sphinxcontrib-bibtex Documentation

Release 2.3.0

Matthias C. M. Troffaes

Jun 01, 2021

CONTENTS

1 Contents **3**

1.1 Getting Started . . . 3
1.2 Usage . . . 4
1.3 Extension API . . . 16
1.4 Changes . . . 38
1.5 License . . . 45
1.6 Related Projects . . . 45

2 Indices and tables **47**

Bibliography **49**

Python Module Index **51**

Index **53**

Release 2.3.0

Date Jun 01, 2021

CHAPTER

ONE

CONTENTS

1.1 Getting Started

1.1.1 Overview

Sphinx extension for BibTeX style citations.

The bibtex extension allows BibTeX citations to be inserted into documentation generated by Sphinx, via a `bibliography` directive, along with `:cite:p:` and `:cite:t:` roles. These work similarly to LaTeX's `thebibliography` environment and the `\citet` and `\citep` commands.

For formatting, the extension relies on pybtex written by Andrey Golovizin. The extension is inspired by Matthew Brett's bibstuff.sphinxext.bibref and Weston Nielson's sphinx-natbib.

- Download: https://pypi.org/project/sphinxcontrib-bibtex/#files
- Documentation: https://sphinxcontrib-bibtex.readthedocs.io/en/latest/
- Development: https://github.com/mcmtroffaes/sphinxcontrib-bibtex/

1.1.2 Installation

Install the module with `pip install sphinxcontrib-bibtex`, or from source using `python setup.py install`. Then add:

```
extensions = ['sphinxcontrib.bibtex']
bibtex_bibfiles = ['refs.bib']
```

to your project's Sphinx configuration file `conf.py`.

1.1.3 Minimal Example

In your project's documentation, you can use `:cite:t:` for textual citation references, `:cite:p:` for parenthetical citation references, and `.. bibliography::` for inserting the bibliography. For example:

```
See :cite:t:`1987:nelson` for an introduction to non-standard analysis.
Non-standard analysis is fun :cite:p:`1987:nelson`.

.. bibliography::
```

where `refs.bib` would contain an entry:

```
@Book{1987:nelson,
  author = {Edward Nelson},
  title = {Radically Elementary Probability Theory},
  publisher = {Princeton University Press},
  year = {1987}
}
```

In the default style, this will get rendered as:

See Nelson [Nel87a] for an introduction to non-standard analysis. Non-standard analysis is fun [Nel87a].

Citations in sphinx are resolved globally across all documents. Typically, you have a single *bibliography* directive across your entire project which collects all citations. Advanced use cases with multiple *bibliography* directives across your project are also supported, but some care needs to be taken from your end to avoid duplicate citations.

In contrast, footnotes in sphinx are resolved locally per document. To achieve local bibliographies per document, you can use citations represented by footnotes as follows:

```
Non-standard analysis is lovely. :footcite:`1987:nelson`

.. footbibliography::
```

which will get rendered as:

Non-standard analysis is lovely.[1]

Typically, you have a single *footbibliography* directive at the bottom of each document that has *footcite* citations. Advanced use cases with multiple *footbibliography* directives per document are also supported. Since everything is local, there is no concern with duplicate citations when using footnotes.

1.2 Usage

1.2.1 Configuration

Bibliography Files and Encoding

New in version 2.0.0.

To configure the extension, in your `conf.py` file, set `bibtex_bibfiles` to your list of bib files. For instance, a minimal configuration may look as follows:

```
extensions = ['sphinxcontrib.bibtex']
bibtex_bibfiles = ['refs.bib']
```

In bib files, LaTeX control characters are automatically converted to unicode characters (for instance, to convert `\'e` into `é`). Be sure to write `\%` when you intend to format a percent sign.

You can set the encoding of the bibliography files, using the `bibtex_encoding` variable in your `conf.py`. If no encoding is specified, `utf-8-sig` is assumed. For example:

```
bibtex_encoding = 'latin'
```

[1] Edward Nelson. *Radically Elementary Probability Theory*. Princeton University Press, 1987.

Bibliography Style

You can change the bibliography style, using the `bibtex_default_style` variable in your `conf.py`. If none is specified, the `alpha` style is used. Other supported styles are `plain`, `unsrt`, and `unsrtalpha`. You can also create your own style (see *Custom Formatting, Sorting, and Labelling*). For example:

```
bibtex_default_style = 'unsrt'
```

Referencing Style

New in version 2.2.0.

You can change the inline referencing style (i.e. the formatting of the citation references themselves) using the `bibtex_reference_style` variable in your `conf.py`. Currently available built-in styles are:

- `label`: Use the labels generated by the bibliography style. Similar to natbib's `numbers` style and biblatex's `numeric` and `alphabetic` styles (depending on the labelling style of your bibliography style). This is the default style.
- `author_year`: Use the author and year. Similar to natbib's and biblatex's `authoryear` style. Note that this does not remove labels from bibliographies. This is because, in docutils, every citation must have a label.
- `super`: Use the labels generated by the bibliography style as superscripts. This works best with numeric bibliography styles such as `plain`. Similar to natbib's `super` style and biblatex's `\supercite` command.

The inline referencing style for footnote citations can be configured through the `bibtex_foot_reference_style` variable in your `conf.py`. Currently available built-in styles are:

- `foot`: Use footnotes for parenthetical citations, and author with footnote for a textual citations. This is the default style (and currently also the only built-in style).

Python packages can make new styles available through the `sphinxcontrib.bibtex.style.referencing` entry point group. See sphinxcontrib-bibtex's own `setup.py` script for examples.

1.2.2 Roles and Directives

`:cite:p:`

New in version 2.2.0.

Create a parenthetical citation reference to a bibliographic entry. This will put the citation reference information (author and year, or label, depending on the style) between brackets. Similar to natbib's `\citep` command, or biblatex's `\parencite` command. For example:

```
We will make use of non-standard analysis :cite:p:`1987:nelson`.
```

which would be equivalent to the following LaTeX code:

```
We will make use of non-standard analysis \citep{1987:nelson}.
```

Multiple keys can be specified at once:

```
I love analysis :cite:p:`1987:nelson,2001:schechter`!
```

`:cite:t:`

New in version 2.2.0.

Create a textual citation. This will typically render the name of the first author followed by the year or by the label, depending on the citation reference style. Similar to natbib's `\citet` command, or biblatex's `\textcite` command. For example:

```
See :cite:t:`1987:nelson` for an introduction to non-standard analysis.
```

which would be equivalent to the following LaTeX code:

```
See \citet{1987:nelson} for an introduction to non-standard analysis.
```

Here too, multiple keys can be specified at once.

`:cite:ps:`

`:cite:ts:`

`:cite:ct:`

`:cite:cts:`

New in version 2.2.0.

All these roles modify *`cite:p`* and *`cite:t`*. The ones starting with `c` will capitalize the first letter. The ones ending with `s` will give the full author list.

`:cite:`

This is an alias for the *`cite:p`* role, and will create a parenthetical citation reference. Provided for convenience and compatibility with older versions.

`:cite:label:`

`:cite:labelpar:`

New in version 2.2.0.

Create a citation using just the label. Use the `par` version to include brackets.

`:cite:year:`

`:cite:yearpar:`

New in version 2.2.0.

Create a citation using just the year. Use the `par` version to include brackets.

`:cite:author:`

`:cite:authors:`

`:cite:authorpar:`

`:cite:authorpars:`

`:cite:cauthor:`

`:cite:cauthors:`

New in version 2.2.0.

Create a citation using just the author(s). Use the `par` version to include brackets, and the `c` version to capitalize the first letter.

`:cite:empty:`

New in version 2.3.0.

Register a citation key as being cited without generating a reference, similar to LaTeX's nocite command.

.. bibliography::

Create bibliography for all cited references. Citations in sphinx are resolved globally across all documents. Typically, you have a single bibliography directive across your entire project which collects all citations. Citation keys can also be explicitly listed under the directive; see *Listing Citation Keys*.

> **Warning:** Sphinx will attempt to resolve references to the bibliography across all documents, so you must take care that no citation key is included more than once.

The following options are recognized (all are optional).

:all:

Include all references, instead of just the cited ones (equivalent to `\nocite{*}` in LaTeX). For example:

```
.. bibliography::
   :all:
```

:notcited:

Causes all references that were not cited to be included. Listed references remain included.

:cited:

This is the default and need not be specified.

:style:

Overrides the default bibliography style. For example:

```
.. bibliography::
   :style: unsrt
```

:list:

:enumtype:

:start:

See *Bullet Lists and Enumerated Lists*.

:labelprefix:

See *Label Prefixing*.

:keyprefix:

See *Key Prefixing*.

:filter:

See *Filtering*. Note that listed references are always included, regardless of any filtering.

:footcite:p:

New in version 2.3.0.

Create a parenthetical footnote reference to a bibliographic entry. For example:

```
We will make use of non-standard analysis :footcite:p:`1987:nelson`.
```

which would be equivalent to the following LaTeX code:

```
We will make use of non-standard analysis \footcite{1987:nelson}.
```

As with all citation roles, multiple keys can be specified:

```
I love analysis :footcite:p:`1987:nelson,2001:schechter`!
```

:footcite:t:

New in version 2.3.0.

Create a textual footnote reference to a bibliographic entry. For example:

```
See :footcite:t:`1987:nelson` for an introduction to non-standard analysis.
```

which would be equivalent to the following LaTeX code:

```
See Nelson\footcite{1987:nelson} for an introduction to non-standard analysis.
```

Here too, multiple keys can be specified at once.

:footcite:ps:

:footcite:ts:

:footcite:ct:

:footcite:cts:

New in version 2.3.0.

All these roles modify *footcite:p* and *footcite:t*. The ones starting with `c` will capitalize the first letter. The ones ending with `s` will give the full author list.

:footcite:

New in version 2.0.0.

This is an alias for the *footcite:p* role, and will create a parenthetical footnote citation reference. Provided for convenience and compatibility with older versions.

.. footbibliography::

New in version 2.0.0.

Create footnotes at this location for all references that are cited in the current document up to this point. Typically, you have a single footbibliography directive at the bottom of each document that has footcite citations.

If specified multiple times in the same document, footnotes are only created for references that do not yet have a footnote earlier in the document.

1.2.3 Advanced Features

Splitting Bibliographies Per Bib File

New in version 2.0.0.

If you want multiple bibliographies each of which only contains references from specific bib files, you can specify the relevant bib files as an optional argument to the directive.

The next example shows how to split your citations between articles and books, assuming your articles are in `articles.bib` and your books are in `books1.bib` and `books2.bib`.

```
.. rubric:: Articles

.. bibliography:: articles.bib

.. rubric:: Books

.. bibliography:: books1.bib books2.bib
```

The bib files must be specified as a path that is relative to the containing document.

Bullet Lists and Enumerated Lists

New in version 0.2.4.

You can change the type of list used for rendering the bibliography. By default, a paragraph of standard citations is generated. However, instead, you can also generate a bullet list, or an enumerated list.

```
.. bibliography::
   :list: bullet
   :all:

.. bibliography::
   :list: enumerated
   :all:
```

Note that citations to these types of bibliography lists will not be resolved.

