Installing the Doxygen Documentation Generator for Eclipse C/C++ IDE

Philip Gust
2017-01-16

Modern development environments provide a way for programmers to comment their code in a way that allows external documentation to be automatically generated from the comments. These document generators rely on a certain commenting style and structure that can be processed. Generating useful documentation from structured comments (JavaDoc) is a well-known feature of the Java programming language that enables other developers to easily understand the purpose of classes, methods and declarations through the external documentation.

Although no such capability is part of C or C++, there are several external document generators available that offer the the same benefits to C or C++ programs. We will use an external documentation generator called Doxygen to generate external documentation for C and C++ programs based on structured comments on methods and other declarations. Doxygen is a popular document generator that supports most of the same comment styles that are available through JavaDoc. This will enable you to learn and use a style of comments across C, C++, and Java applications that you write for class projects.

Requirements

To install Eclipse, you will need to already have the Eclipse IDE and the C/C++ Development Tools (CDT) already installed. Doxygen is integrated into Eclipse through a plugin that enables its options to be configured.

Download and Install Doxygen

The first step is to download and install the Doxygen document generator. There are versions for MacOS, Windows, and Linux

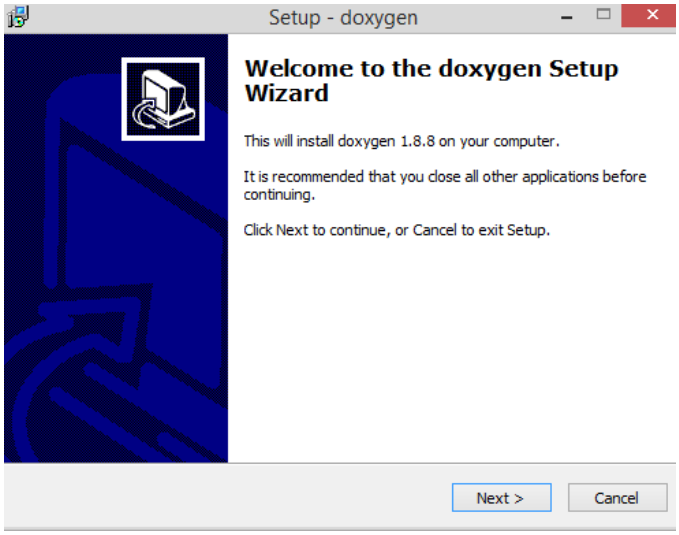
MacOS

Download the MacOS version of Doxygen from http://www.stack.nl/~dimitri/doxygen/download.html. When you open the downloaded .dmg file, you should see a new window with the “Doxygen.app” icon and a Readme file. Move the Doxygen.app to your applications folder, */Applications*.

Doxygen document generator for MacOS is now ready for Eclipse integration.

Windows

Download the Windows Doxygen from http://www.stack.nl/~dimitri/doxygen/download.html. When you run the downloaded *.exe*, you will see an installer wizard and follow the instructions.



Doxygen document generator for Windows is now ready for Eclipse integration.

Linux

Download the Linux-version of Doxygen from http://www.stack.nl/~dimitri/doxygen/download.html. Open a shell and run these commands:

```
tar -xzvf doxygen-1.8.8.linux.bin.tar.gz
cd doxygen-version
./configure
```

```
make
make install
```

If you see an error message when executing that last command:

