
orsopy Documentation

Release 1.1.0

Andrew R. McCluskey

Aug 31, 2023

CONTENTS:

1	Features	3
	Python Module Index	21
	Index	23

orsopy is a Python library that implements ORSO functionality, which currently includes the [reduced data file format](#). The orsopy package is used by a range of data reduction and analysis packages for the writing and reading of reduced reflectometry data. This data is written following the [ORSO defined specification](#), enabling a metadata-rich and flexible file to be created.

[ORSO](#) is an open organisation aimed at improving the scientific techniques of neutron and X-ray reflectometry. In the interest of transparency, all minutes from orsopy developer meetings are available in the [Documents](#) in the sidebar of this page. If you are interested in getting involved in developing orsopy, please feel free to [contribute](#) or get in touch on the [ORSO Slack](#) (where there is a channel dedicated to orsopy).

FEATURES

- Reading and writing of ORSO specification reduced reflectivity files.

1.1 Installation

1.1.1 Stable release

To install orsopy, run this command in your terminal:

```
$ pip install orsopy
```

This is the preferred method to install orsopy, 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.

1.1.2 From sources

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

You can either clone the public repository:

```
$ git clone git://github.com/reflectivity/orsopy
```

Or download the [tarball](#):

```
$ curl -OJL https://github.com/reflectivity/orsopy/tarball/master
```

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

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

1.2 Usage

1.2.1 Usage examples

- Producing metadata rich reduced datasets

1.3 Modules

1.3.1 fileio

The role of the `fileio` module is to enable the creation of and parsing from ORSO reduced data files. All public classes and functions in the `fileio` module are available directly from `fileio` without needing to specify a particular submodule.

`orsopy.fileio.base`

Implementation of the base classes for the ORSO header.

`orsopy.fileio.base.orsodataclass(cls)`

exception `orsopy.fileio.base.ORSOResolveError`

Bases: `ValueError`

class `orsopy.fileio.base.Header`

Bases: `object`

The super class for all of the items in the orso module.

property `user_data`

classmethod `empty()`

Create an empty instance of this item containing all non-option attributes as `None`.

Return type

`Header`

Returns

Empty class.

static `asdict(header)`

Static method for `to_dict()`.

Parameters

header (`Header`) – Object to convert to dictionary.

Return type

`dict`

Returns

Dictionary result.

to_dict()

Produces a clean dictionary of the Header object, removing any optional attributes with the value `None`.

Return type

`dict`

Returns

Cleaned dictionary.

to_yaml()

Return the yaml string for the Header item

Return type

str

Returns

Yaml string

yaml_representer(dumper)**yaml_representer_compact(dumper)**

```
class orsopy.fileio.base.OrsoDumper(stream, default_style=None, default_flow_style=False,
                                     canonical=None, indent=None, width=None, allow_unicode=None,
                                     line_break=None, encoding=None, explicit_start=None,
                                     explicit_end=None, version=None, tags=None, sort_keys=True)
```

Bases: SafeDumper

represent_data(data)

```
class orsopy.fileio.base.ErrorValue(error_value, error_type=None, value_is=None, distribution=None,
                                     comment=None, **user_kwds)
```

Bases: [Header](#)

Information about errors on a value.

error_value: float**error_type:** Optional[Literal['uncertainty', 'resolution']] = None**value_is:** Optional[Literal['sigma', 'FWHM']] = None**distribution:** Optional[Literal['gaussian', 'triangular', 'uniform', 'lorentzian']] = None**yaml_representer(dumper)****property sigma**

Return value converted to standard deviation.

The conversion factors can be found in common statistics and experimental physics text books or derived manually solving the variance definition integral. (e.g. Dekking, Michel (2005). A modern introduction to probability and statistics : understanding why and how. Springer, London, UK:) Values and some references available on Wikipedia, too.

comment: Optional[str] = None

```
class orsopy.fileio.base.Value(magnitude, unit=None, error=None, comment=None, **user_kwds)
```

Bases: [Header](#)

A value or list of values with an optional unit.

magnitude: float**unit:** Optional[str] = None

error: Optional[[ErrorValue](#)] = None

yaml_representer(*dumper*)

as_unit(*output_unit*)