For enumerated lists, you can also specify the type (default is `arabic`), and the start of the sequence (default is 1).

```
.. bibliography::
   :list: enumerated
   :enumtype: upperroman
   :start: 3
   :all:
```

The enumtype can be any of `arabic` (1, 2, 3, . . .), `loweralpha` (a, b, c, . . .), `upperalpha` (A, B, C, . . .), `lowerroman` (i, ii, iii, . . .), or `upperroman` (I, II, III, . . .).

The start can be any positive integer (1, 2, 3, . . .) or `continue` if you wish the enumeration to continue from the last *`bibliography`* directive. This is helpful if you split up your bibliography but still want to enumerate the entries continuously.

Listing Citation Keys

New in version 2.3.0.

If you have many citations to include that are not referenced anywhere, then instead of using *`cite:empty`* it can be more convenient to simply list the citation keys directly under the bibliography directive where you want them to appear. Such references can be listed by having one bibtex key per line under the directive. The keys should not have a key prefix if you are using that option (see *Key Prefixing*). For example:

```
.. bibliography::

   nelson1987
   boole1854
```

This would cause the bibliography to generate citations for all cited references, in addition to citations with bibtex keys `nelson1987` and `boole1854`. The listed keys are always included regardless of filtering. So, if you only want the listed keys to be included, you can use the `:filter:  False` option:

```
.. bibliography::
   :filter: False

```

(continues on next page)

(continued from previous page)

```
   nelson1987
   boole1854
```

See *Filtering* for more information on filtering.

Label Prefixing

New in version 0.2.5.

If you have multiple bibliographies, and experience duplicate labels, use the `labelprefix` option.

```
.. rubric:: References

.. bibliography::
   :cited:
   :labelprefix: A

.. rubric:: Further reading

.. bibliography::
   :notcited:
   :labelprefix: B
```

Key Prefixing

New in version 0.3.3.

If you have multiple bibliographies, and you would like entries to be repeated in different documents, then use the `keyprefix` option.

For example, suppose you have two documents, and you would like to cite `boole1854` in both of these documents, with the bibliography entries showing in both of the documents. In one document you could have:

```
See :cite:`a-boole1854`

.. bibliography::
   :labelprefix: A
   :keyprefix: a-
```

whilst in the other document you could have:

```
See :cite:`b-boole1854`

.. bibliography::
   :labelprefix: B
   :keyprefix: b-
```

The bibliographies will then both generate an entry for `boole1854`, with links and backlinks as expected.

If you list citation keys, you should include those *without* key prefix. For example:

```
.. bibliography::
   :labelprefix: B
   :keyprefix: b-

   nelson1987
```

See also:

Local Bibliographies

Filtering

New in version 0.2.7.

Whilst the `cited`, `all`, and `notcited` options along with *Listing Citation Keys* will cover many use cases, sometimes more advanced selection of bibliographic entries is desired. For this purpose, you can use the `filter` option:

```
.. bibliography::
   :list: bullet
   :filter: author % "Einstein"
```

The string specified in the filter option must be a valid Python expression.

Note: The expression is parsed using `ast.parse()` and then evaluated using an `ast.NodeVisitor`, so it should be reasonably safe against malicious code.

The filter expression supports:

- The boolean operators `and`, `or`.
- The unary operator `not`.
- The comparison operators ==, <=, <, >=, and >.
- Regular expression matching using the `%` operator, where the left hand side is the string to be matched, and the right hand side is the regular expression. Matching is case insensitive. For example:

  ```
  .. bibliography::
     :list: bullet
     :filter: title % "relativity"
  ```

  would include all entries that have the word "relativity" in the title.

  Note: The implementation uses `re.search()`.

- Single and double quoted strings, such as `'hello'` or `"world"`.
- Set literals, such has `{"hello", "world"}`, as well as the set operators `&`, `|`, `in`, and `not in`.

  New in version 0.3.0.

- Various identifiers, such as:
  - `type` is the entry type, as a lower case string (i.e. `"inproceedings"`).
  - `key` is the entry key, as a lower case string (this is because keys are considered case insensitive).

- `cited` evaluates to `True` if the entry was cited in the document, and to `False` otherwise.
- `docname` evaluates to the name of the current document.

  New in version 0.3.0.
- `docnames` evaluates to a set of names from which the entry is cited.

  New in version 0.3.0.
- `True` and `False`.
- `author` is the entry string of authors in standard format (last, first), separated by "and".
- `editor` is similar to `author` but for editors.
- Any other (lower case) identifier evaluates to a string containing the value of the correspondingly named field, such as `title`, `publisher`, `year`, and so on. If the item is missing in the entry then it evaluates to the empty string. Here is an example of how one would typically write an expression to filter on an optional field:

  ```
  .. bibliography::
     :list: bullet
     :filter: cited and year and (year <= "2003")
  ```

  which would include all cited entries that have a year that is less or equal than 2003; any entries that do not specify a year would be omitted.

Local Bibliographies

The easiest way to have a local bibliography per document is to use *footcite* along with *footbibliography*.

If you prefer to have regular citations instead of footnotes, both the `keyprefix` and `filter` options can be used to achieve local bibliographies with *cite* and *bibliography*.

The `filter` system for local bibliographies can only be used if no citation key is used in more than one document. This is not always satisfied. If you need to cite the same reference in multiple documents with references to multiple local bibliographies, use the `keyprefix` system; see *Key Prefixing*.

To create a bibliography that includes only citations that were cited in the current document, use the following filter:

```
.. bibliography::
   :filter: docname in docnames
```

More generally, you can create bibliographies for citations that were cited from specific documents only:

```
.. bibliography::
   :filter: {"doc1", "doc2"} & docnames
```

This bibliography will include all citations that were cited from `doc1.rst` or `doc2.rst`. Another hypothetical example:

```
.. bibliography::
   :filter: cited and ({"doc1", "doc2"} >= docnames)
```

This bibliography will include all citations that were cited in `doc1.rst` or `doc2.rst`, but nowhere else.

Custom Formatting, Sorting, and Labelling

pybtex provides a very powerful way to create and register new styles, using setuptools entry points, as documented here: https://docs.pybtex.org/api/plugins.html

Simply add the following code to your conf.py:

```
import pybtex.plugin
from pybtex.style.formatting.unsrt import Style as UnsrtStyle
from pybtex.style.template import toplevel # ... and anything else needed

class MyStyle(UnsrtStyle):
    def format_XXX(self, e):
        template = toplevel [
            # etc.
        ]
        return template.format_data(e)

pybtex.plugin.register_plugin('pybtex.style.formatting', 'mystyle', MyStyle)
```

Now mystyle will be available to you as a formatting style:

```
bibtex_default_style = 'mystyle'
```

An minimal example is available here: https://github.com/mcmtroffaes/sphinxcontrib-bibtex/tree/develop/test/roots/test-bibliography_style_nowebref

The formatting code uses a very intuitive template engine. The source code for unsrt provides many great examples: https://bitbucket.org/pybtex-devs/pybtex/src/master/pybtex/style/formatting/unsrt.py?at=master&fileviewer=file-view-default

The above example only demonstrates a custom formatting style plugin. It is also possible to register custom author/editor naming plugins (using the pybtex.style.names group), labelling plugins (using the pybtex.style.labels group), and sorting plugins (using the pybtex.style.sorting group). A few minimal examples demonstrating how to create custom label styles are available here:

- https://github.com/mcmtroffaes/sphinxcontrib-bibtex/tree/develop/test/roots/test-bibliography_style_label_1
- https://github.com/mcmtroffaes/sphinxcontrib-bibtex/tree/develop/test/roots/test-bibliography_style_label_2

Custom Inline Citation References

You can create and register your own referencing styles. For instance, say we wish to use the author-year style with round brackets instead of the default square brackets. Simply add the following code to your conf.py:

```
import dataclasses
import sphinxcontrib.bibtex.plugin

from sphinxcontrib.bibtex.style.referencing import BracketStyle
from sphinxcontrib.bibtex.style.referencing.author_year \
    import AuthorYearReferenceStyle

my_bracket_style = BracketStyle(
    left='(',
    right=')',
```

(continues on next page)

(continued from previous page)

```
)


@dataclasses.dataclass
class MyReferenceStyle(AuthorYearReferenceStyle):
    bracket_parenthetical: BracketStyle = my_bracket_style
    bracket_textual: BracketStyle = my_bracket_style
    bracket_author: BracketStyle = my_bracket_style
    bracket_label: BracketStyle = my_bracket_style
    bracket_year: BracketStyle = my_bracket_style


sphinxcontrib.bibtex.plugin.register_plugin(
    'sphinxcontrib.bibtex.style.referencing',
    'author_year_round', MyReferenceStyle)
```

Warning: You must decorate your style as a dataclass, and **include a type annotation with every field**, to ensure these values are correctly passed to the constructor when sphinxcontrib-bibtex instantiates your style.

Now `author_year_round` will be available to you as a formatting style:

```
bibtex_reference_style = 'author_year_round'
```

An minimal example is available here: https://github.com/mcmtroffaes/sphinxcontrib-bibtex/tree/develop/test/roots/test-citation_style_round_brackets

Custom Bibliography Header

By default, the *bibliography* and *footbibliography* directives simply insert a paragraph. The `bibtex_bibliography_header` and `bibtex_footbibliography_header` configuration variables can be set to add a header to this. For example, in your `conf.py` you could have:

```
bibtex_bibliography_header = ".. rubric:: References"
bibtex_footbibliography_header = bibtex_bibliography_header
```

This adds a rubric title to every bibliography.

1.2.4 Known Issues and Workarounds

Encoding: Percent Signs

Be sure to write \% for percent signs at all times (unless your file contains a genuine comment), otherwise the parser will ignore the remainder of the line.

Duplicate Labels When Using `:style: plain`

With `:style: plain`, labels are numeric, restarting at `[1]` for each *`bibliography`* directive. Consequently, when inserting multiple *`bibliography`* directives with `:style: plain`, you are bound to get duplicate labels for entries. There are a few ways to work around this problem:

- Use a single bibliography directive for all your references.
- Use the `labelprefix` option, as documented above.
- Use a style that has non-numeric labelling, such as `:style: alpha`.

LaTeX Backend Fails with Citations In Figure Captions

Sphinx generates `\phantomsection` commands for references, however LaTeX does not support these in figure captions. You can work around this problem by adding the following code to your `conf.py`:

```
latex_elements = {
 'preamble': r'''
     % make phantomsection empty inside figures
     \usepackage{etoolbox}
     \AtBeginEnvironment{figure}{\renewcommand{\phantomsection}{}}
 '''
}
```

Mismatch Between Output of HTML/Text and LaTeX Backends

Sphinx's LaTeX writer currently collects all citations together, and puts them on a separate page, with a separate title, whereas the html and text writers puts citations at the location where they are defined. This issue will occur also if you use regular citations in Sphinx: it has nothing to do with sphinxcontrib-bibtex per se.

To get a closer match between the two outputs, you can tell Sphinx to generate a rubric title only for html or text outputs:

```
.. only:: html or text

   .. rubric:: References

.. bibliography::
```

This code could be placed in a `references.rst` file that you include at the end of your toctree.

