



# **DoxyDoxygen User Guide**

*Release 0.57.0*

**20Tauri**

**Feb 05, 2017**



<b>1</b>	<b>Welcome to DoxyDoxygen</b>	<b>1</b>
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Software installation with Package Control . . . . .	3
2.2	Manual software installation . . . . .	3
2.3	License installation . . . . .	3
2.4	EULA (End User License Agreement) . . . . .	4
2.4.1	Licenses . . . . .	4
2.4.2	Description of other rights and limitations . . . . .	4
2.4.3	No warranties . . . . .	4
<b>3</b>	<b>Usage</b>	<b>5</b>
3.1	Create a documentation block . . . . .	5
3.2	Update / wrap an existing documentation block . . . . .	6
3.3	Switch between comment styles . . . . .	7
3.4	Extend a documentation block . . . . .	7
3.4.1	Auto-completion . . . . .	7
3.4.2	Comment continuation . . . . .	8
3.5	Navigate in documentation . . . . .	8
3.5.1	Move to the right column . . . . .	8
3.5.2	Follow references . . . . .	8
3.6	Fold / Unfold comments . . . . .	8
3.7	Translate . . . . .	9
3.8	Generate documentation . . . . .	9
<b>4</b>	<b>Customization</b>	<b>11</b>
4.1	Settings . . . . .	11
4.1.1	Understanding settings . . . . .	11
4.1.2	Settings references . . . . .	12
4.1.3	Translation services settings . . . . .	18
4.2	Add your own doc-style . . . . .	20
4.3	Key bindings . . . . .	20
4.3.1	Key bindings on Windows and Linux . . . . .	20
4.3.2	Key bindings on OS X . . . . .	21
4.4	Commands from the palette . . . . .	21
4.5	Commands from the menu . . . . .	22
<b>5</b>	<b>Glossary</b>	<b>23</b>
<b>6</b>	<b>Appendices</b>	<b>25</b>
6.1	Features Comparison . . . . .	25

6.2	Supported Documentation Tools . . . . .	26
6.3	Supported Languages . . . . .	26
6.4	Surveys . . . . .	28
<b>7</b>	<b>Known issues</b>	<b>29</b>
7.1	Incorrect syntaxes . . . . .	29
7.2	Conflict with others plugins . . . . .	29
7.3	Incompatible Sublime Text versions . . . . .	29
<b>8</b>	<b>FAQ</b>	<b>31</b>
8.1	Can I call DoxyDoxygen from the command-line ? . . . . .	31
8.2	Is it possible to disable default parameter description ? . . . . .	32
8.3	How can I switch to a different <code>preferred_comment_style</code> ? . . . . .	32
8.4	Is it possible to add tags dynamically ? . . . . .	32
8.5	Why there's no alignment on <code>Enter</code> ? . . . . .	33
<b>9</b>	<b>Support</b>	<b>35</b>
9.1	Contact from GitHub . . . . .	35
9.2	Contact from Web Site . . . . .	35
9.3	Contact from Sublime Text . . . . .	35

## WELCOME TO DOXYDOXYGEN

```
1  /*-----**
2  * Calculates the rectangle area.
3  *
4  * @param    {number}    width    The width (in m)
5  * @param    {number}    height   The height (in m)
6  * @param    {boolean}   isUseless Not used
7  * @return   {number}    The rectangle area (in m²).
8  */
9
10 function calcRectangleArea(width, height, isUseless) {
11     return width * height;
12 }
13
```

4. Press Alt+Q to update comment

DoxyDoxygen is a *plug-in* for [Sublime Text](#) that aims to save a lot of time and effort when creating and updating documentation comments in source code.

**How does it work ?**

- Write your code
- Press Alt+Q (or /\*\* + Enter), code is parsed and a skeleton documentation is written for you
- Update your code
- Press Alt+Q, documentation is updated

**DoxyDoxygen can be easily configured to suit your needs.**

- no matter your *programming language*
- no matter your *documentation generator* : ApiDoc, AsDoc, Doxygen, Drupal Api Module, Google Closure, JavaDoc, JsDoc, PhpDocumentor, SassDoc, Sphinx, XmlDoc, YuiDoc...
- no matter your *comment style* : /\*\*, ///  
...
- no matter your preferred layout for tags...

Documentation is generated... Descriptions are written in *your native language*...

And, reading this manual you will discover even more features like *on demand translation*...



## INSTALLATION

### 2.1 Software installation with Package Control

To install it with Package Control:

- Install the [Package Control](#) (if you haven't already)
- Open the *Command Palette* (on Windows or Linux, press `Ctrl+Shift+P`, on OS X, press `Super+Shift+P`)
- Type `install` to get to the command *Package Control: Install Package*
- Type `DoxyDoxygen` and you'll see this package.
- Hit `Enter` to install it.

### 2.2 Manual software installation

**Warning:** Manual installation is not recommended

---

**Tip:** Manual installation can be used to (re)install an old version. But, if you do not like something new, please *let us know* before reinstalling.

---

To install it manually:

- Click the *Preferences* → *Browse Packages* menu to open package directory
- Download the package `DoxyDoxygen.sublime-package` from [released versions on GitHub](#)
- Rename the downloaded file to `DoxyDoxygen.sublime-package`
- Copy it into the package directory
- Restart Sublime Text to complete the installation

### 2.3 License installation

DoxyDoxygen may be downloaded and evaluated for free, however a license must be purchased for continued use.

**Once license is purchased, take the following steps to install it:**

- Go to *Preferences* → *Package Settings* → *DoxyDoxygen*
- Select *Enter License*
- Paste your license key in the text-box appearing in the bottom
- Hit `Enter` to validate it.