> /usr/bin/install: cannot stat ‘bin/doxytag’: No such file or directory

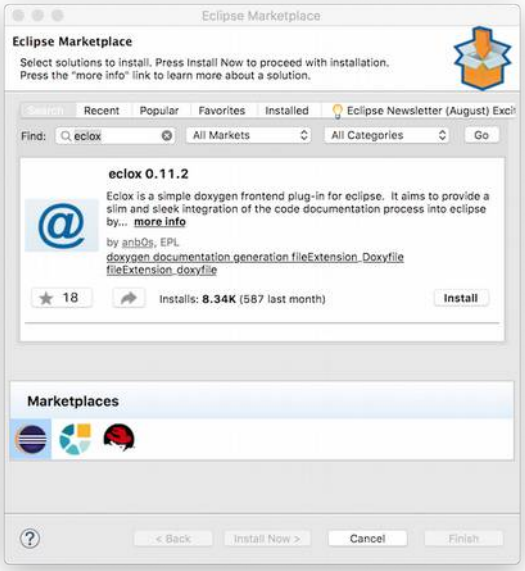
do not worry about it. It is a well-known issue that you can read about here:

http://stackoverflow.com/questions/15020691/installation-of-doxygen-and-error-in-make-command

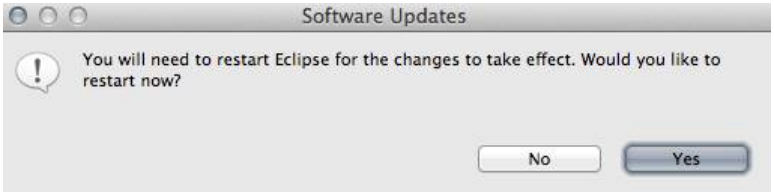
Doxygen document generator for Linux is now ready for Eclipse integration.

Integrating Doxygen with Eclipse

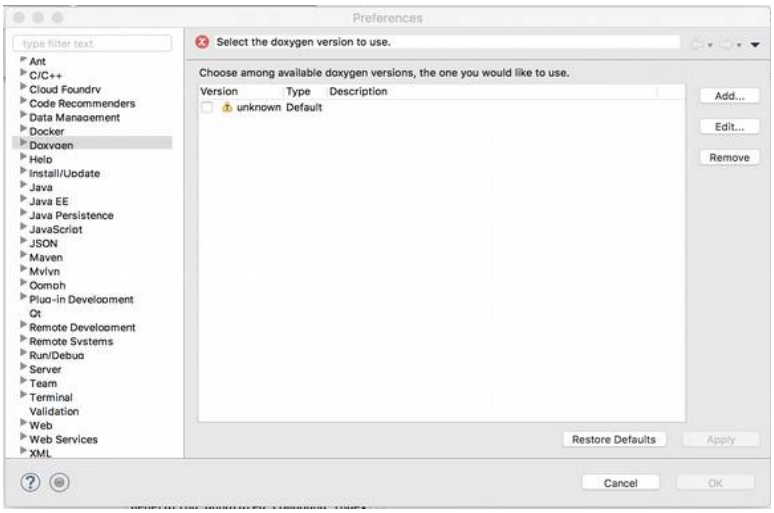
Integrating Doxygen with Eclipse requires installing an integration plugin that provides a user interface to Doxygen within Eclipse. To do this start Eclipse and navigate to *Help > Eclipse Marketplace* and search for ‘eclox’.



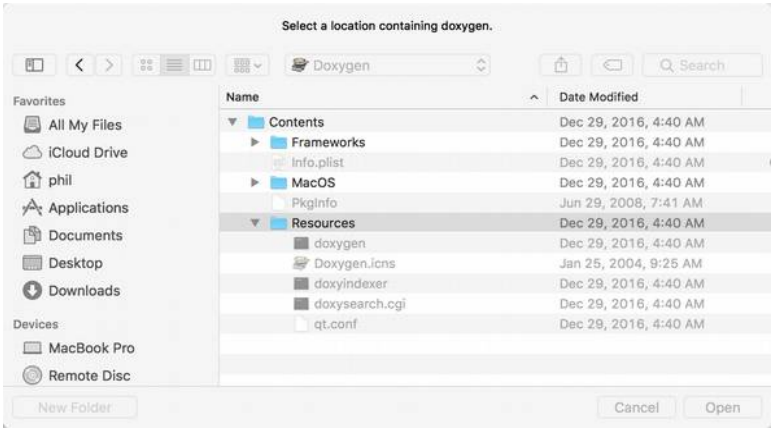
Select the “Install” button to begin the installation. A dialog will ask you to restart Eclipse to complete the integration. Select “Ok.”



