni balise

`user$ doxygen Date.h`

### Date.h

```
Fichier  Édition  Affichage  Aller  Signets  Sessions  Outils  Configuration  Aide
Nouveau  Ouvrir  Précédent  Suivant  Enregistrer

class Date {
private:
    int _jour;
    int _mois;
    int _annee;
public:
    //-----
    // Constructeurs/destructeurs
    Date();
    Date(int iJour, int iMois, int iAnnee);
    Date(const Date & iDate);
    ~Date();

    //-----
    // Selecteurs
    //-----
    int getJour() const;
    int getMois() const;
    int getAnnee() const;
    void setJour(int iJour);
    void setMois(int iMois);
    void setAnnee(int iAnnee);

    //-----
    // Methodes
    //-----
    void saisir();
    void afficher() const;

    //-----
    // Surcharges d'operateurs
    //-----
    bool operator < (const Date& d) const;
    bool operator > (const Date& d) const;
    bool operator <= (const Date& d) const;
    bool operator >= (const Date& d) const;
    bool operator == (const Date& d) const;
    bool operator != (const Date& d) const;

friend istream& operator >> (istream& is, Date& d);
friend ostream& operator << (ostream& os, const Date& d);
```

### Documentation générée (html)

Fichier Édition Affichage Aller Signets Configuration Fenêtre Aide

Main Page Classes Files Search

Class List

## Date Class Reference

List of all members.

### Public Member Functions

	Date (int iJour, int iMois, int iAnnee)
	Date (const Date &iDate)
int	getJour () const
int	getMois () const
int	getAnnee () const
void	setJour (int iJour)
void	setMois (int iMois)
void	setAnnee (int iAnnee)
void	saisir ()
void	afficher () const
bool	operator< (const Date &d) const
bool	operator> (const Date &d) const
bool	operator<= (const Date &d) const
bool	operator>= (const Date &d) const
bool	operator== (const Date &d) const
bool	operator!= (const Date &d) const

### Friends

istream &	operator>> (istream &is, Date &d)
ostream &	operator<< (ostream &os, const Date &d)

The documentation for this class was generated from the following file:

- [date.h](#)

# Informations Doxygen

## Et Commentaires interprétables

Informations pour **Doxygen** dans des commentaires **interprétables par Doxygen**

En C/C++ :

- Style C avec avec deux \*  
/\*\*  
\* Documentation pour doxygen  
\*/
- Style C avec avec un !  
/\*!  
\* Documentation pour doxygen  
\*/
- Style C++ avec avec trois /  
///  
/// Documentation pour doxygen  
///  
///
- Style C++ avec avec un !  
///  
/\*!Documentation pour doxygen  
///  
/\*!

## Balises

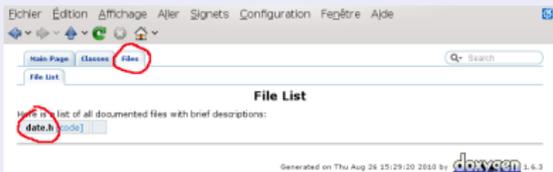
A placer à l'intérieur des commentaires interprétables par Doxygen.

- **\file**  
Description d'un fichier source ou d'en-tête
- **\brief**  
Description courte (peut être complétée par un lien vers la description détaillée)
- **\author**  
Auteur(s)
- **\version**  
Version
- **\date**  
Date
- **\param**  
Description de paramètre(s) d'une fonction (/méthode)
- **\return**  
Description de la valeur retournée
- **\bug**  
Description d'un bug
- **\deprecated**  
Description d'une partie de code obsolète
- **\class**  
Description d'une classe

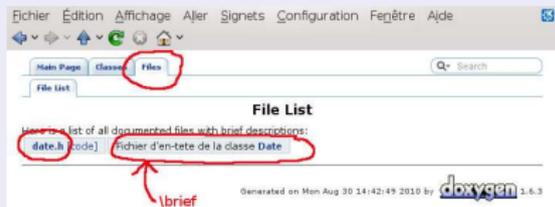
# Informations sur les fichiers d'en-tête

# Informations d'en-tête (1)

## Sans balise doxygen



## Avec balises d'en-tête doxygen



# Informations d'en-tête (2)

## Balises d'en-tête

```
Fichier  Édition  Affichage  Aller  Signets  Sessions  Outils  Configuration  Aide
Nouveau  Ouvrir  Précédent  Suivant  Enregistrer
Explorateur de systèmes de fichiers  Documents
#ifndef DATE_H_
#define DATE_H_

/**
 * \file date.h
 * \author F. Langrognnet
 * \version 1.1
 * \date 12 janvier 2009
 * \brief Fichier d'en-tete de la classe Date
 * +*/

#include <iostream>
using namespace std;

class Date {
public:
private:
    int _jour;
    int _mois;
    int _annee;

public:
    //-----
    // Constructeurs/destructeurs
    //-----
    Date();
    Date(int iJour, int iMois, int iAnnee);
    Date(const Date & iDate);
    ~Date();

    //-----
    // Selecteurs
    //-----
    int getJour() const;
    int getMois() const;
    int getAnnee() const;
    void setJour(int iJour);
    void setMois(int iMois);
    void setAnnee(int iAnnee);

    //-----
    // Methodes
    //-----
    void saisir();
    void afficher() const;
```

```
Fichier  Édition  Affichage  Aller  Signets  Configuration  Fenêtre  Aide
Main Page  Classes  Files
File Link
date.h File Reference
Fichier d'en-tete de la classe Date. More...
#include <iostream>
Go to the source code of this file.
Classes
class Date
Detailed Description
Fichier d'en-tete de la classe Date.
Author:
F. Langrognnet
Version:
1.1
Date:
12 janvier 2009
Generated on Mon Aug 30 15:16:46 2010 by doxygen 1.6.3
```

Description courte / détaillée

# Description courte / détaillée (1)

## Description courte / détaillée

On peut avoir :

- une description courte (**brief**)
- une description détaillée (**sans balise**)

```
Fichier  Édition  Affichage  Aller  Signets  Sessions  Outils  Configuration  Aide
Nouveau  Ouvrir  Précédent  Suivant  Enregistrer
Explorateur de systèmes de fichiers / Documents
#ifdef DATE_H
#define DATE_H
...
+ Fichier décrivant la classe Date \n
+ La classe Date est compose de 3 attributs, ... \n
... \n
+ \file date.h
+ author F. Langrognet
+ version 1.1
+ date 12 janvier 2009
+ \brief Fichier d'en-tete de la classe Date
+ */
#include <iostream>
using namespace std;
class Date {
private:
    int _jour;
    int _mois;
    int _annee;
public:
    // .....
    // Constructeurs/destructeurs
    // .....
    Date();
    Date(int iJour, int iMois, int iAnnee);
    Date(const Date & iDate);
    ~Date();
    // .....
    // Selecteurs
}
```

Fichier Édition Affichage Aller Signets Configuration Fenêtre Aide

Main Page Classes Files

File List

### date.h File Reference

Fichier d'en-tete de la classe Date More...

