



Example Documentation

Release arno.2015.1.0 (2356f56)

OPNFV

August 12, 2016

CONTENTS

1	How to create documentation for your OPNFV project	1
1.1	Directory Structure	1
1.2	Index File	1
1.3	Source Files	2
1.4	Writing RST Markdown	2
1.5	Configuration	2
1.6	Versioning	3
1.7	Testing	3
1.8	Jenkins Jobs	4
1.9	Sphinx Extensions	4
1.10	Hooks	4

HOW TO CREATE DOCUMENTATION FOR YOUR OPNFV PROJECT

1.1 Directory Structure

This is the directory structure of the docs/ directory which have to be placed in the root of your project directory.

```
./how-to-use-docs/documentation-example.rst
./how-to-use-docs/index.rst
```

To create your own document, create any number of directories (depending on your need, e.g. manual) under the docs/ and place an index.rst in each directories. The depth of all directory should be one, so that you can make sure that all directory names are unique. If you want to have set of all documents in your repo, create new docs/all/index.rst and list document links in OPNFV artifact server (artifact.opnfv.org) instead of including all other rst files or having docs/index.rst, in order to avoid having duplicated contents in your documents.

Note: You may have “docs/how-to-use-docs/” in you project repo. You can delete it, since it is sample and master version is stored in opnfvdocs repo.

Note: During the document build process, ‘docs_build’ and ‘docs_output’ will be created in the project root directory. To avoid conflict, you should not use those names and should add the following entries in ‘.gitignore’ file so that git can ignore built files.

```
/docs_build/
/docs_output/
```

1.2 Index File

This index file must reference your other rst files in that directory.

Here is an example index.rst :

```
*****
Documentation Title
*****

.. toctree::
   :numbered:
   :maxdepth: 2

   documentation-example.rst
```

1.3 Source Files

Document source files have to be written in reStructuredText format (rst). Each file would be build as an html page and a chapter in PDF.

Here is an example source rst file :

```
=====  
Chapter Title  
=====  
  
Section Title  
=====  
  
Hello!
```

1.4 Writing RST Markdown

See <http://sphinx-doc.org/rest.html> .

Hint: Table and its contents won't be adjusted, so you may need to fix your source text when your table is truncated in PDF version. Or, you can use 'longtable' option that splits your table vertically (by rows) in multiple pages. It is useful if you have trouble in rendering table containing many rows.

```
.. table::  
   :class: longtable  
  
   +-----+-----+-----+-----+  
   | Header row, column 1 | Header 2 | Header 3 | Header 4 |  
   +-----+-----+-----+-----+  
   | body row 1, column 1 | column 2 | column 3 | column 4 |  
   +-----+-----+-----+-----+  
   | body row 2          | ...     | ...     |          |  
   +-----+-----+-----+-----+
```

Hint: You can add dedicated contents by using 'only' directive with build type ('html' and 'pdf') for OPNFV document. But, this is not encouraged to use since this may make different views in HTML and PDF version.

```
.. only:: html  
   This line will be shown only in html version.
```

1.5 Configuration

If you need to change the default configuration for document build, create new conf.py in the document directory (e.g. 'docs/how-to-use-docs/conf.py') that will be used in build process instead of default for OPNFV document build. During the build process, the following default parameters are automatically added if they are not set in the conf.py.

- **extensions** = ['sphinxcontrib.httpdomain', 'sphinx.ext.autodoc', 'sphinx.ext.viewcode', 'sphinx.ext.napoleon']
- **needs_sphinx** = '1.3'
- **numfig** = True
- **master_doc** = 'index'
- **pygments_style** = 'sphinx'

- `html_use_index = False`
- `html_logo = 'opnfv-logo.png'`
- `latex_domain_indices = False`
- `latex_logo = 'opnfv-logo.png'`
- `latex_documents = [('index', 'document directory name.tex', 'document title in index.rst', 'OPNFV', 'manual'),]`
- `release = 'git last tag name (git last commit hash)'`
- `version = 'git last tag name (git last commit hash)'`
- `project = 'git repo name'`
- `copyright = 'year, OPNFV'`
- `rst_epilog = 'Revision:git last commit hashn Build date:build date in year-month-date format'`

Note: You can leave the file path for OPNFV logo image which will be prepared before each document build.

Hint: In PDF, figures will be floated to get better view. If you want to avoid such automated fixes, just add this option to your `conf.py` after copying the default configuration in to the document directory.

```
latex_elements = {'figure_align': 'H'}
```

1.6 Versioning

The relevant release and version information will be added to your documents by using tags from your project's git repository. The tags will be applied by Releng Project.

1.7 Testing

You can test document build in your laptop by using build script which is also used in document build jobs below:

```
$ sudo pip install virtualenv
$ cd /local/repo/path/to/project
$ git clone https://git.opnfv.org/opnfvdocs docs_build/_opnfvdocs
$ ./docs_build/_opnfvdocs/scripts/docs-build.sh
```

Then, you can see the docs in 'docs_output' directory if build succeeded.

If you have a problem of missing python package, you can make sure all necessary packages are installed as follows:

```
$ sudo pip install Sphinx==1.3.1 doc8 sphinxcontrib-httpdomain
```

Note: Developers are encouraged to use "ssh://<username>@gerrit.opnfv.org:29418/opnfvdocs" instead of "https://git.opnfv.org/opnfvdocs", so that you can quickly start development in opnfvdocs. See <https://wiki.opnfv.org/display/DEV/Developer+Getting+Started> for more detail.

1.8 Jenkins Jobs

1.8.1 Enabling Jenkins Jobs

Jenkins in OPNFV infra performs the jobs to verify and update your documents. To make your project repository watched by Jenkins to execute those jobs, you have to add your project name in 'project-pattern' of the following jobs by sending patch to update `jjb/opnfv/opnfv-docs.yml` on Gerrit.

1.8.2 Verify Job

The verify job name is `opnfv-docs-verify`.

When you send document changes to Gerrit, Jenkins will create your documents in HTML and PDF formats to verify that new document can be built successfully. Please check the Jenkins log and artifact carefully. You can improve your document even though if the build job succeeded.

Documents will be uploaded to `http://artifacts.opnfv.org/review/<Change Number>/` for review. Those documents will be replaced if you update the change by sending new patch set to Gerrit, and deleted after the change is merged. Document link(s) can be found in your change page on Gerrit as a review comment.

Note: Currently, the job reports 'SUCCESS' as result of document build even if the PDF creation failed. This is a provisional workaround, since many projects are not ready for PDF creation yet.

1.8.3 Merge Job

The merge job name is `opnfv-docs-merge`.

Once you are happy with the look of your documentation, you can submit the change. Then, the merge job will upload latest build documents to `http://artifacts.opnfv.org/<Project Name>/docs/`. You can put links in your project wiki page, so that everyone can see the latest document always.

1.9 Sphinx Extensions

You can see available Sphinx extension(s) in `opnfvdocs/etc/requirements.txt`.

You can use other Sphinx extensions to improve your documents. To share such improvements, we encourage you to enable the extension in OPNFV infra by asking releng and opnfvdocs teams to add new Sphinx extension via Gerrit (proposing change in `opnfvdocs/scripts/docs-build.sh` and `opnfvdocs/etc/requirements.txt`). After quick sanity checks, we'll merge the patch to make it available in OPNFV document build.

1.10 Hooks

The document builder script supports pre-hook. If you need to execute some commands before starting to build process (e.g. generate API rst from codes), create '`<your repo>/docs/pre-hook.sh`' which will be executed (using 'source') once before all document build start.