Returns the value as converted to the given unit.

comment: Optional[str] = None

class orsopy.fileio.base.**ComplexValue**(*real, imag=None, unit=None, error=None, comment=None, **user_kwds*)

Bases: [Header](#)

A value or list of values with an optional unit.

real: float

imag: Optional[float] = None

unit: Optional[str] = None

error: Optional[[ErrorValue](#)] = None

yaml_representer(*dumper*)

as_unit(*output_unit*)

Returns the complex value as converted to the given unit.

comment: Optional[str] = None

class orsopy.fileio.base.**ValueRange**(*min, max, unit=None, comment=None, **user_kwds*)

Bases: [Header](#)

A range or list of ranges with mins, maxs, and an optional unit.

min: float

max: float

unit: Optional[str] = None

yaml_representer(*dumper*)

as_unit(*output_unit*)

Returns a (min, max) tuple of values as converted to the given unit.

comment: Optional[str] = None

class orsopy.fileio.base.**ValueVector**(*x, y, z, unit=None, error=None, comment=None, **user_kwds*)

Bases: [Header](#)

A vector or list of vectors with an optional unit. For vectors relating to the sample, such as polarisation, the follow definitions are used.

Parameters

- **x** (float) – is defined as parallel to the radiation beam, positive going with the beam direction.
- **y** (float) – is defined from the other two based on the right hand rule.

- **z** (float) – is defined as normal to the sample surface, positive direction in scattering direction.
- **unit** (Optional[str]) – SI unit string.

x: float

y: float

z: float

unit: Optional[str] = None

error: Optional[ErrorValue] = None

yaml_representer(dumper)

as_unit(output_unit)

Returns a (x, y, z) tuple of values as converted to the given unit.

comment: Optional[str] = None

class orsopy.fileio.base.**Person**(name, affiliation, contact=None, comment=None, **user_kwds)

Bases: [Header](#)

Information about a person, including name, affiliation(s), and contact information.

name: str

affiliation: str

contact: Optional[str] = None

comment: Optional[str] = None

class orsopy.fileio.base.**Column**(name, unit=None, physical_quantity=None, comment=None, **user_kwds)

Bases: [Header](#)

Information about a data column.

name: str

unit: Optional[str] = None

physical_quantity: Optional[str] = None

yaml_representer(dumper)

comment: Optional[str] = None

class orsopy.fileio.base.**ErrorColumn**(error_of, error_type=None, value_is=None, distribution=None, comment=None, **user_kwds)

Bases: [Header](#)

Information about a data column.

error_of: str

error_type: Optional[Literal['uncertainty', 'resolution']] = None

```
value_is: Optional[Literal['sigma', 'FWHM']] = None
```

```
distribution: Optional[Literal['gaussian', 'triangular', 'uniform', 'lorentzian']]  
= None
```

```
yaml_representer(dumper)
```

property name

A convenience property to allow programs to get a valid name attribute for any column.

property to_sigma

The multiplicative factor needed to convert a FWHM to sigma.

The conversion factors can be found in common statistics and experimental physics text books or derived manually solving the variance definition integral. (e.g. Dekking, Michel (2005). A modern introduction to probability and statistics : understanding why and how. Springer, London, UK:) Values and some references available on Wikipedia, too.

```
comment: Optional[str] = None
```

```
class orsopy.fileio.base.File(file, timestamp=None, comment=None, **user_kwds)
```

Bases: [Header](#)

A file with file path and a last modified timestamp.

```
file: str
```

```
timestamp: Optional[datetime] = None
```

```
comment: Optional[str] = None
```

```
exception orsopy.fileio.base.NotOrsoCompatibleFileError
```

Bases: [ValueError](#)

orsopy.fileio.data_source

Implementation of the data_source for the ORSO header.

```
class orsopy.fileio.data_source.Experiment(title, instrument, start_date, probe, facility=None,  
                                           proposalID=None, doi=None, comment=None,  
                                           **user_kwds)
```

Bases: [Header](#)

A definition of the experiment performed.