Once Eclipse restarts, open the select the new “Doxygen” item in the resource tree.Installing the Doxygen Documentation Generator for Eclipse C/C++ IDEInstalling the Doxygen Documentation Generator for Eclipse C/C++ IDE

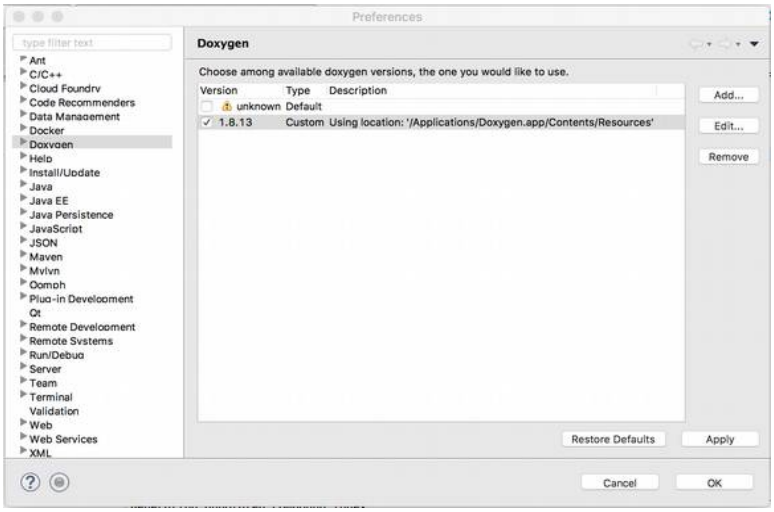


Select “Add...” to add the location where you installed Doxygen. On MacOS navigate to “/Applications/Doxygen” and select “Resources” under “Contents” and select “Open.”



On Windows, navigate to "C:\Program Files\doxygen\bin", and on Linux, navigate to “/usr/local/bin” and select “Open.”

Check the box next to your added directory and select “Apply” and then “Ok.”

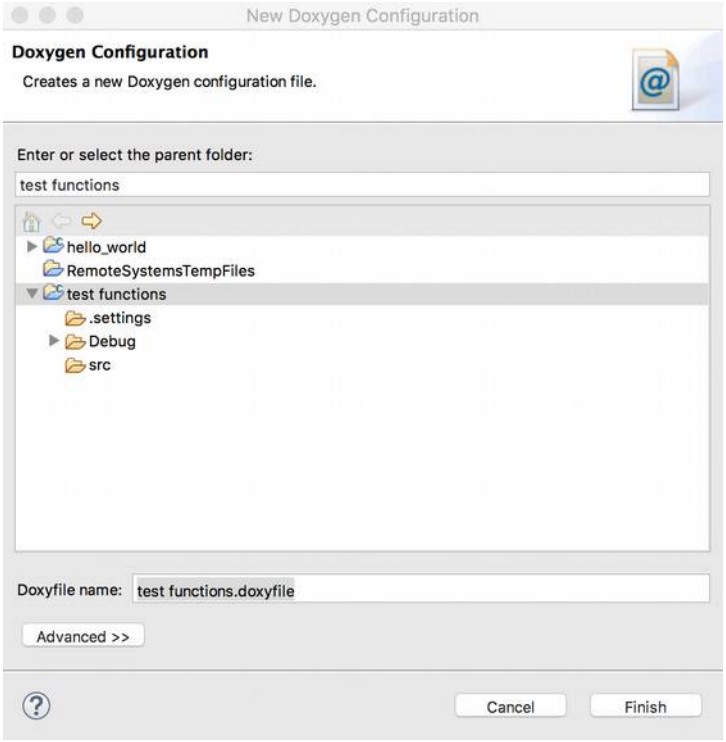


Doxygen is now integrated into Eclipse.