Alternatively, to remove the bibliography section title from the LaTeX output, you can add the following to your LaTeX preamble:

```
\usepackage{etoolbox}
\patchcmd{\thebibliography}{\section*{\refname}}{}{}{}
```

1.3 Extension API

1.3.1 Sphinx Interface

`sphinxcontrib.bibtex.`**`setup`**(*app: sphinx.application.Sphinx*) → Dict[str, Any]

Set up the bibtex extension:

- register config values
- register directives
- register nodes
- register roles
- register transforms
- connect events to functions

1.3.2 New Roles

`class` `sphinxcontrib.bibtex.roles.`**`CitationRef`**(*citation_ref_id: str*, *docname: str*, *line: int*, *keys: List[str]*)

Information about a citation reference.

`property citation_ref_id`

Unique id of this citation reference.

`property docname`

Document name.

`property keys`

Citation keys (including key prefix).

`property line`

Line number.

`class` `sphinxcontrib.bibtex.roles.`**`CiteRole`**(*fix_parens: bool = False*, *lowercase: bool = False*, *nodeclass: Optional[Type[docutils.nodes.Element]] = None*, *innernodeclass: Optional[Type[docutils.nodes.TextElement]] = None*, *warn_dangling: bool = False*)

Bases: `sphinx.roles.XRefRole`

Class for processing the *`cite`* role.

`result_nodes`(*document*, *env*, *node*, *is_ref*)

Associate the pending_xref with the cite domain, and note the cited citation keys.

`class` `sphinxcontrib.bibtex.foot_roles.`**`FootCiteRole`**(*fix_parens: bool = False*, *lowercase: bool = False*, *nodeclass: Optional[Type[docutils.nodes.Element]] = None*, *innernodeclass: Optional[Type[docutils.nodes.TextElement]] = None*, *warn_dangling: bool = False*)

Bases: `sphinx.roles.XRefRole`

Class for processing the *`footcite`* role.

result_nodes(*document: docutils.nodes.document*, *env: BuildEnvironment*, *node: docutils.nodes.Element*, *is_ref: bool*) → Tuple[List[docutils.nodes.Node], List[docutils.nodes.system_message]]

Transform node into footnote references, and add footnotes to a node stored in the environment's temporary data if they are not yet present.

See also:

The node containing all footnotes is inserted into the document by `foot_directives.FootBibliographyDirective.run()`.

1.3.3 New Docutils Nodes

`class sphinxcontrib.bibtex.nodes.`**bibliography**(*rawsource=''*, **children*, ***attributes*)

Node for representing a bibliography. Replaced by a list of citations by `BibliographyTransform`.

`class sphinxcontrib.bibtex.nodes.`**raw_latex**(*rawsource=''*, *text=''*, **children*, ***attributes*)

Node for representing raw latex data.

`sphinxcontrib.bibtex.nodes.`**visit_raw_latex**(*self: sphinx.writers.latex.LaTeXTranslator*, *node: sphinxcontrib.bibtex.nodes.raw_latex*)

Called when entering a raw_latex node. Appends the node's raw source to the latex body.

`sphinxcontrib.bibtex.nodes.`**depart_raw_latex**(*self: sphinx.writers.latex.LaTeXTranslator*, *node: sphinxcontrib.bibtex.nodes.raw_latex*)

Called when leaving a raw_latex node.

1.3.4 New Docutils Directives

`class sphinxcontrib.bibtex.directives.`**BibliographyKey**(*docname: str*, *id_: str*)

Unique key for each bibliography directive.

property docname

Name of the document where the bibliography resides.

property id_

The id of the bibliography node in the document.

`class sphinxcontrib.bibtex.directives.`**BibliographyValue**(*line: int*, *bibfiles: List[str]*, *style: str*, *list_: str*, *enumtype: str*, *start: int*, *labelprefix: str*, *keyprefix: str*, *filter_: _ast.AST*, *citation_nodes: Dict[str, docutils.nodes.citation]*, *keys: List[str]*)

Contains information about a bibliography directive.

property bibfiles

List of bib files for this directive.

property citation_nodes

key -> citation node

property enumtype

The sequence type (for enumerated lists).

property filter_

Parsed filter expression.

property keyprefix

String prefix for citation keys.

`property keys`
: Keys listed as content of the directive.

`property labelprefix`
: String prefix for pybtex generated labels.

`property line`
: Line number of the directive in the document.

`property list_`
: The list type.

`property start`
: The start of the sequence (for enumerated lists).

`property style`
: The pybtex style.

`class sphinxcontrib.bibtex.directives.BibliographyDirective`(*name*, *arguments*, *options*, *content*, *lineno*, *content_offset*, *block_text*, *state*, *state_machine*)

Class for processing the *bibliography* directive.

Produces a *bibliography* node, along with (empty) citation nodes that will be formatted later in the *env-updated* stage, and inserted into the document in a post-transform. We cannot insert the citation nodes here because we do not yet know which keys have been cited.

See also:

Further processing of the resulting *bibliography* node is done by *BibliographyTransform*.

`run()`
: Process .bib files, set file dependencies, and create a node that is to be transformed to the entries of the bibliography.

`class sphinxcontrib.bibtex.foot_directives.FootBibliographyDirective`(*name*, *arguments*, *options*, *content*, *lineno*, *content_offset*, *block_text*, *state*, *state_machine*)

Class for processing the *footbibliography* directive.

`run()`
: Set file dependencies, and insert the footnotes that were created earlier by *foot_roles.FootCiteRole.result_nodes()*.

1.3.5 New Transforms

`class sphinxcontrib.bibtex.transforms.BibliographyTransform`(*document*, *startnode=None*)

Bases: `sphinx.transforms.post_transforms.SphinxPostTransform`

A docutils transform to generate citation entries for bibliography nodes.

`default_priority = 5`
: Numerical priority of this transform, 0 through 999 (override).

`run`(***kwargs*)
: Transform each *bibliography* node into a list of citations.

`sphinxcontrib.bibtex.transforms.node_text_transform`(*node*, *transform*)

Apply transformation to all Text nodes within node.

`sphinxcontrib.bibtex.transforms.`**`transform_url_command`**(*textnode*)

Convert '\url{. . . }' into a proper docutils hyperlink.

1.3.6 New Domains

Classes and methods to maintain any bibtex information that is stored outside the doctree.

`class sphinxcontrib.bibtex.domain.`**`Citation`**(*citation_id: str, bibliography_key: BibliographyKey, key: str, entry: Entry, formatted_entry: FormattedEntry*)

Information about a citation.

`property bibliography_key`

Key of its bibliography directive.

`property citation_id`

Unique id of this citation.

`property entry`

Entry from pybtex.

`property formatted_entry`

Entry as formatted by pybtex.

`property key`

Key (with prefix).

`class sphinxcontrib.bibtex.domain.`**`SphinxReferenceInfo`**(*builder: Builder, fromdocname: str, todocname: str, citation_id: str, title: str*)

Tuple containing reference info to enable sphinx to resolve a reference to a citation.

`property builder`

The Sphinx builder.

`property citation_id`

Unique id of the citation within the bibliography.

`property fromdocname`

Document name of the citation reference.

`property title`

Title attribute for reference node.

`property todocname`

Document name of the bibliography.

`class sphinxcontrib.bibtex.domain.`**`SphinxReferenceText`**(*info: sphinxcontrib.bibtex.richtext.ReferenceInfo, *parts: pybtex.richtext.BaseText*)

Pybtex rich text class for citation references with the docutils backend, for use with *`SphinxReferenceInfo`*.

`render`(*backend: BaseBackend*)

Render this `Text` into markup.

Parameters **`backend`** – The formatting backend (an instance of `pybtex.backends.BaseBackend`).

`class sphinxcontrib.bibtex.domain.`**`BibtexDomain`**(*env: BuildEnvironment*)

Sphinx domain for the bibtex extension.

`property bibfiles: Dict[str, sphinxcontrib.bibtex.bibfile.BibFile]`

Map each bib filename to some information about the file (including the parsed data).

property bibliographies: Dict[BibliographyKey, BibliographyValue]
Map storing information about each bibliography directive.

property citation_refs: List[CitationRef]
Citation reference data.

property citations: List[sphinxcontrib.bibtex.domain.Citation]
Citation data.

clear_doc(*docname: str*) → None
Remove traces of a document in the domain-specific inventories.

get_all_cited_keys(*docnames*)
Yield all citation keys for given *docnames* in order, then ordered by citation order.

get_entries(*bibfiles: List[str]*) → Iterable[Entry]
Return all bibliography entries from the bib files, unsorted (i.e. in order of appearance in the bib files.

get_filtered_entries(*bibliography_key: BibliographyKey*) → Iterable[Tuple[str, Entry]]
Return unsorted bibliography entries filtered by the filter expression.

get_formatted_entries(*bibliography_key: BibliographyKey*, *docnames: List[str]*) → Iterable[Tuple[Entry, FormattedEntry]]
Get sorted bibliography entries along with their pybtex labels, with additional sorting and formatting applied from the pybtex style.

get_sorted_entries(*bibliography_key: BibliographyKey*, *docnames: List[str]*) → Iterable[Tuple[str, Entry]]
Return filtered bibliography entries sorted by citation order.

merge_domaindata(*docnames: List[str]*, *otherdata: Dict*) → None
Merge in data regarding *docnames* from a different domaindata inventory (coming from a subprocess in parallel builds).

resolve_any_xref(*env: BuildEnvironment*, *fromdocname: str*, *builder: Builder*, *target: str*, *node: pending_xref*, *contnode: docutils.nodes.Element*) → List[Tuple[str, docutils.nodes.Element]]
Replace node by list of citation references (one for each key), provided that the target has citation keys.

resolve_xref(*env: BuildEnvironment*, *fromdocname: str*, *builder: Builder*, *typ: str*, *target: str*, *node: pending_xref*, *contnode: docutils.nodes.Element*) → docutils.nodes.Element
Replace node by list of citation references (one for each key).

1.3.7 Bib Files

Classes and methods to work with bib files.

class sphinxcontrib.bibtex.bibfile.**BibFile**(*mtime: float*, *data: BibliographyData*)
Contains information about a parsed bib file.

property data
parsed data from pybtex

property mtime
modification time of bib file when last parsed

sphinxcontrib.bibtex.bibfile.**normpath_filename**(*env: BuildEnvironment*, *filename: str*) → str
Return normalised path to *filename* for the given environment *env*.

sphinxcontrib.bibtex.bibfile.**parse_bibfile**(*bibfilename: str*, *encoding: str*) → BibliographyData
Parse *bibfilename* with given *encoding*, and return parsed data.

`sphinxcontrib.bibtex.bibfile.`**`process_bibfile`**(*bibfiles: Dict[str, sphinxcontrib.bibtex.bibfile.BibFile], bibfilename: str, encoding: str*) → None

Check if *bibfiles* is still up to date. If not, parse *bibfilename* and store parsed data in *bibfiles*.

`sphinxcontrib.bibtex.bibfile.`**`get_bibliography_entry`**(*bibfiles: Dict[str, sphinxcontrib.bibtex.bibfile.BibFile], key: str*) → Optional[Entry]

Return bibliography entry from *bibfiles* for the given *key*.

