

SAS® GLOBAL FORUM 2021

Paper 1077-2021

Introduction to Doxygen

Tom Bellmer

ABSTRACT

A good programming practice is to document your code via the use of comments in the header area. Comments support the understanding of code when you have been away from it or when others need to support it.

Generating external documentation from existing code comments is highly desirable as to do otherwise will inevitably cause things to no longer be synchronized. A free open source utility named doxygen does what is needed and this paper describes how to use it.

INTRODUCTION

To reinforce the concept of adding comments to a program header, I wrote a SAS® stored process to generate a standardized program header (see below). The idea was to be able to dissect the contents and create documentation from those comments. Below is a very simple SAS® macro function that uses the header, accepts three parameters and returns a hex string:

```
/******  
Program: rgbtohex.sas  
Author: Tom Bellmer  
Created: 20180521 @ 15:18:13  
SAS Version: SAS 9.4 (TS1M3)  
Purpose: returns the SAS (CX) hex value from R, G, B values  
Usage: %let x = %rgb(255, 128, 0); /* returns CXFF8000 */  
*****/  
  
%macro rgbtohex(r, g, b);  
  %sysfunc(compress(CX%sysfunc(putn(&r., hex2.))%sysfunc(putn(&g., hex2.))  
    %sysfunc(putn(&b., hex2.))))  
%mend;  
  
/*EOF: rgbtohex.sas */
```

It was my desire to create HTML based documentation with a list of macros in a treeview on the left similar to what is used on the support.sas.com website (see Figure 1).

Unfortunately, that was not an easy endeavor requiring the use of JSON (JavaScript Object Notation) data and some advanced JavaScript code. After that I would still have to apply some custom CSS styles for aesthetics. When I discussed these issues with well-known SAS® guru, Allan Bowe, he recommended a product named doxygen that he had just started using.