Parameters

- **title** (str) – Proposal or project title.
- **instrument** (str) – Reflectometer identifier.
- **start_date** (datetime) – Start date for the experiment.
- **probe** (Literal['neutron', 'x-ray']) – Radiation probe, either 'neutron' or 'x-ray'.
- **facility** (Optional[str]) – Facility where the experiment was performed.
- **proposalID** (Optional[str]) – Identifier for experiment at a facility.

- **doi** (Optional[str]) – Digital object identifier for the experiment, possibly provided by the facility.

title: str

instrument: str

start_date: datetime

probe: Literal['neutron', 'x-ray']

facility: Optional[str] = None

proposalID: Optional[str] = None

doi: Optional[str] = None

comment: Optional[str] = None

```
class orsopy.fileio.data_source.Sample(name, category=None, composition=None, description=None,
                                       size=None, environment=None, sample_parameters=None,
                                       model=None, comment=None, **user_kwds)
```

Bases: [Header](#)

A description of the sample measured.

Parameters

- **name** (str) – An identified for the individual sample or the subject and state being measured.
- **category** (Optional[str]) – Simple sample description, front (beam side) / back, each side should be one of 'solid/liquid', 'liquid/solid', 'gas/liquid', 'liquid/liquid', 'solid/gas', 'gas/solid'.
- **composition** (Optional[str]) – Notes on the nominal composition of the sample e.g. Si | SiO2 (20 angstrom) | Fe (200 angstrom) | air (beam side).
- **description** (Optional[str]) – Further details of the sample, e.g. size.
- **size** (Optional[[ValueVector](#)]) – Sample size in x, y, z direction, where z is parallel to the surface normal and x is along the beam direction (important for footprint correction).
- **environment** (Optional[List[str]]) – Name of the sample environment device(s).
- **sample_parameters** (Optional[Dict[str, Union[[Value](#), [ValueRange](#), [ValueVector](#), [ComplexValue](#)]]]) – Dictionary of sample parameters.

name: str

category: Optional[str] = None

composition: Optional[str] = None

description: Optional[str] = None

size: Optional[[ValueVector](#)] = None

environment: Optional[List[str]] = None

sample_parameters: Optional[Dict[str, Union[[Value](#), [ValueRange](#), [ValueVector](#), [ComplexValue](#)]]] = None

model: Optional[SampleModel] = None

comment: Optional[str] = None

class orsopy.fileio.data_source.Polarization(*value*)

Bases: str, Enum

Polarization of the beam used for the reflectivity.

Neutrons: The first symbol indicates the magnetisation direction of the incident beam, the second symbol indicates the direction of the scattered beam. If either polarization or analysis are not employed the symbol is replaced by “o”.

X-rays: Uses the conventional names pi, sigma, left and right. In experiments with polarization analysis the incident and outgoing polarizations are separated with an underscore “_”.

unpolarized = 'unpolarized'

po = 'po'

mo = 'mo'

op = 'op'

om = 'om'

mm = 'mm'

mp = 'mp'

pm = 'pm'

pp = 'pp'

pi = 'pi'

sigma = 'sigma'

left = 'left'

right = 'right'

pi_pi = 'pi_pi'

sigma_sigma = 'sigma_sigma'

pi_sigma = 'pi_sigma'

sigma_pi = 'sigma_pi'

yaml_representer(*dumper*)

class orsopy.fileio.data_source.InstrumentSettings(*incident_angle*, *wavelength*, *polarization*=None, *configuration*=None, *comment*=None, ***user_kwds*)

Bases: [Header](#)

Settings associated with the instrumentation.

Parameters

- **incident_angle** (Union[[Value](#), [ValueRange](#)]) – Angle (range) of incidence.

- **wavelength** (Union[[Value](#), [ValueRange](#)]) – Neutron/x-ray wavelength (range).
- **polarization** (Union[[Polarization](#), [ValueVector](#), None]) – Radiation polarization as one of 'unpolarized', 'p', 'm', 'pp', 'pm', 'mp', 'mm', or a [orsopy.fileio.base.ValueVector](#).
- **configuration** (Optional[str]) – Description of the instrument configuration (full polarized/liquid surface/etc).

incident_angle: Union[[Value](#), [ValueRange](#)]

wavelength: Union[[Value](#), [ValueRange](#)]

polarization: Union[[Polarization](#), [ValueVector](#), None] = None

configuration: Optional[str] = None

comment: Optional[str] = None

```
class orsopy.fileio.data_source.Measurement(instrument_settings, data_files, additional_files=None,
                                         scheme=None, comment=None, **user_kwds)
```

Bases: [Header](#)

The measurement elements for the header.