1.3.8 Referencing Styles

Base Classes For Composing Styles

`class sphinxcontrib.bibtex.style.referencing.`**`BaseReferenceStyle`**

Base class for citation reference styles.

For consistency, all subclasses of this class must be decorated as a `dataclasses.dataclass`, and must provide a type annotation and default value for all attributes (unless `init=False` is used, in which case they can be initialized in `__post_init__()`).). This allows client code to instantiate any reference style without needing to specify any arguments through the constructor.

`inner`(*role_name: str*) → Node

Returns inner template for formatting the references.

`outer`(*role_name: str, children: List[BaseText]*) → Node

Returns outer template for formatting the references.

`role_names`() → Iterable[str]

Get list of role names supported by this style.

`class sphinxcontrib.bibtex.style.referencing.`**`BracketStyle`**(*left: Union[BaseText, str] = '[', right: Union[BaseText, str] = ']', sep: Union[BaseText, str] = ', ', sep2: Optional[Union[BaseText, str]] = None, last_sep: Optional[Union[BaseText, str]] = None*)

A class which provides brackets, as well as separators and a function to facilitate formatting of the outer template.

`last_sep: Optional[Union[BaseText, str]] = None`

Separator for outer template, for last item if three or more items.

`left: Union[BaseText, str] = '['`

Left bracket.

`outer`(*children: List[BaseText], brackets=False, capfirst=False*) → Node

Creates an outer template with separators, adding brackets if requested, and capitalizing the first word if requested.

`right: Union[BaseText, str] = ']'`

Right bracket.

`sep: Union[BaseText, str] = ', '`

Separators used for outer template (i.e. in between references if multiple keys are referenced in a single citation).

`sep2: Optional[Union[BaseText, str]] = None`

Separator for outer template, if only two items.

class sphinxcontrib.bibtex.style.referencing.**GroupReferenceStyle**(*styles: List[sphinxcontrib.bibtex.style.referencing.BaseRe = <factory>, role_style: Dict[str, sphinxcontrib.bibtex.style.referencing.BaseReferenceStyle] = <factory>*)

Composes a group of reference styles into a single consistent style.

inner(*role_name: str*) → Node

Gets the inner template associated with *role_name* in one of the `styles`.

outer(*role_name: str, children: List[BaseText]*) → Node

Gets the outer template associated with *role_name* in one of the `styles`.

role_names()

Get list of role names supported by this style.

role_style: Dict[str, sphinxcontrib.bibtex.style.referencing.BaseReferenceStyle]

Dictionary from role names to styles. Automatically initialized from `styles`.

styles: List[sphinxcontrib.bibtex.style.referencing.BaseReferenceStyle]

List of style types.

class sphinxcontrib.bibtex.style.referencing.**PersonStyle**(*style: str = 'last', abbreviate: bool = True, sep: Union[BaseText, str] = ', ', sep2: Optional[Union[BaseText, str]] = ' and ', last_sep: Optional[Union[BaseText, str]] = ', and ', other: Optional[Union[BaseText, str]] = Text(' ', Tag('em', 'et al.'))*)

A class providing additional data and helper functions to facilitate formatting of person names.

abbreviate: bool = True

Whether or not to abbreviate first names.

last_sep: Optional[Union[BaseText, str]] = ', and '

Separator between persons, for last person if three or more persons.

names(*role: str, full: bool*) → Node

Returns a template formatting the persons with correct separators and using the full person list if so requested.

other: Optional[Union[BaseText, str]] = Text(' ', Tag('em', 'et al.'))

Abbreviation text if three or more persons.

sep: Union[BaseText, str] = ', '

Separator between persons.

sep2: Optional[Union[BaseText, str]] = ' and '

Separator between persons, if only two persons.

style: str = 'last'

Plugin name of the style used for formatting person names.

style_plugin: BaseNameStyle

Plugin class instance used for formatting person names. Automatically initialised from `style`.

`sphinxcontrib.bibtex.style.referencing.`**`format_references`**(*style:* sphinxcontrib.bibtex.style.referencing.BaseReferenceStyle, *reference_text_class: Type[*sphinxcontrib.bibtex.richtext.BaseReferenceText*[*sphin
role_name: *str*, *references: Iterable[Tuple[Entry, FormattedEntry,* sphinxcontrib.bibtex.richtext.ReferenceInfo*]]*) → BaseText

Format the list of references according to the given role.

First formats each reference using the style's `get_inner()` method, then joins all these formatted references together using the style's `get_outer()` method.

Basic Styles

Basic styles that support both textual and parenthetical citations. Should provide roles with names `p`, `ps`, `t`, `ts`, `ct`, and `cts`. Here, `t` stands for textual and `p` for parenthetical. The `c` prefix causes the first letter to be capitalized, and the `s` suffix causes all authors to be named rather than shortening the list using "et al." or some other suffix as specified by the style.

class sphinxcontrib.bibtex.style.referencing.basic_label.**BasicLabelParentheticalReferenceStyle**(*bracket: sphinxcontrib.bibtex. = BracketStyle(left= right=']', sep=', ', sep2=None last_sep=N person: sphinxcontrib.bibtex. = PersonStyle(style= style_plugi object>, abbreviate=True, sep=', ', sep2=' and ', last_sep=', and ', other=Text ', Tag('em', 'et al.'))))*

Reference by label if parenthetical, and by author and label if textual.

bracket: *sphinxcontrib.bibtex.style.referencing.BracketStyle* **= BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)**

Bracket style.

inner(*role_name: str*) → Node

Returns inner template for formatting the references.

outer(*role_name: str, children: List[BaseText]*) → Node

Returns outer template for formatting the references.

person: *sphinxcontrib.bibtex.style.referencing.PersonStyle* = PersonStyle(style='last', style_plugin=<sphinxcontrib.bibtex.style.names.last.LastNameStyle object>, abbreviate=True, sep=', ', sep2=' and ', last_sep=', and ', other=Text(' ', Tag('em', 'et al.')))

Person style.

role_names() → Iterable[str]

Get list of role names supported by this style.

class sphinxcontrib.bibtex.style.referencing.basic_label.**BasicLabelTextualReferenceStyle**(*bracket: sphinxcontrib.bibtex.style.ref = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None), person: sphinxcontrib.bibtex.style.ref = PersonStyle(style='last', style_plugin=<sph object>, abbreviate=True, sep=', ', sep2=' and ', last_sep=', and ', other=Text(' ', Tag('em', 'et al.'))), text_reference_sep: Union[BaseText, str] = ' '*)

Reference by label if parenthetical, and by author and label if textual.

bracket: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = **BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)**

Bracket style.

`inner`(*role_name: str*) → Node

Returns inner template for formatting the references.

`outer`(*role_name: str*, *children: List[BaseText]*) → Node

Returns outer template for formatting the references.

`person: sphinxcontrib.bibtex.style.referencing.PersonStyle = PersonStyle(style='last', style_plugin=<sphinxcontrib.bibtex.style.names.last.LastNameStyle object>, abbreviate=True, sep=', ', sep2=' and ', last_sep=', and ', other=Text(' ', Tag('em', 'et al.')))`

Person style.

`role_names`() → Iterable[str]

Get list of role names supported by this style.

`text_reference_sep: Union[BaseText, str] = ' '`

Separator between text and reference.

class sphinxcontrib.bibtex.style.referencing.basic_author_year.**BasicAuthorYearParentheticalReferenceSty**

Parenthetical reference by author-year.

author_year_sep: Union[BaseText, str] = ', '

Separator between author and year.

bracket: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = **BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)**

Bracket style.

inner(*role_name: str*) → Node

Returns inner template for formatting the references.

outer(*role_name: str, children: List[BaseText]*) → Node

Returns outer template for formatting the references.

person: *sphinxcontrib.bibtex.style.referencing.PersonStyle* = **PersonStyle(style='last', style_plugin=<sphinxcontrib.bibtex.style.names.last.LastNameStyle object>, abbreviate=True, sep=', ', sep2=' and ', last_sep=', and ', other=Text(' ', Tag('em', 'et al.')))**

Person style.

role_names() → Iterable[str]

Get list of role names supported by this style.

class sphinxcontrib.bibtex.style.referencing.basic_author_year.**BasicAuthorYearTextualReferenceStyle**(*brac
sphi
con-
trib.
=
Brac
et-
Style
righ
sep=
',
sep2
last_
per-
son:
sphi
con-
trib.
=
Per-
son-
Style
style
ob-
ject>
ab-
bre-
vi-
ate=
sep=
',
sep2
and
',
last_
and
',
othe
',
Tag(
'et
al.')
text_
Unic
str]
=
'
')*

Textual reference by author-year.

bracket: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = **BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)**

Bracket style.

inner(*role_name: str*) → Node
Returns inner template for formatting the references.

outer(*role_name: str, children: List[BaseText]*) → Node
Returns outer template for formatting the references.

person: *sphinxcontrib.bibtex.style.referencing.PersonStyle* = PersonStyle(style='last', style_plugin=<sphinxcontrib.bibtex.style.names.last.LastNameStyle object>, abbreviate=True, sep=', ', sep2=' and ', last_sep=', and ', other=Text(' ', Tag('em', 'et al.')))
Person style.

role_names() → Iterable[str]
Get list of role names supported by this style.

text_reference_sep: Union[BaseText, str] = ' '
Separator between text and reference.

Extra Styles

For styles providing additional roles, e.g. for citations that specifically use the label, the author, the year, etc. The convention for these styles is to have one role for producing whichever text needs to be had, and to have a `par` suffix in the role name if the citation text needs to be embedded in brackets (for example `label` and `labelpar`).

class sphinxcontrib.bibtex.style.referencing.extra_label.**ExtraLabelReferenceStyle**(*bracket: sphinxcontrib.bibtex.style.referencing. = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)*)

Reference just by label.

bracket: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)
Bracket style.

inner(*role_name: str*) → Node
Returns inner template for formatting the references.

outer(*role_name: str, children: List[BaseText]*) → Node
Returns outer template for formatting the references.

role_names() → Iterable[str]
Get list of role names supported by this style.

class sphinxcontrib.bibtex.style.referencing.extra_author.**ExtraAuthorReferenceStyle**(*bracket: sphinxcontrib.bibtex.style.referenci = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None), person: sphinxcontrib.bibtex.style.referenci = PersonStyle(style='last', style_plugin=<sphinxcon object>, abbreviate=True, sep=', ', sep2=' and ', last_sep=', and ', other=Text(' ', Tag('em', 'et al.'))))*

Reference just by author names.

bracket: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)

Bracket style.

inner(*role_name: str*) → Node

Returns inner template for formatting the references.

outer(*role_name: str, children: List[BaseText]*) → Node

Returns outer template for formatting the references.

person: *sphinxcontrib.bibtex.style.referencing.PersonStyle* = PersonStyle(style='last', style_plugin=<sphinxcontrib.bibtex.style.names.last.LastNameStyle object>, abbreviate=True, sep=', ', sep2=' and ', last_sep=', and ', other=Text(' ', Tag('em', 'et al.')))