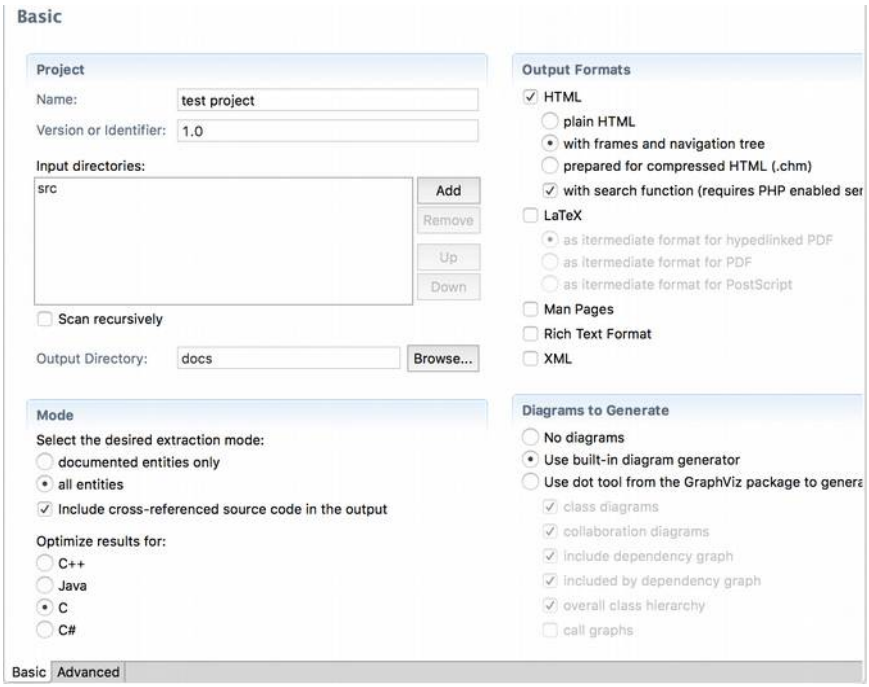
Generating Documentation for a Project

You can add documentation to an Eclipse project by selecting the project from the project tree on the left of the workspace, and selecting the @ button to the left of the debug button. You will be prompted to create and customize a Doxygen configuration file for the project. Select “Yes.”

It’s best to store the file in the root of your project, meaning it should be in the same folder as "Debug" and "src". This will allow version control software like GitHub to backup the Doxygen configuration file and allow multiple developers the ability to generate documentation. In the example below, the root of the “test functions” project is selected. Use the default name <*project_name*>.doxyfile. In this case, “test functions.doxyfile.”

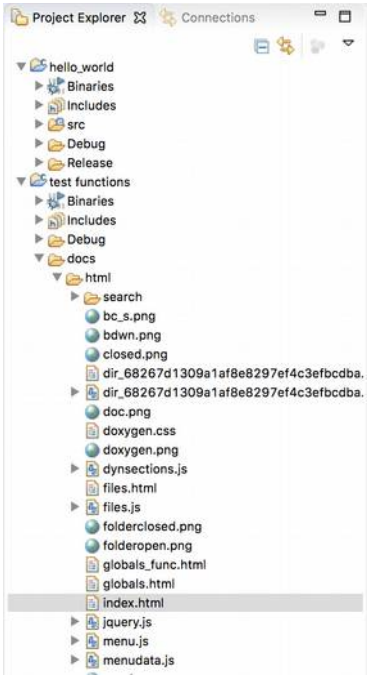


Open the configuration file in the project resource tree to customize its options. For the extraction mode, select “all entities” and optimize results for “C.” For now, we’ll only generate documentation in HTML format that is convenient for browsing on the web, including frames and a navigation tree.

