Introduction to Source Code Documentation Using Doxygen and HTML Help Workshop

Joe Pardue August 22, 2008, be sure and visit www.smileymicros.com .

Doxygen is a very flexible tool for documenting software, so flexible in fact that the first problem is settling on a subset of its features to use. One of Doxygen's main values it that it allows us to keep the documentation in one place tied to the source code. If we change the code, we've got the documentation right there and can change it also.

Lets take a simple view and just do the minimum to get some acceptable documentation generated, then if you feel up to it you can RTFM and make things fancy. We will first generate a directory with HTML files, then we will use a feature of Doxygen to generate files suitable for HTML Help Workshop and then generate a single .chm help file familiar to Windows users.

Putting Doxygen readable comments in the source code

We will look at one way to add Doxygen readable comments to our source code that will generate a useful document. We will create a project in AVRStudio, DoxyTest and write a C program that will do nothing but show how to provide Doxygen readable comments. The source code and AVRStudio project for this program is in the .zip file along with this document.

```
/*********************************************************************
 *
 * DoxyTest.c
 * Version: 1.00, August, 22 2008
 * Copyright 2008 Joe Pardue
 *
 * This code is guaranteed to destroy any machine it is run on.
 * AND it will kill all life in a 1 kilometer radius.
 *
*********************************************************************/

#include <avr\io.h>
#include "DoxyTest.h"

int main()
{
	uint8_t test1 = 1;
	uint8_t test2 = 2;
	uint8_t test3 = 3;

	char strTest1[] = "Hello";
	char strTest2[] = "World";
	char strTest3[]     = "whazzup?";


	while(1)
	{
		DoxyTest1(test1, strTest1);
		DoxyTest2(test2, strTest2);
```

```
            DoxyTest3(test3, strTest3);
      }
}

/** \fn void DoxyTest1(uint8_t test, char strTest[])
 *  \brief Does nothing but test Doxygen comments.
 *
 *  This function doesn't do anything but test Doxygen comments.<br>
 *  Yes that is all it does.<br>
 *  The quick brown fox jumped over the lazy dog.<br>
 *
 *  \param test is a byte for testing
 *  \param strTest is a string for testing.
 */
void DoxyTest1(uint8_t test, char strTest[])
{
      // Test Doxygen comments
}

/** \fn void DoxyTest2(uint8_t test, char strTest[])
 *  \brief Does nothing but test Doxygen comments.
 *
 *  This function doesn't do anything but test Doxygen comments.<br>
 *  Yes that is all it does.<br>
 *  The quick brown fox jumped over the lazy dog.<br>
 *
 *  \param test is a byte for testing
 *  \param strTest is a string for testing.
 */
void DoxyTest2(uint8_t test, char strTest[])
{
      // Test Doxygen comments

}

/** \fn void DoxyTest3(uint8_t test, char strTest[])
 *  \brief Does nothing but test Doxygen comments.
 *
 *  This function doesn't do anything but test Doxygen comments.<br>
 *  Yes that is all it does.<br>
 *  The quick brown fox jumped over the lazy dog.<br>
 *
 *  \param test is a byte for testing
 *  \param strTest is a string for testing.
 */
void DoxyTest3(uint8_t test3, char strTest3[])
{
      // Test Doxygen comments
}
```

Function header document format

If a function header is written as shown above. Doxygen will note the ‘/**’ as the start of a block to interpret. The \fn, \brief, and \param text are interpreted so that the associated text will appear in the resulting HTML file as shown in Figure 6: HTML output for function. Pretty, isn’t it?

