

Doxygen howto

1. Preface

A lot o project documentations have the same problem: they are not actual. Function names may have changed, complete interfaces are gone or were marked as deprecated or the complete design has changed. One way to avoid an obselete docu is to generate the call reference ans design document directly from the source code.

Java and python have integrated tools to generate documentation. For c++ projects you can use doxygen. This document will show you the basic ideas of doxygen and how to use it in your actual project.

2. Basic idea of doxygen

To get an overview over functions the most common developers write comments. This comments should make the code mush more readable and should increase the reusable of the software. Doxygen creates a documentation by reading this comments and builds a complete overview over the file structure and the existing classes, functions and namespaces.

The advantage of this method should be obvious: The developer must actualize the comments any time he will refactorize his code base. To actualize an external documentation is an unpopular task.

3. Preconditions for doxygen

Enshure that doxygen is installed on your computer. You can get the source and binaries at [3]. To start doxygen you can use the wizard or or console. At first you need a configfile:

```
> doxygen -g <config-file>
```

Doxygen will create a basic template file where you can change your own project. The following keys are the most important ones:

```
PROJECT_NAME      = <Name_of project documentation>
OUTPUT_DIRECTORY = <Output path for documentation>
OUTPUT_LANGUAGE   = <Used language>
INPUT             = <Path to the sources of your project>
FILE_PATTERNS     = File extention, for example .h,.cpp for a c++ project
```

After specifiying you project settings you can create your documentation by typing:

```
> doxygen <Config-File>
```

4. Doxygen command tags

To generate a doxygen documentation you must use special doxygen tags in your documentation.

To show doxygen that special format commands are following use the c++ comments in those special form:

- 1.) /**
 * ...Text...
 */
- 2.) /*!
 * ...Text...
 */
- 3.) /*!

```
    ...Text...
*/
```

This types a derived form the c and java comment stile. You can also use c++ comment block for doxygen:

- 1.)

```
///
/// ...Text
///
```
- 2.)

```
//!
//! ...text...
//!
```
- 3.)

```
////////////////////////////////////
/// ...Text...
////////////////////////////////////
```

If you want to add special detailed infos about a function or a method use the brief tag:

```
/*! \brief Description
 *     more infos
 *
 * Much more detailed infos are following here
 */
```

If you want to aboid the ! symbol you can use the following way:

```
/// Brief Description.
/** Much more detailed infos are following after the point symbol. */
```

To add a reference use the sa-tag:

```
\sa Parent()
```

Write parameter documentation with the param-tag:

```
/// A arbitrary method.
/*!
    Much more detailed infos are following here
    \sa Parent()
    \param pa1 Parameter 1
    \param pa2 Parameter 2
*/
```

To document the several c/c++ statements doxygen supports a lot of special tags like:

```
\struct    ... Documentation for a struct statement
\union     ... Documentation for an union statement
\enum      ... Documentation for an enum statement
\fn        ... Documentation for a function
\var       ... Documentation for a simple variable / attribute
\def       ... Documentation for a #define statement
\file      ... Documentation for a code file
\namespace ... Documentation for a c++ namespace
```

5. Remarks

To get the complete doxygen manual and a reference with all supported stags follow [3].

If you have any questions, fixes or comments just write me an email ([4]).

6. Links

[1] <http://www.gnu.org>

[2] <http://www.zfx.info>

[3] <http://www.stack.nl/~dimitri/doxygen/>

[4] sir_kimmi@gmx.de