doconv Documentation

Release 0.1.7

Jacob Mourelos

Jan 30, 2019

Contents

Contents:

CHAPTER 1

Introduction

Warning: doconv is immature software.

Currently doconv only allows to convert from AsciiDoc and DocBook to DITA, but new format conversions are to be added.

CHAPTER 2

Features

- Conversion from AsciiDoc to DITA
- Conversion from DocBook to DITA
- Plugins' orchestration to perform conversions not achievable by a single plugin
- Easily extensible by adding new plugins

2.1 Available Format Conversions

Each node of the graph represents a document format. Each arrow represents the conversion between two formats (being provided by the plugin labeling the arrow). Note that some arrows are unidirectional indicating that the conversion only works in one way.

CHAPTER 3

Installation

3.1 Pre-requisites

In addition to any dependency described in the installation section some plugins require additional dependencies to be enabled. See the *plugins section* for more information.

3.2 Arch Linux

doconv is available in AUR, it can be installed using a tool like yaourt:

```
$ sudo yaourt -S doconv
```

3.3 Other distributions (generic installation)

Some development dependencies need to be installed. In Ubuntu would be:

```
$ sudo apt-get install libxml2-dev libxslt1-dev python-dev
```

Then install latest stable doconv version using pip:

```
$ pip install doconv
```

or, if administrative privileges are needed:

```
$ sudo pip install doconv
```

CHAPTER 4

Plugins

4.1 Introduction

Some of the plugins provide overlapping functionality, like asciidoc and asciidoctor plugins. In that cases it suffices to install the dependencies for one of the overlapping plugins.

Usually a plugin has some pre-requisites to be used. If doconv detects in the system the dependencies needed for the plugin to work, the plugin will be enabled.

4.2 asciidoc

The asciidoc plugin converts from AsciiDoc to DocBook.

4.2.1 Pre-requisites for activation

asciidoc should be installed:

Arch Linux:

```
$ pacman -S asciidoc
```

Debian/Ubuntu:

```
$ apt-get install asciidoc
```

CentOS/Fedora:

```
$ yum install asciidoc
```

See asciidoc website for more information.

Note: asciidoc requires Python 2.x to be available in the system.

4.3 asciidoctor

The asciidoctor plugin converts from AsciiDoc to DocBook.

4.3.1 Pre-requisites for activation

asciidoctor should be installed:

See asciidoctor website for installation instructions.

4.4 docbooktodita

The docbooktodita plugin converts from DocBook to DITA.

4.4.1 Pre-requisites for activation

There are no pre-requisites.

CHAPTER 5

Usage

doconv CLI is quite straight-forward to use:

```
$ doconv
```

```
usage: doconv convert [-h] [-o OUTPUT_FILE]
                      <input_file> <input_format> <output_format>

positional arguments:
  <input_file>          input file to be converted
  <input_format>        format of <input_file>
  <output_format>       format to convert <input_file> to

optional arguments:
  -h, --help            show this help message and exit
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                        write output of the conversion to OUTPUT_FILE
```

A typical usage of doconv could be:

```
$ doconv convert asciidoc_file.txt asciidoc dita
$
$ Conversion successful: file asciidoc_file.dita generated
```

Generating a DITA file as result.

CHAPTER 6

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

6.1 Types of Contributions

6.1.1 Report Bugs

Report bugs at https://github.com/jmourelos/doconv/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

6.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" is open to whoever wants to implement it.

6.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

6.1.4 Write Documentation

doconv could always use more documentation, whether as part of the official doconv docs, in docstrings, or even on the web in blog posts, articles, and such.

6.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/jmourelos/doconv/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

6.2 Get Started!

Ready to contribute? Here's how to set up *doconv* for local development.

1. Fork the *doconv* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/doconv.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv doconv
$ cd doconv/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
  $ flake8 doconv tests
  $ python setup.py test
  $ tox

To get flake8 and tox, just pip install them into your virtualenv.
```

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

6.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for the Python versions specified in tox.ini. Check https://travis-ci.org/jmourelos/doconv/pull_requests and make sure that the tests pass for all supported Python versions.

CHAPTER 7

Creating a plugin

Creating a plugin should be quite easy, it requires 3 steps:

- Extend the base plugin class, implementing these abstract methods:

  class doconv.plugin.base.**PluginBase**

  check_dependencies()
  Check that all neccessary dependencies for a particular plugin are available in the system, raise an exception otherwise.

  convert(*input_file*, *output_file*, *input_format*, *output_format*)
  Convert a input_file from a specified format to another

  get_supported_conversions()
  Return the conversions provided by the plugin implementing this method.

- If the plugin requires some external files, for example XSLT files, store them in a directory with the plugin's name and remember to add the directory to the MANIFEST.in.

- Add an entry point for your new plugin to *setup.py*:

```
entry_points={
    'console_scripts': [
        'doconv = doconv.doconv:main'
        ],
    'doconv.converter': [
        'asciidoc = doconv.plugin.asciidoc:AsciiDoc',
        'docbooktodita = doconv.plugin.docbooktodita:DocBookToDita',
        'yournewplugin = doconv.plugin.yournewplugin:YourNewPlugin',
        ],
    },
```

You are done! In case you would like to contribute your new plugin, see *how to contribute*.

CHAPTER 8

Credits

8.1 Development Lead

- Jacob Mourelos <jacob.mourelos@gmail.com>

8.2 Contributors

None yet. Why not be the first?

8.3 Attributions

Most docbooktodita plugin's code is under the Apache License Version 2.0 as it was mercilessly copied from the docbook2dita plugin of the DITA Open Toolkit. As required in section 4.a of the forementioned license, a copy of the license is available in the "misc" directory, being also available online.

CHAPTER 9

Release History

9.1 0.1.0 (2013-10-16)

- Initial public release.
- Conversion from AsciiDoc and DocBook to DITA.
- Plugins' orchestration to perform conversions not achievable by a single plugin.

9.2 0.1.1 (2013-10-20)

- Fix several installation issues.
- Remove temporarily python 3 support.

9.3 0.1.2 (2013-11-09)

- Expand and improve documentation.
- Proper logging (available using –verbose CLI option).
- New plugins are now easier to develop.

9.4 0.1.3 (2013-12-26)

- Add asciidoctor plugin.
- Add doconv command to query available document formats.

9.5 0.1.4 (2014-03-05)

- Preparations for native packaging for Ubuntu and Arch Linux.

9.6 0.1.5 (2016-10-17)

- Fixed issue converting notes from AsciiDoc to DITA.
- Added support for Python 3.4 and 3.5.

9.7 0.1.6 (2016-10-17)

- Improve confusing error message under certain circumstances when using Python 3.

9.8 0.1.7 (2019-01-29)

- Drop support for Python 2.6 and 3.3
- Update release process
- Fix bug preventing doconv to work with latest version of some dependencies

CHAPTER 10

Indices and tables

- genindex
- modindex
- search

Index