Person style.

`role_names()` → Iterable[str]

Get list of role names supported by this style.

class `sphinxcontrib.bibtex.style.referencing.extra_year.`**`ExtraYearReferenceStyle`**(*bracket: sphinxcontrib.bibtex.style.referencing.Br = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)*)

Reference just by year.

`bracket: sphinxcontrib.bibtex.style.referencing.BracketStyle = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)`

Bracket style.

`inner`(*role_name: str*) → Node

Returns inner template for formatting the references.

`outer`(*role_name: str, children: List[BaseText]*) → Node

Returns outer template for formatting the references.

`role_names()` → Iterable[str]

Get list of role names supported by this style.

Full Styles

For styles that combine a basic style with one or more extra styles.

class sphinxcontrib.bibtex.style.referencing.label.**LabelReferenceStyle**(*styles: List[sphinxcontrib.bibtex.style.referencing = <factory>, role_style: Dict[str, sphinxcontrib.bibtex.style.referencing.BaseReference = <factory>, bracket_textual: sphinxcontrib.bibtex.style.referencing.BracketStyle = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None), bracket_parenthetical: sphinxcontrib.bibtex.style.referencing.BracketStyle = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None), bracket_author: sphinxcontrib.bibtex.style.referencing.BracketStyle = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None), bracket_label: sphinxcontrib.bibtex.style.referencing.BracketStyle = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None), bracket_year: sphinxcontrib.bibtex.style.referencing.BracketStyle = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None), person: sphinxcontrib.bibtex.style.referencing.PersonStyle = PersonStyle(style='last', style_plugin=<sphinxcontrib.bibtex.style.r object>, abbreviate=True, sep=', ', sep2=' and ', last_sep=', and ', other=Text(' ', Tag('em', 'et al.'))), text_reference_sep: Union[BaseText, str] = ' '*)

Textual or parenthetical reference by label, or just by author, label, or year.

bracket_author: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = **BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)**

Bracket style for author citations (:cite:author: and variations).

bracket_label: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = **BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)**

Bracket style for label citations (:cite:label: and variations).

bracket_parenthetical: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = **BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)**

Bracket style for parenthetical citations (:cite:p: and variations).

bracket_textual: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = **BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)**

Bracket style for textual citations (:cite:t: and variations).

bracket_year: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = **BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)**

Bracket style for year citations (:cite:year: and variations).

person: *sphinxcontrib.bibtex.style.referencing.PersonStyle* = **PersonStyle(style='last', style_plugin=<sphinxcontrib.bibtex.style.names.last.LastNameStyle object>, abbreviate=True, sep=', ', sep2=' and ', last_sep=', and ', other=Text(' ', Tag('em', 'et al.')))**

Person style.

text_reference_sep: Union[BaseText, str] = ' '

Separator between text and reference for textual citations.

class sphinxcontrib.bibtex.style.referencing.author_year.**AuthorYearReferenceStyle**(styles: List[sphinxcontrib.bibtex.sty = <factory>, role_style: Dict[str, sphinxcontrib.bibtex.style.referencing. = <factory>, bracket_textual: sphinxcontrib.bibtex.style.referencing. = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None), bracket_parenthetical: sphinxcontrib.bibtex.style.referencing. = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None), bracket_author: sphinxcontrib.bibtex.style.referencing. = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None), bracket_label: sphinxcontrib.bibtex.style.referencing. = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None), bracket_year: sphinxcontrib.bibtex.style.referencing. =

author_year_sep: Union[BaseText, str] = ', '

Separator between author and year for parenthetical citations.

bracket_author: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)

Bracket style for author citations (:cite:author: and variations).

bracket_label: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)

Bracket style for label citations (:cite:label: and variations).

bracket_parenthetical: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)

Bracket style for parenthetical citations (:cite:p: and variations).

bracket_textual: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)

Bracket style for textual citations (:cite:t: and variations).

bracket_year: *sphinxcontrib.bibtex.style.referencing.BracketStyle* = BracketStyle(left='[', right=']', sep=', ', sep2=None, last_sep=None)

Bracket style for year citations (:cite:year: and variations).

person: *sphinxcontrib.bibtex.style.referencing.PersonStyle* = PersonStyle(style='last', style_plugin=<sphinxcontrib.bibtex.style.names.last.LastNameStyle object>, abbreviate=True, sep=', ', sep2=' and ', last_sep=', and ', other=Text(' ', Tag('em', 'et al.')))

Person style (applies to all relevant citation commands).

text_reference_sep: Union[BaseText, str] = ' '

Separator between text and reference for textual citations.

1.3.9 Plugins

sphinxcontrib.bibtex.plugin.**find_plugin**(*group: str*, *name: str*) → Type[Any]

Load a sphinxcontrib-bibtex plugin.

sphinxcontrib.bibtex.plugin.**register_plugin**(*group: str*, *name: str*, *klass: Type[Any]*, *force: bool = False*) → bool

Register a sphinxcontrib-bibtex plugin at runtime.

1.3.10 Pybtex Extensions

New Text Elements

class sphinxcontrib.bibtex.richtext.**BaseReferenceText**(*info: sphinxcontrib.bibtex.richtext.ReferenceInfo*, **parts: pybtex.richtext.BaseText*)

Generic rich text element for citation references. Instances store some extra reference info that can be used when formatting. This base class renders its children without further formatting. Implementations must create a derivation from this class which overrides the *render* method to create the desired output. See for instance *SphinxReferenceText*.

sphinxcontrib.bibtex.richtext.**ReferenceInfo**

Generic type parameter for types that store reference information. To be implemented by clients. See for instance *SphinxReferenceInfo*.

alias of TypeVar('ReferenceInfo')

New Template Nodes

`sphinxcontrib.bibtex.style.template.join`(*sep=''*, *sep2=None*, *last_sep=None*, *other=None*)
Join text fragments together.

`sphinxcontrib.bibtex.style.template.sentence`(*capfirst=False*, *capitalize=False*, *add_period=True*, *sep=', '*, *sep2=None*, *last_sep=None*, *other=None*)
Join text fragments, capitalize the first letter, and add a period to the end.

`sphinxcontrib.bibtex.style.template.names`(*role*, *sep=''*, *sep2=None*, *last_sep=None*, *other=None*)
Return formatted names.

`sphinxcontrib.bibtex.style.template.entry_label`()
Node for inserting the label of a formatted entry.

`sphinxcontrib.bibtex.style.template.reference`()
Node for inserting a citation reference. The children of the node comprise the content of the reference, and any referencing information is stored in the *reference_info* key of the *data*. The data must also contain a *style* key pointing to the corresponding `BaseReferenceStyle`.

New Names Styles

`class sphinxcontrib.bibtex.style.names.last.LastNameStyle`
A simple name style for formatting the last name of an author.

`format`(*person*, *abbr=True*)
Format last names.

1.4 Changes

1.4.1 2.3.0 (1 June 2021)

- Add `:footcite:p:` and `:footcite:t:` roles. For capitalizing the first letter and/or listing the full author list, you can use `:footcite:ct:`, `:footcite:ts:`, `:footcite:cts:`, and `:footcite:ps:`.
- To configure your footnote referencing style, an optional config setting `bibtex_foot_reference_style` has been added. If not specified, this defaults to the `foot` style, which will use plain footnote references for citation references, matching the referencing style as in previous versions. Footnote reference styles can be fully customized to your heart's desire, similar to regular citation reference styles.
- New `:cite:empty:` role which registers a citation without generating a reference, similar to LaTeX's nocite command (see issue #131).
- Citation keys can now be listed directly under the bibliography directive, one key per line; such citations will always be included, regardless of any filter settings (see issue #54).
- A plain text preview of the full citation information will be shown when hovering over a citation reference (see issue #198, requested by eric-weiser).
- The separator between the text and the reference of all textual citation styles can now be customized.

1.4.2 2.2.1 (16 May 2021)

- The LaTeX output now uses hyperlink instead of sphinxcite. This fixes issues with double brackets and other mismatches between LaTeX and HTML outputs (see issue #244 reported by zhi-wang).
- The setup function now also returns the version of the extension (see issue #239 reported by lcnittl).

1.4.3 2.2.0 (5 March 2021)

- Support the `:any:` role (see issue #232).
- New natbib/biblatex inspired roles for textual and parenthetical citation references (see issue #203 reported by matthew-brett). For textual citation references, use `:cite:t:` and for parenthetical citation references, use `:cite:p:`. The old `:cite:` role is an alias for `:cite:p:`.
- Use the `s` suffix to include the full author list rather than abbreviating it with "et al.": `:cite:ts:`, `:cite:ps:`.
- For textual citation references, use the `c` prefix to capitalize the first letter: `:cite:ct:`, `:cite:cts:`.
- New natbib inspired roles for citing just the author, year, or label, optionally with brackets, and optionally capitalizing the first letter of the author: `:cite:author:`, `:cite:authorpar:`, `:cite:cauthor:`, `:cite:cauthorpar:` `:cite:year:`, `:cite:yearpar:`, `:cite:label:`, `:cite:labelpar:` (see issue #71 reported by bk322).
- To configure your referencing style, an optional config setting `bibtex_reference_style` has been added. If not specified, this defaults to the `label` style, which will use the label to format citation references, matching the referencing style as in previous versions. The other style currently available is `author_year`, for author-year style referencing.
- Reference styles can be fully customized to your heart's desire (see issue #203 reported by amichuda). They are based on pybtex's template system, which was already used for customizing bibliography styles. Refer to the user documentation for examples, and to the API documentation for full details.
- Other packages can register custom reference styles through entry points. Refer to the user documentation for details.
- Propagate pybtex FieldIsMissing exception as a warning (see issue #235 reported by Zac-HD).

1.4.4 2.1.4 (8 January 2021)

- Fix ValueError exception when having citations from orphans (see issue #228, reported by VincentRouvreau).

1.4.5 2.1.3 (1 January 2021)

- Sphinx 2.1 or later is now formally required (up from 2.0).
- Fix unresolved references when running the latex build immediately after the html build, or when rerunning the html build after deleting the generated html files without deleting the pickled doctrees/environment (see issue #226, reported by skirpichev).
- No longer insert user defined header for bibliography directives if there are no citations in it.
- Warnings now consistently provide source file and line number of where the issue originated.
- Simpler and faster implementation of footcite and footbibliography.
- Improved type annotations throughout the API, now using forward declarations where possible.

1.4.6 2.1.2 (30 December 2020)

- Fix KeyError exception when building documents with footbibliography directives but without any footnotes needing to be generated for this directive (see issue #223, reported by drammock).

1.4.7 2.1.1 (29 December 2020)

- Fix latex builder KeyError exception (see issue #221, reported by jedbrown).
- Fix citation references across documents in latex build.