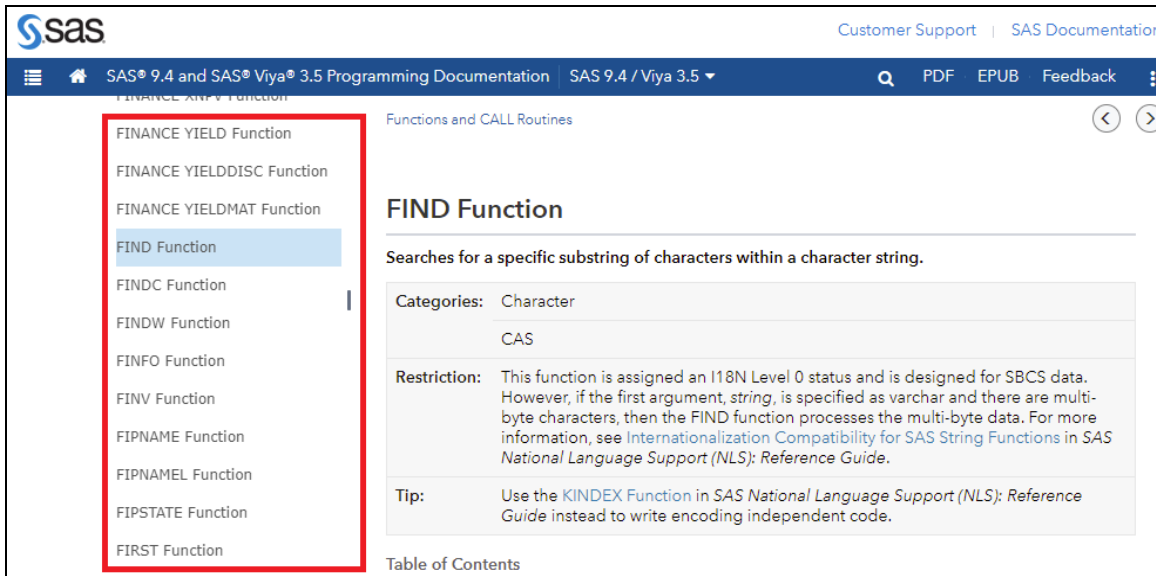


Figure 1 - Treeview from support.sas.com

DOXYGEN SOFTWARE

Doxygen is free software, released under terms of the GNU General Public License version 2 (GPLv2). It is used to scan key annotated comments in source code to create standardized documentation. Dimitri van Heesch created doxygen in 1997 as a cross-platform program written in C++. As a result, you can run doxygen under Linux, MacOS or Windows. Doxygen is the defacto documentation standard for C++ but also supports other languages such as C, C#, Java, Python and Fortran. SAS® is supported by the EXTENSION_MAPPING tag name. This paper uses Windows 10 but most everything will port to other platforms unchanged.

With doxygen you can generate on-line documentation using HTML or in Latex, RTF, PDF and UNIX man pages. This paper will only cover HTML output. In all cases the documentation is extracted directly from the source code, making it much easier to keep things synchronized.

Doxygen got its name from the words *document* and *generator* where document was referenced as docs then dox while generator became gen. According to the [doxygen FAQ](#), van Heesch was looking into lex and yacc, where a lot of things start with 'yy', so the 'y' slipped in to make it more pronounceable (as docs-ee-gen with a long 'e').

INSTALLATION

The first step is to install doxygen by visiting the <https://www.doxygen.nl/download.html> page for installation on your platform. At this time the latest release is version 1.9.1 (released

January 8, 2021). From the downloads page, select sources and binaries and select the appropriate installer for your platform.

Doxygen uses a non-formatted ASCII configuration file (named Doxyfile by default) to store settings. To create the template configuration, run the following from the command line (if you omit the <config-file>, one named Doxyfile will be created): **doxygen -g <config-file>**

CONFIGURATION SETTINGS

Figure 2 provides an overall flow of how doxygen works. The main areas of interest are the config parser and tag file parser.