When license key is installed successfully, confirmation message will appear.

## **2.4 EULA (End User License Agreement)**

The SOFTWARE PRODUCT (DoxyDoxygen) is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

### **2.4.1 Licenses**

Licenses are per user and valid for use on all supported operating systems. License keys may be used on multiple computers and operating systems, provided the license key holder is the primary user. Businesses must purchase at least as many licenses as the number of people using DoxyDoxygen.

#### **Backup Copies**

You may make copies of the license key and / or DoxyDoxygen for backup and archival purposes.

### **2.4.2 Description of other rights and limitations**

#### **Maintenance of Copyright Notices**

You must not remove or alter any copyright notices on any copy of DoxyDoxygen

#### **Distribution**

You may not distribute or sell license keys to third parties. Licenses will be revoked if distributed or sold to third parties.

#### **Rental**

You may not rent, lease, or lend the license key.

### **2.4.3 No warranties**

DoxyDoxygen is provided as is without any express or implied warranty of any kind, including but not limited to any warranties of merchantability, non-infringement, or fitness of a particular purpose.



### 3.1 Create a documentation block

Start a *documentation block* (usually `/**`) before a declaration, then press `Enter`. The corresponding documentation will automatically be inserted. There are no keyboard shortcuts to memorize.

To be more efficient, you may also press `Alt+Q` (or `Super+Alt+Q` on OS X) after the function definition. A *documentation block* is written for you.

```
1
2  #-----
3  ## @brief      Calculates the area.
4  ##
5  ## @param      weight  The weight
6  ## @param      height  The height
7  ##
8  ## @return     The area.
9  ##
10 calc_area = lambda weight, height: (weight * height)
11
```

Tab

Types are automatically deduced from the code:

```
1
2  /*-----*/
3  * { function_description }
4  *
5  * @param      {number}    count    The count
6  * @param      {Function}  callable The callable
7  */
8  function repeat(count, callable) {
9      for (let i=0; i<count; i++) {
10         callable()
11     }
12 }
```

Guess parameters type  
Tab

Even difficult to analyze programming languages are properly supported:

If a function has a template parameter, a `@tparam` property is automatically added:

And, of course, classes (with template or not) are also supported.

```

1
2  /*-----*/
3  * @brief    Calculates the perimeter.
4  *
5  * @param[in] width    The width
6  * @param[in] height   The height
7  *
8  * @return    The perimeter.
9  */
10 size_t calc_perimeter(size_t width, size_t height);
11

```

Tab

```

1
2  /*-----*/
3  * @brief    Dumps an array.
4  *
5  * @param[in] array    The array
6  *
7  * @tparam    N          { description }
8  */
9  template<int N> void dumpArray(const Array<auto, N>& array)
10 {
11     // ...

```

Tab

## 3.2 Update / wrap an existing documentation block

To update a comment, press Alt+Q (or Super+Alt+Q on OS X). As DoxyDoxygen knows the Doxygen commands, no invalid line break will be inserted.

Even better, with default settings, Alt+Q also reexamine the documented object and detects missing, renamed or moved parameters:

```

1
2  /*-----*/
3  * @brief    Draws a rectangle.
4  *
5  * @param[in] rgb        The rgb color
6  * @param      dst        The destination buffer
7  * @param[in] dst_line_size  The destination line size
8  * @param[in] x          The horizontal coordinate
9  * @param[in] y          The horizontal coordinate
10 * @param[in] w          The width of the rectangle to draw
11 * @param[in] h          The height of the rectangle to draw
12 */
13 static void draw_rectangle(unsigned rgb, uint8_t *dst, int dst_line_size,
14                             int x, int y, int w, int h)
15 {
16

```

Tab

DoxyDoxygen preserves list with hierarchy. On update, spaces before an item are kept. A valid list item is a line that start with -#, -, + or \*.

Listing 3.1: Example of valid list

```

/**
 * @return Error code
 *
 *      - E_OK
 *      - E_ACCESS_DENIED
 *      - E_INTERNAL
 */

```

Listing 3.2: Example of invalid list

```
/**
 * @return Error code:
 *         E_OK
 *         E_ACCESS_DENIED
 *         E_INTERNAL
 */
```

Listing 3.3: Invalid list after an update

```
/**
 * @return Error code: E_OK E_ACCESS_DENIED E_INTERNAL
 */
```

## 3.3 Switch between comment styles

To switch between your preferred *comment styles*, press Shift+Alt+Q (or Super+Shift+Alt+Q on OS X). You can also find more flexible commands in the *Command Palette*.

## 3.4 Extend a documentation block

### 3.4.1 Auto-completion

DoxyDoxygen allows auto-completion. A large set of commands is available,

**Available commands depends of *doc-style*:**

- Commands list for ApiDoc
- Commands list for AsDoc
- Commands list for Doxygen
- Commands list for Drupal Api Module
- Commands list for Google Closure
- Commands list for JavaDoc
- Commands list for JsDoc
- Commands list for PhpDocumentor
- Commands list for SassDoc
- Commands list for Sphinx
- Commands list for XmlDoc
- Commands list for YuiDoc

Only commands matching your configured doc-styles are suggested.

For example, to get the list of available commands, press @. Then, press Ctrl+Space to display the completion list. Ctrl+Space is optional, but Sublime Text defaults settings deactivate completion in comment (see `auto_complete_selector` settings).

```
1
2  /**
3   * @mainpage A short story
4   *
5   * @par (paragraph_title)
6   * { paragraph
7   */
8
```

Doxygen Completion

Tab

### 3.4.2 Comment continuation

As you can see on previous example, pressing Enter consecutively automatically continues the comment.