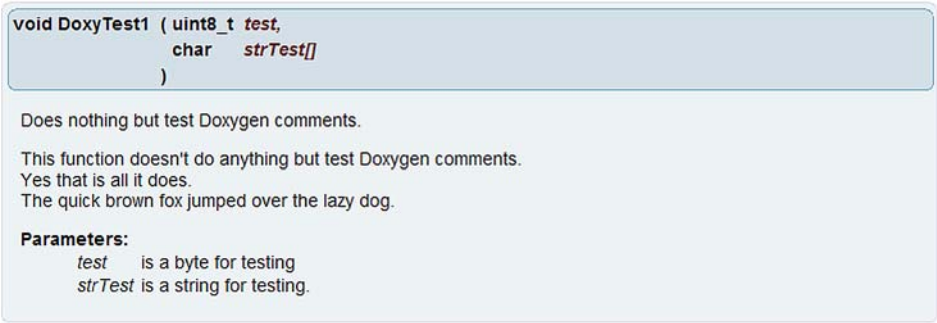

Figure 6: HTML output for function

Creating the HTML Document

You can you can download Doxygen from: www.doxygen.org . Look for: C:\Program Files\doxygen\bin\doxywizard.exe and click it.

You will see the Doxygen GUI frontend shown in Figure 1.

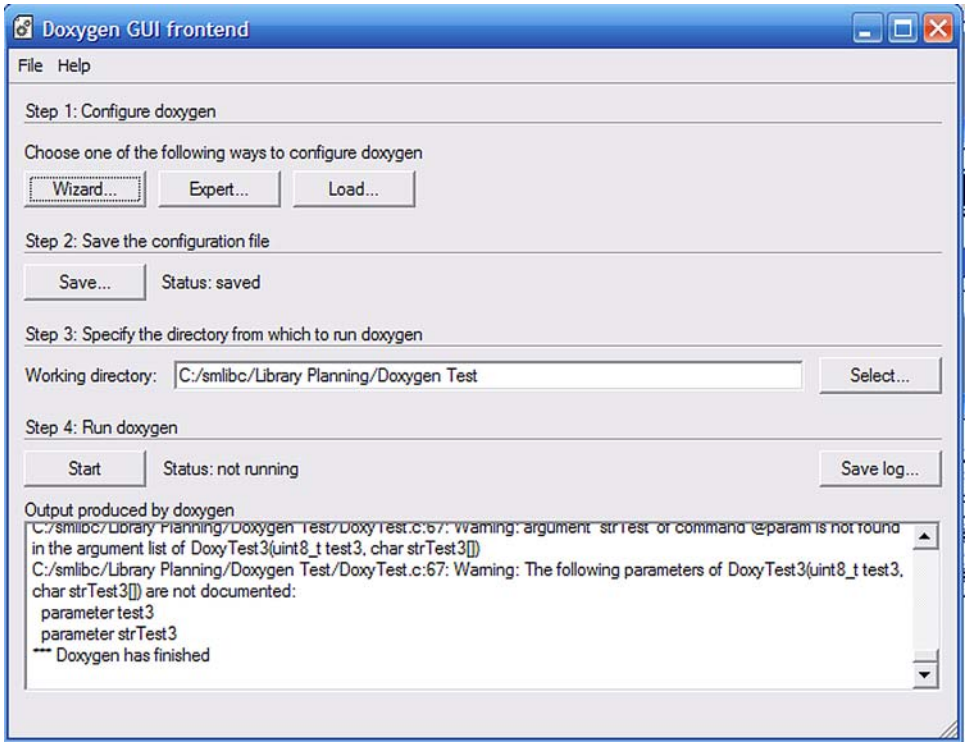

Figure 1: Doxygen GUI frontend

Notice that there are four steps shown. In Step1: Configure doxygen, click the 'Wizard…' button and you will see Figure 2: doxywizard Project.

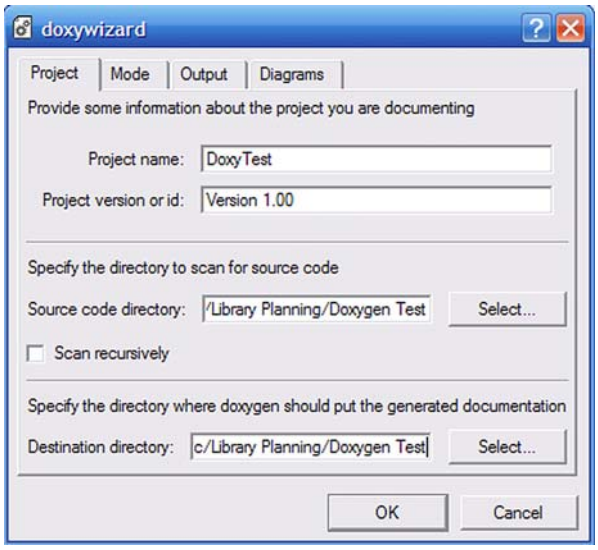

Figure 2: doxywizard Project.

