
Docs4NIST

Jonathan E. Guyer

Mar 18, 2025

CONTENTS:

1	Usage	3
2	Indices and tables	19
	Python Module Index	21
	Index	23

This [GitHub action](#)¹ uses the [Sphinx](#) tool to build documentation in <https://github.com/usnistgov> projects and then host that documentation on <https://pages.nist.gov>, as an approximation of [ReadTheDocs](#).

¹ Certain commercial firms and trade names are identified in this document in order to specify the installation and usage procedures adequately, or to provide context. Such identification is not intended to imply recommendation or endorsement by the [National Institute of Standards and Technology](#), nor is it intended to imply that related products are necessarily the best available for the purpose.

USAGE

- Configure your repo for publishing on pages.nist.gov.
- Enable GitHub Actions for your repo and select “Allow all actions and reusable workflows” (you may need to submit a request to devops@nist.gov to enable this for you).
- For each branch of your repository where you want to host documentation, add a workflow, such as `.github/workflows/Docs4NIST.yml`:

```
name: "Build Documentation"

on: [push, pull_request, delete]

jobs:
  docs:
    runs-on: ubuntu-latest
    steps:
      - uses: usnistgov/Docs4NIST@0.7
        with:
          docs-folder: docs/
          formats: |-
            epub
            pdf
```

See *Docs4NIST Action* for more information about configuration of this workflow.

1.1 Customization

Docs4NIST uses two sets of templates that determine the appearance of the hosted documentation.

1.1.1 Assimilation templates

The first set of templates is used by `assimilate_theme()` to modify the Sphinx configuration.

These templates can be customized by forking this repository.

`conf.py`

This template file overlays the `html_theme` you chose for your documentation with the `ntd2d` theme, described next.

`ntd2d/`

This template directory provides a `Sphinx theme` that modifies your chosen documentation theme.

`ntd2d/layout.html`

This template file inserts standard NIST headers and footers and a dropdown menu that allows selecting different versions of your documentation.

`ntd2d/static/ntd2d.css_t`

This template file controls the appearance of the dropdown menu.

1.1.2 Pages templates

The second set of templates is used by `update_pages()` to create the pages on the hosting site that enable switching between different documentation variants.

You can customize any of these templates by copying them to a `_templates/` directory at the root of your *nist-pages* branch and editing them to suit.

These templates use `Python string format syntax` (Sphinx and `pages.nist.gov` already fight over Jekyll templating, so we're not getting fancy, here).

`downloads.html`

A section inserted into `menu.html` if a documentation variant has any downloadable output, e.g., PDF or ePUB.

Available substitution keywords are:

- `downloads`: A pre-formatted string with each downloadable output formatted by `download_item.html`.

`download_item.html`

Formats a link to a single downloadable output.

Available substitution keywords are:

- `href`: URL of the downloadable output.
- `kind`: Type of downloadable output, e.g., PDF or ePUB.

`index.html`

The default page for your documentation displayed at <https://pages.nist.gov/{repository}>.

Available substitution keywords are:

- `owner`: The GitHub [user](#) or [organization](#) for your repository.
- `repository`: The name of your repository.
- `variants`: The result of filling the [variants.html](#) template.

`menu.html`

The dropdown menu that allows selecting different branches and tags of your documentation.

Available substitution keywords are:

- `downloads`: The result of filling the [downloads.html](#) template.
- `tree_url`: The GitHub URL corresponding to the active tag or branch.
- `variant`: The name of the active tag or branch.
- `variants`: The result of filling the [variants.html](#) template.

`ntd2d_active.css`

Style sheet that controls the appearance of the active tag or branch in the dropdown menu.

Available substitution keywords are:

- `variant`: The name of the active tag or branch.

`variants.html`

Lists tags and branches that are [configured](#) to serve documentation with this Action.

