

Doxygen

Jeremy Schultz
University of Lethbridge
Software Engineering (CPSC3720)

February 7, 2008

What is Doxygen?

Doxygen

- A Documentation generator
- It uses comments from source code to create documentation (Similar to Javadoc)
- It supports C++, C, Java, Objective-C, Python, IDL (Corba and Microsoft flavors), Fortran, VHDL, PHP, C#, and to some extent D
- It create documentation in HTML, \LaTeX , RTF (MS-Word), PostScript, hyperlinked PDF, compressed HTML, and Unix man pages
- Documentation available at <http://www.stack.nl/~dimitri/doxygen/download.html#latestman>

Create a config file

- There are two ways to create a config file
 - 1 Use doxywizard
This is a GUI front-end that can create, read and write a config file
 - 2 Use `doxygen -g filename`
`# doxygen -g Doxyfile`
- If no filename is used, the default is Doxyfile
- The file consists of
`TAGNAME = VALUE`
or
`TAGNAME = VALUE1 VALUE2 ...`

Running doxygen

- On the command line simply run `doxygen`
`# doxygen`
- Once again with no filename, it assumes Doxyfile is the config file
- It will create the different outputs specified in the config file
 - HTML** Goto the html output directory and open `index.html`
 - LaTeX** Goto the latex output directory and run **make**.
If `USE_PDFLATEX` is set to no, then `make` will create `refman.dvi`. This can be used to create a PostScript file or simply run **make ps**
If `USE_PDFLATEX` is set to yes, then **make** will create a pdf. For best results set `PDF_HYPERLINKS` to yes.

Documenting the source

There are 3 flavours to mark a comment block for doxygen.

- 1 **JavaDoc style**, which are like C-style comment blocks

```
/**
 * .. text ..
 */
```

- 2 **Qt style**, which are like C-style comment blocks but with an exclamation mark

```
/*!
 * .. text ..
 */
```

- 3 **A block of at least two C++ comment lines**, where each line starts with an additional slash or exclamation mark

```
///
/// .. text ..
///
or
///!
///! .. text ..
///!
```

Basic Examples (pt2)

File

```
#include <string>

/** \file foo.c
 * \author John Doe
 * \brief A brief description of a foo.c.
 *
 * A more extensive description of foo.c.
 * Which can be many lines
 */
```

Class

```
/** \brief A brief description of a BarClass.
 *
 * A more extensive description of BarClass.
 */
class BarClass {
```

Basic Examples

Function

```
/** \brief A brief description of a foobar().
 *
 * A more extensive description of foobar().
 * \param aParameter A brief description of aParameter.
 * \return A brief description of what foobar() returns.
 */
int foobar(string aParameter);
```

Variables

```
int num; /**< Brief description of num. */
```

or

```
/** \brief A brief description of num.
 *
 * A more extensive description of num.
 */
int num;
```

Basic Examples (pt3)

Lists

```
/**
 * Text before the list
 * - list item 1
 *   - sub item 1
 *     -# sub sub item 1 (numbered list)
 *     -# sub sub item 2 (numbered list)
 *     .
 *     The dot above ends the sub sub item list.
 *     More text for the first sub item
 *     .
 *     The dot above ends the first sub item.
 *     More text for the first list item
 *   - sub item 2
 *   - sub item 3
 * - list item 2
 * .
 * More text in the same paragraph.
 *
 * More text in a new paragraph.
 */
```