1.4.8 2.1.0 (28 December 2020)

- The extension no longer relies on the `bibtex.json` method. Instead, the extension now postpones identifying all citation cross-references to Sphinx's consistency check phase. The actual citation references and bibliography citations are then generated in the resolve phase using post-transforms. As a result, `bibtex.json` is no longer needed and thus Sphinx no longer needs to run twice as in the past if the file did not exist (closes issues #214 and #215). *Thanks to everyone who chimed in on this, especially everyone who made helpful suggestions to find better implementation approaches, and everyone who helped with testing.*
- Citations with multiple keys will now reside in the same bracket (closes issue #94).
- Consistent use of doctutils note_explicit_target to set ids, to ensure no clashing ids.
- Improved and robustified test suite, using regular expressions to verify generated html.
- The test suite now includes a patched version of the awesome but abandoned sphinx-natbib extension, to help comparing and testing implementations and features. The long term intention is to fully support sphinx-natbib style citations.
- **BACKWARD INCOMPATIBLE** The API has been refactored to accommodate the new design. Refer to the API documentation for details.

1.4.9 2.0.0 (12 December 2020)

- There is a new `footcite` role and a new `footbibliography` directive, to allow easy and simple local (per document) bibliographies through footnotes. See issues #184 and #185.
- Parallel builds are now finally supported. See issues #80, #96, and #164, as well as pull request #210.
- **BACKWARD INCOMPATIBLE** To enable parallel builds, a new mandatory config setting `bibtex_bibfiles` has been added. This setting specifies all bib files used throughout the project, relative to the source folder.
- **BACKWARD INCOMPATIBLE** The encoding of bib files has been moved to an optional config setting `bibtex_encoding`. The `:encoding:` option is no longer supported.
- Headers for `bibliography` and `footbibliography` directives can be configured via the `bibtex_bibliography_header` and `bibtex_footbibliography_header` config setting.
- The `bibliography` directive no longer requires the bib files to be specified as an argument. However, if you do, citations will be constrained to those bib files.
- Support newlines/whitespace around cite keys when multiple keys are specified. Thanks to dizcza for help with testing. See issue #205 and pull request #206.
- Improve citation ordering code (reported by ukos-git, see issue #182).

- The unresolved citations across documents issue has been resolved. The extension stores all citation information in a `bibtex.json` file. If it does not exist, the file will be created on your first sphinx build, and you will have to rerun the build to make use of it. The file is automatically kept up to date, with a warning whenever you need to rerun the build. Thanks to dizcza for help with testing. See issues #197 and #204. Also see pull request #208.
- Migrate test suite to pytest, using sphinx's testing fixtures.
- **BACKWARD INCOMPATIBLE** The API has been refactored. Some functions have moved to different modules. Refer to the API documentation for details.
- Drop Python 3.5 support.
- Add Python 3.9 support.

1.4.10 1.0.0 (20 September 2019)

- Drop Python 2.7 and 3.4 support (as upstream sphinx has dropped support for these as well).
- Add Python 3.8 support (contributed by hroncok).
- Update for Sphinx 2.x, and drop Sphinx 1.x support (as there is too much difference between the two versions).
- Non-bibtex citations will now no longer issue warnings (fix contributed by chrisjsewell).
- Switch to codecov for coverage reporting.

1.4.11 0.4.2 (7 January 2018)

- Drop Python 3.3 support, add Python 3.7 support.
- Work around issue with sphinx-testing on Fedora (reported by jamesjer in issue #157, fix contributed by mitya57 in pull request #158).

1.4.12 0.4.1 (28 November 2018)

- Disable tinkerer test due to upstream bug.
- Remove crossref test due to changed upstream behaviour in pybtex.
- Fix latex test to match new upstream code generation.
- Fix documentation of encoding option (contributed by Kai Mühlbauer).
- Migrate to sphinx.util.logging in favour of old deprecated logging method.

1.4.13 0.4.0 (19 April 2018)

- Remove latexcodec and curly bracket strip functionality, as this is now supported by pybtex natively (see issue #127, reported by erosennin).
- Fix tests failures with Sphinx 1.7 (see pull request #136, reported and fixed by mitya57).

1.4.14 0.3.6 (25 September 2017)

- Real fix for issue #111 (again reported by jamesjer).
- Fix test regressions due to latest Sphinx updates (see issues #115, #120, #121, and #122, reported by ndarmage and ghisvail).
- Fix test regressions on ascii locale (see issue #121, reported by ghisvail).
- Support and test Python 3.6.

1.4.15 0.3.5 (22 February 2017)

- Fix extremely high memory usage when handling large bibliographies (reported by agjohnson, see issue #102).
- Fix tests for Sphinx 1.5.1 (see issue #111, reported by jamesjer).

1.4.16 0.3.4 (20 May 2016)

- Document LaTeX workaround for `:cite:` in figure captions (contributed by xuhdev, see issue #92 and pull request #93).
- Add `bibtex_default_style` config value to override the default bibliography style (see issue #91 and pull request #97).
- Support Python 3.5 (see issue #100).

1.4.17 0.3.3 (23 October 2015)

- Add per-bibliography key prefixes, enabling local bibliographies to be used in isolation from each other (see issue #87, reported by marscher).
- Documentation now points to new location of pybtex on bitbucket.
- Simplified testing code by using the new sphinx_testing package.

1.4.18 0.3.2 (20 March 2015)

- Document how to create custom label styles (see issue #77, reported by tino).
- Disable parallel_read_safe for Sphinx 1.3 and later (see issue #80, reported by andreacassioli).

1.4.19 0.3.1 (10 July 2014)

- Fix for `type_.lower()` bug: pybtex 0.18 expects type to be a string (this fixes issue #68 reported by jluttine).

1.4.20 0.3.0 (4 May 2014)

- **BACKWARD INCOMPATIBLE** The alpha style is now default, so citations are labelled in a way that is more standard for Sphinx. To get the old behaviour back, add `:style: plain` to your bibliography directives.
- **BACKWARD INCOMPATIBLE** `is_cited()` has been removed. Use `get_cited_docnames()` instead, which will return an empty list for keys that are not cited.
- Improved support for local bibliographies (see issues #52, #62, and #63; test case provided by Boris Kheyfets):
  - New `docname` and `docnames` filter identifiers.
  - Filter expressions now also support set literals and the operators `in`, `not in`, `&`, and `|`.

  See documentation for details.
- Multiple comma-separated citation keys per cite command (see issue #61, suggested by Boris Kheyfets).
- Add support for pypy and Python 3.4.
- Drop support for Python 2.6 and Python 3.2.
- Drop 2to3 and instead use six to support both Python 2 and 3 from a single code base.
- Simplify instructions for custom styles.
- Various test suite improvements.

1.4.21 0.2.9 (9 October 2013)

- Upgrade to the latest pybtex-docutils to produce more optimal html output (specifically: no more nested `<span>`s).
- Remove latex codec code, and rely on latexcodec package instead.
- `FilterVisitor` has been removed from the public API. Use `get_bibliography_entries()` instead.
- Fix upstream Sphinx bug concerning LaTeX citation hyperlinks (contributed by erikb85; see pull request #45).
- Fix most pylint warnings, refactor code.

1.4.22 0.2.8 (7 August 2013)

- Use pybtex-docutils to remove dependency on pybtex.backends.doctree.

1.4.23 0.2.7 (4 August 2013)

- Integrate with coveralls.io, first release with 100% test coverage.
- Minor bug fixes and code improvements.
- Remove ordereddict dependency for Python 2.7 and higher (contributed by Paul Romano, see pull requests #27 and #28).
- New `:filter:` option for advanced filtering (contributed by d9pouces, see pull requests #30 and #31).
- Refactor documentation of advanced features.
- Document how to create custom pybtex styles (see issues #25, #29, and #34).
- Code is now mostly pep8 compliant.

1.4.24 0.2.6 (2 March 2013)

- For unsorted styles, citation entries are now sorted in the order they are cited, instead of following the order in the bib file, to reflect more closely the way LaTeX handles unsorted styles (addresses issue #15).
- Skip citation label warnings on Sphinx [source] links (issue #17, contributed by Simon Clift).

1.4.25 0.2.5 (18 October 2012)

- Duplicate label detection (issue #14).
- New `:labelprefix:` option to avoid duplicate labels when having multiple bibliographies with a numeric label style (addresses issue #14).

1.4.26 0.2.4 (24 August 2012)

- New options for the bibliography directive for rendering the bibliography as bullet lists or enumerated lists: `:list:`, `:enumtype:`, and `:start:`.
- Minor latex codec fixes.
- Turn exception into warning when a citation cannot be relabeled (fixes issue #2).
- Document LaTeX encoding, and how to turn it off (issue #4).
- Use pybtex labels (fixes issue #6 and issue #7).
- Cache tracked citation keys and labels, and bibliography enumeration counts (fixes issues with citations in repeated Sphinx runs).
- Bibliography ids are now unique across documents (fixes issue that could cause the wrong bibliography to be inserted).
- The plain style is now the default (addresses issue #9).

1.4.27 0.2.3 (30 July 2012)

- Document workaround for Tinkerer (issue #1).
- Use tox for testing.
- Full 2to3 compatibility.
- Document supported versions of Python (2.6, 2.7, 3.1, and 3.2).

1.4.28 0.2.2 (6 July 2012)

- Documentation and manifest fixes.

1.4.29 0.2.1 (19 June 2012)

- First public release.

1.5 License

1.6 Related Projects

Below is a list of projects which include functionality that is similar or related to sphinxcontrib-bibtex. If you know of any other, leave a message on the issue tracker.

- Andrey Golovizin's pybtex, a general purpose Python library for working with bibtex files. Drives sphinxcontrib-bibtex.
- Matthew Brett's bibstuff. Includes a Sphinx extension similar to sphinxcontrib-bibtex, as well as an assorted collection of bibtex tools. This is a fork of Dylan W. Schwilk and Alan G. Isaac's dschwilk/bibstuff.
- Weston Nielson's sphinx-natbib. This extension is similar to sphinxcontrib-bibtex, and aims to support natbib style citations. Sadly, sphinx-natbib appears no longer maintained and the original repository is no longer available. A patched version of sphinx-natbib, with various bug fixes, is maintained in the test suite of sphinxcontrib-bibtex, for the purpose of comparison. Additionally, a few forks of the original repository can be found on github.
- Jeff Terrace's Sphinx Thesis Resource sphinxtr, is a fork of Sphinx which includes a fork of sphinx-natbib.

CHAPTER TWO

INDICES AND TABLES

- genindex
- modindex
- search

BIBLIOGRAPHY

[Nel87a] Edward Nelson. *Radically Elementary Probability Theory*. Princeton University Press, 1987.