#include <iostream>

Go to the source code of this file.

Description courte

### Classes

class Date

### Detailed Description

Fichier d'en-tete de la classe Date

Fichier décrivant la classe Date  
La classe Date est compose de 3 attributs, ...

Author: F. Langrognet

Version: 1.1

Date: 12 janvier 2009

Generated on Mon Aug 30 15:41:12 2010 by **doxygen** 1.6.3

# Description courte / détaillée (2)

## Description courte / détaillée

On peut aussi générer automatiquement la **description courte** à partir de la **description détaillée** (option JAVADOC\_AUTOBRIEF à YES).

La description courte s'arrête au 1er point trouvé dans la description détaillée.

```
Fichier  Édition  Affichage  Aller  Signets  Sessions  Outils  Configuration  Aide
Nouveau  Ouvrir  Précédent  Suivant  Enregistrer

#ifndef DATE_H_
#define DATE_H_

/**
 * Fichier décrivant la classe Date
 * La classe Date est compose de 3 attributs, ...\n
 * ...
 *
 * \file date.h
 * \author F. Langrognet
 * \version 1.1
 * \date 12 janvier 2009
 */

#include <iostream>
using namespace std;

class Date {
public:
private:
    int _jour;
    int _mois;
    int _annee;

public:
    //-----
    // Constructeurs/destructeurs
    //-----
    Date();
    Date(int iJour, int iMois, int iAnnee);
    Date(const Date & iDate);
    ~Date();

    //-----
    // Selecteurs
    //-----
    int getJour() const;
```

```
Fichier  Édition  Affichage  Aller  Signets  Configuration  Fenêtre  Aide
Main Page  Classes  Files
Q- Search
File List

date.h File Reference

Fichier décrivant la classe Date. More...
#include <iostream>
Go to the source code of this file.

Classes
class Date

Detailed Description
Fichier décrivant la classe Date.
La classe Date est compose de 3 attributs, ...
...
Author:
F. Langrognet
Version:
1.1
Date:
12 janvier 2009

Generated on Mon Aug 30 16:00:34 2010 by doxygen 1.4.3
```

Description d'une méthode / fonction

# Description d'une méthode / fonction (1)

## Description d'une méthode

### Utilisation de `\param` et `\return`.

```
Fichier  Edition  Affichage  Aller  Signets  Sessions  Outils  Configuration  Aide
Nouveau  Ouvrir  Précédent  Suivant  Enregistrer
Date(int iJour, int iMois, int iAnnee);
/** Constructeur par copie.
 */
Date(const Date &iDate);
/** Destructeur
 */
~Date();
/**
 * Selecteur (get) de l'attribut _jour.
 */
return la valeur du jour
int getJour() const;
/**
 * Selecteur (get) de l'attribut _mois.
 */
return la valeur du mois
int getMois() const;
/**
 * Selecteur (get) de l'attribut _annee.
 */
return la valeur de l'annee
int getAnnee() const;
/**
 * Selecteur (set) de l'attribut _jour.
 */
param la valeur du jour
void setJour(int iJour);
/**
 * Selecteur (set) de l'attribut _mois.
 */
param la valeur du mois
void setMois(int iMois);
/**
 * Selecteur (set) de l'attribut _annee.
 */
param la valeur de l'annee
```

```
Fichier  Edition  Affichage  Aller  Signets  Configuration  Fenêtre  Aide
creer une nouvelle date (iJour/Mois/iAnnee).
Parameters:
    iJour entier representant le jour
    iMois entier representant le mois
    iAnnee entier representant l'annee

Member Function Documentation
int Date::getAnnee ( ) const
Selecteur (get) de l'attribut _annee.
Returns:
    la valeur de l'annee
int Date::getJour ( ) const
Selecteur (get) de l'attribut _jour.
Returns:
    la valeur du jour
int Date::getMois ( ) const
Selecteur (get) de l'attribut _mois.
Returns:
    la valeur du mois
void Date::setAnnee ( int iAnnee )
Selecteur (set) de l'attribut _annee.
Parameters:
    la valeur de l'annee
void Date::setJour ( int iJour )
Selecteur (set) de l'attribut _jour.
Parameters:
    la valeur du jour
void Date::setMois ( int iMois )
Selecteur (set) de l'attribut _mois.
Parameters:
    la valeur du mois
```

# Description d'une méthode / fonction (2)

## Balises pour une méthode / fonction

```
Fichier  Édition  Affichage  Aller  Signets  Sessions  Outils  Configuration  Aide
Nouveau  Ouvrir  Précédent  Suivant  Enregistrer  Enregistrer sous
Explorateur de systèmes de fichiers  Documents
#ifndef DATE_H
#define DATE_H

#include <iostream>
using namespace std;

class Date {
private:
    int _jour;
    int _mois;
    int _annee;
public:
    /** Constructeur par défaut.
     * cree une nouvelle date (01/01/0000). */
    Date();

    /** Constructeur par initialisation.
     * cree une nouvelle date (iJour/iMois/iAnnee).
     * @param iJour entier representant le jour
     * @param iMois entier representant le mois
     * @param iAnnee entier representant l'annee
     */
    Date(int iJour, int iMois, int iAnnee);

    /** Constructeur par copie.
     */
    Date(const Date & iDate);

    /** Destructeur
     */
    ~Date();

    //-----
    // Selecteurs
    //-----
    int getJour() const;
    int getMois() const;
    int getAnnee() const;
    void setJour(int iJour);
    void setMois(int iMois);
    void setAnnee(int iAnnee);

    //-----
    // Methodes
    //-----
    void saisir();
    void afficher() const;
```

Description courte et détaillée

Description courte uniquement

```
Fichier  Édition  Affichage  Aller  Signets  Configuration  Fenêtre  Aide
Public Member Functions
Date ()
Constructeur par défaut.
Date (int iJour, int iMois, int iAnnee)
Constructeur par initialisation.
Date (const Date &iDate)
Constructeur par copie.
~Date ()
Destructeur.
int getJour () const
int getMois () const
int getAnnee () const
void setJour (int iJour)
void setMois (int iMois)
void setAnnee (int iAnnee)
void saisir ()
void afficher () const
bool operator= (const Date &d) const
Friends
istream & operator>> (istream &is, Date &d)
ostream & operator<< (ostream &os, const Date &d)
Constructor & Destructor Documentation
Date::Date ( )
Constructeur par défaut.
cree une nouvelle date (01/01/0000).
Date::Date ( int iJour,
int iMois,
int iAnnee
)
Constructeur par initialisation.
cree une nouvelle date (iJour/Mois/Annee).
Parameters:
iJour entier representant le jour
iMois entier representant le mois
iAnnee entier representant l'annee
The documentation for this class was generated from the following file:
```