Available substitution keywords are:

- `branches`: A pre-formatted string with each git branch formatted by [variant_item.html](#).
- `latest`: A pre-formatted string with the HEAD of the default GitHub branch formatted by [variant_item.html](#).
- `stable`: A pre-formatted string with the `stable_version` that has the highest version identifier, as formatted by [variant_item.html](#).
- `stable_versions`: A pre-formatted string with the tags or branches that satisfy the **PEP 440** version specification and aren't [pre-releases](#), each formatted by [variant_item.html](#).
- `versions`: A pre-formatted string with the tags or branches that satisfy the **PEP 440** version specification, each formatted by [variant_item.html](#).

`variant_item.html`

Formats a link to a single tag or branch.

Available substitution keywords are:

- `href`: URL of the downloadable output.
- `kind`: Type of downloadable output, e.g., PDF or ePUB.

1.2 Q & A

1.2.1 Can't I just use Read the Docs?

Nope.

1.2.2 Is there a reason?

Yep.

1.2.3 Is it a good reason?

Nope.

1.3 API

<code>ntd2d_action</code>	Workhorse code for modifying and building documentation.
---------------------------	--

1.3.1 `ntd2d_action`

Workhorse code for modifying and building documentation.

Modules

<code>ntd2d_action.files</code>	Representations of file objects.
<code>ntd2d_action.repository</code>	Interface to a git/GitHub repositories.
<code>ntd2d_action.sphinxdocs</code>	Sphinx configuration directory.
<code>ntd2d_action.variants</code>	Representations of a branches and tags.

ntd2d_action.files

Representations of file objects.

Modules

ntd2d_action.files.borgedconffile

ntd2d_action.files.conffile

ntd2d_action.files.cssfile

ntd2d_action.files.file

ntd2d_action.files.indexfile

ntd2d_action.files.menufile

ntd2d_action.files.nojekyllfile

ntd2d_action.files.pagesfile

ntd2d_action.files.sphinxlog

ntd2d_action.files.template

ntd2d_action.files.templatehierarchy

ntd2d_action.files.variantsfile

ntd2d_action.files.borgedconffile

Classes

class `ntd2d_action.files.borgedconffile.BorgedConfFile`(*source_dir, config, original_docs*)

Bases: *ConfFile*

Sphinx configuration file that overlays the html theme.

property `html_theme`

html_theme Sphinx configuration value.

property `html_theme_path`

html_theme_path Sphinx configuration value.

property `language`

language Sphinx configuration value.

property `project`

project Sphinx configuration value.

read()

Evaluate Sphinx configuration values.

The Sphinx docs says that it reads `conf.py` with `importlib.import_module()`. It doesn't.

Returns

namespace – Configuration values.

Return type

dict

ntd2d_action.files.conffile**Classes**

class `ntd2d_action.files.conffile.ConfFile`(*source_dir*, *config*)

Bases: *File*

Sphinx configuration file.

property html_theme

html_theme Sphinx configuration value.

property html_theme_path

html_theme_path Sphinx configuration value.

property language

language Sphinx configuration value.

property project

project Sphinx configuration value.

read()

Evaluate Sphinx configuration values.

The Sphinx docs says that it reads `conf.py` with `importlib.import_module()`. It doesn't.

Returns

namespace – Configuration values.

Return type

dict

ntd2d_action.files.cssfile**Classes**

class `ntd2d_action.files.cssfile.CSSFile`(*variant*)

Bases: *PagesFile*

ntd2d_action.files.file

Classes

class ntd2d_action.files.file.**File**

Bases: `object`

Base file class

ntd2d_action.files.indexfile

Classes

class ntd2d_action.files.indexfile.**IndexFile**(*repo*, *variants_url*)

Bases: *PagesFile*

get_contents()

build index.html with available documentation variants

ntd2d_action.files.menufile

Classes

class ntd2d_action.files.menufile.**MenuFile**(*variant*, *variants_url*)

Bases: *PagesFile*

ntd2d_action.files.nojekyllfile

Classes

class ntd2d_action.files.nojekyllfile.**NoJekyllFile**(*repo*)

Bases: *PagesFile*

ntd2d_action.files.pagesfile

Classes

class ntd2d_action.files.pagesfile.**PagesFile**(*repo*)

Bases: *File*

ntd2d_action.files.sphinxlog

Classes

class ntd2d_action.files.sphinxlog.**SphinxLog**(*docs, path*)

Bases: *File*

parse_sphinx_warnings()

Parse and emit a sphinx file containing warnings and errors.