PYTHON MODULE INDEX

S

sphinxcontrib.bibtex, 16
sphinxcontrib.bibtex.bibfile, 20
sphinxcontrib.bibtex.directives, 17
sphinxcontrib.bibtex.domain, 19
sphinxcontrib.bibtex.foot_directives, 18
sphinxcontrib.bibtex.foot_roles, 16
sphinxcontrib.bibtex.nodes, 17
sphinxcontrib.bibtex.plugin, 37
sphinxcontrib.bibtex.richtext, 37
sphinxcontrib.bibtex.roles, 16
sphinxcontrib.bibtex.style.names.last, 38
sphinxcontrib.bibtex.style.referencing, 21
sphinxcontrib.bibtex.style.referencing.author_year, 35
sphinxcontrib.bibtex.style.referencing.basic_author_year, 27
sphinxcontrib.bibtex.style.referencing.basic_label, 23
sphinxcontrib.bibtex.style.referencing.extra_author, 31
sphinxcontrib.bibtex.style.referencing.extra_label, 31
sphinxcontrib.bibtex.style.referencing.extra_year, 33
sphinxcontrib.bibtex.style.referencing.label, 33
sphinxcontrib.bibtex.style.template, 38
sphinxcontrib.bibtex.transforms, 18

INDEX

Symbols

- :all: (*directive option*)
  - bibliography (*directive*), 7
- :cited: (*directive option*)
  - bibliography (*directive*), 7
- :enumtype: (*directive option*)
  - bibliography (*directive*), 7
- :filter: (*directive option*)
  - bibliography (*directive*), 7
- :keyprefix: (*directive option*)
  - bibliography (*directive*), 7
- :labelprefix: (*directive option*)
  - bibliography (*directive*), 7
- :list: (*directive option*)
  - bibliography (*directive*), 7
- :notcited: (*directive option*)
  - bibliography (*directive*), 7
- :start: (*directive option*)
  - bibliography (*directive*), 7
- :style: (*directive option*)
  - bibliography (*directive*), 7

A

- abbreviate (*sphinxcontrib.bibtex.style.referencing.PersonStyle attribute*), 22
- author_year_sep (*sphinxcontrib.bibtex.style.referencing.author_year.AuthorYearReferenceStyle attribute*), 36
- author_year_sep (*sphinxcontrib.bibtex.style.referencing.basic_author_year.BasicAuthorYearParentheticalReferenceStyle attribute*), 28
- AuthorYearReferenceStyle (*class in sphinxcontrib.bibtex.style.referencing.author_year*), 35

B

- BaseReferenceStyle (*class in sphinxcontrib.bibtex.style.referencing*), 21
- BaseReferenceText (*class in sphinxcontrib.bibtex.richtext*), 37
- BasicAuthorYearParentheticalReferenceStyle (*class in sphinxcontrib.bibtex.style.referencing.basic_author_year*), 27
- BasicAuthorYearTextualReferenceStyle (*class in sphinxcontrib.bibtex.style.referencing.basic_author_year*), 29
- BasicLabelParentheticalReferenceStyle (*class in sphinxcontrib.bibtex.style.referencing.basic_label*), 23
- BasicLabelTextualReferenceStyle (*class in sphinxcontrib.bibtex.style.referencing.basic_label*), 25
- BibFile (*class in sphinxcontrib.bibtex.bibfile*), 20
- bibfiles (*sphinxcontrib.bibtex.directives.BibliographyValue property*), 17
- bibfiles (*sphinxcontrib.bibtex.domain.BibtexDomain property*), 19
- bibliographies (*sphinxcontrib.bibtex.domain.BibtexDomain property*), 19
- bibliography (*class in sphinxcontrib.bibtex.nodes*), 17
- bibliography (*directive*), 6
  - :all: (*directive option*), 7
  - :cited: (*directive option*), 7
  - :enumtype: (*directive option*), 7
  - :filter: (*directive option*), 7
  - :keyprefix: (*directive option*), 7
  - :labelprefix: (*directive option*), 7
  - :list: (*directive option*), 7
  - :notcited: (*directive option*), 7
  - :start: (*directive option*), 7
  - :style: (*directive option*), 7
- bibliography_key (*sphinxcontrib.bibtex.domain.Citation property*), 19
- BibliographyDirective (*class in sphinxcontrib.bibtex.directives*), 18
- BibliographyKey (*class in sphinxcontrib.bibtex.directives*), 17
- BibliographyTransform (*class in sphinxcon-*

trib.bibtex.transforms), 18

BibliographyValue (*class in sphinxcontrib.bibtex.directives*), 17

BibtexDomain (*class in sphinxcontrib.bibtex.domain*), 19

bracket (*sphinxcontrib.bibtex.style.referencing.basic_author_year.BasicAuthorYearParentheticalReferenceStyle attribute*), 28

bracket (*sphinxcontrib.bibtex.style.referencing.basic_author_year.BasicAuthorYearTextualReferenceStyle attribute*), 30

bracket (*sphinxcontrib.bibtex.style.referencing.basic_label.BasicLabelParentheticalReferenceStyle attribute*), 24

bracket (*sphinxcontrib.bibtex.style.referencing.basic_label.BasicLabelTextualReferenceStyle attribute*), 26

bracket (*sphinxcontrib.bibtex.style.referencing.extra_author.ExtraAuthorReferenceStyle attribute*), 32

bracket (*sphinxcontrib.bibtex.style.referencing.extra_label.ExtraLabelReferenceStyle attribute*), 31

bracket (*sphinxcontrib.bibtex.style.referencing.extra_year.ExtraYearReferenceStyle attribute*), 33

bracket_author (*sphinxcontrib.bibtex.style.referencing.author_year.AuthorYearReferenceStyle attribute*), 37

bracket_author (*sphinxcontrib.bibtex.style.referencing.label.LabelReferenceStyle attribute*), 34

bracket_label (*sphinxcontrib.bibtex.style.referencing.author_year.AuthorYearReferenceStyle attribute*), 37

bracket_label (*sphinxcontrib.bibtex.style.referencing.label.LabelReferenceStyle attribute*), 35

bracket_parenthetical (*sphinxcontrib.bibtex.style.referencing.author_year.AuthorYearReferenceStyle attribute*), 37

bracket_parenthetical (*sphinxcontrib.bibtex.style.referencing.label.LabelReferenceStyle attribute*), 35

bracket_textual (*sphinxcontrib.bibtex.style.referencing.author_year.AuthorYearReferenceStyle attribute*), 37

bracket_textual (*sphinxcontrib.bibtex.style.referencing.label.LabelReferenceStyle attribute*), 35

bracket_year (*sphinxcontrib.bibtex.style.referencing.author_year.AuthorYearReferenceStyle attribute*), 37

bracket_year (*sphinxcontrib.bibtex.style.referencing.label.LabelReferenceStyle attribute*), 35

BracketStyle (*class in sphinxcontrib.bibtex.style.referencing*), 21

builder (*sphinxcontrib.bibtex.domain.SphinxReferenceInfo property*), 19

C

Citation (*class in sphinxcontrib.bibtex.domain*), 19

citation_id (*sphinxcontrib.bibtex.domain.Citation property*), 19

citation_id (*sphinxcontrib.bibtex.domain.SphinxReferenceInfo property*), 19

citation_nodes (*sphinxcontrib.bibtex.directives.BibliographyValue property*), 17

citation_ref_id (*sphinxcontrib.bibtex.roles.CitationRef property*), 16

citation_refs (*sphinxcontrib.bibtex.domain.BibtexDomain property*), 20

CitationRef (*class in sphinxcontrib.bibtex.roles*), 16

citations (*sphinxcontrib.bibtex.domain.BibtexDomain property*), 20

cite (*role*), 6

cite:author (*role*), 6

cite:authorpar (*role*), 6

cite:authorpars (*role*), 6

cite:authors (*role*), 6

cite:cauthor (*role*), 6

cite:cauthors (*role*), 6

cite:ct (*role*), 6

cite:cts (*role*), 6

cite:empty (*role*), 6

cite:label (*role*), 6

cite:labelpar (*role*), 6

cite:p (*role*), 5

cite:ps (*role*), 6

cite:t (*role*), 5

cite:ts (*role*), 6

cite:year (*role*), 6

cite:yearpar (*role*), 6

CiteRole (*class in sphinxcontrib.bibtex.roles*), 16

clear_doc() (*sphinxcontrib.bibtex.domain.BibtexDomain method*), 20

D

data (*sphinxcontrib.bibtex.bibfile.BibFile property*), 20

default_priority (*sphinxcontrib.bibtex.transforms.BibliographyTransform attribute*), 18

depart_raw_latex() (*in module sphinxcontrib.bibtex.nodes*), 17

docname (*sphinxcontrib.bibtex.directives.BibliographyKey property*), 17

docname (*sphinxcontrib.bibtex.roles.CitationRef property*), 16

E

entry (*sphinxcontrib.bibtex.domain.Citation property*), 19

entry_label() (*in module sphinxcontrib.bibtex.style.template*), 38

enumtype (*sphinxcontrib.bibtex.directives.BibliographyValue property*), 17

ExtraAuthorReferenceStyle (*class in sphinxcontrib.bibtex.style.referencing.extra_author*), 31

ExtraLabelReferenceStyle (*class in sphinxcontrib.bibtex.style.referencing.extra_label*), 31

ExtraYearReferenceStyle (*class in sphinxcontrib.bibtex.style.referencing.extra_year*), 33

F

filter_ (*sphinxcontrib.bibtex.directives.BibliographyValue property*), 17

find_plugin() (*in module sphinxcontrib.bibtex.plugin*), 37

footbibliography (*directive*), 8

FootBibliographyDirective (*class in sphinxcontrib.bibtex.foot_directives*), 18

footcite (*role*), 8

footcite:ct (*role*), 8

footcite:cts (*role*), 8

footcite:p (*role*), 7

footcite:ps (*role*), 8

footcite:t (*role*), 7

footcite:ts (*role*), 8

FootCiteRole (*class in sphinxcontrib.bibtex.foot_roles*), 16

format() (*sphinxcontrib.bibtex.style.names.last.LastNameStyle method*), 38

format_references() (*in module sphinxcontrib.bibtex.style.referencing*), 22

formatted_entry (*sphinxcontrib.bibtex.domain.Citation property*), 19

fromdocname (*sphinxcontrib.bibtex.domain.SphinxReferenceInfo property*), 19

G

get_all_cited_keys() (*sphinxcontrib.bibtex.domain.BibtexDomain method*), 20

get_bibliography_entry() (*in module sphinxcontrib.bibtex.bibfile*), 21

get_entries() (*sphinxcontrib.bibtex.domain.BibtexDomain method*), 20

get_filtered_entries() (*sphinxcontrib.bibtex.domain.BibtexDomain method*), 20

get_formatted_entries() (*sphinxcontrib.bibtex.domain.BibtexDomain method*), 20

get_sorted_entries() (*sphinxcontrib.bibtex.domain.BibtexDomain method*), 20

GroupReferenceStyle (*class in sphinxcontrib.bibtex.style.referencing*), 21

I

id_ (*sphinxcontrib.bibtex.directives.BibliographyKey property*), 17

inner() (*sphinxcontrib.bibtex.style.referencing.BaseReferenceStyle method*), 21

inner() (*sphinxcontrib.bibtex.style.referencing.basic_author_year.BasicAu method*), 29

inner() (*sphinxcontrib.bibtex.style.referencing.basic_author_year.BasicAu method*), 30

inner() (*sphinxcontrib.bibtex.style.referencing.basic_label.BasicLabelPar method*), 24

inner() (*sphinxcontrib.bibtex.style.referencing.basic_label.BasicLabelTex method*), 26

inner() (*sphinxcontrib.bibtex.style.referencing.extra_author.ExtraAuthorR method*), 32

inner() (*sphinxcontrib.bibtex.style.referencing.extra_label.ExtraLabelRefe method*), 31

inner() (*sphinxcontrib.bibtex.style.referencing.extra_year.ExtraYearRefere method*), 33

inner() (*sphinxcontrib.bibtex.style.referencing.GroupReferenceStyle method*), 22