**Warning:** On single line comment, comment continuation may appear as strange on the last line comment (///). The behavior is optional (see parameter `continuation_on_last_comment`). If activated, you can press Shift+Enter to stop continuation.

## 3.5 Navigate in documentation

### 3.5.1 Move to the right column

To ease navigation, press End (Super+Right on OS X) on end-of-line to go to the next column.

```
1
2  ///
3  /// @brief This block is too short
4  ///
5  ///
6
```

Move easily to the right column

End

### 3.5.2 Follow references

**Warning:** Doxygen file only (.dox)

You can move from a @ref tag to the referenced page or section using the `goto_definition` command (press F12 using Sublime Text default key bindings)

## 3.6 Fold / Unfold comments

You can Fold / Unfold comments blocks from the *Command Palette* or using Sublime Text standard shortcuts.

**On Windows and Linux:**

- Ctrl+Shift+[ : Fold
- Ctrl+Shift+] : Unfold

**On OS X:**

- Super+Alt+[: Fold
- Super+Alt+]: Unfold

```

1
2 //-----
3 /// Earth's second moon ?
4 ///
5 /// The name Cruithne is from Old Irish and refers to the early Picts (Irish:
6 /// Cruithin) in the Annals of Ulster and their eponymous king ("Cruide, son
7 /// of Cinge") in the Pictish Chronicle
8 ///
9 void Let_me_introduce_Cruithne();
10

```

Comment folding Ctrl+Shift+[

## 3.7 Translate

**Warning:** Translations use network service. If you are behind a proxy, don't forget to configure it before using those features.

To translate selections, go to the *Command Palette* (Ctrl+Shift+P), then select *DoxyDoxygen: Translate* or *DoxyDoxygen: Translate To* to translate them.

```

1
2 A new feature in DoxyDoxygen.
3 Nova trajto en DoxyDoxygen.
4
5 Now, enjoy the translation of the selected text.
6 Nun ĝui la tradukado de la elektita teksto.
7

```

To French (fr)

To Afrikaans (af)

Translate to... Ctrl+Shift+P

**Tip:** If a cursor is in a comment block (without selection), all descriptions of this comment will be translated.

## 3.8 Generate documentation

If you use Doxygen, you can generate your documentation directly from the *Command Palette*. An assistant will help you to download tools and configure your project.

**Note:** New in version 0.46.4.

Before command execution, DoxyDoxygen parses the Doxyfile file and extract all heading @INCLUDE. For each included file, an environment variable is generated. The name of this variable is: DOXYDOXYGEN\_GENERATED\_<base\_name\_without\_extension>\_PATH and its value is the path where the file is stored. This allows relative inclusion inside each included file (useful for footer...)

If the filename contains non alpha-numeric characters, they are replaced with `_`.

- `../path/filename.ext` defines a variable `DOXYDOXYGEN_GENERATED_FILENAME_PATH` with the value `../path`
  - `path/A@STRANGE!VALUE.ext` defines a variable `DOXYDOXYGEN_GENERATED_A_STRANGE_VALUE_PATH` with the value `path`
- 

---

**Tip:** If you want to include it in your build chain, you can *call this command from the command-line*.

---

## CUSTOMIZATION

### 4.1 Settings

#### 4.1.1 Understanding settings

DoxyDoxygen is highly configurable. This section aims to help to understand the global philosophy.

Take an example: You develop in Php and you want to use the `Drupal API Module` style. `Drupal` has NOT the same set of commands that `PhpDoc`, so you have to change the *doc-style*.

The job to map programming language to a doc-style is done by *profiles*. The new section will look like this:

```
"profiles": [
  {
    "languages": [ "php" ],
    "parsers": [ "LanguagePhp" ],
    "doc_styles": [ "Drupal" ] // instead of "PhpDoc"
  }
],
```

- `languages` is the list of sublime text affected syntaxes. To get a language name, use the *integrated agent*, then look for *doxy.language* and add the value to the list.
- `parsers` is the list of programming language parsers. First is preferred. Copy this settings from “Default Settings” or use `LanguageGeneric` for aliens ones.
- `doc_styles` is the list of doc-styles. First item is use on comment creation. Others only add commands for completion.

Now consider that you want to add information about exceptions. The presence and the order of the tags is defined by `block_layout`. The new section will look like this:

```
"block_layout": {
  "Drupal": [
    "",
    "@_brief",
    "",
    "@param",
    "",
    "@return",
    "", // added line (to force a blank line before @throws)
    "@throws", // added line
    ""
  ]
},
```

The `block_layout` section don't help to fix command formatting... For example, to remove columns alignment, you have to set `preferred_tabs_sizes`:

```
"preferred_tabs_sizes": [ 0 ],
```

### 4.1.2 Settings references

The default configuration file is highly commented, this section is an extract of those informations.

**debug\_enabled** (*default is false*) Turn on logs in console.

**completion\_enabled** (*default is true*) Enable/Disable internal commands completion.

**completion\_use\_aliases** (*default is true*) Add aliases to completions suggestions.

**completion\_snippet\_format** (*default is "default"*) Change the way snippets are generated.

Possibles values are:

- `default`: current preferred (may be one of the following values)
- `pedagogic`: optional arguments generate a tab stop
- `remove_optional`: assumed optional is not optional

**continuation\_on\_last\_comment** (*default is true*) If true, pressing on `Enter` on the last comment line of a group inserts a new comment line. To break this sequence, you have to press `Shift+Enter` or `Down`.

**distance\_for\_bottom\_line\_over\_top\_line** (*default is 3*) The `DoxyCommentNearestEntity` command uses this value to locate the “nearest” function. If the value is 3, the plug-in prefers go up to  $2*3-1=5$  lines up, instead of 2 lines down.