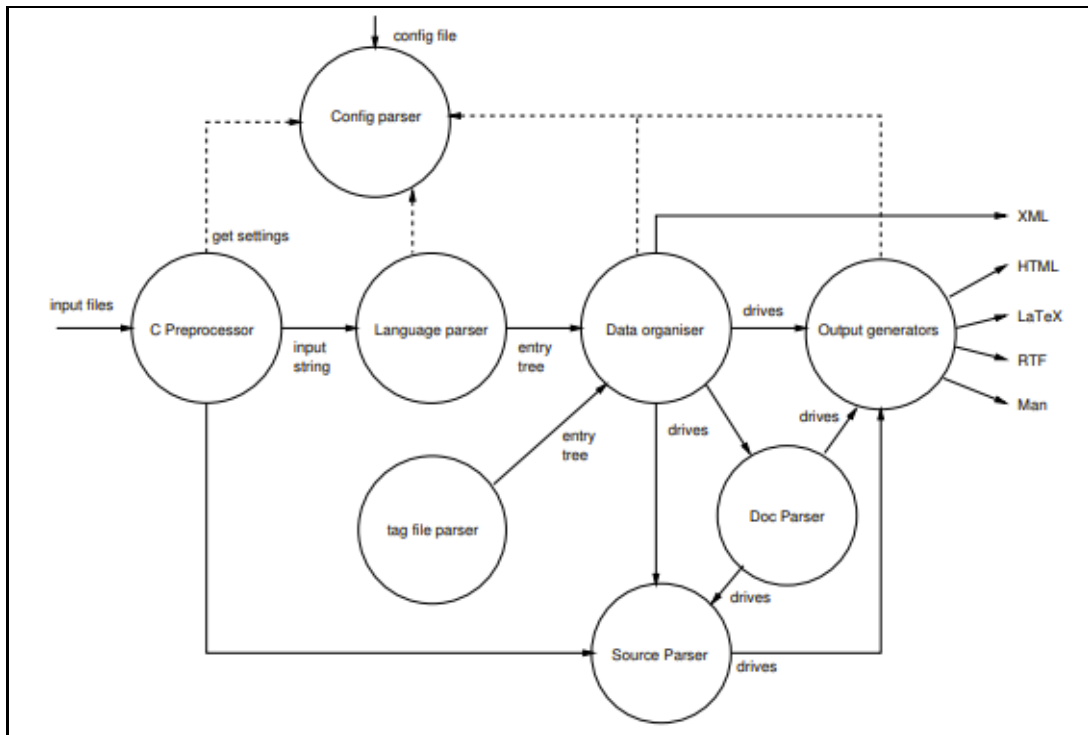


Figure 2- Doxygen flow diagram

The case sensitive uppercase tag names are separated from the associated values by an equal sign (=). Reading in the 1.91 release of the generated doxyfile, there were 302 unique tags. You can edit those settings in a text editor or in **doxywizard**.

Doxywizard is a GUI front-end (see Figure 3) for configuring and running doxygen. You can specify a configuration file by specifying it from the *File | Open...* or *Open Recent* options. The important areas are the **Wizard**, **Expert** and **Run** tabs highlighted in red.

