

---

# **sphinxcontrib-doxylink Documentation**

***Release 1.11.2***

**Matt Williams**

**Apr 29, 2022**



---

## Contents

---

<b>1</b>	<b>Function reference</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>5</b>
2.1	Namespaces, classes etc. . . . .	6
2.2	Functions . . . . .	6
2.3	Files . . . . .	6
<b>3</b>	<b>Setup</b>	<b>7</b>
<b>4</b>	<b>Configuration values</b>	<b>9</b>
<b>5</b>	<b>Bug reports</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>
	<b>Index</b>	<b>15</b>



Doxylink is a Sphinx extension to link to external Doxygen API documentation.

It allows you to specify C++ symbols and it will convert them into links to the HTML page of their Doxygen documentation.



## Function reference

```
class sphinxcontrib.doxylink.doxylink.Entry(kind, file)
```

<b>file</b>	Alias for field number 1
-------------	--------------------------

<b>kind</b>	Alias for field number 0
-------------	--------------------------

```
class sphinxcontrib.doxylink.doxylink.FunctionList
    A FunctionList maps argument lists to specific entries
```

```
class sphinxcontrib.doxylink.doxylink.SymbolMap (xml_doc:  
                                                xml.etree.ElementTree.ElementTree)
```

A SymbolMap maps symbols to Entries or FunctionLists

`sphinxcontrib.doxylink.doxylink.fetch_file(app, source, output_path)`  
Fetches file and puts it in the desired location if it does not exist yet.

Local files will be copied and remote files will be downloaded. Directories in the `output_path` get created if needed.

**Args:** app: Sphinx' application instance  
source (str): Path to local file or URL to remote file  
output\_path (str): Path with filename to copy/download the source to, relative to Sphinx' output directory

<code>sphinxcontrib.doxylink.doxylink.is_url(str_to_validate)</code>	Helper function to check if string contains URL
--	---

**Args:** str\_to\_validate (str): String to validate as URL

**Returns:** bool: True if given string is a URL, False otherwise

```
sphinxcontrib.doxylink.doxylink.match_pieewise(candidates: set, symbol: str, sep: str = ':') → set
```

Match the requested symbol reverse piecewise (split on `:`) against the candidates. This allows you to under-specify the base namespace so that `"MyClass"` can match `my_namespace::MyClass`

**Args:** candidates: set of possible matches for symbol symbol: the symbol to match against sep: the separator between identifier elements

**Returns:** set of matches

sphinxcontrib.doxylink.doxylink.**parse\_tag\_file** (*doc: xml.etree.ElementTree.ElementTree*)  
→ dict  
Takes in an XML tree from a Doxygen tag file and returns a dictionary that looks something like:

```
{'PolyVox': Entry(...),
 'PolyVox::Array': Entry(...),
 'PolyVox::Array1DDouble': Entry(...),
 'PolyVox::Array1DFloat': Entry(...),
 'PolyVox::Array1DInt16': Entry(...),
 'QScriptContext::throwError': FunctionList(...),
 'QScriptContext::toString': FunctionList(...)
}
```

Note the different form for functions. This is required to allow for ‘overloading by argument type’.

#### Parameters

**doc** [xml.etree.ElementTree] The XML DOM object

**Returns** a dictionary mapping fully qualified symbols to files

sphinxcontrib.doxylink.doxylink.**process\_configuration** (*app, tag\_filename, rootdir,*  
*pdf\_filename*)

Processes the configured values for doxylink and doxylink\_pdf\_files and warns about potential issues.

The type of builder decides which values shall be used.

**Args:** app: Sphinx’ application instance tag\_filename (str): Path to the Doxygen tag file rootdir (str): Path to the root directory of Doxygen HTML documentation pdf\_filename (str): Path to the pdf file; may be empty when LaTeX builder is not used

sphinxcontrib.doxylink.doxylink.**report\_info** (*env, msg, docname=None, lineno=None*)  
Convenience function for logging an informational

**Args:** msg (str): Message of the warning docname (str): Name of the document on which the error occurred  
lineno (str): Line number in the document on which the error occurred

sphinxcontrib.doxylink.doxylink.**report\_warning** (*env, msg, docname=None,*  
*lineno=None*)

Convenience function for logging a warning

**Args:** msg (str): Message of the warning docname (str): Name of the document on which the error occurred  
lineno (str): Line number in the document on which the error occurred

sphinxcontrib.doxylink.parsing.**normalise** (*symbol: str*) → Tuple[str, str]  
Takes a c++ symbol or function and splits it into symbol and a normalised argument list.