Inputs look like this:

/media/sf_shared/workspace/sphinx-action/tests/test_projects/warnings_and_errors/index.rst:19:

WARNING: Error in “code-block” directive: maximum 1 argument(s) allowed, 2 supplied.

/cpython/Doc/distutils/_setuptools_disclaimer.rst: WARNING: document isn’t included in any toctree
/cpython/Doc/contents.rst:5: WARNING: toctree contains reference to nonexistent document ‘ayyl-mao’

ntd2d_action.files.template

Classes

class ntd2d_action.files.template.**FileTemplate**(*name*)

Bases: *Template*

class ntd2d_action.files.template.**PagesTemplate**(*working_dir, name*)

Bases: *FileTemplate*

class ntd2d_action.files.template.**Template**(*template_path*)

Bases: *object*

ntd2d_action.files.templatehierarchy

Classes

class ntd2d_action.files.templatehierarchy.**TemplateHierarchy**(*name, destination_dir,*
***substitutions*)

Bases: *FileTemplate*

ntd2d_action.files.variantsfile

Classes

class ntd2d_action.files.variantsfile.**VariantsFile**(*repo, variants, pages_url*)

Bases: *PagesFile*

ntd2d_action.repository

Interface to a git/GitHub repositories.

Classes

class ntd2d_action.repository.**Repository**(*server_url, repository, branch, default_branch, docs, pages_url*)

Bases: object

Interface to a git/GitHub repository.

Parameters

- **server_url** (*url*) – The URL of the GitHub server (GITHUB_SERVER_URL).
- **repository** (*str*) – The owner and repository name (GITHUB_REPOSITORY).
- **branch** (*str*) – The branch linked to your documentation server (*pages-branch*).
- **default-branch** (*str*) – The default branch configured in GitHub (*default-branch*).
- **docs** ([SphinxDocs](#)) – The documentation being built.
- **pages_url** (*url*) – URL of the web server for served documentation (*pages-url*).

update_pages(*branch, sha*)

Commit built documentation to *pages-branch*.

Parameters

- **branch** (*str*) – The sanitized (‘/’ removed) short ref name of the branch or tag that triggered the workflow (GITHUB_REF_NAME).
- **sha** (*str*) – The commit SHA that triggered the workflow (GITHUB_SHA).

ntd2d_action.sphinxdocs

Sphinx configuration directory.

Classes

class ntd2d_action.sphinxdocs.**BorgedSphinxDocs**(*original_docs*)

Bases: [SphinxDocs](#)

Sphinx configuration directory modified by Docs4NIST

Parameters

- **original_docs** ([SphinxDocs](#)) – The configuration directory stored in the *Repository*.

assimilate_theme(*name, insert_header_footer=True*)

Replace configuration directory with customized html theme.

build_docs(*build_command*)

Build Sphinx Documentation

Adapted from https://github.com/ammraskar/sphinx-action/blob/master/sphinx_action/action.py (Apache-2.0).

property inherited_layout

Find inherited layout.html

Inherited theme may not define its own `layout.html`, but rely on a theme that it, in turn, derives from. `{% extends "!layout.html" %}` should work, but doesn't. <https://github.com/sphinx-doc/sphinx/issues/12049>

```
class ntd2d_action.sphinxdocs.SphinxDocs(docs_dir)
```

Bases: object

Sphinx configuration directory.

```
build_docs(build_command)
```

Build Sphinx Documentation

Adapted from https://github.com/ammraskar/sphinx-action/blob/master/sphinx_action/action.py (Apache-2.0).

ntd2d_action.variants

Representations of a branches and tags.