Description courte

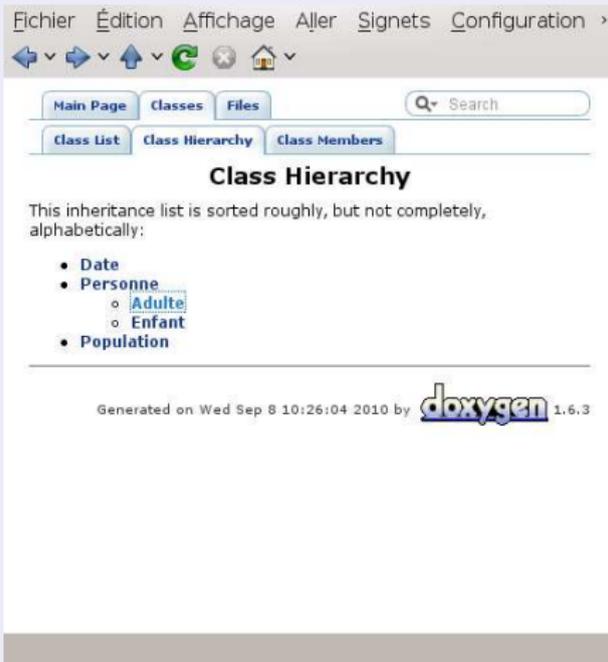
Description détaillée

# PLAN

- 1 Introduction
- 2 Comment ça marche ?
  - Sur quels types de fichiers ?
  - Pour quelles sorties ?
- 3 **Un exemple détaillé : Doxygen**
  - Fiche d'identité
  - Flux de données
  - Comment utiliser Doxygen ?
  - Balises
  - **Diagrammes et graphes**
  - Personnalisation des sorties
  - Enrichir la documentation
- 4 Pour aller plus loin ...

## Fonctionnement par défaut

### Hiérarchie de classes



Fichier Édition Affichage Aller Signets Configuration >

Main Page Classes Files Search

Class List Class Hierarchy Class Members

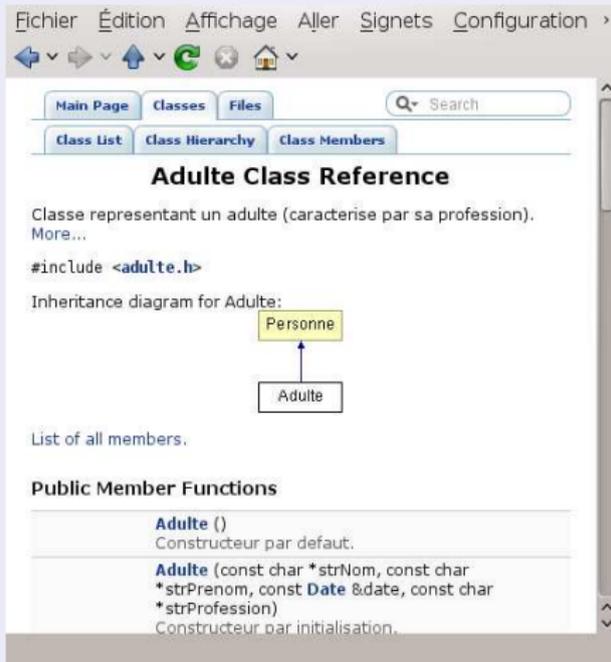
### Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- [Date](#)
- [Personne](#)
  - [Adulte](#)
  - [Enfant](#)
- [Population](#)

---

Generated on Wed Sep 8 10:26:04 2010 by  1.6.3



Fichier Édition Affichage Aller Signets Configuration >

Main Page Classes Files Search

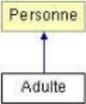
Class List Class Hierarchy Class Members

### Adulte Class Reference

Classe representant un adulte (caracterise par sa profession).  
More...

```
#include <adulte.h>
```

Inheritance diagram for Adulte:



```
graph BT;
    Adulte --> Personne;
```

List of all members.

#### Public Member Functions

<b>Adulte</b> () Constructeur par défaut.
<b>Adulte</b> (const char *strNom, const char *strPrenom, const <a href="#">Date</a> &date, const char *strProfession) Constructeur par initialisation.

## Personnalisation

Avec le logiciel **graphviz** (et l'option HAVE\_DOT=YES)

- Hiérarchie de classes
- Diagramme de collaboration

The screenshot shows the Doxygen web interface for a 'Graphical Class Hierarchy'. The browser window has a menu bar with 'Fichier', 'Édition', 'Affichage', 'Aller', 'Signets', 'Configuration', 'Fenêtre', and 'Aide'. Below the menu is a navigation bar with 'Main Page', 'Classes', and 'Files' tabs, and a search box. The main content area is titled 'Graphical Class Hierarchy' and includes a link 'Go to the textual class hierarchy'. The diagram shows a hierarchy where 'Date' is a base class, and 'Adulte' and 'Enfant' inherit from 'Personne'. 'Population' is also shown as a separate class. At the bottom, it says 'Generated on Wed Sep 8 10:23:42 2010 by doxygen 1.6.3'.

The screenshot shows the Doxygen web interface for the 'Personne Class Reference'. The browser window has the same menu bar as the previous screenshot. The navigation bar shows 'Main Page', 'Classes', and 'Files' tabs, with 'Class Hierarchy' and 'Class Members' tabs selected. The main content area is titled 'Personne Class Reference' and includes a description: 'Classe representant une personne: nom, prenom et date de naissance. More...'. It shows the include directive '#include <personne.h>' and the text 'Inheritance diagram for Personne:'. The inheritance diagram shows 'Personne' at the top, with 'Adulte' and 'Enfant' below it, both pointing to 'Personne'. Below this is the text 'Collaboration diagram for Personne:' followed by a diagram showing 'Date' pointing to 'Personne' with the label 'dateDeNaissance'. At the bottom, it says 'List of all members.' and 'Public Member Functions'.





# PLAN

## 1 Introduction

## 2 Comment ça marche ?

- Sur quels types de fichiers ?
- Pour quelles sorties ?