**doxygen\_command\_prefix** (*default is "@"*) Preferred command prefix (@ is more common and shouldn't be change, but can be `\\` if you use QT Doc).

**doxygen\_discard\_param\_directions** (*default is false*) Doxygen allows to specify parameters directions (in, out). Sets this parameter to `true` to discard this information.

**doxygen\_paragraph\_prefix** (*default is "\\t"*) Prefix before a paragraph.

Examples:

- `"\\t"`: move to next column
- `"\\n"`: move to a new line
- `"\\n "`: move to a new line with an indentation
- `" "`: add a space

Currently unaffected commands: `@par`, `@todo`, `@xrefitem`

**doxygen\_cmd** (*default is "doxygen"*) Path to the Doxygen command.

Rules:

- Can be absolute or relative
- If path is relative, the plugin will look for the module in `PATH`
- On Windows machines, the `.exe` suffix will be added if missing

**jsdoc\_paragraph\_prefix** (*default is "\\t"*) Same as `doxygen_paragraph_prefix` for JsDoc (for example, Drupal uses `"\\n "`).



**phpdoc\_paragraph\_prefix** (*default is "\t"*) Same as `doxygen_paragraph_prefix` for JsDoc (for example, Drupal uses "\n ").

**phpdoc\_short\_primitives** (*default is false*) If true, the primitives “Boolean” and “Integer” are shortened to “Bool” and “Int”

Some tools, like PHP CODE SNIFFER, may require this option to work.

**php\_generate\_fully\_qualified\_name** (*default is true*) Replace relative name with fully qualified name.

More details can be found in the [GitHub request](#).

**python\_func\_annotations\_are\_types** (*default is true*) Python annotation may be everything. Set this parameters to true to allow to use those values as parameter types.

**remove\_template\_used\_as\_type** (*default is false*) If true, `template <typename T1> void Bar(T1 x);` will not generate a `@tparam` entry as `T1` appear as type in parameters list.

**brief\_mode** (*default is "preserve"*) Behavior for `@brief`: remove / preserve existing one.

**indentation\_max** (*default is 99999999*) Maximum indentation within a comment.

---

**Note:** This setting does NOT affect list items

---

Listing 4.1: Example with the value ‘2’

```
/**
 * Find players in the game.
 *
 * @param string $filter Name of the filter. 'registered' to return only
 *   registered players. Leave empty to return all players.
 * ^
 * +----- indentation is 2 spaces here
 */
```

Listing 4.2: Example with the value ‘9999999’ (or any number greater than 22)

```
/**
 * @param string $filter Name of the filter. 'registered' the return only
 *   registered players. Leave empty to return all
 *   players.
 * ^
 * +----- indentation is 22 spaces here
 */
```

**min\_spaces\_between\_columns** (*default is [ 1, 2 ]*) Minimum spaces between columns.

**Warning:** The value CANNOT be an empty list.

**Rules:**

- First number is for first column
- Second number is for second column
- ...
- Last number is applied to all next columns

Listing 4.3: Example with the value '[1, 2]'

```
      +-- At least 1 space here (but according ``preferred_tabs_sizes`` it
→ 's more)
/**      V      V-- At least 2 spaces here
 * @param[in]  foo      Lorem ipsum dolor sit amet.
 * @param[out] long_bar  Nullam fringilla feugiat pretium. Quisque
 */
```

**preferred\_tabs\_sizes** (default is [ 12, 6, 8, 8, 8, 8 ]) Default tabs size in documentation block (last value is repeated if needed). You can use [ 0 ] to treat tabs as spaces (and disable alignment).

**Warning:** The value CANNOT be an empty list.

Listing 4.4: Example with the value '[ 13, 6, 8, 8, 8, 8 ]'

```
/**
 * @param[in]  a      { parameter_description }
 * 123456789012      12345678...and continue because there is no more tab
 *
 *      123456
 */
```

**max\_line\_reducing** (default is 12) Maximum number of characters that a line may be reduced to respect the rulers.

---

**Note:** There is no DoxyDoxygen parameter to set the preferred line length.

The value used is the first value of the SublimeText `rulers` parameter (default: 80 if empty).

---

**merge\_matching\_ratio** (default is 0.6) Matching ratio to consider two keys similar.

**It's a value from 0 to 1:**

- 1 is exact match,
- 0.6 is close match.

For example:

```
/** @param  mistake  Name in comment has two 'ss'
 */
void function(int mistake);
```

With `merge_matching_ratio` set to 0.6, on update, as `misstake` similar to `mistake`, the comment will directly become:

```
/* @param  mistake  Name in comment has two 'ss'
 */
void function(int mistake);
```

**preserve\_aliases** (default is false) If false, replace aliases with the main command.

**preferred\_comments\_styles** List of possible styles for comments (first match is better).

**block\_layout\_default** Positioning of the commands that can be present in the documentation block.

See `block_layout` for further informations on format.

**Warning:** SHOULD NOT BE OVERLOAD BY USER (use `block_layout` instead)

**block\_layout** (*default is {}*) Used to overload `block_layout_default` values.

**Warning:** The section define commands positionning (and optionally default value). Do NOT try to use it to format command.