Give the project a name, in our case: DoxyTest, and a version if you so choose. Next specify the directory to scan for source code, and then the directory to put the generated documentation.

Click on the ‘Mode’ tab and you will see Figure 3: doxywizard Mode.

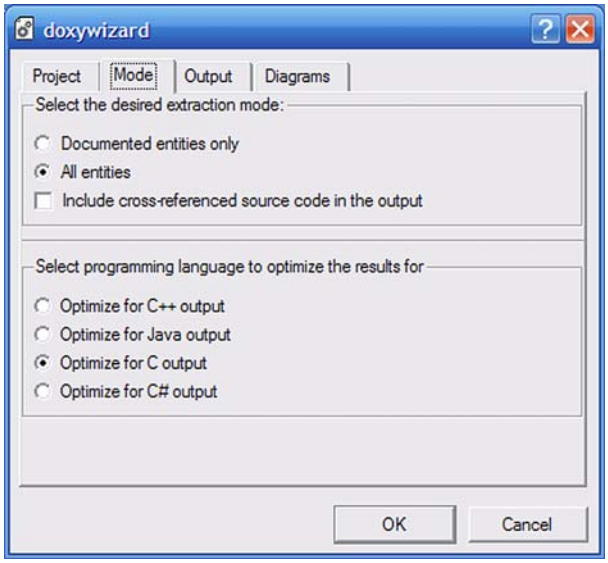

Figure 3: doxywizard Mode

In the ‘Select the desired extraction mode’ box, select ‘All entries’ and ‘Optimize for C output’. Click on the ‘Output’ tab and you will see Figure 4: doxywizard Output. Change the HTML from plain HTML to ‘with frames and a navigation tree’ then uncheck the ‘Latex’ selection.

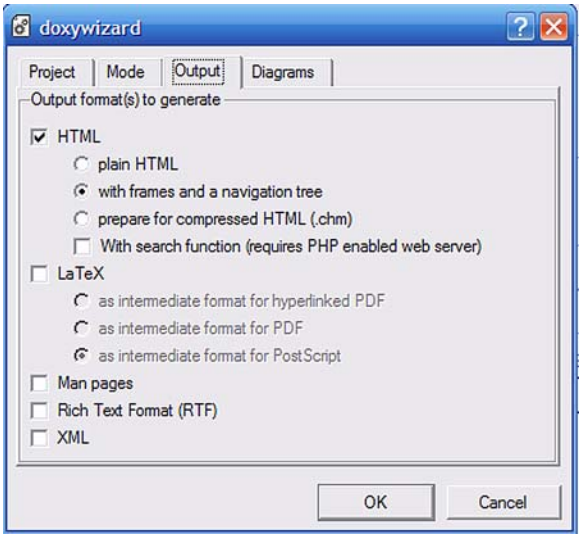

Figure 4: doxywizard Output

Click ‘OK’ to return to the GUI, and follow Step 2: ‘Save the configuration file’ by clicking the ‘Save’ button. In Step3: Specify the directory from which to run doxygen, click the ‘Select’ button and select a place to put it. In Step 4: Run Doxygen, you realize that this is a very intuitive interface and can figure out what to do next without my help.

If everything went okay, you should have a directory with a bunch of html files in it. Click on index.html and your browser will open and you’ll see something like Figure 5: Example browser output.