Parameters

- **instrument_settings** ([InstrumentSettings](#)) – Instrumentation details.
- **data_files** (List[Union[[File](#), str]]) – Raw data files produced in the measurement.
- **references** – Raw reference files used in the reduction.
- **scheme** (Optional[Literal['angle- and energy-dispersive', 'angle-dispersive', 'energy-dispersive']]) – Measurement scheme (one of 'angle-dispersive', 'energy-dispersive'/'angle- and energy-dispersive').

instrument_settings: [InstrumentSettings](#)

data_files: List[Union[[File](#), str]]

additional_files: Optional[List[Union[[File](#), str]]] = None

scheme: Optional[Literal['angle- and energy-dispersive', 'angle-dispersive', 'energy-dispersive']] = None

comment: Optional[str] = None

```
class orsopy.fileio.data_source.DataSource(owner, experiment, sample, measurement, comment=None,
                                         **user_kwds)
```

Bases: [Header](#)

The data_source object definition.

Parameters

- **owner** ([Person](#)) – This refers to the actual owner of the data set, i.e. the main proposer or the person doing the measurement on a lab reflectometer.
- **experiment** ([Experiment](#)) – Details of the experimental.
- **sample** ([Sample](#)) – Sample information.

- **measurement** (*Measurement*) – Measurement specifics.

owner: *Person*

experiment: *Experiment*

sample: *Sample*

measurement: *Measurement*

comment: Optional[str] = None

orsopy.fileio.reduction

The reduction elements for the ORSO header

```
class orsopy.fileio.reduction.Software(name, version=None, platform=None, comment=None,
                                       **user_kwds)
```

Bases: *Header*

Software description.

Parameters

- **name** (str) – Software name.
- **version** (Optional[str]) – Version identified for the software.
- **platform** (Optional[str]) – Operating system.

name: str

version: Optional[str] = None

platform: Optional[str] = None

yaml_representer(*dumper*)

comment: Optional[str] = None

```
class orsopy.fileio.reduction.Reduction(software, timestamp=None, creator=None, corrections=None,
                                          computer=None, call=None, script=None, binary=None,
                                          comment=None, **user_kwds)
```

Bases: *Header*

A description of the reduction that has been performed.

Parameters

- **software** (*Software*) – Software used for reduction.
- **timestamp** (Optional[datetime]) – Datetime of reduced file creation.
- **creator** (Optional[*Person*]) – The person or routine who created the reduced file.
- **corrections** (Optional[List[str]]) – A list of the corrections that have been performed.
- **computer** (Optional[str]) – Name of the reduction machine.
- **call** (Optional[str]) – Command line call or similar.
- **script** (Optional[str]) – Path to reduction script or notebook.
- **binary** (Optional[str]) – Path to full reduction information file.

```

software: Software

timestamp: Optional[datetime] = None

creator: Optional[Person] = None

corrections: Optional[List[str]] = None

computer: Optional[str] = None

call: Optional[str] = None

script: Optional[str] = None

binary: Optional[str] = None

comment: Optional[str] = None

```

orsopy.fileio.orso

Implementation of the top level class for the ORSO header.

```
class orsopy.fileio.orso.Orso(data_source, reduction, columns, data_set=None, **user_data)
```

Bases: *Header*

The Orso object collects the necessary metadata.

Parameters

- **data_source** (*DataSource*) – Information about the origin and ownership of the raw data.
- **reduction** (*Reduction*) – Details of the data reduction that has been performed. The content of this section should contain enough information to rerun the reduction.
- **columns** (List[Union[*Column*, *ErrorColumn*]]) – Information about the columns of data that will be contained in the file.
- **data_set** (Union[int, str, None]) – An identifier for the data set, i.e. if there is more than one data set in the object.

```
data_source: DataSource
```

```
reduction: Reduction
```

```
columns: List[Union[Column, ErrorColumn]]
```

```
data_set: Union[int, str, None] = None
```

```
classmethod empty()
```

Create an empty instance of the ORSO header with all non-optional attributes as None.