Values can be either:

- an array of tags for rigid layout
- an array of dictionaries for context dependent layout

If value is an array of tags (strings):

- Tags may be command or blank lines.
- Doxygen like commands must be prefixed with @ or \
- Commands arguments are in [the python format\(\) style](#)
- **In addition of standard format specifiers, some have been added:**
  - `doxy_words (start[,end[,step]])` ; (work as python arrays [start:end:step])
  - `doxy_chars (start[,end[,step]])` ; (work as python arrays [start:end:step])
  - `doxy_capitalize()` ;
  - `doxy_upper()` ;
  - `doxy_lower()` ;
  - `doxy_a()` ; (add *a* or *an* before the word)
  - `doxy_A()` ; (add *A* or *An* before the word)
  - `doxy_the()` ; (add *the* before the word)
  - `doxy_The()` ; (add *The* before the word)
  - `doxy_plural()` ;
  - `doxy_conjugate()` ;
  - `doxy_undecorate()` ; (remove prefix: Hungarian notation)
- **Commands arguments may include:**
  - documented item information: `name`, `kind`, `nb_params`, `nb_tparams`...
  - data information: `now` or `utc_now` (see example below)
  - path information: `file_base_name`, `file_full_name`, `file_name`, `file_path`
  - project information: `project_base_name`, `project_full_name`, `project_name`, `project_path`
  - context information: `user_name`, `host_name`, `language`
- Special command `@_brief` may be used to ask for item description without prefix

If value is a dictionary, it should have 2 keys:

- `tags`: same format as if it's an array of strings

- `context`: similar to `sublime text` “context” format

Each **context** is a list of conditions. Each condition is composed of

- `key`: Name of the context whose value you want to query.
- `operator`: Type of test to perform against `key`’s value.

Possibles values for **key**:

- `name`: Name of the entity to document
- **kind**: Kind of the entity to document
  - `function`, `lambda`, `generator`, `constructor`, `destructor`,
  - `class`, `struct`, `union`, `enum`,
  - `var`, `constant`
- `nb_params`: Number of parameters (destructing params group “{ first, second }” count as 1 parameters)
- `nb_tparams`: Number of template parameters
- `row`: Cursor row
- `col`: Cursor column

Possibles values for **operator** (Defaults to `equal`):

- `equal`
- `not_equal`
- `regex_match`: Match against a regular expression (full match).
- `not_regex_match`
- `regex_contains`: Match against a regular expression (partial match).
- `not_regex_contains`
- `greater_than`
- `lower_than`

**proxies** Defines for each protocol a proxy settings.

If the dictionary is empty, system defaults are used.

Possibles values:

- `<package_control>` uses package control settings
- `joe:password@proxy.example.com:3128` define specific proxy

**translators** The list of translation services (first is preferred).

**source\_languages** (*default is []*) The list of languages that can be present in your source (or `auto`)

If the list is not empty and not the target language, a conversion may be perform (this can slow down update)

Examples:

- `auto` Auto-detection
- `[]` Disable translation of non-generated text
- `["de", "fr"]` German or French (currently same as `auto`)

**target\_language** (*default is "en"*) Preferred language in generated comments (also used as default translation target).

The format is based on ISO\_639-1 format with Google extension.

**Examples:**

- ar Arabic
- de German
- es Spanish
- fr French
- it Italian
- pt Portuguese
- ru Russian
- zh-CN Chinese (simplified)
- zh-TW Chinese (traditional)
- hmn Hmong
- ceb Cebuano

**autofill\_rules\_default** DoxyDoxygen can automatic fill some descriptions. Here the default rules used to do it.

To disallow auto-fill, you may override it with an empty array.

**autofill\_rules** Used to overload `autofill_rules_default` values.

If you want to disable parameters description (common request), you can try:

```
"autofill_rules": [
  {
    "parameter_description": [
      "."
    ]
  }
],
```

**autofill\_reliability\_percentage** (*default is 60*) Default reliability for generated text.

**Should be:**

- 0: disable auto-fill (default and custom)
- 60: text if generated only if none found in source WITH a tab stop
- 100: text if generated only if none found in source WITHOUT tab stop
- 101: generated text override source one

**profiles\_default** For each Sublime Text language, give the preferred parser and documentation style (also called “documentation generator”).

**profiles** Used to overload `profiles_default` values.

### 4.1.3 Translation services settings

To add a translation service, append an entry to `translators` settings. The first entry is the preferred one. Each entry is a dictionary. This section describes for each translator service, the available keys and their meaning.

#### Google

- Used by default

##### Settings description:

- `provider`: "google" (mandatory)
- `api_key`: Your API key

#### Bing

- [Require a Bing account](#)
- Use your key as the built-in key have limit of 2,000,000 characters per month.

##### Settings description:

- `provider`: "bing" (mandatory)
- `client_id`: Your client identifier
- `client_secret`: Your secret identifier for this service

#### SDL Language Cloud

- [Require a SDL account](#)
- For Machine Translation you can use up to 500,000 characters per month in Sandbox for free.

##### Settings description:

- `provider`: "sdl" (mandatory)
- `api_key`: Your API key

#### Frengly

- [Require a Frengly account](#)

##### Settings description:

- `provider`: "frengly" (mandatory)
- `email`: Your email
- `password`: Your password for this service

## Yandex

- [Require a Yandex account](#)

### Settings description:

- provider: "yandex" (mandatory)
- key: Your API key
- user: Your client identifier
- pass: Your password for this service

## MyMemories

- [Limited count of free translations per day without key \(1000-10000 words/day\)](#)
- email is optional, 10000 words/days if email provided
- user is optional

**Warning:** Contrary to MyMemories statements, email and user seem to have no effect on translation limitation

### Settings description:

- provider: "mymemories" (mandatory)
- email: Your email
- user: Your client identifier
- pass: Your password for this service

## Baidu

- [Require a Baidu account](#)

### Settings description:

- provider: "baidu" (mandatory)
- app\_id: Your client identifier
- secret\_key: Your secret identifier for this service

## Youdao

- Web interface and [Youdao account supported](#)
- 1000 requests per hour. [If you need more, please contact Youdao](#)

### Settings description:

- provider: "youdao" (mandatory)
- key\_from: Your client identifier
- secret\_key: Your secret identifier for this service

## Hablaa

- Currently word by word but large languages support.