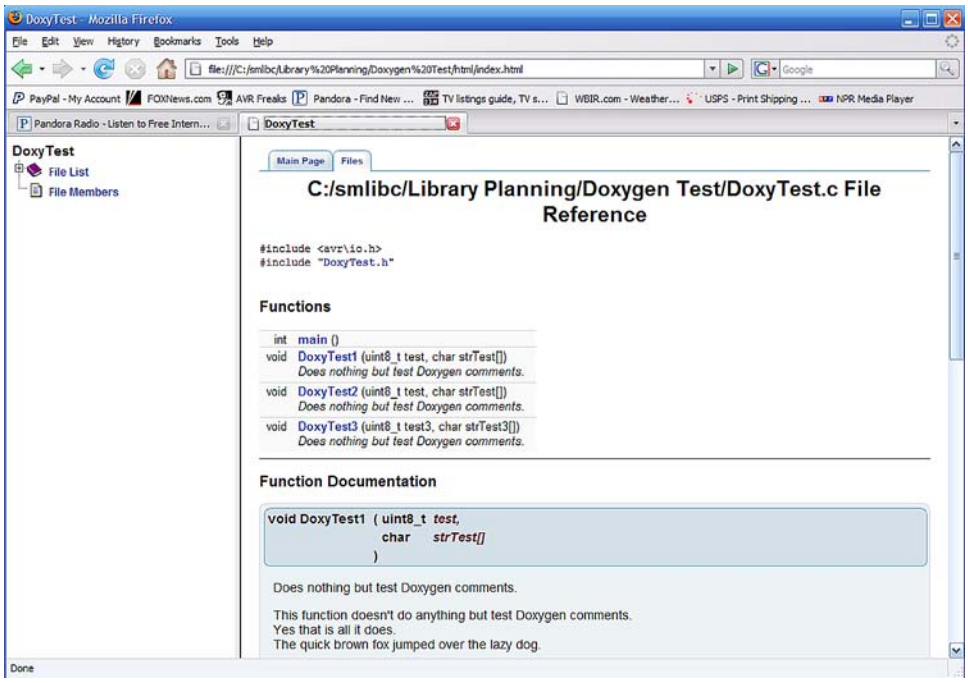

Figure 5: example browser output.

This gives us a simple way to generate documentation from our source code.

Converting to HTML Help .chm file

You will notice that the HTML output is about 40 files and that you need to click on index.html to open the browser to access them. You can convert these files into a single compressed HTML (.chm) file, the typical Microsoft Help file.

Set Doxygen to prepare the HTML output for compressed HTML (.chm) by selecting as shown in Figure 6.

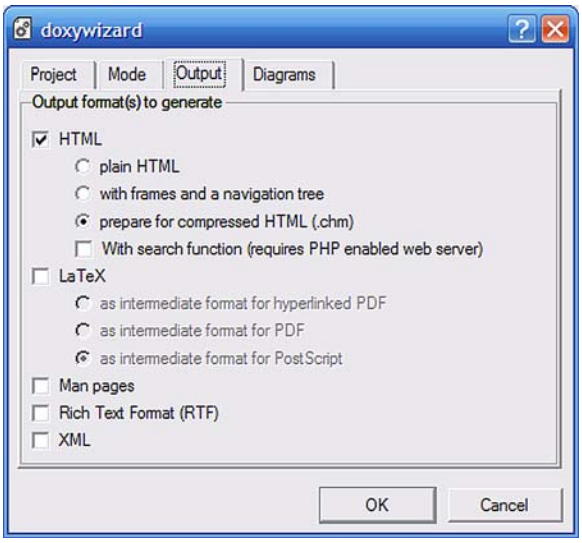

Figure 6: doxywizard Output compressed HTML

Download Microsoft HTML Help Workshop from: http://msdn.microsoft.com/en-us/library/ms669985.aspx . Open HTML Help Workshop and click 'File\Open', browse to select the index.hhp generated by Doxygen which will fill out the IDE as shown in Figure 7: HTML Help Workshop.