## 3 Un exemple détaillé : Doxygen

- Fiche d'identité
- Flux de données
- Comment utiliser Doxygen ?
- Balises
- Diagrammes et graphes
- **Personnalisation des sorties**
- Enrichir la documentation

## 4 Pour aller plus loin ...

# Personnalisation des sorties

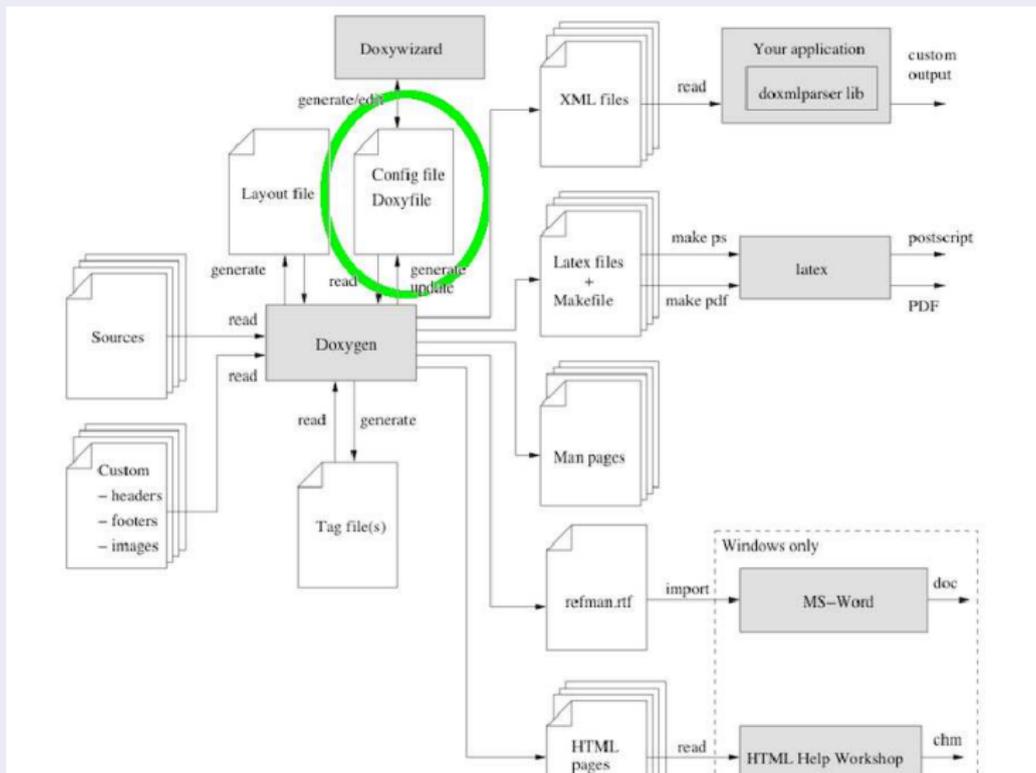
## 3 niveaux de personnalisation

- Avec les **fichiers de configuration** (modifications mineures)
- Avec le **fichier de Layout**
- Avec **XML** (pour une personnalisation totale)

# Personnalisation des sorties

## 1. Avec les fichiers de configuration (1)

### Fichier de configuration



# Personnalisation des sorties

## 1. Avec les fichiers de configuration (2)

### Exemples

#### ● Couleurs

- ▶ HTML\_COLORSTYLE\_HUE
- ▶ HTML\_COLORSTYLE\_SAT
- ▶ HTML\_COLORSTYLE\_GAMMA

Note : peut être msie à jour via DoxyWizard

#### ● Navigation

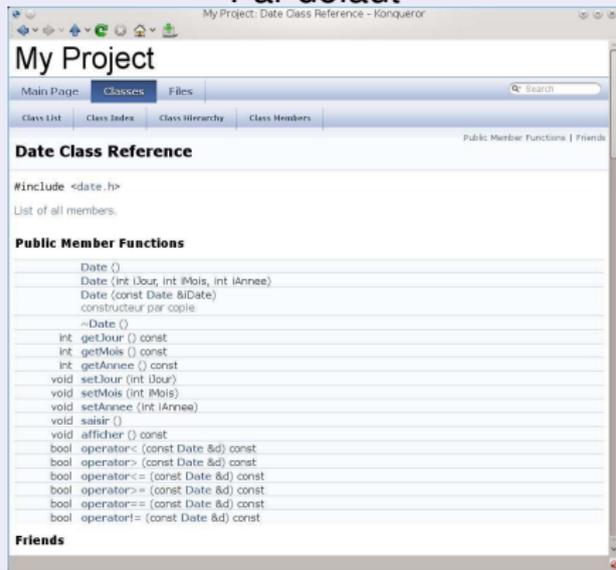
- ▶ DISABLE\_INDEX (=NO par défaut)
- ▶ GENERATE\_TREEVIEW (=NO par défaut)

# Personnalisation des sorties

## 1. Avec les fichiers de configuration (3)

### Couleurs

#### Par défaut



My Project: Date Class Reference - Konqueror

My Project

Main Page | **Classes** | Files

Class List | Class Index | Class Hierarchy | Class Members

Date Class Reference

```
#include <date.h>
```

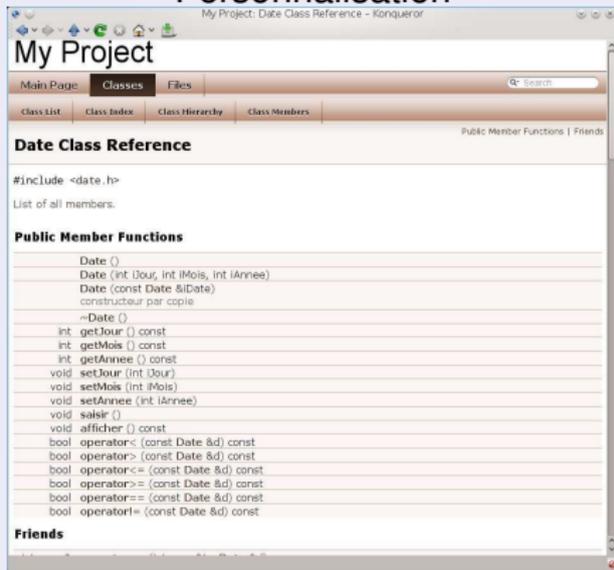
List of all members.

