
sphinx.rfc2119 Documentation

Release 0.1

Chris Gough

September 10, 2015

1	Introduction	3
2	Requirement Directives	5
2.1	Mandatory Requirements	5
2.2	Recommendations	5
2.3	Options	6
3	Requirement List Directives	7
3.1	Example Mandatory List	7
3.2	Example Recommended List	8
3.3	Example Optional List	9
4	Code	11
5	Indices and tables	15
	Python Module Index	17

Contents:

Introduction

The purpose of this sphinx extension is enable markup of requirements per the definitions in RFC 2119.

- <http://sphinx-doc.org>
- <https://www.ietf.org/rfc/rfc2119.txt>

This is supported by a suite of requirement directives that correspond to the RFC 2119 keywords. There are also a suite of requirement list directives that support documenting requirements (that have been defined with the rfc2119 directives).

This module is in ALPHA status. Feedback / pull requests very welcome. Development occurs on GitHub:

- <https://github.com/monkeypants/sphinx.rfc2119/>

Documentation is a bit strange - I used the directives to document what it is supposed to do. Which was handy as I made it, but probably not very readable.

Basically, add the module to your the *conf.py* of your repo then use the directives. They have lower case names.

The motivation for this module is in response to this ticket:

- <https://github.com/AusDTO/apiguide/issues/11>

Must

As per RFC 2119, users of this module should include a block of boilerplate near the beginning of their sphinx document. A *rfc2119interpretation* directive is required for this purpose.

RFC 2119 keywords

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119.

May

Optionally, this module should support governance parameters for documenting requirement status, valid from date, review date, etc.

None of these features are implemented yet, sorry. If there's something you actually need, please raise a ticket on GitHub.

Requirement Directives

RFC 2119 defines:

- 5 mandatory requirement labels
- 5 recommendation labels
- 2 optional labels

This sphinx extension includes 12 corresponding “requirements directives”. They have lowercase names (with words separated by underscores for two word phrases).

2.1 Mandatory Requirements

Must

A *must* directive is required, with semantic equivalence to the MUST keyword in RFC 2119.

Must Not

The absence of a *must_not* directive is forbidden. This directive has semantic equivalence to the MUST NOT keyword in RFC 2119

Shall

A *shall* directive is required. It is an alias for the *must* directive.

Shall Not

The absence of a *shall_not* directive is forbidden. It is a necessary alias for the *must_not* directive.

Required

A *required* directive is something we need. It is an alias for the *must* directive.

2.2 Recommendations

Should

A *should* directive is important. It has semantic equivalence to the SHOULD keyword in RFC 2119.

Recommended

A *recommended* directive is very useful. It is an alias for the *should* directive.

Should Not

The absence of a *should_not* directive is not something we want. It is equivalent to the SHOULD NOT key word in RFC 2119.

Not Recommended

Because the *not_recommended* is an alias for *should_not* directive, it's absence should be avoided.

2.3 Options

Optional

It might be nice to have an *optional* directive, which would be equivalent of OPTIONAL keyword in RFC 2119.

May

As an alias for the *optional* directive, a *may* directive might also be good to have.

Requirement List Directives

Must

A *mandatorylist* directive is required. It should produce a list of all MUST and MUST NOT directives (and their aliases).

Must

A *recommendedlist* directive is required. It should produce a list of all SHOULD and SHOULD NOT directives (and their aliases).

Must

A *optionallist* directive is required. It should produce a list of all OPTIONAL and MAY directives.

3.1 Example Mandatory List

The following list of mandatory requirements includes every MUST, MUST NOT, SHALL, SHALL NOT and REQUIRED semantic in this suite of documentation, because they are makted up with the appropriate directives.

Must

As per RFC 2119, users of this module should include a block of boilerplate near the beginning of their sphinx document. A *rfc2119interpretation* directive is required for this purpose.

(The original entry is located in introduction.rst, line 24 and can be found [here](#).)

Must

A *must* directive is required, with semantic equivalence to the MUST keyword in RFC 2119.

(The original entry is located in requirement_directives.rst, line 17 and can be found [here](#).)