Classes

```
class ntd2d_action.variants.Variant(repo, name, rebuild_menu=False, true_name=None)
```

Bases: object

Representation of a branch or tag.

Parameters

- **repo** ([Repository](#)) – Git repository containing this branch or tag.
- **name** (*str*) – Ref name of branch or tag.
- **rebuild_menu** (*bool*) – Whether to rebuild variant menu after updating variants (default: False).
- **true_name** (*str*) – Name of actual branch or tag associated with *stable* or *latest* variants.

property css_name

Escape ref name to satisfy css class naming requirements

Used to escape characters that are - allowed by *man git-check-ref-format* - disallowed by <https://www.w3.org/International/questions/qa-escapes#cssescapes>

```
class ntd2d_action.variants.VariantCollection(initlist=None)
```

Bases: [UserList](#)

A collection of [Variant](#) objects.

```
append(item)
```

S.append(value) – append value to the end of the sequence

```
clear() → None -- remove all items from S
```


count(*value*) → integer -- return number of occurrences of value

extend(*other*)

S.extend(iterable) – extend sequence by appending elements from the iterable

index(*value*[, *start*[, *stop*]]) → integer -- return first index of value.

Raises ValueError if the value is not present.

Supporting start and stop arguments is optional, but recommended.

insert(*i*, *item*)

S.insert(index, value) – insert value before index

pop([*index*]) → item -- remove and return item at index (default last).

Raise IndexError if list is empty or index is out of range.

remove(*item*)

S.remove(value) – remove first occurrence of value. Raise ValueError if the value is not present.

reverse()

S.reverse() – reverse *IN PLACE*

class ntd2d_action.variants.**VariantCollector**(*repo*, *current_variant*)

Bases: object

Tool that aggregates branches and versions.

Parameters

- **repo** ([Repository](#)) – Git repository to be scanned for branches and tags.
 - **current_variant** ([Variant](#)) – Branch or tag which triggered the workflow.
-

class ntd2d_action.variants.**Version**(*repo*, *name*, *rebuild_menu*=False)

Bases: [Variant](#)

A [Variant](#) that satisfies the [PEP 440](#) version specification

Raises

InvalidVersion – If the name is not parsable by `packaging.version`.

property *css_name*

Escape ref name to satisfy css class naming requirements

Used to escape characters that are - allowed by *man git-check-ref-format* - disallowed by <https://www.w3.org/International/questions/qa-escapes#cssescapes>

1.4 Docs4NIST Action

This [GitHub action](#) uses the [Sphinx](#) tool to build documentation in <https://github.com/usnistgov> projects and then host on <https://pages.nist.gov> as an approximation of [ReadTheDocs](#).

1.4.1 Usage

This action is invoked by adding a workflow file to your repository, such as `.github/workflows/Docs4NIST.yml`:

```
name: "Build Documentation"

on: [push, pull_request, delete]

jobs:
  docs:
    runs-on: ubuntu-latest
    steps:
      - uses: usnistgov/Docs4NIST@0.7
        with:
          docs-folder: docs/
          pages-branch: 'nist-pages'
          pages-url: 'https://pages.nist.gov'
          formats: ''
          build-html-command: make html
          build-epub-command: make epub
          build-pdf-command: make latexpdf
          pre-build-command: ''
          apt-packages: ''
          pip-requirements: ''
          conda-environment: ''
          push-pulls-pages: false
          include-header-footer: true
```

1.4.2 Inputs

`docs-folder`

The folder containing your Sphinx configuration.

`pages-branch`

The branch linked to your documentation server.

`pages-url`

URL of the web server for served documentation.

formats

Type(s) of output desired in addition to html (pdf, and/or epub).