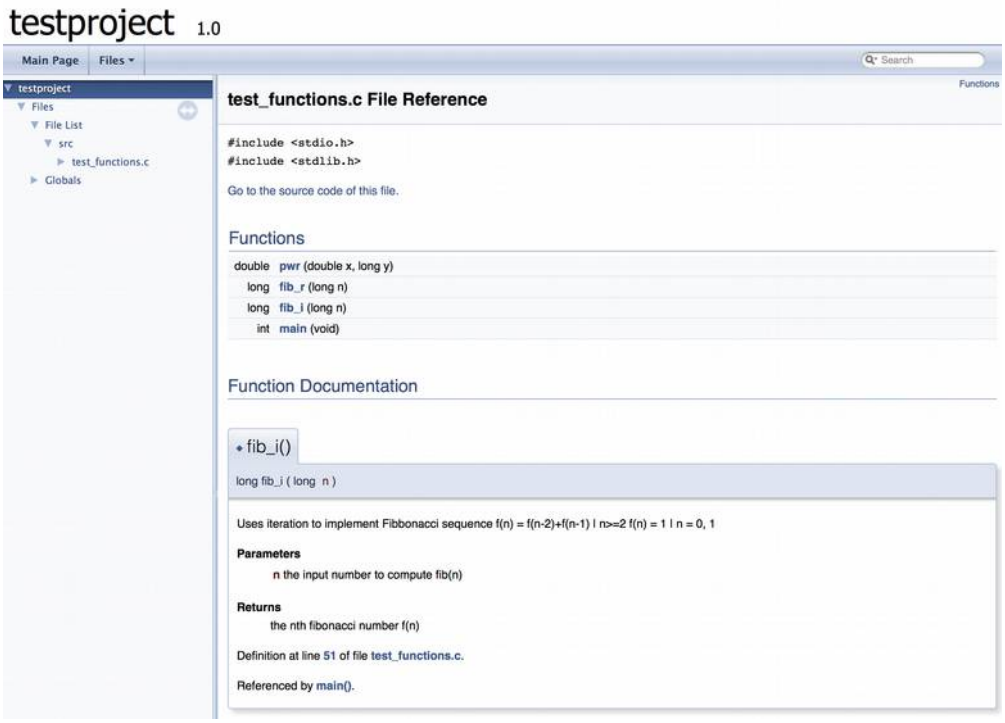


Save the changes by pressing the key sequence "*[CMD]* s" on MacOS or "*[CTRL]* s" on Windows or Linux.

To test the installation, select the project in the workspace project tree and select the  button to the left of the debug button to generate the documentation for the project.



You will see a new "docs" folder with an "html" subfolder." This contains the generated documentation for this project. If you view this HTML in an external web browser, you will see documentation for the functions in the "test_functions.c" file. These files are typically placed on a webs server to document the functions you have written.



Here is a section of commented code for the *fib_i()* function from which the documentation was generated.

```
/**
*  Uses iteration to implement Fibbonacci sequence
 *                  f(n) = f(n-2)+f(n-1) | n>=2,
 *                  f(n) = 1 | n = 0, 1
 * @param n the input number to compute fib(n)
 * @return the nth fibonacci number f(n)
 */
long fib_i(long n) {
        // fn is f(n); fn1 is f(n-1), fn2 is f(n-2)
        long fn = 1, fn1 = 1, fn2 = 1;
        while (n > 1) {
                fn = fn2 + fn1;

                fn2=fn1;
                fn1=fn;
                n--;
        }
        return fn;
}
```

The function comments use standard JavaDoc style notation. The first line begins with a special comment “/**” to indicate a function documentaiton block. The function description comes next, followed by documentation about the parameters (“@param”) and the return value (“@return”). The comments can be formatted with HTML tags.

For more information on options available with the Doxygen documentation generator, see the online manual at http://www.stack.nl/~dimitri/doxygen/manual/index.html.