Must Not

The absence of a *must_not* directive is forbidden. This directive has semantic equivalence to the MUST NOT keyword in RFC 2119

(The original entry is located in requirement_directives.rst, line 23 and can be found [here](#).)

Shall

A *shall* directive is required. It is an alias for the *must* directive.

(The original entry is located in requirement_directives.rst, line 30 and can be found [here](#).)

Shall Not

The absence of a *shall_not* directive is forbidden. It is a necessary alias for the *must_not* directive.

(The original entry is located in requirement_directives.rst, line 36 and can be found [here](#).)

Required

A *required* directive is something we need. It is an alias for the *must* directive.

(The original entry is located in requirement_directives.rst, line 42 and can be found [here](#).)

Must

A *mandatorylist* directive is required. It should produce a list of all MUST and MUST NOT directives (and their aliases).

(The original entry is located in requirement_list_directives.rst, line 4 and can be found [here](#).)

Must

A *recommendedlist* directive is required. It should produce a list of all SHOULD and SHOULD NOT directives (and their aliases).

(The original entry is located in requirement_list_directives.rst, line 10 and can be found [here](#).)

Must

A *optionallist* directive is required. It should produce a list of all OPTIONAL and MAY directives.

(The original entry is located in requirement_list_directives.rst, line 17 and can be found [here](#).)

Must

As per RFC 2119, users of this module should include a block of boilerplate near the beginning of their sphinx document. A *rfc2119interpretation* directive is required for this purpose.

(The original entry is located in introduction.rst, line 24 and can be found [here](#).)

3.2 Example Recommended List

The following list of recommendations includes every SHOULD, SHOULD NOT, RECOMMENDED etc.

Should

A *should* directive is important. It has semantic equivalence to the SHOULD keyword in RFC 2119.

(The original entry is located in requirement_directives.rst, line 51 and can be found [here](#).)

Recommended

A *recommended* directive is very useful. It is an alias for the *should* directive.

(The original entry is located in requirement_directives.rst, line 57 and can be found [here](#).)

Should Not

The absence of a *should_not* directive is not something we want. It is equivalent to the SHOULD NOT key word in RFC 2119.

(The original entry is located in requirement_directives.rst, line 63 and can be found [here](#).)

Not Recommended

Because the *not_recommended* is an alias for *should_not* directive, it's absence should be avoided.

(The original entry is located in requirement_directives.rst, line 69 and can be found [here](#).)

3.3 Example Optional List

Optional

It might be nice to have an *optional* directive, which would be equivalent of OPTIONAL keyword in RFC 2119.

(The original entry is located in requirement_directives.rst, line 78 and can be found [here](#).)

May

As an alias for the *optional* directive, a *may* directive might also be good to have.

(The original entry is located in requirement_directives.rst, line 84 and can be found [here](#).)

May

Optionally, this module should support governance parameters for documenting requirement status, valid from date, review date, etc.

None of these features are implemented yet, sorry. If there's something you actually need, please raise a ticket on GitHub.

(The original entry is located in introduction.rst, line 35 and can be found [here](#).)

Code

```

class sphinx_rfc2119.MandatoryListDirective(name, arguments, options, content,
                                             lineno, content_offset, block_text, state,
                                             state_machine)

    has_content = True

    run()

class sphinx_rfc2119.MayDirective(name, arguments, options, content, lineno, content_offset,
                                   block_text, state, state_machine)

    label = 'May'
    requirement_class = 'optional'

class sphinx_rfc2119.MustDirective(name, arguments, options, content, lineno, content_offset,
                                    block_text, state, state_machine)

    label = 'Must'
    requirement_class = 'mandatory'

class sphinx_rfc2119.MustNotDirective(name, arguments, options, content, lineno, con-
                                       tent_offset, block_text, state, state_machine)

    label = 'Must Not'
    requirement_class = 'mandatory'

class sphinx_rfc2119.NotRecommendedDirective(name, arguments, options, content,
                                              lineno, content_offset, block_text, state,
                                              state_machine)

    label = 'Not Recommended'
    requirement_class = 'recommendation'

class sphinx_rfc2119.OptionalDirective(name, arguments, options, content, lineno, con-
                                       tent_offset, block_text, state, state_machine)

    label = 'Optional'
    requirement_class = 'optional'

class sphinx_rfc2119.OptionalListDirective(name, arguments, options, content,
                                             lineno, content_offset, block_text, state,
                                             state_machine)

    has_content = True

    run()

