Sphinx Tests Documentation

Release 0.6alpha1

Georg Brandl **someone else**

May 25, 2016

Table of Contents

Contents:

1 Extension API tests

Testing directives: **from function: Foofrom class: Bar**

2 Sphinx image handling

3 Image including source in subdir

4 Including in subdir

```
print("line 1")
```

```
print("line 2")
```

Absolute `/img.png` download.

This is an include file.

5 Testing downloadable files

Download `img.png` here. Download `this` there. Don't download `this`.

6 Test file and literal inclusion

```
# Literally included file using Python highlighting
# -*- coding: utf-8 -*-

foo = "Including Unicode characters: üöä"

class Foo:
    pass

class Bar:
    def baz():
        pass

def bar(): pass
```

Encoding 'utf-8-sig' used for reading included file u'/Users/tkomiya/work/sphinx/tests/build/latex_howto/wrongenc.inc' seems to be wrong, try giving an :encoding: option

```
This file is encoded in latin-1 but at first read as utf-8.

Max StrauSS aSS in München eine Leberkässemmel.
```

This file is encoded in latin-1 but at first read as utf-8.

Max StrauSS aSS in München eine Leberkässemmel.

7 Literalinclude options

```
class Foo:
    pass
```

```
    def baz():
        pass
```

```
6 class Foo:
7     pass
8 class Bar:
```

```
foo = "Including Unicode characters: üöä"
```

```
START CODE
# Literally included file using Python highlighting
# -*- coding: utf-8 -*-

foo = "Including Unicode characters: üöä"

class Foo:
    pass

class Bar:
    def baz():
        pass

def bar(): pass
END CODE
```

```
foo = "Including Unicode characters: üöä"

class Foo:
    pass

class Bar:
    def baz():
        pass

def bar(): pass
```

```
# Literally included file using Python highlighting
# -*- coding: utf-8 -*-

foo = "Including Unicode characters: üöä"
```

```
--- literal_orig.inc
+++ literal.inc
@@ -1,12 +1,12 @@
 # Literally included file using Python highlighting
```

```
 # -*- coding: utf-8 -*-

-foo = "Including Unicode characters: üöä"  # This will be changed
+foo = "Including Unicode characters: üöä"

-class FooOrig:
+class Foo:
     pass

-class BarOrig:
+class Bar:
     def baz():
         pass
```

```
Tabs include file test
----------------------

The next line has a tab:
-| |-
```

```
Tabs include file test
----------------------

The next line has a tab:
-|       |-
```

```
6 class Foo:
7     pass
```

```
6 class Foo:
7     pass
```

```
3 foo = "Including Unicode characters: üöä"
```

Test if dedenting before parsing works.

```
    def baz():
        pass
```

8 Docutils include with “literal”

While not recommended, it should work (and leave quotes alone).

```
Testing "quotes" in literal 'included' text.
```

9 Testing various markup

9.1 Meta markup

Section author: Georg Brandl

Module author: Georg Brandl

TOC

9.2 Generic reST

A global substitution (the definition is in rst_epilog).

```
some code
```

Option list:

-h help

–help also help

Line block:

line1
    line2
        line3
            line4
    line5
line6
    line7

Body directives

Title

Topic body.

Sidebar

Sidebar subtitle

Sidebar body.

Test rubric

Epigraph title

Epigraph body.

—Author

Highlights

Highlights body.

Pull-quote

Pull quote body.

a

b

`with some *markup* inside`

Admonitions

My Admonition

Admonition text.

Note: Note text.

Warning: Warning text.

Tip: Tip text.

Indirect hyperlink targets

9.3 Inline markup

Generic inline markup

Adding n to test unescaping.

- **`command\n`**
- *dfn\n*
- *guilabel with accelerator and \n*
- `kbd\n`
- *`mailheader\n`*
- **`makevar\n`**
- *`manpage\n`*
- *`mimetype\n`*
- *`newsgroup\n`*
- **`program\n`**
- `regexp\n`
- *File → Close\n*
- *File → Print*
- `a/varpart/b\n`
- `print i\n`

Linking inline markup

- **PEP 8**
- **Python Enhancement Proposal #8**
- **RFC 1**
- **Request for Comments #1**
- *`HOME`*
- *`with`*
- *`try statement`*
- *Admonitions*
- *here*
- *there*
- *My caption of the figure*
- *My caption of the figure*
- *my table*
- *my table*
- *my ruby code*
- *my ruby code*
- Fig. **??**

- Fig. **??**
- Table **??**
- Table **??**
- Listing **??**
- Listing **??**
- *Including in subdir*
- `:download:` is tested in includes.txt
- *`Python -c option`*
- This used to crash: `&option`

Test ABBR (abbreviation) and another ABBR.

Testing the index role, also available with explicit title.

9.4 With

(Empty section.)

9.5 Tables

Table 1: my table

1	• Block elems • In table	x
2	Empty cells:	

Table 2: empty cell in table header

1	2
3	4

Tables with multirow and multicol:

1	test!		c
2 y	col	col	
	multi-column cell		x

x 1

Fig. 1: My caption of the figure
My description paragraph of the figure.
Description paragraph is wraped with legend node.

