

Generating Documentation with Doxygen

CS 2ME3/SE 2AA4

Steven Palmer

Department of Computing and Software
McMaster University

January 9 - 13

Outline

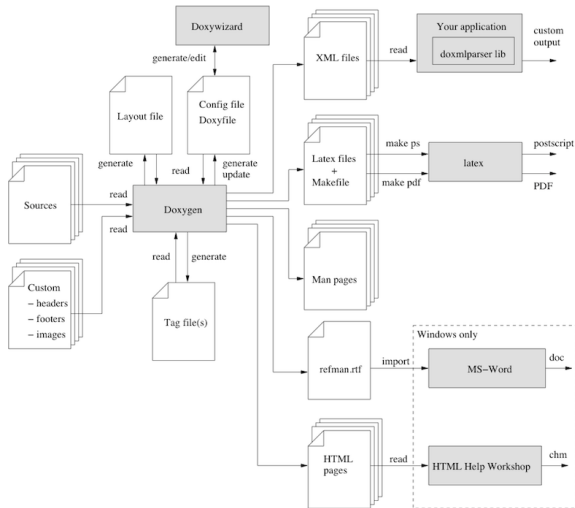
- 1 What is Doxygen?
- 2 Prerequisite Software
- 3 Using Doxygen
 - Doxygen Style Comments
 - Building the Documentation

What is Doxygen?

- Doxygen is a tool used to generate documentation for code modules/classes.
- Comments with special syntax are used in source files to mark information that Doxygen should use.
- When Doxygen is run, it extracts the marked information from the source file(s) and compiles it into selected output formats.

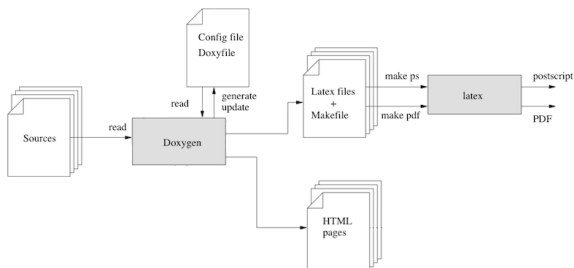
└ What is Doxygen?

Doxygen Information Flow



└ What is Doxygen?

Doxygen Information Flow



- We will focus on the generation of LaTeX and HTML documentation.

Installing TeX Distribution

- You will need a TeX distribution.
 - TeX Live is recommended.
 - For Ubuntu-based Linux distributions you can install via `apt-get install texlive-full`.
 - For other operating systems see <https://www.tug.org/texlive/> for installation details.

Installing Graphviz

- Doxygen uses dot, which is part of the graphviz suite of graph visualization tools.
 - For Ubuntu-based Linux distributions you can install via `apt-get install graphviz`.
 - For other operating systems see <http://www.graphviz.org/> for installation details.
 - Depending on your operating system, you may need to add the graphviz bin folder to your path so that dot can be accessed via the command-line.

Installing Make

- You will need `make` to build a pdf from the Doxygen LaTeX output.
 - `make` should be available by default on Linux systems.
 - If you are using OS X and `make` is not available, you will need to install the Command Line Tools package provided by Apple.
 - If you are using Windows, you should install the MinGW environment (<http://www.mingw.org/>). Make sure to add the MinGW bin folder to your path so that `make` is runnable from the command-line.

Installing Doxygen

- Finally, you will of course need to install Doxygen itself.
 - For Ubuntu-based Linux distributions you can install via `apt-get install doxygen`.
 - For other operating systems see <http://www.stack.nl/~dimitri/doxygen/download.html> for installation details.

Doxygen Comments in Python

- Doxygen comments in Python use the following simple structure:

```
## @command1 args
#  @command2 args
#  @command3 args
      ⋮
#  @commandn args
```

- In general, a Doxygen comment block directly precedes either a class definition, a function definition, or a field definition.

Documenting Classes

- Python classes are documented as follows:

```
## @brief A brief description of ClassX  
#  @details A more detailed description of ClassX  
class ClassX:  
    ...
```

- Depending on the complexity of the class, @details may not be necessary.

Documenting Functions

- Python functions are documented as follows:

```
## @brief A brief description of methodX
# @details A more detailed description of methodX
# @param p1 A description of parameter p1
# @param p2 A description of parameter p2
# @return A description of the returned value
def methodX(p1, p2):
    ...
    return x
```

- Depending on the complexity of the function, `@details` may not be necessary.
- There should be an `@param` entry for every parameter of the function (possibly none). The parameter `self` in class functions should be omitted.
- `@return` is not necessary for void functions.

Documenting Fields

- Python fields are documented as follows:

```
## A brief description of fieldX  
fieldX = ...
```

Example

- See Box3D.py for a small example of Python code with Doxygen style comments.

Additional Commands

- The Doxygen snippets given in this tutorial as well as the Box3D.py example file provide the basics for documenting your code.
- Sometimes you may want to use additional commands to capture more details in your documentation.
- Consult <http://www.stack.nl/~dimitri/doxygen/manual/commands.html> for the full listing of available documentation commands and descriptions.

Configuration File

- Producing documentation for a given set of source files with Doxygen requires a configuration file.
- A default configuration file can be generated via the command line with:

```
doxygen -g configFileName
```

- configFileName can be whatever you want.
- The configuration file needs to be edited to define your project.

Configuration File

- Of particular importance for a new configuration are the `PROJECT_NAME` (line 35) and `INPUT` (line 774) fields.
- `PROJECT_NAME` should be replaced with the name of the program you are documenting.
- `INPUT` should list all of the source files you wish to be included in the documentation. Alternatively, you can list a directory as `INPUT` and use `FILE_PATTERNS` (line 799) to determine which files will be included.
- There are several other options you can use to customize your generated documentation. Refer to <http://www.stack.nl/~dimitri/doxygen/manual/config.html>.

Document Generation

- Documentation is generated using the following command:

```
doxygen configFileName
```

- HTML and LaTeX documentation are default generated outputs.
- By default, the HTML documentation will be found in a new directory called html: look for index.html.
- The LaTeX documentation will be found in a new directory called latex. This folder will contain a makefile – you must call make to generate a pdf which will be called refman.pdf by default.

Reference

- Refer to the Doxygen documentation (<http://www.stack.nl/~dimitri/doxygen/manual/index.html>) for further details about using Doxygen.