### Settings description:

- provider: "hablaa" (mandatory)

## LiguaSys

- **Require a LiguaSys account, 3 steps:**
  - [Register](#)
  - Check your email
  - [Choose your product](#)
- Limited free translations (up to 20 API calls a minute, and up to 500 calls a month).

### Settings description:

- provider: "linguasy" (mandatory)
- secret\_key: Your secret identifier for this service

## 4.2 Add your own doc-style

DoxyDoxygen may be extended by *plug-in*. Plug-in currently allow to define your own *doc-style*. See [HeaderDoc example](#) for further informations.

## 4.3 Key bindings

This plug-in come with few keyboard shortcuts (to prevent conflict with others plug-ins). But, key bindings are configurable. Please refer to [Sublime Text official documentation](#) for further informations.

This section describes the key bindings added by default in DoxyDoxygen.

### 4.3.1 Key bindings on Windows and Linux

- Enter : doxy\_enter()
- Enter (keypad) : doxy\_enter()
- End : doxy\_go\_to\_eol()
- Ctrl+Shift+[ : doxy\_fold\_comments(action=fold)
- Ctrl+Shift+] : doxy\_fold\_comments(action=unfold)
- Alt+Q : doxy\_comment\_nearest\_entity()
- Shift+Alt+Q : doxy\_update\_comments(reparse=True,new\_style=roll\_2)
- Alt+Q : doxy\_update\_comments(reparse=True)



### 4.3.2 Key bindings on OS X

In the table below Super refers to the Command key (marked with the ⌘ symbol)

- Enter : doxy\_enter()
- Enter (keypad) : doxy\_enter()
- Super+Right : doxy\_go\_to\_eol()
- Super+Alt+[ : doxy\_fold\_comments(action=fold)
- Super+Alt+] : doxy\_fold\_comments(action=unfold)
- Super+Alt+Q : doxy\_comment\_nearest\_entity()
- Super+Shift+Alt+Q : doxy\_update\_comments(reparse=True, new\_style=roll\_2)
- Super+Alt+Q : doxy\_update\_comments(reparse=True)

## 4.4 Commands from the palette

A lot of commands are only available from the *Command Palette*.

**To open the Command Palette**

- on Windows or Linux, press Ctrl+Shift+P
- on OS X, press Super+Shift+P

This section describes the commands added by default in DoxyDoxygen.

- *DoxyDoxygen: Comment Wrap* : doxy\_update\_comments(reparse=False)
- *DoxyDoxygen: Comment Update Preserving Style* : doxy\_update\_comments(reparse=True)
- *DoxyDoxygen: Update Comment With Preferred Style* : doxy\_update\_comments(reparse=False, new\_style=pref)
- *DoxyDoxygen: Update Comment With User-Selected Style (interactive)...* :  
doxy\_update\_comments(reparse=False, new\_style=user\_select)
- *DoxyDoxygen: Comment All* : doxy\_comment\_all()
- *DoxyDoxygen: Select Comments (interactive)...* : doxy\_select\_comments(kind=user\_select)
- *DoxyDoxygen: Fold All Comments* : doxy\_fold\_comments(action=fold, scope=file)
- *DoxyDoxygen: Unfold All Comments* : doxy\_fold\_comments(action=unfold, scope=file)
- *DoxyDoxygen: Send Report* : doxy\_send\_report()
- *DoxyDoxygen: Build Documentation* : doxy\_build\_documentation()
- *DoxyDoxygen: Tutorial* : doxy\_tutorial()
- *DoxyDoxygen: Translate Selection* : doxy\_translate()
- *DoxyDoxygen: Translate Selection To...* : doxy\_translate(target\_language=user\_select)
- *Preferences: DoxyDoxygen Settings* : doxy\_edit\_settings(base\_file=\${packages}/\${module\_name}/Doxy.
- *Preferences: DoxyDoxygen Key Bindings* : doxy\_edit\_settings(base\_file=\${packages}/\${module\_name}/De  
(\${platform}).sublime-keymap, user\_file=\${packages}/User/Default  
(\${platform}).sublime-keymap)

## 4.5 Commands from the menu

Alternatively to the *Command Palette*, some commands are also available from the menu. Go to *Preferences* → *Package Settings* → *DoxyDoxygen* to launch them.

Available commands:

- *Settings*: `doxy_edit_settings (base_file=${packages}/${module_name}/Doxy.sublime-settings, us`
- *Key Bindings* : `doxy_edit_settings (base_file=${packages}/${module_name}/Default ($ {platform}) .sublime-keymap, user_file=${packages}/User/Default ($ {platform}) .sublime-keymap)`
- *Commands from the palette*: `doxy_edit_settings (base_file=${packages}/${module_name}/Doxy.sublim`
- *Purchase License* : `doxy_buy ()`
- *Enter License* : `doxy_register ()`
- *Remove License* : `doxy_unregister ()`

---

**Note:** License commands are only available from to menu

---

**GLOSSARY**

**comment style** A decoration around the comment.

```
/* *****  
 * comment  
 ***** */  
  
////////////////////  
/// comment  
////////////////////
```

**doc-style** A markup language stored in special comment blocks or python doc-string used by a specific *documentation generator*.

**documentation block** A documentation block is a comment handled by a *documentation generator*

**documentation generator** A programming tool that generates software documentation intended for programmers (API documentation) or/and end users (End-user Guide).