#### Parameters

**symbol** [string] A C++ symbol or function definition like PolyVox::Volume,  
Volume::printAll() const

**Returns** a tuple consisting of two strings: (qualified function name or symbol,  
normalised argument list)



You use Doxylink like:

```
:polyvox:`PolyVox::Volume`  
You use :qtogre:`QtOgre::Log` to log events for the user.  
:polyvox:`PolyVox::Array::operator[]`
```

Where polyvox and qtogre roles are defined by the *doxylink* configuration value.

Like any interpreted text role in Sphinx, if you want to display different text to what you searched for, you can include some angle brackets `<...>`. In this case, the text inside the angle brackets will be used to match up with Doxygen and the part in front will be displayed to the user:

```
:polyvox:`Array <PolyVox::Array>`.  
:polyvox:`tidyUpMemory <tidyUpMemory(int)>` will reduce memory usage.
```

---

**Note:** In C++, it is common that classes and functions will be templated and so will have angle brackets themselves. For example, the C++ class:

```
PolyVox::Array<0,ElementType>
```

would be naively linked to with Doxylink with:

```
:polyvox:`PolyVox::Array<0,ElementType>`
```

but that would result in Sphinx parsing it as you wanting to search for `0,ElementType` and display `PolyVox::Array` as the text to the user. To avoid this misparsing you must escape the opening `<` by prepending it with a `\`:

```
:polyvox:`PolyVox::Array\<0,ElementType>`
```

If you want to use templated symbols inside the angle brackets like:

```
:polyvox:`Array <PolyVox::Array<0,ElementType>>`
```

then that will work without having to escape anything.

---

## 2.1 Namespaces, classes etc.

For non-functions (i.e. namespaces, classes, enums, variables) you simply pass in the name of the symbol. If you pass in a partial symbol, e.g. ``Volume`` when you have a symbol in C++ called `PolyVox::Utils::Volume` then it would be able to match it as long as there is no ambiguity (e.g. with another symbol called `PolyVox::Old::Volume`). If there is ambiguity then simply enter the fully qualified name like:

```
:polyvox:`PolyVox::Utils::Volume` or :polyvox:`PolyVox::Utils::Volume <Volume>`
```

## 2.2 Functions

For functions there is more to be considered due to C++'s ability to overload a function with multiple signatures. If you want to link to a function and either that function is not overloaded or you don't care which version of it you link to, you can simply give the name of the function with no parentheses:

```
:polyvox:`PolyVox::Volume::getVoxelAt`
```

Depending on whether you have set the `add_function_parentheses` configuration value, Doxylink will automatically add on parentheses so that it will be printed as `PolyVox::Volume::getVoxelAt()`.

If you want to link to a specific version of the function, you must provide the correct signature. For a requested signature to match on in the tag file, it must exactly match a number of features:

- The types must be correct, including all qualifiers, e.g. `unsigned const int`
- You must include any pointer or reference labeling, e.g. `char*`, `const QString &` or `int **`
- You must include whether the function is const, e.g. `getX() const`

The argument list is not whitespace sensitive (any more than C++ is anyway) and the names of the arguments and their default values are ignored so the following are all considered equivalent:

```
:myapi:`foo( const QString & text, bool recalc, bool redraw = true )`  
:myapi:`foo(const QString &foo, bool recalc, bool redraw = true )`  
:myapi:`foo( const QString& text, bool recalc, bool redraw )`  
:myapi:`foo(const QString&,bool,bool)`
```

When making a match, Doxylink splits up the requested string into the function symbol and the argument list. If it finds a match for the function symbol part but not for the argument list then it will return a link to any one of the function versions.