**Public Member Functions**

```
Date ()
Date (int Jour, int Mois, int lAnnee)
Date (const Date &Date)
    constructeur par copie
~Date ()
int getJour () const
int getMois () const
int getAnnee () const
void setJour (int Jour)
void setMois (int Mois)
void setAnnee (int lAnnee)
void saisir ()
void afficher () const
bool operator< (const Date &d) const
bool operator> (const Date &d) const
bool operator<= (const Date &d) const
bool operator>= (const Date &d) const
bool operator== (const Date &d) const
bool operator!= (const Date &d) const
```

**Friends**

#### Personnalisation



My Project: Date Class Reference - Konqueror

My Project

Main Page | **Classes** | Files

Class List | Class Index | Class Hierarchy | Class Members

Date Class Reference

```
#include <date.h>
```

List of all members.

**Public Member Functions**

```
Date ()
Date (int Jour, int Mois, int lAnnee)
Date (const Date &Date)
    constructeur par copie
~Date ()
int getJour () const
int getMois () const
int getAnnee () const
void setJour (int Jour)
void setMois (int Mois)
void setAnnee (int lAnnee)
void saisir ()
void afficher () const
bool operator< (const Date &d) const
bool operator> (const Date &d) const
bool operator<= (const Date &d) const
bool operator>= (const Date &d) const
bool operator== (const Date &d) const
bool operator!= (const Date &d) const
```

**Friends**

# Personnalisation des sorties

## 1. Avec les fichiers de configuration (4)

### Navigation

#### Par défaut

My Project: Date Class Reference - Konqueror

## My Project

Main Page | **Classes** | Files

Class List | Class Index | Class Hierarchy | Class Members

### Date Class Reference

```
#include <date.h>
```

List of all members.

#### Public Member Functions

- Date ()
- Date (int lJour, int lMois, int lAnnee)
- Date (const Date &Date)
- constructeur par copie
- ~Date ()
- int getJour () const
- int getMois () const
- int getAnnee () const
- void setJour (int lJour)
- void setMois (int lMois)
- void setAnnee (int lAnnee)
- void saisir ()
- void afficher () const
- bool operator< (const Date &d) const
- bool operator> (const Date &d) const
- bool operator<= (const Date &d) const
- bool operator>= (const Date &d) const
- bool operator== (const Date &d) const
- bool operator!= (const Date &d) const

#### Friends

#### Personnalisation

My Project: Date Class Reference - Konqueror

## My Project

Search

### Date Class Reference

```
#include <date.h>
```

List of all members.

#### Public Member Functions

- Date ()
- Date (int lJour, int lMois, int lAnnee)
- Date (const Date &Date)
- constructeur par copie
- ~Date ()
- int getJour () const
- int getMois () const
- int getAnnee () const
- void setJour (int lJour)
- void setMois (int lMois)
- void setAnnee (int lAnnee)
- void saisir ()
- void afficher () const
- bool operator< (const Date &d) const
- bool operator> (const Date &d) const
- bool operator<= (const Date &d) const
- bool operator>= (const Date &d) const
- bool operator== (const Date &d) const
- bool operator!= (const Date &d) const

#### Friends

- istream & operator>> (istream &is, Date &d)
- ostream & operator<< (ostream &os, const Date &d)

#### Detailed Description

Classe representant une date, de la forme jour/mois/annee.

#### Constructor & Destructor Documentation

Generated on Wed Jan 16 2013 10:32:28 for My Project by [doxygen](#) 1.7.6.3



# Personnalisation des sorties

## 1. Avec les fichiers de configuration (6)

### Créer les 3 fichiers et les modifier

`user$ doxygen -w html header.html footer.html customdoxygen.css`

#### Feuille de style

```
body, table, div, p, dl {
  font-family: Lucida Grande, Verdana, Geneva, Arial,
  sans-serif;
  font-size: 10px;
}

/* adjust heading levels */
h2 {
  text-align: center;
  font-size: 100%;
}
h3 {
  font-size: 120%;
}
h4 {
  font-size: 100%;
}
p {
  font-weight: bold;
}

div.Multirow {
  -moz-column-gap: 5px;
  -webkit-column-gap: 5px;
  -ms-column-gap: 5px;
  -moz-column-count: 2;
  -webkit-column-count: 2;
  -ms-column-count: 2;
}

div[id$=1, id$=99, id$=start] {
  margin-top: 20px;
}

div[id$=1] {
  margin-bottom: 50px;
}
```

#### En-tête

```

<!-- This comment will not be in the output -->
<DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0
  Transitional//EN" "http://www.w3.org/TR/html40/
  transitional.dtd" [
  <meta name="http://www.w3.org/1999/xhtml">
  <!--
  <meta http-equiv="Content-Type"
    content="text/html; charset=utf-8" />
  </-->
  <title>Title</title>
  <link href="stylesheet.css" rel="stylesheet"
    type="text/css" />
  <link href="stylesheet.css" rel="stylesheet"
    type="text/css" />
  <script type="text/javascript"
    src="search.js">
  </script>
  <link href="searchdoxygen.css" rel="stylesheet"
    type="text/css" />
  </head>
  <body onload="searchBox.DeflectInit();">
```

#### Pied de page

```



---



Generated on date for project by doxygen  

  href="http://www.doxygen.org/index.html">http://www.doxygen.org/index.html


</body>
</html>
```

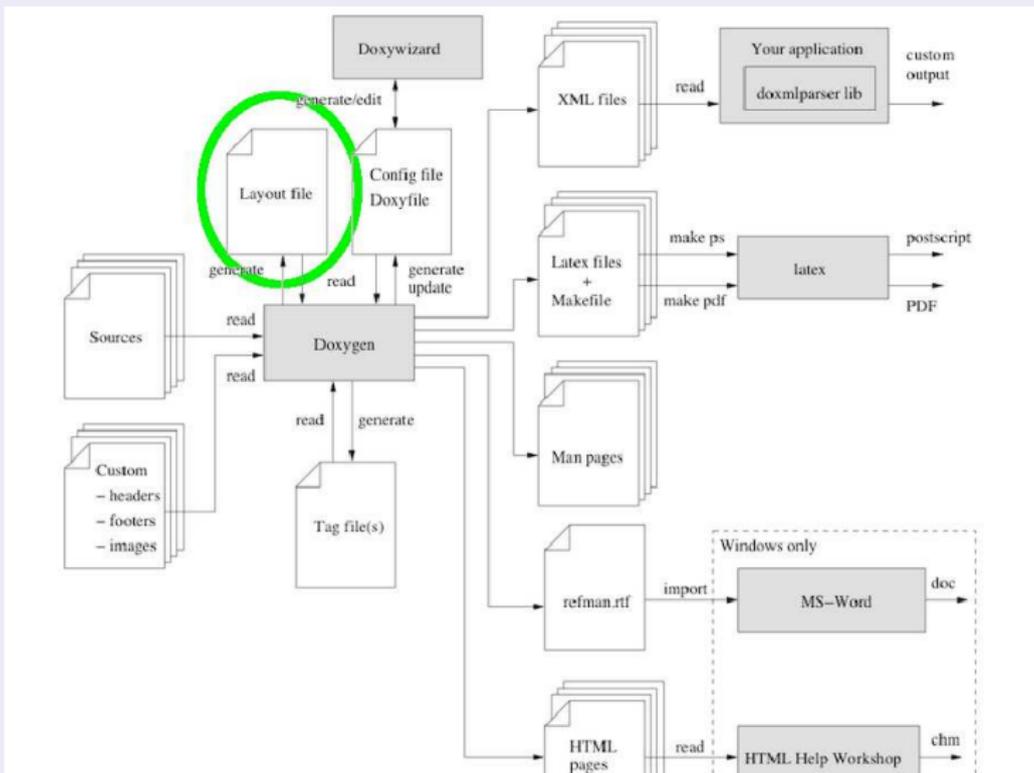
## Les référencer dans le fichier de configuration