J

join() (*in module sphinxcontrib.bibtex.style.template*), 38

K

key (*sphinxcontrib.bibtex.domain.Citation property*), 19

keyprefix (*sphinxcontrib.bibtex.directives.BibliographyValue property*), 17

keys (*sphinxcontrib.bibtex.directives.BibliographyValue property*), 17

keys (*sphinxcontrib.bibtex.roles.CitationRef property*), 16

L

labelprefix (*sphinxcontrib.bibtex.directives.BibliographyValue property*), 18

LabelReferenceStyle (*class in sphinxcontrib.bibtex.style.referencing.label*), 33

last_sep (*sphinxcontrib.bibtex.style.referencing.BracketStyle attribute*), 21

last_sep (*sphinxcontrib.bibtex.style.referencing.PersonStyle attribute*), 22
LastNameStyle (*class in sphinxcontrib.bibtex.style.names.last*), 38
left (*sphinxcontrib.bibtex.style.referencing.BracketStyle attribute*), 21
line (*sphinxcontrib.bibtex.directives.BibliographyValue property*), 18
line (*sphinxcontrib.bibtex.roles.CitationRef property*), 16
list_ (*sphinxcontrib.bibtex.directives.BibliographyValue property*), 18

M

merge_domaindata() (*sphinxcontrib.bibtex.domain.BibtexDomain method*), 20
module
- sphinxcontrib.bibtex, 16
- sphinxcontrib.bibtex.bibfile, 20
- sphinxcontrib.bibtex.directives, 17
- sphinxcontrib.bibtex.domain, 19
- sphinxcontrib.bibtex.foot_directives, 18
- sphinxcontrib.bibtex.foot_roles, 16
- sphinxcontrib.bibtex.nodes, 17
- sphinxcontrib.bibtex.plugin, 37
- sphinxcontrib.bibtex.richtext, 37
- sphinxcontrib.bibtex.roles, 16
- sphinxcontrib.bibtex.style.names.last, 38
- sphinxcontrib.bibtex.style.referencing, 21
- sphinxcontrib.bibtex.style.referencing.author_year, 35
- sphinxcontrib.bibtex.style.referencing.basic_author_year, 27
- sphinxcontrib.bibtex.style.referencing.basic_label, 23
- sphinxcontrib.bibtex.style.referencing.extra_author, 31
- sphinxcontrib.bibtex.style.referencing.extra_label, 31
- sphinxcontrib.bibtex.style.referencing.extra_year, 33
- sphinxcontrib.bibtex.style.referencing.label, 33
- sphinxcontrib.bibtex.style.template, 38
- sphinxcontrib.bibtex.transforms, 18

mtime (*sphinxcontrib.bibtex.bibfile.BibFile property*), 20

N

names() (*in module sphinxcontrib.bibtex.style.template*), 38
names() (*sphinxcontrib.bibtex.style.referencing.PersonStyle method*), 22
node_text_transform() (*in module sphinxcontrib.bibtex.transforms*), 18
normpath_filename() (*in module sphinxcontrib.bibtex.bibfile*), 20

O

other (*sphinxcontrib.bibtex.style.referencing.PersonStyle attribute*), 22
outer() (*sphinxcontrib.bibtex.style.referencing.BaseReferenceStyle method*), 21
outer() (*sphinxcontrib.bibtex.style.referencing.basic_author_year.BasicAu method*), 29
outer() (*sphinxcontrib.bibtex.style.referencing.basic_author_year.BasicAu method*), 31
outer() (*sphinxcontrib.bibtex.style.referencing.basic_label.BasicLabelPar method*), 24
outer() (*sphinxcontrib.bibtex.style.referencing.basic_label.BasicLabelTex method*), 27
outer() (*sphinxcontrib.bibtex.style.referencing.BracketStyle method*), 21
outer() (*sphinxcontrib.bibtex.style.referencing.extra_author.ExtraAuthorR method*), 32
outer() (*sphinxcontrib.bibtex.style.referencing.extra_label.ExtraLabelRefe method*), 31
outer() (*sphinxcontrib.bibtex.style.referencing.extra_year.ExtraYearRefere method*), 33
outer() (*sphinxcontrib.bibtex.style.referencing.GroupReferenceStyle method*), 22

P

parse_bibfile() (*in module sphinxcontrib.bibtex.bibfile*), 20
person (*sphinxcontrib.bibtex.style.referencing.author_year.AuthorYearRefe attribute*), 37
person (*sphinxcontrib.bibtex.style.referencing.basic_author_year.BasicAut attribute*), 29
person (*sphinxcontrib.bibtex.style.referencing.basic_author_year.BasicAut attribute*), 31
person (*sphinxcontrib.bibtex.style.referencing.basic_label.BasicLabelParen attribute*), 24
person (*sphinxcontrib.bibtex.style.referencing.basic_label.BasicLabelTextu attribute*), 27
person (*sphinxcontrib.bibtex.style.referencing.extra_author.ExtraAuthorRe attribute*), 32
person (*sphinxcontrib.bibtex.style.referencing.label.LabelReferenceStyle attribute*), 35
PersonStyle (*class in sphinxcontrib.bibtex.style.referencing*), 22
process_bibfile() (*in module sphinxcontrib.bibtex.bibfile*), 20

R

raw_latex (*class in sphinxcontrib.bibtex.nodes*), 17 reference() (*in module sphinxcontrib.bibtex.style.template*), 38

ReferenceInfo (*in module sphinxcontrib.bibtex.richtext*), 37

register_plugin() (*in module sphinxcontrib.bibtex.plugin*), 37

render() (*sphinxcontrib.bibtex.domain.SphinxReferenceText method*), 19

resolve_any_xref() (*sphinxcontrib.bibtex.domain.BibtexDomain method*), 20

resolve_xref() (*sphinxcontrib.bibtex.domain.BibtexDomain method*), 20

result_nodes() (*sphinxcontrib.bibtex.foot_roles.FootCiteRole method*), 16

result_nodes() (*sphinxcontrib.bibtex.roles.CiteRole method*), 16

right (*sphinxcontrib.bibtex.style.referencing.BracketStyle attribute*), 21

role_names() (*sphinxcontrib.bibtex.style.referencing.BaseReferenceStyle method*), 21

role_names() (*sphinxcontrib.bibtex.style.referencing.basic_author_year.BasicAuthorYearParentheticalReferenceStyle method*), 29

role_names() (*sphinxcontrib.bibtex.style.referencing.basic_author_year.BasicAuthorYearTextualReferenceStyle method*), 31

role_names() (*sphinxcontrib.bibtex.style.referencing.basic_label.BasicLabelParentheticalReferenceStyle method*), 25

role_names() (*sphinxcontrib.bibtex.style.referencing.basic_label.BasicLabelTextualReferenceStyle method*), 27

role_names() (*sphinxcontrib.bibtex.style.referencing.extra_author.ExtraAuthorReferenceStyle method*), 32

role_names() (*sphinxcontrib.bibtex.style.referencing.extra_label.ExtraLabelReferenceStyle method*), 31

role_names() (*sphinxcontrib.bibtex.style.referencing.extra_year.ExtraYearReferenceStyle method*), 33

role_names() (*sphinxcontrib.bibtex.style.referencing.GroupReferenceStyle method*), 22

role_style (*sphinxcontrib.bibtex.style.referencing.GroupReferenceStyle attribute*), 22

run() (*sphinxcontrib.bibtex.directives.BibliographyDirective method*), 18

run() (*sphinxcontrib.bibtex.foot_directives.FootBibliographyDirective method*), 18

run() (*sphinxcontrib.bibtex.transforms.BibliographyTransform method*), 18

S

sentence() (*in module sphinxcontrib.bibtex.style.template*), 38

sep (*sphinxcontrib.bibtex.style.referencing.BracketStyle attribute*), 21

sep (*sphinxcontrib.bibtex.style.referencing.PersonStyle attribute*), 22

sep2 (*sphinxcontrib.bibtex.style.referencing.BracketStyle attribute*), 21

sep2 (*sphinxcontrib.bibtex.style.referencing.PersonStyle attribute*), 22

setup() (*in module sphinxcontrib.bibtex*), 16

sphinxcontrib.bibtex
module, 16

sphinxcontrib.bibtex.bibfile
module, 20

sphinxcontrib.bibtex.directives
module, 17

sphinxcontrib.bibtex.domain
module, 19

sphinxcontrib.bibtex.foot_directives
module, 18

sphinxcontrib.bibtex.foot_roles
module, 16

sphinxcontrib.bibtex.nodes
module, 17

sphinxcontrib.bibtex.plugin
module, 37

sphinxcontrib.bibtex.richtext
module, 37

sphinxcontrib.bibtex.roles
module, 16

sphinxcontrib.bibtex.style.names.last
module, 38

sphinxcontrib.bibtex.style.referencing
module, 21

sphinxcontrib.bibtex.style.referencing.author_year
module, 35

sphinxcontrib.bibtex.style.referencing.basic_author_year
module, 27

sphinxcontrib.bibtex.style.referencing.basic_label
module, 23

sphinxcontrib.bibtex.style.referencing.extra_author
module, 31

sphinxcontrib.bibtex.style.referencing.extra_label
module, 31

sphinxcontrib.bibtex.style.referencing.extra_year
module, 33

sphinxcontrib.bibtex.style.referencing.label
module, 33

sphinxcontrib.bibtex.style.template
module, 38
sphinxcontrib.bibtex.transforms
module, 18
SphinxReferenceInfo (*class in sphinxcontrib.bibtex.domain*), 19
SphinxReferenceText (*class in sphinxcontrib.bibtex.domain*), 19
start (*sphinxcontrib.bibtex.directives.BibliographyValue property*), 18
style (*sphinxcontrib.bibtex.directives.BibliographyValue property*), 18
style (*sphinxcontrib.bibtex.style.referencing.PersonStyle attribute*), 22
style_plugin (*sphinxcontrib.bibtex.style.referencing.PersonStyle attribute*), 22
styles (*sphinxcontrib.bibtex.style.referencing.GroupReferenceStyle attribute*), 22

T

text_reference_sep (*sphinxcontrib.bibtex.style.referencing.author_year.AuthorYearReferenceStyle attribute*), 37
text_reference_sep (*sphinxcontrib.bibtex.style.referencing.basic_author_year.BasicAuthorYearTextualReferenceStyle attribute*), 31
text_reference_sep (*sphinxcontrib.bibtex.style.referencing.basic_label.BasicLabelTextualReferenceStyle attribute*), 27
text_reference_sep (*sphinxcontrib.bibtex.style.referencing.label.LabelReferenceStyle attribute*), 35
title (*sphinxcontrib.bibtex.domain.SphinxReferenceInfo property*), 19
todocname (*sphinxcontrib.bibtex.domain.SphinxReferenceInfo property*), 19
transform_url_command() (*in module sphinxcontrib.bibtex.transforms*), 18

V

visit_raw_latex() (*in module sphinxcontrib.bibtex.nodes*), 17