```

```
class sphinx_rfc2119.RecommendationListDirective (name, arguments, options, content,
                                                lineno, content_offset, block_text,
                                                state, state_machine)

    has_content = True

    run ()

class sphinx_rfc2119.RecommendedDirective (name, arguments, options, content, lineno, con-
                                          tent_offset, block_text, state, state_machine)

    label = 'Recommended'

    requirement_class = 'recommendation'

class sphinx_rfc2119.RequiredDirective (name, arguments, options, content, lineno, con-
                                       tent_offset, block_text, state, state_machine)

    label = 'Required'

    requirement_class = 'mandatory'

class sphinx_rfc2119.ShallDirective (name, arguments, options, content, lineno, con-
                                    tent_offset, block_text, state, state_machine)

    label = 'Shall'

    requirement_class = 'mandatory'

class sphinx_rfc2119.ShallNotDirective (name, arguments, options, content, lineno, con-
                                       tent_offset, block_text, state, state_machine)

    label = 'Shall Not'

    requirement_class = 'mandatory'

class sphinx_rfc2119.ShouldDirective (name, arguments, options, content, lineno, con-
                                     tent_offset, block_text, state, state_machine)

    label = 'Should'

    requirement_class = 'recommendation'

class sphinx_rfc2119.ShouldNotDirective (name, arguments, options, content, lineno, con-
                                       tent_offset, block_text, state, state_machine)

    label = 'Should Not'

    requirement_class = 'recommendation'

sphinx_rfc2119.depart_rfc2119_node (self, node)

class sphinx_rfc2119.mandatory (rawsource='', *children, **attributes)

class sphinx_rfc2119.mandatorylist (rawsource='', *children, **attributes)

class sphinx_rfc2119.optional (rawsource='', *children, **attributes)

class sphinx_rfc2119.optionallist (rawsource='', *children, **attributes)

sphinx_rfc2119.process_rfc2119_nodes (app, doctree, fromdocname)

sphinx_rfc2119.purge_rfc2119_mandatory (app, env, docname)

sphinx_rfc2119.purge_rfc2119_optional (app, env, docname)

sphinx_rfc2119.purge_rfc2119_recommendation (app, env, docname)

class sphinx_rfc2119.recommendationlist (rawsource='', *children, **attributes)

class sphinx_rfc2119.recommended (rawsource='', *children, **attributes)
```



```
class sphinx_rfc2119.rfc2119Directive (name, arguments, options, content, lineno, con-  
                                     tent_offset, block_text, state, state_machine)  
    An abstract base for rfc2199 requirements.  
  
    has_content = True  
    label = 'rfc2119'  
    requirement_class = 'rfc2119'  
    run ()  
  
class sphinx_rfc2119.rfc2119InterpretationDirective (name, arguments, op-  
                                                         tions, content, lineno, con-  
                                                         tent_offset, block_text, state,  
                                                         state_machine)  
  
    has_content = False  
    run ()  
  
class sphinx_rfc2119.rfc2119interpretation (rawsource='', *children, **attributes)  
sphinx_rfc2119.setup (app)  
sphinx_rfc2119.visit_rfc2119_node (self, node)
```

Indices and tables

- `genindex`
- `modindex`
- `search`

S

`sphinx_rfc2119`, [11](#)