- HTML\_HEADER = header.html
- HTML\_FOOTER = footer.html
- HTML\_STYLESHEET = customdoxygen.css

# Personnalisation des sorties

## 2. Modification de la structure avec le fichier layout (1)

### Fichier de configuration



# Personnalisation des sorties

## 2. Modification de la structure avec le fichier layout (2)

### Layout

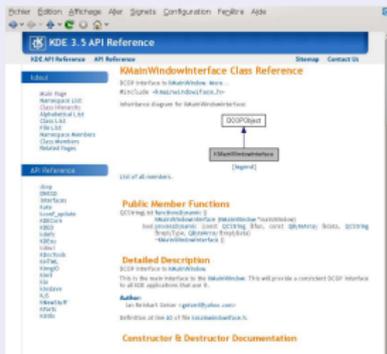
- Créer un fichier layout et le modifier

`user$ doxygen -l`

- Référencer ce fichier layout dans le fichier de configuration (`LAYOUT_FILE = DoxygenLayout.xml`) :

- Execution de doxygen

`user$ doxygen configWithLayout.txt`



# PLAN

- 1 Introduction
- 2 Comment ça marche ?
  - Sur quels types de fichiers ?
  - Pour quelles sorties ?
- 3 Un exemple détaillé : Doxygen
  - Fiche d'identité
  - Flux de données
  - Comment utiliser Doxygen ?
  - Balises
  - Diagrammes et graphes
  - Personnalisation des sorties
  - **Enrichir la documentation**
- 4 Pour aller plus loin ...

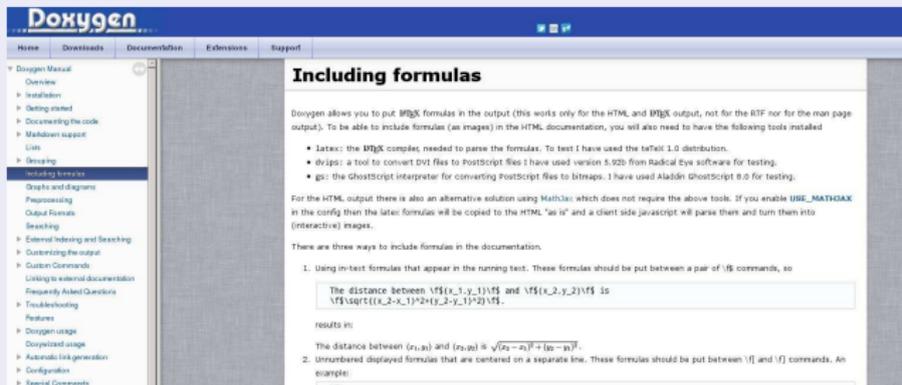
# Enrichir la documentation

## Quelques outils

- Utilisation de balises Doxygen spécifiques à la mise en page (listes, ....)
- Insertion de balises HTML
- Utilisation de Latex (pour les formules par exemple)

Motivations pour utiliser Doxygen pour gérer les documentations scientifiques et/ou pour utilisateurs

## Guide utilisateur Doxygen produit par ... Doxygen



The screenshot shows the Doxygen user guide interface. The top navigation bar includes 'Home', 'Downloads', 'Documentation', 'Extensions', and 'Support'. The left sidebar contains a tree view with categories like 'Doxygen Manual', 'Installation', 'Building', 'Customizing', and 'Special Commands'. The main content area is titled 'Including formulas' and contains the following text:

Doxygen allows you to put  $\LaTeX$  formulas in the output (this works only for the HTML, and  $\LaTeX$  output, not for the RTF nor for the man page output). To be able to include formulas (as images) in the HTML documentation, you will also need to have the following tools installed

- latex: the  $\LaTeX$  compiler, needed to parse the formulas. To test I have used the TeXLive 1.0 distribution.
- dvips: a tool to convert DVI files to PostScript files I have used version 5.92b from Radical Eye software for testing.
- gs: the GhostScript interpreter for converting PostScript files to bitmaps. I have used Aladdin Ghostscript 8.0 for testing.

For the HTML output there is also an alternative solution using MathJax which does not require the above tools. If you enable USE\_MATHJAX in the config then the latex formulas will be copied to the HTML "as is" and a client side javascript will parse them and turn them into (interactive) images.

There are three ways to include formulas in the documentation.

1. Using in-text formulas that appear in the running text. These formulas should be put between a pair of  $\$$  commands, so

The distance between  $\sqrt{(x_1 - y_1)^2}$  and  $\sqrt{(x_2 - y_2)^2}$  is  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ .

RESULTS:

The distance between  $(x_1, y_1)$  and  $(x_2, y_2)$  is  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ .

2. Unnumbered displayed formulas that are centered on a separate line. These formulas should be put between  $\[$  and  $\]$  commands. An example:

# Enrichir la documentation (2)

## Exemple avec une formule

```
truc.h - Kate
File Edit View Bookmarks Sessions Tools Settings Help
#ifndef __TRUC_H__
#define __TRUC_H__
/**
 * \file truc.h
 * \author F. Langrognet
 * \version 1.2
 * \date 16/1/2013
 * \f{
 * |I_2| = \left| \int_0^T \int_{\Omega} u_t(x,t) \psi(x) dx dt - \int_0^T \int_{\Omega} \frac{\partial u}{\partial t}(x,t) \int_{\Omega} c(x) u(x,t) dx dt \right|
 * \f}
 * */
class Truc {
public:
    Truc();
};
#endif // ifndef __TRUC_H__
```

My Project: truc.h File Reference - Konqueror

# My Project

Main Page | Classes | **Files** | Search

File List

## truc.h File Reference

Go to the source code of this file.