9.6 Figures

Fig. 2: figure with align option

9.7 Version markup

New in version 0.6: Some funny **stuff**.

Changed in version 0.6: Even more funny stuff.

Deprecated since version 0.6: Boring stuff.

New in version 1.2: First paragraph of versionadded.

Changed in version 1.2: First paragraph of versionchanged.

Second paragraph of versionchanged.

Fig. 3: figure with align & figwidth option

9.8 Code blocks

Listing 9.1: my ruby code

```ruby
def ruby?
    false
end
```

```
import sys

sys.stdout.write('hello world!\n')
```

9.9 Misc stuff

Stuff [1]

Reference lookup: *[Ref1]* (defined in another file). Reference lookup underscore: *[Ref_1]*

See also:

something, something else, something more

Google For everything.

- This
- is
- a horizontal
- list
- with several
- items

Side note

This is a side note.

This tests `role names in uppercase`.

LICENSE AGREEMENT

Terry Pratchett, Tolkien, Monty Python.

änhlich Dinge

boson Particle with integer spin.

fermion Particle with half-integer spin.

tauon

myon

electron Examples for fermions.

über Gewisse

```
try_stmt   ::=   try1_stmt | try2_stmt
try1_stmt  ::=   "try" ":"  suite
```

[1] Like footnotes.

```
                  ("except" [expression ["," target]] ":"  suite)+
                  ["else" ":"  suite]
                  ["finally" ":"  suite]
try2_stmt  ::=    "try" ":"  suite
                  "finally" ":"  suite
```

9.10 Index markup

Invalid index markup...

Main

9.11 Ö... Some strange characters

Testing öää...

9.12 Only directive

In LaTeX.

In both.

9.13 Any role

Test referencing to *headings* and *objects*. Also *modules* and *classes*.

More domains:

- *JS*
- *C*
- *myobj* (user markup)
- *n::Array*
- *perl -c*

10 Testing object descriptions

func_without_module(*a*, *b*, **c*[, *d*])
Does something.

func_without_body()

func_with_unknown_field()
: :

: empty field name:

Field_name

Field_name all lower

FIELD_NAME

FIELD_NAME ALL CAPS

Field_Name

Field_Name All Word Caps

Field_name

Field_name First word cap

FIELd_name

FIELd_name PARTial caps

func_noindex()

foolib.**func_with_module**()

Referring to `func with no index`. Referring to `nothing`.

mod.**func_in_module**()

class mod.**Cls**

meth1()

static meths()

attr

Cls.**meth2**()

exception errmod.**Error**(*arg1*, *arg2*)

mod.**var**

func_without_module2() → annotation

long(parameter, list)

another one

class TimeInt

Has only one parameter (triggers special behavior...)

Parameters **moo** (Moo) – Moo

class Time(*hour*, *minute*, *isdst*)

Parameters

- **year** (TimeInt) – The year.
- **minute** (TimeInt) – The minute.
- **isdst** – whether it's DST
- **hour** (*DuplicateType*) – Some parameter
- **hour** – Duplicate param. Should not lead to crashes.
- **extcls** (Cls) – A class from another module.

Returns a new *Time* instance

Return type *Time*

Raises **ValueError** – if the values are out of range

Variables

- **hour** (*int*) – like *hour*
- **minute** (*int*) – like *minute*

11 C items

Sphinx_DoSomething()

SphinxStruct.member

SPHINX_USE_PYTHON

SphinxType

sphinx_global

12 Javascript items

foo()

bar

bar.**baz**(*href*, *callback*[, *errback*])

Arguments

- **href** (*string*) – The location of the resource.
- **callback** – Get's called with the data returned by the resource.

Throws **InvalidHref** – If the *href* is invalid.

Returns *undefined*

bar.**spam**

13 References

Referencing *mod.Cls* or *mod.Cls* should be the same.

With target: *Sphinx_DoSomething()* (parentheses are handled), *SphinxStruct.member*, *SPHINX_USE_PYTHON*, *SphinxType ** (pointer is handled), *sphinx_global*.

Without target: CFunction(). malloc().

foo() foo()

bar bar.baz() bar.baz() baz()

bar.baz

14 Others

HOME

-c command

-c

+p

arg

Link to *perl +p* and *arg*

commit

-p

Link to *hg commit* and *git commit -p*.

15 User markup

myobj (*parameter*)
Description of userdesc.

Referencing *myobj*.

16 CPP domain

class n::**Array**

T &**operator[]**(unsigned *j*)

const T &**operator[]**(unsigned *j*) **const**

17 File with UTF-8 BOM

This file has a UTF-8 “BOM”.

18 Test math extensions $E = mc^2$

This is inline math: $a^2 + b^2 = c^2$.

$$a^2 + b^2 = c^2$$

$$a + 1 < b$$

$$e^{i\pi} = 1 \tag{1}$$

$$e^{ix} = \cos x + i \sin x$$

$$n \in \mathbb{N}$$

a + 1 < b Referencing equation (**??**).