Note: GitHub Actions' YAML implementation does not support list or array elements, so use a multiline string to declare formats, e.g.,

```
formats: |-  
  epub  
  pdf
```

build-html-command

The command used by [usnistgov/sphinx-action](#) to build your html documentation.

build-epub-command

The command used by [usnistgov/sphinx-action](#) to build your ePUB documentation.

build-pdf-command

The command used by [usnistgov/sphinx-action](#) to build your PDF documentation.

pre-build-command

Run by [usnistgov/sphinx-action](#) before the build command. You can use this to install system level dependencies, for example, with “`apt-get update -y && apt-get install -y perl`”, although those are better installed with *apt-packages*.

apt-packages

List of any APT packages that should be installed.

pip-requirements

The path to the pip requirements file, relative to the root of the project.

conda-environment

The path to the Conda environment file, relative to the root of the project.

push-pulls-pages

Whether the results of pull requests should be pushed to *pages-branch*. For security reasons, this is impossible for pull requests from repository forks, but it is generally undesirable in any case (they appear with cryptic names like *merge_1234* and are redundant to the branch the pull is from). As long as this action is set to run *on: push*, then any build products from branches in the same repository will appear at *pages-url*.

insert-header-footer

When true, automatically inserts the NIST header and footer. Setting to true conflicts with, e.g., *sphinx_rtd_theme*.

1.4.3 Implementation

This action implements a composite workflow with the following major steps:

1. `actions/checkout`
2. *ntd2d Sub-Action*
3. `ad-m/github-push-action`
4. `actions/upload-artifact`

1.5 ntd2d Sub-Action

This GitHub action is invoked by the main *Docs4NIST Action* to move built documentation into your designated *pages-branch*.

1.5.1 Usage

This action is invoked as a step of the composite workflow of the *Docs4NIST Action*. There is no reason to invoke this action yourself.

```
- name: Commit documentation changes
  uses: ../../../../_actions/current/ntd2d
  with:
    docs-folder: docs/
    default-branch: 'main'
    pages-branch: 'nist-pages'
    pages-url: 'https://pages.nist.gov'
    formats: ''
    build-html-command: make html
    build-epub-command: make epub
    build-pdf-command: make epub
    pre-build-command: ''
    apt-packages: ''
    pip-requirements: ''
    conda-environment: ''
    include-header-footer: true
```

Note: This action must be synchronized with the invoked version of the *Docs4NIST Action*, but “usnistgov/Docs4NIST/borg_the_docs@\${{ github.action_ref }}" *doesn't work*, hence the clumsy `uses:` statement.

1.5.2 Inputs

`docs-folder`

The folder containing your Sphinx configuration.

`default-branch`

The default branch *configured in GitHub*.

`pages-branch`

The branch linked to your documentation server.

`pages-url`

URL of the web server for served documentation.

`formats`

Type(s) of output desired in addition to html (pdf, and/or epub).

Note: GitHub Actions' YAML implementation does not support list or array elements, so use a multiline string to declare formats, e.g.,

```
formats: |-  
  epub  
  pdf
```

`build-html-command`

The command used to build your html documentation.

build-epub-command

The command used to build your ePUB documentation.

build-pdf-command

The command used to build your PDF documentation.

pre-build-command

Run before the build command. You can use this to install system level dependencies, for example, with “`apt-get update -y && apt-get install -y perl`”, although those are better installed with *apt-packages*.

apt-packages

List of any *APT* packages that should be installed.

pip-requirements

The path to the pip requirements file, relative to the root of the project.

conda-environment

The path to the Conda environment file, relative to the root of the project.

include-header-footer

Whether to insert the *NIST header and footer*. This content conflicts with, e.g., *sphinx_rtd_theme*.

1.5.3 Outputs

borged-build-folder

The folder containing the Sphinx build outputs.

1.5.4 Implementation

This action implements a *Docker workflow step*. The Docker ENTRYPOINT

- installs any specified *apt-packages*, *pip-requirements*, and *conda-environment*,
- wraps the *Sphinx* configuration directory in a *BorgedSphinxDocs* object,
- invokes *assimilate_theme()*
- executes any *pre-build-command*,
- invokes *build_docs()* for html and any other formats specified in *formats*,
- wraps the GITHUB_REPOSITORY in a *Repository* object,
- invokes *update_pages()*.