### Classes

class Truc

### Detailed Description

Author:  
F. Langrognet

Version:  
1.2

Date:  
16/1/2013

$$|I_2| = \left| \int_0^T \int_{\Omega} u_t(x,t) \psi(x) dx dt - \int_0^T \int_{\Omega} \frac{\partial u}{\partial t}(x,t) \int_{\Omega} c(x) u(x,t) dx dt \right|$$

Generated on Wed Jan 16 2013 11:23:12 for My Project by [doxygen](#) 1.7.6.1

# PLAN

- 1 Introduction
- 2 Comment ça marche ?
  - Sur quels types de fichiers ?
  - Pour quelles sorties ?
- 3 Un exemple détaillé : Doxygen
  - Fiche d'identité
  - Flux de données
  - Comment utiliser Doxygen ?
  - Balises
  - Diagrammes et graphes
  - Personnalisation des sorties
  - Enrichir la documentation
- 4 Pour aller plus loin ...

# Quel outil de génération de documentation choisir ?

[en.wikipedia.org/wiki/Comparison\\_of\\_documentation\\_generators](http://en.wikipedia.org/wiki/Comparison_of_documentation_generators)

## Quel langage ?

	C/C++	Java	C#	VB / VBScript	Delphi / Pascal	Ada	D	IDL	.NET <sup>1</sup>	Access	Fortran
ApiGen	No	No	No	No	No	No	No	No	No	No	No
Autoduck	Yes	No	No	Yes	No	No	No	No	No	No	No
BI Documenter	No	No	No	No	No	No	No	No	Yes	No	No
classdoc	No	Yes	No	No	No	No	No	No	No	No	No
CppDoc	Yes	No	No	No	No	No	No	No	No	No	No
DB Manual	No	No	No	No	No	No	No	No	Yes	No	No
DBDesc	No	No	No	No	No	No	No	No	Yes	No	No
DBScribe	No	No	No	No	No	No	No	No	No	No	No
Ddoc	No	No	No	No	No	No	Yes	No	No	No	No
devescribe	No	No	Yes	No	No	No	No	No	No	No	No
Doc-O-Matic	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No	No
DOC++	Yes	Yes	No	No	No	No	No	Yes	No	No	No
Document! X	C++/CLI only	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes	No
Doxygen	Yes	Yes	Yes	with Plugin <sup>2</sup>	with Plugin <sup>2</sup>	No	Partial	Yes	No	No	Yes
Epydoc	No	No	No	No	No	No	No	No	No	No	No
ForgeDoc	No	No	No	No	No	No	No	Yes	No	No	No
fpdoc	No	No	No	No	Yes	No	No	No	No	No	No
GenHelp	Yes	No	Yes	No	Yes	No	No	Yes	No	No	No
Haddock	No	No	No	No	No	No	No	No	No	No	No
HeaderDoc	Yes	Yes	No	No	Yes	No	No	Yes	No	No	No
Help Generator	Yes	No	Yes	Yes	No	No	No	Yes	Yes	Yes	No
HyperSQL	No	No	No	No	No	No	No	No	No	No	No
Imagix 4D	Yes	Yes	No	No	No	No	No	No	No	No	No
Javadoc	No	Yes	No	No	No	No	No	No	No	No	No
JGrouseDoc	No	No	No	No	No	No	No	No	No	No	No
JSdoc	No	No	No	No	No	No	No	No	No	No	No
JuDoc Toolkit	No	No	No	No	No	No	No	No	No	No	No
KDOC	Yes	No	No	No	No	No	No	Yes	No	No	No
Natural Docs	Partial	Partial	Yes	Partial	Partial	Partial	No	No	No	No	Partial
NDoc	No	No	Yes	No	No	No	No	No	Yes	No	No
phpDocumentor	No	No	No	No	No	No	No	No	No	No	No

# Quel outil de génération de documentation choisir ?

## Quel langage ? ... suite

	PHP	Perl	Python	Ruby	JavaScript	ActionScript	PL/SQL	Tcl	Haskell
ApiGen	Yes	No	No	No	No	No	No	No	No
AutoFuck	No	No	No	No	No	No	No	No	No
BI Documenter	No	No	No	No	No	No	No	No	No
classdoc	No	No	No	No	No	No	No	No	No
CppDoc	No	No	No	No	No	No	No	No	No
DB Manual	No	No	No	No	No	No	No	No	No
DBDoc	No	No	No	No	No	No	No	No	No
DBScribe	No	No	No	No	No	No	Yes	No	No
Ddoc	No	No	No	No	No	No	No	No	No
devscribe	No	No	No	No	No	No	No	No	No
Doc-O-Matic	Yes	No	No	No	Yes	No	No	No	No
DOC++	No	No	No	No	No	No	No	No	No
Document! X	No	No	No	No	No	No	Yes	No	No
Doxygen	Yes	No	Yes	No	No	No	No	Yes	No
Epydoc	No	No	Yes	No	No	No	No	No	No
ForgeDoc	No	No	No	No	No	No	No	No	No
fpdoc	No	No	No	No	No	No	No	No	No
GemHelp	No	No	No	No	No	No	No	No	No
Haddock	No	No	No	No	No	No	No	No	Yes
HeaderDoc	Yes	Yes	Yes	Yes	Yes	No	No	Yes	No
Help Generator	No	No	No	No	No	No	No	No	No
HyperSQL	No	No	No	No	No	No	Yes	No	No
Imagix 4D	No	No	No	No	No	No	No	No	No
Javadoc	No	No	No	No	No	No	No	No	No
JGrouseDoc	No	No	No	No	Yes	No	No	No	No
JSdoc	No	No	No	No	Yes	No	No	No	No
JsDoc Toolkit	No	No	No	No	Yes	No	No	No	No
KDOC	No	No	No	No	No	No	No	No	No
Natural Docs	Partial	Yes	Partial	Partial	Partial	Yes	Partial	Partial	No
NDoc	No	No	No	No	No	No	No	No	No
phpDocumentor	Yes	No	No	No	No	No	No	No	No
phpSimpleDoc	Yes	No	No	No	No	No	No	No	No
Project Analyzer	No	No	No	No	No	No	No	No	No

