Example Documentation

Release test-tag (16a0bf3)

OPNFV

February 15, 2016

CONTENTS

CHAPTER

ONE

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 releng repo.

1.2 Index File

This index file must refence 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 .

You can add dedicated contents by using 'only' directive with build type ('html' and 'pdf') for OPNFV document

Example :

```
.. 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. The OPNFV default configuration can be found in releng repo (see docs/etc/conf.py).

In the build process, the following parameters are automatically added if they are not set in the conf.py .

- **release**, **version** : `git last tag name` (`git last commit hash`)
- **project** : `git repo name`
- **copyright** : `year`, OPNFV
- **latex_documents** (set of pdf configuration) : [('index', '`document directory name`.tex', '`document title in index.rst`', 'OPNFV', 'manual'),]

See http://sphinx-doc.org/config.html to learn sphinx configuration.

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

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 used in document build jobs:

```
$ cd /loacal/repo/path/to/project
$ git clone ssh://gerrit.opnfv.org:29418/releng
$ ./releng/utils/docs-build.sh
```

Then, you can see docs in output directory if build succeeded.

This script will generate files in 'docs_build' and 'docs_output'. You should consider to add the following entries in '.gitignore' file, so that git can ignore built files.

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

Note, you need some python packages which can be installed as follows:

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

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 docs/etc/requirements.txt.

You can use other sphinx extensions to improve your documents. To share such tips, 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 docs/etc/conf.py and docs/etc/requirements.txt). After quick sanity checks, we'll install python package (if needed) and make it available in OPNFV document build.