D

depart_rfc2119_node() (in module sphinx_rfc2119), 12

H

has_content (sphinx_rfc2119.MandatoryListDirective attribute), 11

has_content (sphinx_rfc2119.OptionalListDirective attribute), 11

has_content (sphinx_rfc2119.RecommendationListDirective attribute), 12

has_content (sphinx_rfc2119.rfc2119Directive attribute), 13

has_content (sphinx_rfc2119.rfc2119InterpretationDirective attribute), 13

L

label (sphinx_rfc2119.MayDirective attribute), 11

label (sphinx_rfc2119.MustDirective attribute), 11

label (sphinx_rfc2119.MustNotDirective attribute), 11

label (sphinx_rfc2119.NotRecommendedDirective attribute), 11

label (sphinx_rfc2119.OptionalDirective attribute), 11

label (sphinx_rfc2119.RecommendedDirective attribute), 12

label (sphinx_rfc2119.RequiredDirective attribute), 12

label (sphinx_rfc2119.rfc2119Directive attribute), 13

label (sphinx_rfc2119.ShallDirective attribute), 12

label (sphinx_rfc2119.ShallNotDirective attribute), 12

label (sphinx_rfc2119.ShouldDirective attribute), 12

label (sphinx_rfc2119.ShouldNotDirective attribute), 12

M

mandatory (class in sphinx_rfc2119), 12

mandatorylist (class in sphinx_rfc2119), 12

MandatoryListDirective (class in sphinx_rfc2119), 11

MayDirective (class in sphinx_rfc2119), 11

MustDirective (class in sphinx_rfc2119), 11

MustNotDirective (class in sphinx_rfc2119), 11

N

NotRecommendedDirective (class in sphinx_rfc2119), 11

O

optional (class in sphinx_rfc2119), 12

OptionalDirective (class in sphinx_rfc2119), 11

optionallist (class in sphinx_rfc2119), 12

OptionalListDirective (class in sphinx_rfc2119), 11

P

process_rfc2119_nodes() (in module sphinx_rfc2119), 12

purge_rfc2119_mandatory() (in module sphinx_rfc2119), 12

purge_rfc2119_optional() (in module sphinx_rfc2119), 12

purge_rfc2119_recommendation() (in module sphinx_rfc2119), 12

R

recommendationlist (class in sphinx_rfc2119), 12

RecommendationListDirective (class in sphinx_rfc2119), 11

recommended (class in sphinx_rfc2119), 12

RecommendedDirective (class in sphinx_rfc2119), 12

RequiredDirective (class in sphinx_rfc2119), 12

requirement_class (sphinx_rfc2119.MayDirective attribute), 11

requirement_class (sphinx_rfc2119.MustDirective attribute), 11

requirement_class (sphinx_rfc2119.MustNotDirective attribute), 11

requirement_class (sphinx_rfc2119.NotRecommendedDirective attribute), 11

requirement_class (sphinx_rfc2119.OptionalDirective attribute), 11

requirement_class (sphinx_rfc2119.RecommendedDirective attribute), 12

requirement_class (sphinx_rfc2119.RequiredDirective attribute), 12

requirement_class (sphinx_rfc2119.rfc2119Directive attribute), 13

requirement_class (sphinx_rfc2119.ShallDirective attribute), 12

requirement_class (sphinx_rfc2119.ShallNotDirective attribute), 12

requirement_class (sphinx_rfc2119.ShouldDirective attribute), 12

requirement_class (sphinx_rfc2119.ShouldNotDirective
attribute), [12](#)
rfc2119Directive (class in sphinx_rfc2119), [12](#)
rfc2119interpretation (class in sphinx_rfc2119), [13](#)
rfc2119InterpretationDirective (class in
sphinx_rfc2119), [13](#)
run() (sphinx_rfc2119.MandatoryListDirective
method), [11](#)
run() (sphinx_rfc2119.OptionalListDirective method),
[11](#)
run() (sphinx_rfc2119.RecommendationListDirective
method), [12](#)
run() (sphinx_rfc2119.rfc2119Directive method), [13](#)
run() (sphinx_rfc2119.rfc2119InterpretationDirective
method), [13](#)

S

setup() (in module sphinx_rfc2119), [13](#)
ShallDirective (class in sphinx_rfc2119), [12](#)
ShallNotDirective (class in sphinx_rfc2119), [12](#)
ShouldDirective (class in sphinx_rfc2119), [12](#)
ShouldNotDirective (class in sphinx_rfc2119), [12](#)
sphinx_rfc2119 (module), [11](#)

V

visit_rfc2119_node() (in module sphinx_rfc2119), [13](#)