# Quel outil de génération de documentation choisir ?

## Quel type de fichier en entrée ? ?

	Text	Binary
ApiGen	Yes	No
Autoduck	Yes	No
BI Documenter	No	Yes
DB Manual	No	Yes
classdoc	No	Yes
CppDoc	Yes	No
DBDesc	No	Yes
DBScribe	No	Yes
Ddoc	Yes	No
devacrtbe	No	Yes
Doc-O-Matic	Yes	No
DOC++	Yes	No (except Java class files)
Document! X	Yes	Yes
Doxygen	Yes	No
Epydoc	Yes	No
ForgeDoc	No	Yes
fpdoc	Yes	No
GenHelp	Yes	No
HaalDock	Yes	No
HeaderDoc	Yes	No
Help Generator	Yes	Yes
HyperSQL	Yes	No
IDLdoc	Yes	No (except for sav files)
Imagic 4D	Yes	No
Javadoc	Yes	No
JGrouseDoc	Yes	No
JSDoc	Yes	No
JaDoc Toolkit	Yes	No
KDOC	Yes	No
mkd	Yes	No
Natural Docs	Yes	No
NDoc	No	Yes

# Quel outil de génération de documentation choisir ?

## Quel OS ?

	Windows	Mac OS X	Linux	BSD	Unix
ApiGen	Yes	Yes	Yes	Yes	Yes
Appledoc	No	Yes	No	No	No
Autolink	Yes	No	No	No	No
BI Documenter	Yes	No	No	No	No
classdoc	Yes	Yes	Yes	Yes	Yes
CppDoc	Yes	No	Yes	No	No
DB Manual	Yes	No	No	No	No
DBDesc	Yes	No	No	No	No
DBScribe	Yes	No	No	No	No
Ddoc	Yes	Yes?	Yes	Yes?	No
describe	Yes	No	No	No	No
Doc-O-Matic	Yes	No	No	No	No
DOC++	Yes	Yes?	Yes	Yes	Yes
Document X	Yes	No	No	No	No
Doxygen	Yes	Yes	Yes	Yes	Yes
Epydoc	Yes	Yes	Yes	Yes	Yes
ForgeDoc	Yes	No	No	No	No
fpdoc	Yes	Yes	Yes	Yes	Yes
GeniHelp	Yes	No	No	No	No
HeadDock	Yes	Yes	Yes	Yes	Yes
HeaderDoc	No	Yes	Yes	Yes	Yes
Help Generator	Yes	No	No	No	No
HyperSQL	Yes	Yes	Yes	Yes	Yes
IDLdoc	Yes	Yes	Yes	No	Yes
Image 4D	Yes	No	Yes	No	Yes
Javadoc	Yes	Yes	Yes	Yes	Yes
JGrouseDoc	Yes	Yes	Yes	Yes	Yes
JSdoc	Yes	Yes	Yes	Yes	Yes
JsDoc Toolkit	Yes	Yes	Yes	Yes	Yes
KDOC	Partial	Yes	Yes	Yes	Yes
mkul	Yes	No?	Yes	Yes	Yes
Natural Docs	Yes	Yes	Yes	Yes	Yes

# Quel outil de génération de documentation choisir ?

## Format de sortie

	HTML	CHM	RTF	PDF	LaTeX	PostScript	man pages	DocBook	XML
ApiGen	Yes	No							
Autofuck	Yes	No	Yes	No	No	No	No	No	No
BI Documenter	Yes	Yes	No						
classdoc	Yes	No							
CppDoc	Yes	No							
DB Manual	Yes	Yes	No						
DBDesc	Yes	No	Yes	Yes	No	No	No	No	Yes
DBScribe	Yes	Yes	Yes	No	No	No	No	No	No
Ddoc	Yes	Yes <sup>1</sup>	No	Yes <sup>4</sup>	Yes <sup>4</sup>	Yes <sup>4</sup>	Yes <sup>4</sup>	No	Yes <sup>4</sup>
devscribe	Yes	No	Yes						
Doc-O-Matic	Yes	Yes	Yes	Yes	No	No	No	No	Yes
DOC++	Yes	No	No	Yes <sup>7</sup>	Yes	Yes <sup>7</sup>	No	No	No
Document! X	Yes	Yes	No						
Duxygen	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Epydoc	Yes	No	No	Yes	Indirecty <sup>5</sup>	Indirecty <sup>5</sup>	No	No	No
ForgeDoc	Yes	Yes	No						
fpdoc	Yes	Yes	Yes	No	Yes	No	Yes	No	Yes
GenHelp	Yes	Yes	Yes	No	No	No	No	No	No
Harddock	Yes	Yes	No	No	No	No	No	Partial	No
HeaderDoc	Yes	No	No	No	No	No	Yes	No	Yes
Help Generator	Yes	Yes	Yes	No	No	No	No	No	Yes
HyperSQL	Yes <sup>8</sup>	No	No <sup>8</sup>						
IDLdoc	Yes	No	No	No	Yes	No	No	Yes	Yes
Imagix 4D	Yes	No	Yes	No	No	No	No	No	No
Javadoc	Yes	Indirecty <sup>3</sup>	Indirecty <sup>6</sup>	Indirecty <sup>2</sup>	Indirecty <sup>4</sup>	Indirecty <sup>6</sup>	Indirecty <sup>2</sup>	Indirecty <sup>6</sup>	Indirecty <sup>2</sup>
JGrouseDoc	Yes	No	Yes						
JSDoc	Yes	No							
JsDoc Toolkit	Yes	No	Yes+JSON						
KDOC	Yes	No							
Natural Docs	Yes	No							
NDoc	Yes	Yes	No						
phpDocumentor	Yes	Yes	No	Yes	No	No	No	Yes	Yes

## Quelques références

- Comparaison des outils de génération de documentation :  
[en.wikipedia.org/wiki/Comparison\\_of\\_documentation\\_generators](http://en.wikipedia.org/wiki/Comparison_of_documentation_generators)
- Doxygen :
  - ▶ Doxygen (officiel) :  
[www.doxygen.org](http://www.doxygen.org) <-> [www.stack.nl/~dimitri/doxygen/](http://www.stack.nl/~dimitri/doxygen/)
  - ▶ Manuel d'utilisation Doxygen :  
[www.stack.nl/~dimitri/doxygen/manual.html](http://www.stack.nl/~dimitri/doxygen/manual.html)
  - ▶ Wikipedia :  
[fr.wikipedia.org/wiki/Doxygen](http://fr.wikipedia.org/wiki/Doxygen)
  - ▶ Initiation à Doxygen pour le C++ :  
[franckh.developpez.com/tutoriels/outils/doxygen/](http://franckh.developpez.com/tutoriels/outils/doxygen/)
- javadoc
  - ▶ officiel :  
<http://docs.oracle.com/javase/1.5.0/docs/tooldocs/windows/javadoc.html>
  - ▶ Wikipedia :  
[fr.wikipedia.org/wiki/Javadoc](http://fr.wikipedia.org/wiki/Javadoc)

**FIN**

**Merci de votre attention**

Florent Langrognat