**plug-in** A software component that adds a specific feature to an existing computer program (like Sublime Text).



## APPENDICES

## 6.1 Features Comparison

	DoxyDoxygen	DocBlockr	AutoDocString
Functions Documentation	✓	✓ (creation only)	✓
Class Documentation	✓	✓ (creation only)	✓
Variable Documentation	✓	✓ (creation only)	✓
Type Detection	✓ (body inspection)	☁ (default value or name)	☁ (default value)
Documentation Update	✓	✗	✓
One Key Comment Creation	✓	✗	✓
Auto-filled description	✓	✗	✗
Translation Services	✓	✗	✗
Snippets	✓	✓	✗
Comments Continuation	✓	✓	✗
Comments Folding	✓	✗	✗
Comments Re-wrap	✓	☁ (basic)	✗
Configurable Layout	✓	☁ (limited)	☁
DocString Support	✓	✗ (but fork available)	✓
Plug-ins Support	✓	✗	✗
Documentation Export (html, latex, chm...)	☁ (Doxygen only)	✗	✗

## 6.2 Supported Documentation Tools







	DoxyDoxygen	DocBlockr
ApiDoc	✓	✗
AsDoc	✓	✓
Doxygen	✓	✗
Drupal Api Module	✓	☁
Google Closure	✓	✓
JavaDoc	✓	☁ (partial)
JsDoc	✓	✓
PhpDocumentor	✓	✓
SassDoc	✓	✗
Sphinx	☁	✗
XmlDoc	✓	✗
YuiDoc	✓	✓

## 6.3 Supported Languages

	DoxyDoxygen	DocBlockr
4GL	✓ (generic, # comment style)	✗
ActionScript	✓ (generic)	✓
AngelScript	☁ (alt+q not working on functions)	✗
Apex	✓ (partial, Java based)	✓ (partial, java based)
AppleScript	✓ (generic)	✗
Assembler (x86, arm...)	! (only wrap and continuation)	✗
ASP	✓ (generic)	✗
AutoHotKey	☁ (poor)	✗
Bash	✓	✗
C	✓	✓ (official but poor)
C#	✓	✗
C++	✓ (including C++11)	✓ (official but poor)
Clojure	✓ (generic)	✗
Coffee	✓ (generic)	✓
Crystal	✓ (generic)	✗
D	✓ (generic)	✗
Dot	! (only wrap and continuation)	✗
Doxygen	✓	✗

Continued on next page

Table 6.1 – continued from previous page

	DoxyDoxygen	DocBlockr
Fortran (Modern)	 (generic, poor)	✗
Go		✗
Graddle	 (generic)	✗
Groovy	 (partial, extended Java syntax)	 (partial, java based)
Haskell		✗
Haxe	 (generic)	 (ActionScript based)
Erlang	 (poor)	✗
Elixir	 (poor, no docstring support)	✗
Java		
JavaScript	 (including ES/6)	 (including ES/6)
Jinja2	 (generic, poor)	✗
Julia	 (generic)	✗
Lisp	 (generic)	✗
Lua	 (generic)	✗
Matlab	 (generic)	✗
NSIS	 (generic, poor)	✗
Objective C	 (generic)	 (poor ?)
Objective C++	 (generic)	 (poor ?)
OpenGL Shading Language (GLSL)	 (alt+q not working on functions)	✗
OCaml	 (generic)	✗
MQL4		✗
Pascal	 (generic)	✗
Perl	 (generic, poor only Perl6 syntax)	✗
Php		
Processing		✗
Prolog	 (generic, poor)	✗
Python		✗
QML		✗
R	 (generic)	✗
Racket	 (generic)	✗
Razor	 (generic, poor functions detection)	✗
Ruby	 (Python based with extension)	✗
Rust		
Sass / Scss	 (require <i>Syntax highlight for Sass</i> )	✗
Scala	 (generic)	✗
Scheme	 (generic)	✗

Continued on next page

Table 6.1 – continued from previous page

	DoxyDoxygen	DocBlockr
Shell Script (bash)	! (only wrap and continuation)	×
SQL	✓ (generic)	×
Squirrel (rut)	✓ (generic)	×
Swift	✓	×
SystemVerilog	✓ (generic)	×
TCL	✓ (generic)	×
TypeScript	✓ (generic)	✓
Thrift	✓ (generic)	×
VBScript	✓ (generic)	×
VB.NET	✓ (generic)	×
VHDL	✓ (generic)	×
VEX	✓ (generic)	×
WebIDL	✓ (generic)	×
YAML	! (only wrap and continuation)	×

## 6.4 Surveys



To help to improve this software, I need to know your needs...

**Here, you can find some surveys:**

- [about this user guide.](#)
- [general survey.](#)
- [about auto-filled description.](#)



## KNOWN ISSUES

### 7.1 Incorrect syntaxes

- SublimeText 3083: SystemVerilog last update may cause incompatibility
- SublimeText 3115-3122+: Shell functions are no longer detected
- SublimeText 3115-3122+: C++ template operators are no longer detected
- SublimeText (all versions): Comment continuation doesn't work on Batch if cursor is at the end of line

### 7.2 Conflict with others plugins

- WrapLinePlus: `Alt+Q` redefined ([issue #53](#))
- DoxyDoc: `Enter` redefined. Please uninstall this plugin. It cause trouble and add no fonctionnality.
- CursorRuler: `rulers` is overwrite ([issue #57](#), [Cursor Rulers Issue #13](#) <<https://github.com/icylace/CursorRuler/issues/13>> and [Cursor Rulers Issue #3](#) <<https://github.com/icylace/CursorRuler/issues/3>>). Please do not leave empty to solve problem or use DoxyDoxygen version `>= 0.49.0`
- Shader Syntax (GLSL HSL Cg) 1.0.3: Comments are now handled properly, so trigger now works (you can use "Open/GL Shading Language" instead)

### 7.3 Incompatible Sublime Text versions

- SublimeText 3121-3122: `urllib.parse` is missing on Linux (see <https://github.com/20Tauri/DoxyDoxygen/issues/63>). You have to install Package Control 3.2.0-beta2+ to fix this issue



This section contains the Frequently Asked Questions.