INDICES AND TABLES

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

PYTHON MODULE INDEX

n

- [ntd2d_action](#), 6
- [ntd2d_action.files](#), 7
 - [ntd2d_action.files.borgedconffile](#), 7
 - [ntd2d_action.files.conffile](#), 8
 - [ntd2d_action.files.cssfile](#), 8
 - [ntd2d_action.files.file](#), 9
 - [ntd2d_action.files.indexfile](#), 9
 - [ntd2d_action.files.menufile](#), 9
 - [ntd2d_action.files.nojekyllfile](#), 9
 - [ntd2d_action.files.pagesfile](#), 9
 - [ntd2d_action.files.sphinxlog](#), 10
 - [ntd2d_action.files.template](#), 10
 - [ntd2d_action.files.templatehierarchy](#), 10
 - [ntd2d_action.files.variantsfile](#), 10
- [ntd2d_action.repository](#), 11
- [ntd2d_action.sphinxdocs](#), 11
- [ntd2d_action.variants](#), 12

INDEX

A

`append()` (*ntd2d_action.variants.VariantCollection*
method), 12
`assimilate_theme()` (*ntd2d_action.sphinxdocs.BorgedSphinxDocs*
method), 11

B

`BorgedConfFile` (class in *ntd2d_action.files.borgedconf*), 7
`BorgedSphinxDocs` (class in *ntd2d_action.sphinxdocs*),
11
`build_docs()` (*ntd2d_action.sphinxdocs.BorgedSphinxDocs*
method), 11
`build_docs()` (*ntd2d_action.sphinxdocs.SphinxDocs*
method), 12

C

`clear()` (*ntd2d_action.variants.VariantCollection*
method), 12
`ConfFile` (class in *ntd2d_action.files.conf*), 8
`count()` (*ntd2d_action.variants.VariantCollection*
method), 12
`css_name` (*ntd2d_action.variants.Variant* property), 12
`css_name` (*ntd2d_action.variants.Version* property), 13
`CSSFile` (class in *ntd2d_action.files.css*), 8

E

environment variable
 GITHUB_REF_NAME, 11
 GITHUB_REPOSITORY, 11, 18
 GITHUB_SERVER_URL, 11
 GITHUB_SHA, 11
`extend()` (*ntd2d_action.variants.VariantCollection*
method), 13

F

`File` (class in *ntd2d_action.files*), 9
`FileTemplate` (class in *ntd2d_action.files*), 10

G

`get_contents()` (*ntd2d_action.files.indexfile.IndexFile*
method), 9

GITHUB_REF_NAME, 11
GITHUB_REPOSITORY, 11, 18
GITHUB_SERVER_URL, 11
GITHUB_SHA, 11

H

`html_theme` (*ntd2d_action.files.borgedconf.BorgedConfFile*
property), 7
`html_theme` (*ntd2d_action.files.conf.ConfFile* prop-
erty), 8
`html_theme_path` (*ntd2d_action.files.borgedconf.BorgedConfFile*
property), 7
`html_theme_path` (*ntd2d_action.files.conf.ConfFile*
property), 8

I

`index()` (*ntd2d_action.variants.VariantCollection*
method), 13
`IndexFile` (class in *ntd2d_action.files.index*), 9
`inherited_layout` (*ntd2d_action.sphinxdocs.BorgedSphinxDocs*
property), 11
`insert()` (*ntd2d_action.variants.VariantCollection*
method), 13

L

`language` (*ntd2d_action.files.borgedconf.BorgedConfFile*
property), 7
`language` (*ntd2d_action.files.conf.ConfFile* property),
8