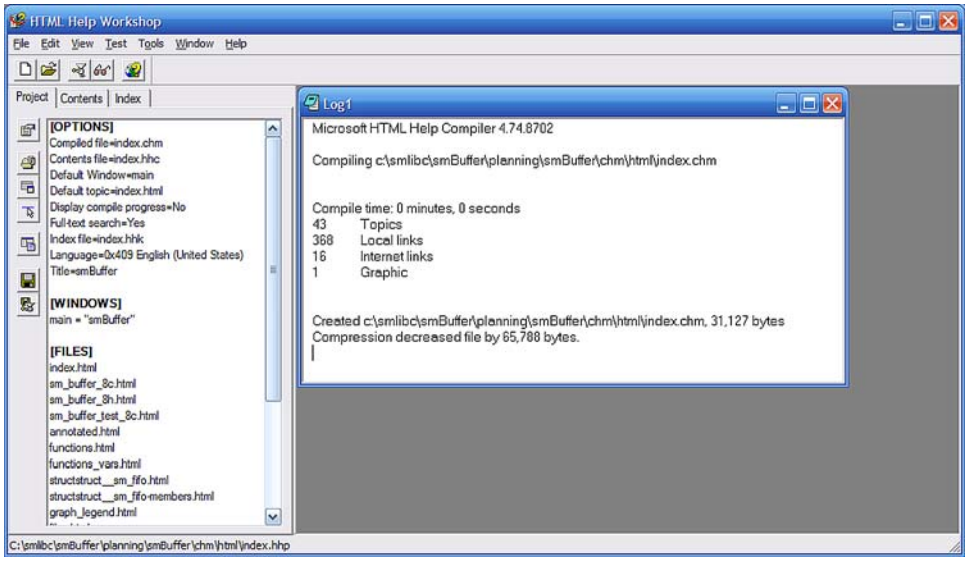

Figure 7: HTML Help Workshop.

Click on the ‘Compile HTML file’ button as shown in Figure 8.

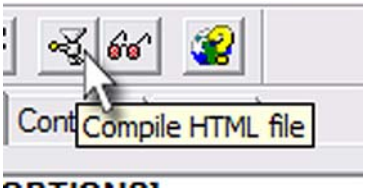

Figure 8: Compile HTML file

The results will be index.chm. Change the name to DoxyTestHelp.chm and click on it to reveal the help file shown in Figure 9.

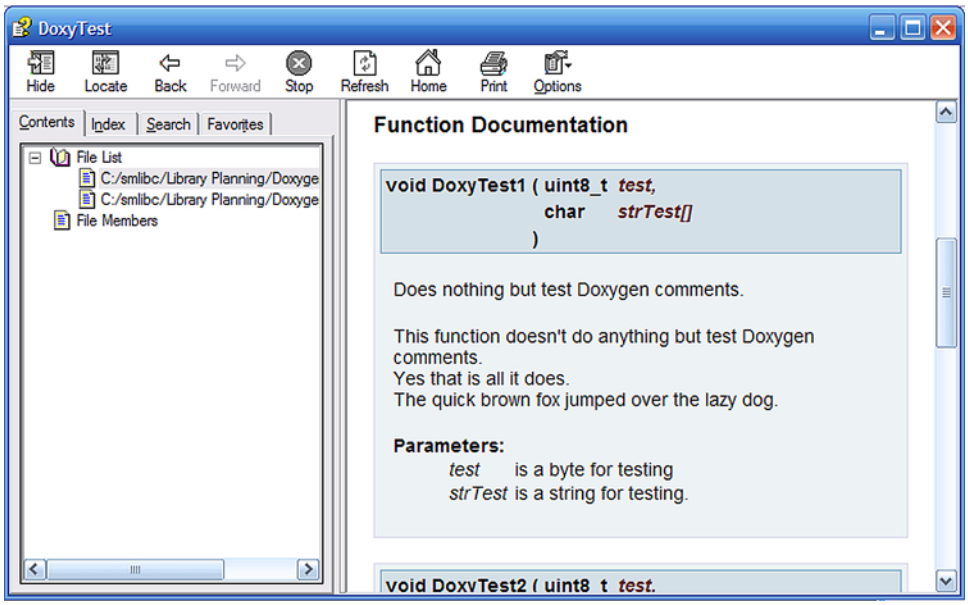

Figure 9: DoxyTest Help file

You will probably want to expand this to full screen. This has the same information in it as the HTML files generated earlier, but it is now all in one file and the Help format provides some search options not available viewing the HTML version in a browser.

Now you have a simple way to generate documents for your source code and you have an introduction to a tool that has many other features you may want to explore. Happy coding!

This tutorial as written by Joe Pardue of www.smileymicros.com - go there and buy something.