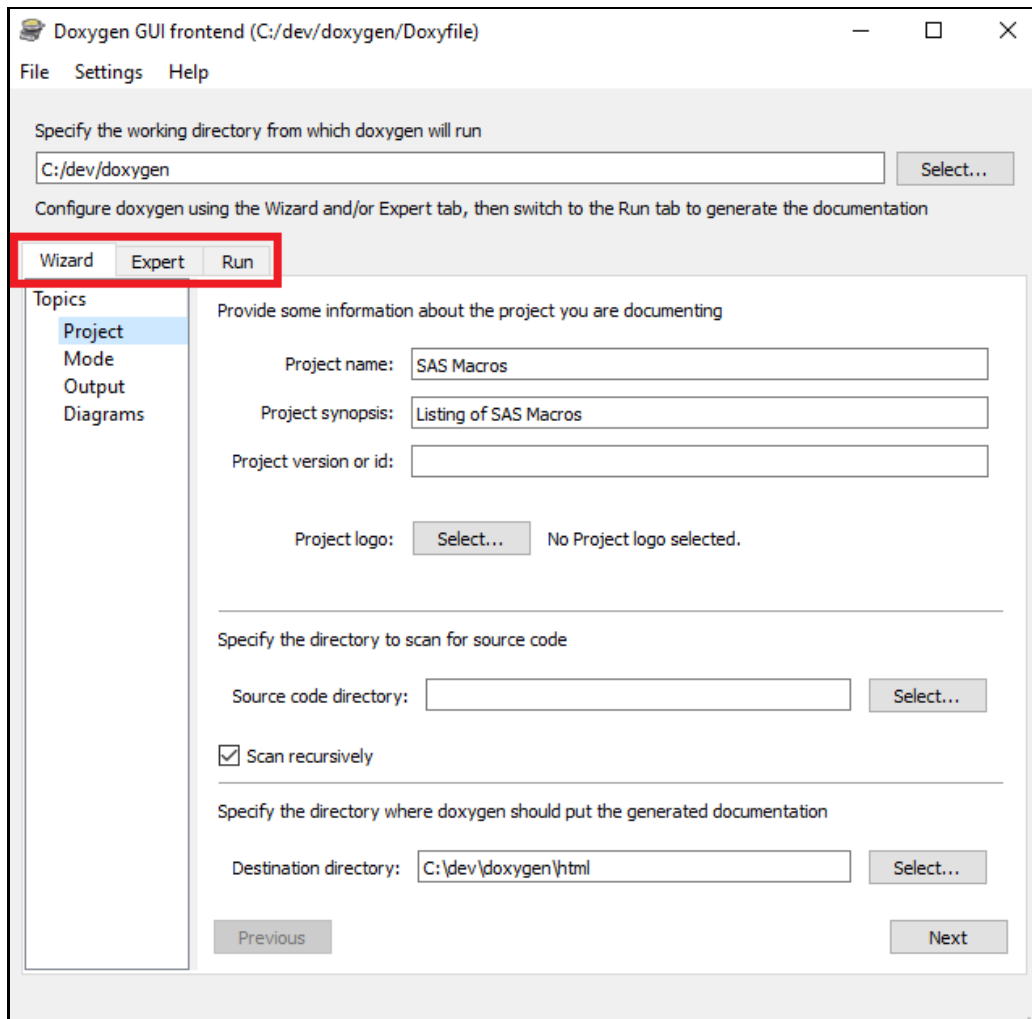


Figure 3- Doxywizard GUI

The **Wizard** tab is used to quickly configure the most important settings while leaving the other options at their defaults. The **Expert** tab provides access to all the available options - it is very detailed and covers all 300+ options. The **Run** tab is used to create the output based on the defined options that have been selected. Read the doxygen manual for complete details on all the options.

SPECIAL COMMANDS

Next we will cover the steps needed in each of the source files to make them compatible with doxygen. This example uses the macro code from the first page and how it is refactored to work within the doxygen environment.

Use the JavaDoc style `/**` (slash followed by two asterisks) in the header section to signify this contains comments to be analyzed by doxygen. Special commands recognized by doxygen can start with a backslash (`\`) or an at sign (`@`).

I like to use the latter and start with a @file that resolves to the name of the file. This is followed by @brief or a one line brief description. Next up is @details to contain a longer, more detailed description of the code. Notice markdown language for the URL with a mouse over title as well as '>' as a blockquote. In order to retain the "%" sign in your output, be sure it is indented not 2 but 4 spaces. That assumes a TAB_SIZE = 2 value in the Doxyfile configuration.

The @param command is used to describe parameters. @return or @returns starts a return value description. An @note command can be used to call out things that users should understand. The @version command can contain whatever you like but I use it to reveal the specific version of SAS® used when it was created. The @author displays just that and the @todo will write out a block quote to make the action stand out as well as create a separate page that is a collection of all @todo occurrences across all input files.

```
/**
  @file
  @brief convert RGB to hex
  @details Usage:

      %put %rgbtohex(255, 128, 0);

  returns:
  > CXFF8000

  Credit Perry Watts' inspiring [SUGI 28 paper](https://bit.ly/3aiKJWo
  "Working with RGB and HLS Color Coding Systems in SAS Software")

  @param r The decimal value (0 to 255) for the color red
  @param g The decimal value (0 to 255) for the color green
  @param b The decimal value (0 to 255) for the color blue

  @returns a hex value preceded by CX used by SAS graphics
  @note values < 0 or > 255 will generate erroneous values
  @version SAS 9.4 (TS1M3)
  @author Tom Bellmer
  @todo Need to create more examples
*/
```

GENERATING OUTPUT

From the doxywizard interface, click on the *Run* tab then *Run doxygen*. You can see the generated file by clicking on *Show HTML Output* or the index.html file located in the destination directory specified in the Wizard tab. See figure 4 for an example of the output.

If you would like to see a more complete site that contains a collection of production ready SAS® macros using doxygen then visit the open source site <https://core.sasis.io/>.

SAS Macros
Listing of SAS Macros

Main Page | Related Pages | Files

SAS Macros
 SAS Macros
 Todo List
 Files
 File List
 allan.sas
 directorylisting.sas
 getattr.sas
 mp_searchdata.sas
 rgbtohex.sas
rgbtohex_doxy.sas

rgbtohex_doxy.sas File Reference

convert RGB to hex [More...](#)

[Go to the source code of this file.](#)

Detailed Description

convert RGB to hex

Usage:

```
%put %rgbtohex(255, 128, 0);
```

returns:

```
CXFF8000
```

Credit Perry Watts' inspiring [SUGI 28 paper](#)

Parameters Working with RGB and HLS Color Coding Systems in SAS Software

- r The decimal value (0 to 255) for the color red
- g The decimal value (0 to 255) for the color green
- b The decimal value (0 to 255) for the color blue

Returns

- a hex value preceded by CX used by SAS graphics

Note

- values < 0 or > 255 will generate erroneous values

Version

SAS 9.4 (TS1M3)

Author

Tom Bellmer

Date

20180521

Todo:

- Need to show more examples

Definition in file [rgbtohex_doxy.sas](#).