M

`MenuFile` (class in *ntd2d_action.files.menu*), 9
module
 ntd2d_action, 6
 ntd2d_action.files, 7
 ntd2d_action.files.borgedconf, 7
 ntd2d_action.files.conf, 8
 ntd2d_action.files.css, 8
 ntd2d_action.files.file, 9
 ntd2d_action.files.indexfile, 9
 ntd2d_action.files.menufile, 9
 ntd2d_action.files.nojekyllfile, 9

[ntd2d_action.files.pagesfile](#), 9
[ntd2d_action.files.sphinxlog](#), 10
[ntd2d_action.files.template](#), 10
[ntd2d_action.files.templatehierarchy](#), 10
[ntd2d_action.files.variantsfile](#), 10
[ntd2d_action.repository](#), 11
[ntd2d_action.sphinxdocs](#), 11
[ntd2d_action.variants](#), 12

N

[NoJekyllFile](#) (class in [ntd2d_action.files.nojekyllfile](#)), 9
[ntd2d_action](#)
 module, 6
[ntd2d_action.files](#)
 module, 7
[ntd2d_action.files.borgedconf](#)file
 module, 7
[ntd2d_action.files.conf](#)file
 module, 8
[ntd2d_action.files.css](#)file
 module, 8
[ntd2d_action.files.file](#)
 module, 9
[ntd2d_action.files.index](#)file
 module, 9
[ntd2d_action.files.menu](#)file
 module, 9
[ntd2d_action.files.nojekyll](#)file
 module, 9
[ntd2d_action.files.pages](#)file
 module, 9
[ntd2d_action.files.sphinx](#)log
 module, 10
[ntd2d_action.files.template](#)
 module, 10
[ntd2d_action.files.templatehierarchy](#)
 module, 10
[ntd2d_action.files.variants](#)file
 module, 10
[ntd2d_action.repository](#)
 module, 11
[ntd2d_action.sphinx](#)docs
 module, 11
[ntd2d_action.variants](#)
 module, 12

P

[PagesFile](#) (class in [ntd2d_action.files.pagesfile](#)), 9
[PagesTemplate](#) (class in [ntd2d_action.files.template](#)),
 10
[parse_sphinx_warnings\(\)](#)
 ([ntd2d_action.files.sphinxlog.SphinxLog](#)
 method), 10

[pop\(\)](#) ([ntd2d_action.variants.VariantCollection](#)
 method), 13
[project](#) ([ntd2d_action.files.borgedconf](#)file.[BorgedConfFile](#)
 property), 7
[project](#) ([ntd2d_action.files.conf](#)file.[ConfFile](#) property),
 8
 Python Enhancement Proposals
 PEP 440, 5, 13

R

[read\(\)](#) ([ntd2d_action.files.borgedconf](#)file.[BorgedConfFile](#)
 method), 7
[read\(\)](#) ([ntd2d_action.files.conf](#)file.[ConfFile](#) method), 8
[remove\(\)](#) ([ntd2d_action.variants.VariantCollection](#)
 method), 13
[Repository](#) (class in [ntd2d_action.repository](#)), 11
[reverse\(\)](#) ([ntd2d_action.variants.VariantCollection](#)
 method), 13

S

[SphinxDocs](#) (class in [ntd2d_action.sphinxdocs](#)), 12
[SphinxLog](#) (class in [ntd2d_action.files.sphinxlog](#)), 10

T

[Template](#) (class in [ntd2d_action.files.template](#)), 10
[TemplateHierarchy](#) (class in [ntd2d_action.files.templatehierarchy](#)), 10

U

[update_pages\(\)](#) ([ntd2d_action.repository.Repository](#)
 method), 11

V

[Variant](#) (class in [ntd2d_action.variants](#)), 12
[VariantCollection](#) (class in [ntd2d_action.variants](#)),
 12
[VariantCollector](#) (class in [ntd2d_action.variants](#)), 13
[VariantsFile](#) (class in [ntd2d_action.files.variantsfile](#)),
 10
[Version](#) (class in [ntd2d_action.variants](#)), 13