Return type

Orso

Returns

Empty Orso class, within minimum required columns

```
property user_data
```

column_header()

An information string that explains what each of the columns in a dataset corresponds to.

Return type

str

Returns

Explanatory string.

from_difference(other_dict)

Constructs another *Orso* instance from self, and a dict containing updated header information.

Parameters

other_dict (dict) – Contains updated header information.

Return type

Orso

Returns

A new *Orso* object constructed from self, and the updated header information.

to_difference(other)

A dictionary containing the difference in header information between two *Orso* objects.

Parameters

other (*Orso*) – Other header to diff with.

Return type

dict

Returns

Dictionary of the header information difference.

to_dict()

Adds the user data to the returned dictionary.

comment: Optional[str] = None

class orsopy.fileio.orsopy.*OrsoDataset*(info, data)

Bases: object

Parameters

- **info** (*Orso*) – The header information for the reflectivity measurement
- **data** (Any) – The numerical data associated with the reflectivity measurement. The data has shape (npnts, ncols).

Raises

ValueError – When ncols != len(self.info.columns).

info: *Orso*

data: Any

header()

The header string for the ORSO file.

Return type

str

Returns

Header string.

diff_header(*other*)

Return a header string that only contains changes to other *OrsoDataset* ensure that data_set is the first entry.

Parameters

other (*OrsoDataset*) – Other *OrsoDataset* to compare against.

Return type

str

Returns

Header string with only changes.

save(*fname*)

Save the *OrsoDataset*.

Parameters

fname (Union[TextIO, str]) – The file name to save to.

orsopy.fileio.orso.**save_orso**(*datasets*, *fname*, *comment=None*, *data_separator=""*)

Saves an ORSO file. Each of the datasets must have a unique *OrsoDataset.info.data_set* attribute. If that attribute is not set, it is given an integer value corresponding to it's position in the list.

Parameters

- **datasets** (List[*OrsoDataset*]) – List of OrsoDataset to save into the Orso file.
- **fname** (Union[TextIO, str]) – The file name to save to.
- **comment** (Optional[str]) – Comment to write at the top of Orso file.
- **data_separator** (str) – Optinal string of newline characters to separate multiple datasets.

Raises

ValueError – If the *OrsoDataset.info.data_set* values are not unique.

Return type

None

orsopy.fileio.orso.**load_orso**(*fname*)

Parameters

fname (Union[TextIO, str]) – The Orso file to load.

Return type

List[*OrsoDataset*]

Returns

OrsoDataset objects for each dataset contained within the ORT file.

1.4 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:

1.4.1 Types of Contributions

Report Bugs

Report bugs at <https://github.com/reflectivity/orsopy/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

orsopy could always use more documentation, whether as part of the official orsopy 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/reflectivity/orsopy/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 :)

1.4.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/orsopy.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 orsopy
$ cd orsopy/
$ 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, auto format the code and check that your changes pass the unit tests and confirms to PEP 8:

```
$ black -l 120 orsopy tests
$ isort -l 120 --lbt 1 orsopy tests
$ flake8 --max-line-length=120 --ignore=F401,W503,E203 --count --show-source --
  ↳statistics orsopy tests
$ pytest
```

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.

1.4.3 Pull Request Guidelines

Before you submit a pull request of your feature branch against the *main* branch of the *orsopy* repository, check that it meets these guidelines:

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`.
1. The pull request should include tests for the new functionality. Run the tests in your local machine with *pytest*. 3. The pull request should work for Python 3.5, 3.6, 3.7 and 3.8, and for PyPy. To make sure that the tests pass for all supported Python versions, you can first create a pull request of your feature branch against the *main* branch of your forked **repository**. If the Github actions pass, it is highly likely that the GitHub actions will also pass for the pull request against the *main* branch of the *orsopy* repository.

1.4.4 Tips

To run a subset of tests:

```
$ pytest tests.test_orsopy
```

1.4.5 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:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

GitHub actions will then deploy to PyPI if tests pass.