Figure 4- Doxygen Output

An even more impressive add on feature is the use of Graphviz and the DOT language to create data driven diagrams, but that is beyond the scope of this paper. However, as a teaser, here is an example of what can be done and is being done in the SAS® based [Data Controller](#) product that can be seen in figure 5 and in this video: <https://vimeo.com/383391622>

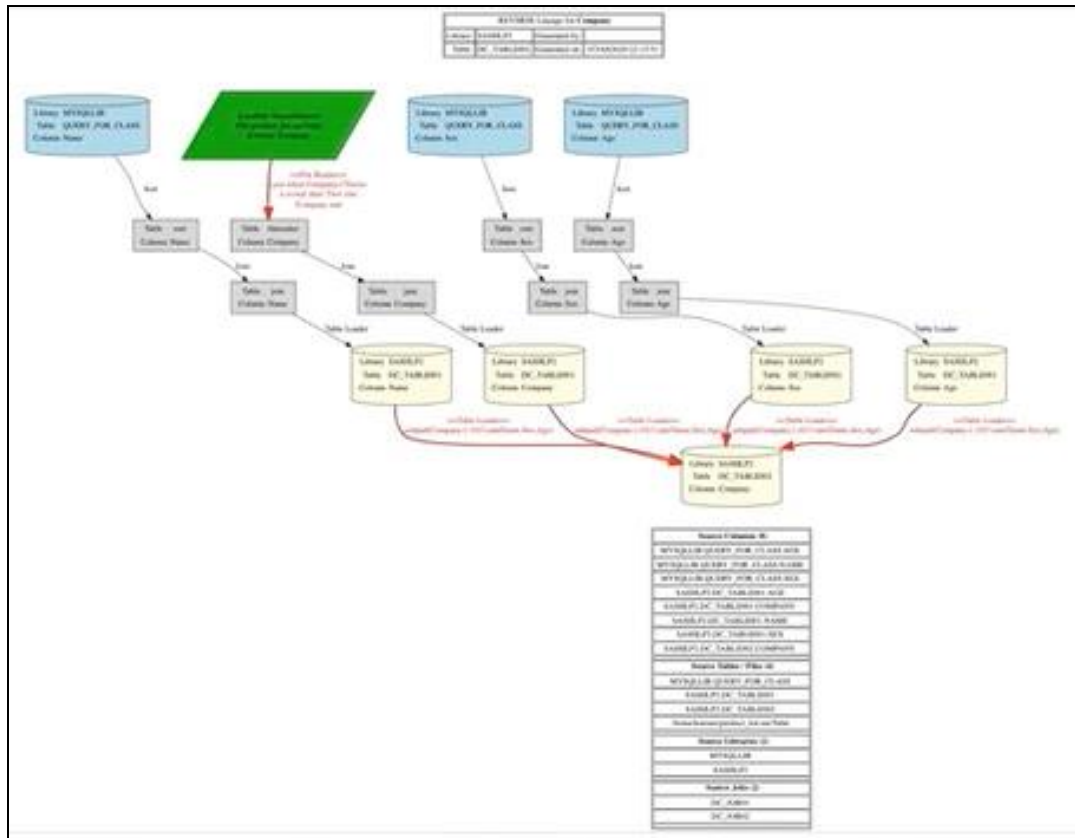


Figure 5- Graphviz output in Data Controller

CONCLUSION

Doxygen is an extremely robust document generator that has been around for nearly 25 years and passed the test of time. There is no need to reinvent the wheel when you have free options like doxygen around.

