

 [Anvil](#) / **bash-doxygen** Public

A doxygen filter for bash scripts


 WTFPL license

 **80** stars    **25** forks

 Star



 Notifications

 **Code**    Issues   8    Pull requests   2    Actions    Projects    Wiki    Sec

master ▾

Go to file

**Damien Nadé** doxygen-bash.sed: hopefully fix indented com... ... ✓ on Feb 25, 2019 ⌚ 45

[View code](#)

☰ README.md



## Releases

build passing

No releases published

# bash-doxygen

## Packages

No packages published

A basic doxygen filter (originally written in GNU sed) allowing you to add inline-documentation to your bash shell scripts.

## Supported shell syntaxes

Contributors 3



- All lines starting with a `##` (without any leading blanks) are passed to doxygen. You can use all the doxygen commands you want in those lines. (see [doxygen documentation](#)).



darnip Darsnit Shan



- Some top-level declarations will be recognized if you use the `declare` primitive:

- `declare -a` for arrays
- `declare -A` for associative arrays
- `declare -i` for integers
- Any other top-level `declare` statement will consider variable is a string.
- Those additional declaration attributes can be combined with `-A/-a/-i/`:
  - `declare -l` will mark the variable as LowerCase
  - `declare -u` will mark the variable as UpperCase
  - `declare -x` will mark the variable as Exported
  - `declare -r` will mark the variable as ReadOnly
- Additionally, declaring a variable with an `export` statement will also be recognized and the variable will be marked as an Exported String.

- Functions declaration will be recognized if all these conditions are met:

- a `## @fn` line is found above the function declaration,
- the function is declared either with or without the non-posix `function` keyword, but always with `()`.
- the body-opening `{` char is on the same line as the `funcname()` instruction.

## How to use it

1. If you do not have a Doxygen configuration file (usually named Doxyfile), you can generate one by simply running `doxygen -g`.
2. Edit the Doxyfile to map shell files to C parser: `EXTENSION_MAPPING = sh=C`
3. Set your shell script file names pattern as Doxygen inputs, like e.g.:  
`FILE_PATTERNS = *.sh`
4. Mention `doxygen-bash.sed` in either the `INPUT_FILTER` or the `FILTER_PATTERN` directive of your Doxyfile. If `doxygen-bash.sed` is in your `$PATH`, then you can just invoke it as is, else use `sed -n -f /path/to/doxygen-bash.sed --`.

**CAREFUL:** If you are a BSD and or a Mac user, you will definitely want to use `gsed` instead of `sed` to make it work.

## Known limitations

---

Yes.

## FAQ

---

Q. Does it actually work ? A. The [bash-argsparse](#) project uses this filter. Check [the result](#). Click on the links. See by yourself.

Q. Is it rock-solid ? A. No.

Q. Do you accept patches ? A. Definitely.

Q. Why is the project named bash-doxxygen while the filter is named doxygen-bash ? A. Yeah, haha. Seriously.

Q. Can I include the `doxygen-bash.sed` file in my own tarball ? A. See the [COPYING](#) file.

Q. Dude. sed ? Seriously ? A. Are you.. Jealous ?

Q. ... ? A. Don't you dare !