1.5 Credits

1.5.1 Contributors

- Andrew R. McCluskey <andrew.mccluskey@ess.eu>
- Andrew R. J. Nelson <andrew.nelson@ansto.gov.au>
- Artur G. Glavic <artur.glavic@psi.ch>
- Brian B. Maranville <brian.maranville@nist.gov>

1.6 History

1.6.1 1.1.0 (2023-02-20)

- Introduction of simple model language that can be used to describe sample structures. The module *orsopy.fileio.model_language* is used to implement and parse the model language. See https://www.reflectometry.org/projects/simple_model for specifications. Sample model examples can be found in the examples folder together with scripts using the orsopy module to parse and plot the data.
- Add polarization channels for x-ray experiments
- Implement ErrorValue class for optional description of errors on values within the file header.
- Update of .ort standard according to discussions with community. (E.g. rename of column attribute “dimension” to “physical_quantity”)

1.6.2 1.0.1 (2022-06-28)

- Fix bug that did allow some dictionary type values to be created in Sample.
- Update the schema files for released .ort standard.
- Sample.sample_parameters keys to be strings and values restricted to Value, ValueRange, ValueVector or ComplexValue.
- Add *as_unit* method to value classes that uses the *pint* library to convert values to supplied unit automatically.

1.6.3 1.0.0 (2022-06-10)

- ORSO general assembly has voted to release the first version of orsopy together with the text representation of the text file (.ort) specification. See https://www.reflectometry.org/workshops/workshop_2022/

1.6.4 0.1.1 (2022-06-08)

- Fix missing data files in distribution

1.6.5 0.1.0 (2022-05-19)

- Revise .ort file header specification according to ORSO discussions.
- Implement option for automatic unit conversion based on pint library
- Improve yaml export to support compact on-line layout for e.g. Value
- Add a ErrorColumn for clear separation between data and error columns and allow specification of type/distribution of error with conversion factors to get standard deviation (sigma)
- Add a ComplexValue class
- Fix some type conversions where e.g. lists have been converted to str

1.6.6 0.0.5 (2022-02-04)

- Merge the slddb package into orsopy for simple query of the database. SLD db will transition to orsopy for its backend.

1.6.7 0.0.4 (2022-01-19)

- Fix a bug preventing usage of fileio on python $\geq 3.10.1$ due to changes in dataclasses internal API
- Replace the metaclass implementation by a decorator behaving similar to dataclass
- Add meeting minutes documenting ORSO decisions
- Define documentation how to auto-format code and execute on source
- More documentation improvements

1.6.8 0.0.3 (2021-11-14)

- Implement user_data from custom keyword arguments
- Improvements to documentation
- Backport to python 3.6 and 3.7
- Allow user defined spaces between multiple datasets

1.6.9 0.0.2 (2021-10-08)

- Integration of PyPI with Github build system

1.6.10 0.0.1 (2021-10-08)

- First release on PyPI as alpha version.

1.7 Documents

In the interest of transparency, here we will host minutes of developer meetings. This should serve as an achieve of **why** particular decisions where made and when.

1.7.1 Meeting minutes

- 2021-11-30

1.8 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

PYTHON MODULE INDEX

O

`orsopy.fileio.base`, [4](#)
`orsopy.fileio.data_source`, [8](#)
`orsopy.fileio.orso`, [13](#)
`orsopy.fileio.reduction`, [12](#)

A

`additional_files` (*orsopy.fileio.data_source.Measurement* attribute), 11
`affiliation` (*orsopy.fileio.base.Person* attribute), 7
`as_unit()` (*orsopy.fileio.base.ComplexValue* method), 6
`as_unit()` (*orsopy.fileio.base.Value* method), 6
`as_unit()` (*orsopy.fileio.base.ValueRange* method), 6
`as_unit()` (*orsopy.fileio.base.ValueVector* method), 7
`asdict()` (*orsopy.fileio.base.Header* static method), 4