19 Autodoc tests

Just testing a few autodoc possibilities...

19.1 Sphinx test suite utilities

copyright Copyright 2007-2016 by the Sphinx team, see AUTHORS.

license BSD, see LICENSE for details.

19.2 test_autodoc

Test the autodoc extension. This tests mainly the Documenters; the auto directives are tested in a test source file translated by test_build.

copyright Copyright 2007-2016 by the Sphinx team, see AUTHORS.

license BSD, see LICENSE for details.

class test_autodoc.**Class**(*arg*)
Class to document.

attr = 'bar'
should be documented – süSS

descr
Descriptor instance docstring.

docattr = 'baz'
should likewise be documented – süSS

excludemeth()
Method that should be excluded.

inst_attr_comment = None
a documented instance attribute

inst_attr_inline = None
an inline documented instance attr

inst_attr_string = None
a documented instance attribute

mdocattr = <StringIO.StringIO instance>
should be documented as well - süSS

meth()
Function.

classmethod moore(*a*, *e*, *f*) → happiness

prop
Property.

skipmeth()
Method that should be skipped.

udocattr = 'quux'
should be documented as well - süSS

test_autodoc.**function**(*foo*, **args*, ***kwds*)
Return spam.

class test_autodoc.**Class**(*arg*)
Class to document.

Additional content.

attr = 'bar'
should be documented – süSS

descr
Descriptor instance docstring.

docattr = 'baz'
should likewise be documented – süSS

excludemeth()
Method that should be excluded.

inheritedmeth()
Inherited function.

inst_attr_comment = None
a documented instance attribute

inst_attr_inline = None
an inline documented instance attr

inst_attr_string = None
a documented instance attribute

mdocattr = <StringIO.StringIO instance>
should be documented as well - süSS

meth()
Function.

classmethod moore(*a*, *e*, *f*) → happiness

prop
Property.

skipmeth()
Method that should be skipped.

udocattr = 'quux'
should be documented as well - süSS

class test_autodoc.**Outer**
Foo

class Inner
Foo

meth()
Foo

Class.**docattr = 'baz'**
should likewise be documented – süSS

exception test_autodoc.**CustomEx**
My custom exception.

f()
Exception method.

class test_autodoc.**CustomDict**
Bases: dict

Docstring.

class autodoc_fodder.**MarkupError**

Note: This is a docstring with a

Explicit markup ends without a blank line; unexpected unindent.

small markup error which should have correct location information.

class test_autodoc.**InstAttCls**
Class with documented class and instance attributes.

All members (5 total)

ca1 = 'a'
Doc comment for class attribute InstAttCls.ca1. It can have multiple lines.

ca2 = 'b'
Doc comment for InstAttCls.ca2. One line only.

ca3 = 'c'
Docstring for class attribute InstAttCls.ca3.

ia1 = None
Doc comment for instance attribute InstAttCls.ia1

ia2 = None
Docstring for instance attribute InstAttCls.ia2.

class test_autodoc.**InstAttCls**
Class with documented class and instance attributes.

Specific members (2 total)

ca1 = 'a'
Doc comment for class attribute InstAttCls.ca1. It can have multiple lines.

ia1 = None
Doc comment for instance attribute InstAttCls.ia1

Dedication

For Docutils users & co-developers.

Abstract

This document is a demonstration of the reStructuredText markup language, containing examples of all basic reStructuredText constructs and many advanced constructs.

20 reStructuredText Demonstration

20.1 Examples of Syntax Constructs

21 Test for diverse extensions

21.1 extlinks

Test diverse links: issue 1000 and http://python.org/dev/, also with explicit caption.

21.2 todo

list of all todos

22 Testing footnote and citation

22.1 numbered footnote

[1]

22.2 auto-numbered footnote

[2]

22.3 named footnote

[3]

22.4 citation

[bar]

22.5 footnotes in table

Table 4: Table caption [4]

name [5]	desription
VIDIOC_CROPCAP	Information about VIDIOC_CROPCAP

[1] numbered
[2] auto numbered
[3] named
[4] footnotes in table caption
[5] footnotes in table

22.6 footenotes

Citations

22.7 missing target

[missing] citation

23 Various kinds of lists

23.1 nested enumerated lists

1. one
2. two

(a) two.1

(b) two.2

3. three

23.2 enumerated lists with non-default start values

0. zero
1. one

---

1. one
2. two

---

2. two
3. three

23.3 enumerated lists using letters

1. a
2. b
3. c
4. d

---

24. x
25. y
26. z
27. {

23.4 definition lists

term1 description

term2 (stronged partially) description

24 Generated section

25 Indices and tables

- genindex
- modindex
- search

26 References

27 Test for issue #1157

This used to crash:

28 Test for issue #1700

Table of Contents

29 Test for indirect hyperlink targets

indirect hyperref

References

[bar] cite

[Ref1] Reference target.

[Ref_1] Reference target 2.

Python Module Index

a

m

t

u