More questions can be found on [GitHub](#). If you're not familiar with GitHub, do not forget to click on *Closed* to consult the archives.

## 8.1 Can I call DoxyDoxygen from the command-line ?

Yes, you can. Use **subl** to do that.

**subl** is part of Sublime Text. It can be found in your Sublime Text install directory. Please refer to the [Sublime Text documentation](#) for further informations.

All commands described in the *Commands from the palette* section can be called.

New in version 0.53.0: It's strongly advised to use `doxy_chain_commands` to chain commands (or use view's ones) because it's help to synchronise them.

Here some examples:

- Build documentation:

```
subl -b --command "doxy_build_documentation"
```

- Update all documentation blocks using the preferred style:

```
subl -b --command "doxy_select_comments {"kind":"doxy"}"  
subl -b --command "doxy_update_comments {"reparse":"False", "new_style":"  
→"preferred"}
```

- Comment the entity near the line 15 of the `example.c` file:

```
rem Carriage returns are only here for readability. Please remove them.  
subl --command "doxy_chain_commands {"commands": [  
  ["window.doxy_open_file", { "file":"path\\example.c" }],  
  ["window.goto_line", { "line":"15" }],  
  ["view.doxy_comment_nearest_entity", {}]  
] }"
```

- Comment all entities of the current file

New in version 0.49.1: `command_all`

```
subl --command "comment_all"
```

## 8.2 Is it possible to disable default parameter description ?

To disable auto-fill, update your configuration file with one of the following lines:

- "autofill\_rules\_default": [], to disable the default rules
- "autofill\_reliability\_percentage": 0, to disable all rules

If you prefer to have no description, you can also try:

```
"autofill_rules": [  
  {  
    "parameter_description": [  
      "."  
    ]  
  }  
],
```

## 8.3 How can I switch to a different preferred\_comment\_style ?

See section *Switch between comment styles*.

Alternatively, using Enter, DoxyDoxygen use the first preferred style that match the language. To use a specific comment style, you have to start your comment style (ex: // =), then press Alt+Q.

**Warning:** For block styles, you have to close it before pressing Alt+Q in it.

## 8.4 Is it possible to add tags dynamically ?

New in version 0.27.

block\_layout parameter may be context dependent. To set up a context dependent, you have to define a list of dictionaries.

Detail format of the block\_layout parameter can be found in the *Settings* section.

Example

```
"block_layout": {  
  "Doxygen": [  
    {  
      "tags": [  
        "@brief          I'm the {name} class"  
      ],  
      "context": [  
        { "key": "name",      "operator": "regex_match",    "operand": "^_.*$" },  
        { "key": "kind",     "operator": "equal",          "operand": "class" }  
      ]  
    },  
    // Auto-fill description for getters  
    {
```

```

        "tags": [
            "@brief          Get {name:doxy_words(1,);doxy_lower();}",
            "@return         {name:doxy_words(1,);doxy_capitalize();}"
        ],
        "context": [
            { "key": "name",          "operator": "regex_match",          "operand":
↪ "get[A-Z_].*" },
        ]
    },

    // File Header integration
    {
        "tags": [
            "@brief          I'm a file header and my name is {file_base_name}",
            "",
            "@author         {user_name}",
            "",
            "@date           {now:%d-%b-%Y}"
        ],
        "context": [
            { "key": "row",          "operator": "equal",          "operand": "0" },
            { "key": "kind",        "operator": "equal",          "operand": "" }
        ]
    },

    // Compact style if there's less than one parameter
    {
        "tags": [
            "@brief",
            "@param",
            "@tparam",
            "@return",
            ""
        ],
        "context": [
            { "key": "nb_params", "operator": "lower_than",          "operand": "2" }
        ]
    }

    // You don't have to be exhaustive.
    // If no rule match, 'block_layout_default' is considered
}

```

## 8.5 Why there's no alignment on Enter ?

This point was first discussed on Git “How to go to next line without breaking TAB order?”.

**Tip:** Not recommended.

You probably want to chain `doxy_enter` and `doxy_go_to_eol`. So, you can try to override the Enter key binding. Install the package [Chain of Command](#). Then, add your custom binding to chain commands:

```
{
    "keys": ["enter"],
```

```
"command": "chain", "args": {
  "commands": [
    ["doxy_enter" ],
    ["doxy_go_to_eol" ],
  ]
},
"context":
[
  { "key": "auto_complete_visible", "operator": "equal", "operand": false,
↪ "match_all": true },
  { "key": "selector", "operator": "equal", "operand": "comment,
↪ string.quoted.double.block.cython, string.quoted.single.block.cython, string.quoted.
↪ double.block.python, string.quoted.single.block.python", "match_all": true }
]
},
```

## 9.1 Contact from GitHub



Issues and public discussions may be found on [GitHub](#). This is a place of choice to know current issues status.

## 9.2 Contact from Web Site



An [online contact form](#) may be used to report everything or to ask private questions (no email required).

## 9.3 Contact from Sublime Text



To ease reporting, DoxyDoxygen includes an integrated agent. It will automatically be opened on crash prompting you to make a report. This agent provides informations about your configuration (version, settings...).

You can also activate it manually from the context menu (right click, then select *DoxyDoxygen Report* → *Anomaly report*).

It's also the fastest and easiest way to provide example for your reporting.

**Attention:** Extra description and email address are optional, but if you don't add it there's no way to contact you for a solution or to get further informations.