B

`binary` (*orsopy.fileio.reduction.Reduction* attribute), 13

C

`call` (*orsopy.fileio.reduction.Reduction* attribute), 13
`category` (*orsopy.fileio.data_source.Sample* attribute), 9
`Column` (class in *orsopy.fileio.base*), 7
`column_header()` (*orsopy.fileio.orso.Orso* method), 13
`columns` (*orsopy.fileio.orso.Orso* attribute), 13
`comment` (*orsopy.fileio.base.Column* attribute), 7
`comment` (*orsopy.fileio.base.ComplexValue* attribute), 6
`comment` (*orsopy.fileio.base.ErrorColumn* attribute), 8
`comment` (*orsopy.fileio.base.ErrorValue* attribute), 5
`comment` (*orsopy.fileio.base.File* attribute), 8
`comment` (*orsopy.fileio.base.Person* attribute), 7
`comment` (*orsopy.fileio.base.Value* attribute), 6
`comment` (*orsopy.fileio.base.ValueRange* attribute), 6
`comment` (*orsopy.fileio.base.ValueVector* attribute), 7
`comment` (*orsopy.fileio.data_source.DataSource* attribute), 12
`comment` (*orsopy.fileio.data_source.Experiment* attribute), 9
`comment` (*orsopy.fileio.data_source.InstrumentSettings* attribute), 11
`comment` (*orsopy.fileio.data_source.Measurement* attribute), 11
`comment` (*orsopy.fileio.data_source.Sample* attribute), 10
`comment` (*orsopy.fileio.orso.Orso* attribute), 14
`comment` (*orsopy.fileio.reduction.Reduction* attribute), 13
`comment` (*orsopy.fileio.reduction.Software* attribute), 12
`ComplexValue` (class in *orsopy.fileio.base*), 6

`composition` (*orsopy.fileio.data_source.Sample* attribute), 9
`computer` (*orsopy.fileio.reduction.Reduction* attribute), 13
`configuration` (*orsopy.fileio.data_source.InstrumentSettings* attribute), 11
`contact` (*orsopy.fileio.base.Person* attribute), 7
`corrections` (*orsopy.fileio.reduction.Reduction* attribute), 13
`creator` (*orsopy.fileio.reduction.Reduction* attribute), 13

D

`data` (*orsopy.fileio.orso.OrsoDataset* attribute), 14
`data_files` (*orsopy.fileio.data_source.Measurement* attribute), 11
`data_set` (*orsopy.fileio.orso.Orso* attribute), 13
`data_source` (*orsopy.fileio.orso.Orso* attribute), 13
`DataSource` (class in *orsopy.fileio.data_source*), 11
`description` (*orsopy.fileio.data_source.Sample* attribute), 9
`diff_header()` (*orsopy.fileio.orso.OrsoDataset* method), 14
`distribution` (*orsopy.fileio.base.ErrorColumn* attribute), 8
`distribution` (*orsopy.fileio.base.ErrorValue* attribute), 5
`doi` (*orsopy.fileio.data_source.Experiment* attribute), 9

E

`empty()` (*orsopy.fileio.base.Header* class method), 4
`empty()` (*orsopy.fileio.orso.Orso* class method), 13
`environment` (*orsopy.fileio.data_source.Sample* attribute), 9
`error` (*orsopy.fileio.base.ComplexValue* attribute), 6
`error` (*orsopy.fileio.base.Value* attribute), 5
`error` (*orsopy.fileio.base.ValueVector* attribute), 7
`error_of` (*orsopy.fileio.base.ErrorColumn* attribute), 7
`error_type` (*orsopy.fileio.base.ErrorColumn* attribute), 7
`error_type` (*orsopy.fileio.base.ErrorValue* attribute), 5
`error_value` (*orsopy.fileio.base.ErrorValue* attribute), 5
`ErrorColumn` (class in *orsopy.fileio.base*), 7

`ErrorValue` (class in `orsopy.fileio.base`), 5
`Experiment` (class in `orsopy.fileio.data_source`), 8
`experiment` (`orsopy.fileio.data_source.DataSource` attribute), 12

F

`facility` (`orsopy.fileio.data_source.Experiment` attribute), 9
`File` (class in `orsopy.fileio.base`), 8
`file` (`orsopy.fileio.base.File` attribute), 8
`from_difference()` (`orsopy.fileio.orso.Orso` method), 14

H

`Header` (class in `orsopy.fileio.base`), 4
`header()` (`orsopy.fileio.orso.OrsoDataset` method), 14