## 2.3 Files

You can also link directly to a header or source file by giving the name of the file:

```
:myapi:`main.cpp`  
:myapi:`MainWindow.h`
```

## CHAPTER 3

---

### Setup

---

When generating your Doxygen documentation, you need to instruct it to create a ‘tag’ file. This is an XML file which contains the mapping between symbols and HTML files. To make Doxygen create this file ensure that you have a line like:

```
GENERATE_TAGFILE = PolyVox.tag
```

in your Doxyfile.



---

## Configuration values

---

### **doxylink**

The environment is set up with a dictionary mapping the interpreted text role to a tuple of tag file and prefix. The keys of this dictionary must be lower-case. The prefix can be an absolute path or a path relative to Sphinx' output directory. An optional third element with the name of a Doxygen pdf file may be added. This will be used when Sphinx uses the LaTeX builder. Otherwise, the second element of the tuple will be used to link to.

```
doxylink = {
    'polyvox' : ('/home/matt/PolyVox.tag', '/home/matt/PolyVox/html/', 'polyvox_
↪doxygen.pdf'),
    'qtogre' : ('/home/matt/QtOgre.tag', '/home/matt/QtOgre/html/', 'qtogre_
↪doxygen.pdf'),
}
```

---

**Note:** The links in your pdf document to your Doxygen pdf file(s) may not work (properly) in a browser or a basic PDF-reader. They should work in Adobe Reader for example.

---

### **add\_function\_parentheses**

A boolean that decides whether parentheses are appended to function and method role text. Default is `True`.

### **doxylink\_pdf\_files**

Doxylink can be configured to download remote Doxygen pdf files or copy them from a local location. You should use the output file name as the third element of the value of the `doxylink` dictionary **and** as key in the `doxylink_pdf_files` dictionary, which should contain the URL to the remote location or local location as value. If the pdf file already exists locally in Sphinx' output directory, it will not be downloaded or overwritten.

```
doxylink_pdf_files = {
    'polyvox_doxxygen.pdf': url_to_remote_doxxygen_pdf,
    'qtogre_doxxygen.pdf': '/home/matt/qtogre/doxygen.pdf',
}
```



## CHAPTER 5

---

### Bug reports

---

If you find any errors, bugs, crashes etc. then please raise an issue [on GitHub](#). If there is a crash please include the backtrace and log returned by Sphinx. If you have a bug, particularly with Doxylink not being able to parse a function, please send the tag file so tat I can reproduce and fix it.

**requires** Python 3.4

**copyright** Copyright 2017 by Matt Williams

**license** BSD, see LICENSE for details.





### S

`sphinxcontrib.doxylink.doxylink`, 3  
`sphinxcontrib.doxylink.parsing`, 4



## A

`add_function_parentheses`  
configuration value, 9

## C

configuration value  
    `add_function_parentheses`, 9  
    `doxylink`, 9  
    `doxylink_pdf_files`, 9

## D

`doxylink`  
    configuration value, 9  
`doxylink_pdf_files`  
    configuration value, 9

## E

`Entry` (class in *sphinxcontrib.doxylink.doxylink*), 3

## F

`fetch_file()` (in module *sphinxcontrib.doxylink.doxylink*), 3  
`file` (*sphinxcontrib.doxylink.doxylink.Entry* attribute), 3  
`FunctionList` (class in *sphinxcontrib.doxylink.doxylink*), 3

## I

`is_url()` (in module *sphinxcontrib.doxylink.doxylink*), 3

## K

`kind` (*sphinxcontrib.doxylink.doxylink.Entry* attribute), 3

## M

`match_piecewise()` (in module *sphinxcontrib.doxylink.doxylink*), 3

## N

`normalise()` (in module *sphinxcontrib.doxylink.parsing*), 4

## P

`parse_tag_file()` (in module *sphinxcontrib.doxylink.doxylink*), 4  
`process_configuration()` (in module *sphinxcontrib.doxylink.doxylink*), 4

## R

`report_info()` (in module *sphinxcontrib.doxylink.doxylink*), 4  
`report_warning()` (in module *sphinxcontrib.doxylink.doxylink*), 4

## S

*sphinxcontrib.doxylink.doxylink* (module), 3  
*sphinxcontrib.doxylink.parsing* (module), 4  
`SymbolMap` (class in *sphinxcontrib.doxylink.doxylink*), 3