
trialbureautools Documentation

Release 0.3.10

Sjoerd Kerkstra

Sep 13, 2019

CONTENTS:

1	Features	3
2	Getting started	5
3	Credits	7
3.1	trialbureautools	7
3.2	Installation	7
3.3	Usage	8
3.4	Modules	10
3.5	Contributing	10
3.6	Credits	12
3.7	History	13

Command line utility for the radboudumc trial bureau.

- Free software: MIT license
- Documentation: <https://trialbureautools.readthedocs.io>.

FEATURES

- Sort DICOM files into folders based on DICOM tag patterns
- Set windows icacis folder permissions

GETTING STARTED

See [Getting started](#)

CREDITS

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

3.1 trialbureautools

Command line utility for the radboudumc trial bureau.

- Free software: MIT license
- Documentation: <https://trialbureautools.readthedocs.io>.

3.1.1 Features

- Sort DICOM files into folders based on DICOM tag patterns
- Set windows icacfs folder permissions

3.1.2 Getting started

See [Getting started](#)

3.1.3 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

3.2 Installation

3.2.1 Stable release

To install trialbureautools, run this command in your terminal:

```
$ pip install trialbureautools
```

This is the preferred method to install trialbureautools, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

3.2.2 From sources

The sources for trialbureautools can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/sjoerdk/trialbureautools
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/sjoerdk/trialbureautools/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

3.3 Usage

3.3.1 Basics

The trial bureau tools are accessed from command line. On windows you can start a command line (also called 'terminal' or 'shell') in the following way:

- Press windows-key + r
- type 'cmd' and press enter

After that, typing 'tbt' + enter will show the different commands available trial bureau tools.

You can add '-help' to any command to see more info on using it. For example `tbt sort --help`

3.3.2 DICOM sorter

To sort all DICOM files in directory `job1` using the `nucmed` pattern:

```
$ tbt sorter sort job1 nucmed
```

To see what the 'nucmed' pattern is exactly:

```
$ tbt sorter pattern list
```

By default, sorting `/folder1` will write sorted output to `/folder1_sorted`. You can also set the output folder manually:

```
$ tbt sorter sort job1 nucmed --output_folder C:/temp/myfolder
```

3.3.3 DICOM path patterns

Typing `tbt sorter pattern list` will by default show the following:

```
$ tbt sorter pattern list
> idis: (0010,0020)/(0008,1030)-(0008,0050)/(0008,103e)-(0008,0060)-(0020,0011)/
↪ (count:SOPInstanceUID)
> nucmed: (0010,0020)/(0008,1030)-(0008,0050)/(0008,0020)/(0008,103e)-(0008,0060)-
↪ (0020,0011)/(count:SOPInstanceUID)
```

These are the DICOM path patterns. These are text strings consisting of any combination of the following elements:

Element	Examples	Description
Text	Folder file- /tmp. .dcm	Just any free text, inc. slash and -
Dicom Tag Code	(0010,0020) (0008,103e)	A dicom tag with colons around it
Dicom Tag Name	(PatientID) (SopInstanceUID)	A dicom tag name
count: marker	(count:PatientID) (count:0008,103e)	Will number the given tag

To see a list of available Dicom tag codes and names, use `tbt sorter pattern list_dicomtags`

Dicom Pattern resolution

For each DICOM file a new path will be created by reading its values and filling in the pattern. For example, for a file with PatientID '1234' and SopInstanceUID '1.1.1':

Pattern	Result (1 file)
/folder1/(PatientID)/(SopInstanceUID).dcm	/folder1/1234/1.1.1.dcm

Counting

Counting can be useful to make paths shorter. By adding `count:` to an element, the sorter will number each unique value. For example, for three files with the same PatientID but separate SopInstanceUIDs:

Pattern without :count	Result (3 files)
/folder1/(PatientID)/(SopInstanceUID).dcm	/folder1/1234/1.3451.35356.4234.1.dcm
	/folder1/1234/1.3451.35356.4234.2.dcm
	/folder1/1234/1.3451.35356.4234.3.dcm

Pattern with :count	Result (3 files)
/folder1/(PatientID)/ file(count:SopInstanceUID).dcm	/folder1/1234/file0.dcm
	/folder1/1234/file1.dcm
	/folder1/1234/file2.dcm

Adding custom patterns

To add a new pattern `/folder1/(PatientID)/file(count:SopInstanceUID).dcm` named `test`:

```
$ tbt sorter pattern add test /folder1/(PatientID)/file(count:SopInstanceUID).dcm
```

To remove this pattern again:

```
$ tbt sorter pattern remove test
```

3.4 Modules

3.4.1 dicomsort

Core

mappers

parser

folderprocessing

3.5 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:

3.5.1 Types of Contributions

Report Bugs

Report bugs at <https://github.com/sjoerdk/trialbureautools/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.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

Write Documentation

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

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/sjoerdk/trialbureautools/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 :)

3.5.2 Get Started!

Ready to contribute? Here’s how to set up *trialbureautools* for local development.

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

```
$ git clone git@github.com:your_name_here/trialbureautools.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 trialbureautools
$ cd trialbureautools/
$ 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 trialbureautools tests
$ python setup.py test or 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.

3.5.3 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 Python 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/sjoerdk/trialbureautools/pull_requests and make sure that the tests pass for all supported Python versions.

3.5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_trialbureautools
```

3.5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch  
$ git push  
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

3.6 Credits

3.6.1 Development Lead

- Sjoerd Kerkstra <sjoerd.kerkstra@radboudumc.nl>

3.6.2 Contributors

None yet. Why not be the first?

3.7 History

3.7.1 0.1.0 (2018-12-05)

- Alpha