REFERENCES

- Doxygen. "Generated Documentation From Source Code". Accessed February 20, 2021. <https://www.doxygen.nl/index.html>
- Macro Core. "Production Ready Macros for SAS Application Developers". Accessed February 20, 2021 <https://core.sasjs.io/>
- Data Controller. "Flexible and Secure Data Modification". Accessed February 20, 2021 <https://datacontroller.io/>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Tom Bellmer

thomas.bellmer@gmail.com

<https://KanSAScode.blogspot.com>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

Appendix A - Common Doxygen [Configuration Tag Names](#)

TAG_NAME	Setting	Description
ALPHABETICAL_INDEX	NO	Alphabetical index of all compounds
ALWAYS_DETAIL_SEC	NO	Only create a detailed section if there is something in that section
DISABLE_INDEX	YES	Control top index (tabs) at top of page
ENABLE_PREPROCESSING	NO	Evaluate all C-preprocessor directives
EXCLUDE		Specify files and/or directories that should be excluded from the <code>INPUT</code> source files.
EXTENSION_MAPPING	sas=Java	Specifies the parser to use for a file extension
EXTRACT_ALL	YES	Ensure that all files in <code>INPUT</code> are processed even those without any doxygen special characters
EXTRACT_LOCAL_CLASSES	NO	Include classes in documentation
FILE_PATTERNS	*.sas	Which file extensions to process in folders
GENERATE_LATEX	NO	Create a Latex file?
GENERATE_TREEVIEW	YES	Create a tree-like index should be created
HIDE_FRIEND_COMPOUNDS	YES	Hide friend compounds
HIDE_IN_BODY_DOCS	YES	Hide documentation inside body of function
HIDE_SCOPE_NAMES	YES	Show members with full namespace scope
HIDE_UNDOC_CLASSES	YES	Hide undocumented classes
HIDE_UNDOC_MEMBERS	YES	Hide undocumented members in files
HTML_OUTPUT	doxy	Location of HTML files
HTML_HEADER		User defined HTML header file
HTML_FOOTER		User defined HTML footer file
HTML_STYLESHEET		User defined CSS file
IMAGE_PATH		Specify one or more files or directories

		# that contain images that are to be included in the documentation
INHERIT_DOCS	NO	Will undocumented member inherit
INLINE_INFO	NO	Insert tag for inline members
INPUT		Directory that contains source files
MAX_INITIALIZER_LINES	0	Max # of lines - if 0 it is ignored
PROJECT_BRIEF		Text in title area
PROJECT_NAME		Title of most generated pages
PROJECT_LOGO		Logo or icon max dimensions of max dimension of 55px by 200 px
RECURSIVE	YES	Search subdirectories
REPEAT_BRIEF	NO	Prepend brief before detailed description
SHOW_NAMESPACES	NO	Create namespace page
SHOW_USED_FILES	NO	List files generated at bottom of page
SOURCE_BROWSER	YES	Create list of source files generated
SOURCE_TOOLTIPS	NO	Enable mouse over tooltips
STRICT_PROTO_MATCHING	YES	Handling when parameters do not match
STRIP_CODE_COMMENTS	NO	Hide comments from source
SUBGROUPING	NO	Show subgrouping
TAB_SIZE	2	Tabbing space, min: 1, max: 16
USE_MDFILE_AS_MAINPAGE		Use a markdown file such as README.md
VERBATIM_HEADERS	NO	Copy header for included code

Appendix B - Common Doxygen [Special Commands](#)

Command	Description
@author {list of authors}	Author's name
@b <word>	Displays text in bold
@brief {brief description}	One line brief description
@date {date description}	Display a date
@details {detailed description}	Detailed description
@e <word>	Show the word in italics
@emoji "name"	Displays an emoji using the specified name. Names can be found here: https://gist.github.com/rxaviers/7360908
@file [<name>]	Name of program
@image[{'option'}] <format> <file>	Inserts an image into the documentation. Doxygen will look for images in the IMAGE_TAG tag. Example: @image html application.png
@li {item-description}	Generates a simple list of items
@mainpage [(title)]	Used to customize the index page. The tag USE_MDFILE_AS_MAINPAGE is an alternative
@n	Force a new line
@note {text}	Indented notes
@param ['dir'] <parameter-name> {parameter description}	Starts a parameter description
@return {description of return value}	Description of the return value
@see {references}	One or more cross-references
@todo {what needs to be done}	Adds a TODO section and also creates a separate TODO list
@version {version number}	Show a version
@warning {warning text}	Displays a warning paragraph