I

`imag` (`orsopy.fileio.base.ComplexValue` attribute), 6
`incident_angle` (`orsopy.fileio.data_source.InstrumentSettings` attribute), 11
`info` (`orsopy.fileio.orso.OrsoDataset` attribute), 14
`instrument` (`orsopy.fileio.data_source.Experiment` attribute), 9
`instrument_settings` (`orsopy.fileio.data_source.Measurement` attribute), 11
`InstrumentSettings` (class in `orsopy.fileio.data_source`), 10

L

`left` (`orsopy.fileio.data_source.Polarization` attribute), 10
`load_orso()` (in module `orsopy.fileio.orso`), 15

M

`magnitude` (`orsopy.fileio.base.Value` attribute), 5
`max` (`orsopy.fileio.base.ValueRange` attribute), 6
`Measurement` (class in `orsopy.fileio.data_source`), 11
`measurement` (`orsopy.fileio.data_source.DataSource` attribute), 12
`min` (`orsopy.fileio.base.ValueRange` attribute), 6
`mm` (`orsopy.fileio.data_source.Polarization` attribute), 10
`mo` (`orsopy.fileio.data_source.Polarization` attribute), 10
`model` (`orsopy.fileio.data_source.Sample` attribute), 9
`module`
 `orsopy.fileio.base`, 4
 `orsopy.fileio.data_source`, 8
 `orsopy.fileio.orso`, 13
 `orsopy.fileio.reduction`, 12
`mp` (`orsopy.fileio.data_source.Polarization` attribute), 10

N

`name` (`orsopy.fileio.base.Column` attribute), 7

`name` (`orsopy.fileio.base.ErrorColumn` property), 8
`name` (`orsopy.fileio.base.Person` attribute), 7
`name` (`orsopy.fileio.data_source.Sample` attribute), 9
`name` (`orsopy.fileio.reduction.Software` attribute), 12
`NotOrsoCompatibleFileError`, 8

O

`om` (`orsopy.fileio.data_source.Polarization` attribute), 10
`op` (`orsopy.fileio.data_source.Polarization` attribute), 10
`Orso` (class in `orsopy.fileio.orso`), 13
`orsodataclass()` (in module `orsopy.fileio.base`), 4
`OrsoDataset` (class in `orsopy.fileio.orso`), 14
`OrsoDumper` (class in `orsopy.fileio.base`), 5
`orsopy.fileio.base`
 module, 4
`orsopy.fileio.data_source`
 module, 8
`orsopy.fileio.orso`
 module, 13
`orsopy.fileio.reduction`
 module, 12
`ORSOResolveError`, 4
`owner` (`orsopy.fileio.data_source.DataSource` attribute), 12

P

`Person` (class in `orsopy.fileio.base`), 7
`physical_quantity` (`orsopy.fileio.base.Column` attribute), 7
`pi` (`orsopy.fileio.data_source.Polarization` attribute), 10
`pi_pi` (`orsopy.fileio.data_source.Polarization` attribute), 10
`pi_sigma` (`orsopy.fileio.data_source.Polarization` attribute), 10
`platform` (`orsopy.fileio.reduction.Software` attribute), 12
`pm` (`orsopy.fileio.data_source.Polarization` attribute), 10
`po` (`orsopy.fileio.data_source.Polarization` attribute), 10
`Polarization` (class in `orsopy.fileio.data_source`), 10
`polarization` (`orsopy.fileio.data_source.InstrumentSettings` attribute), 11
`pp` (`orsopy.fileio.data_source.Polarization` attribute), 10
`probe` (`orsopy.fileio.data_source.Experiment` attribute), 9
`proposalID` (`orsopy.fileio.data_source.Experiment` attribute), 9

R

`real` (`orsopy.fileio.base.ComplexValue` attribute), 6
`Reduction` (class in `orsopy.fileio.reduction`), 12
`reduction` (`orsopy.fileio.orso.Orso` attribute), 13
`represent_data()` (`orsopy.fileio.base.OrsoDumper` method), 5
`right` (`orsopy.fileio.data_source.Polarization` attribute), 10

S

[Sample](#) (class in [orsopy.fileio.data_source](#)), 9
[sample](#) ([orsopy.fileio.data_source.DataSource](#) attribute), 12
[sample_parameters](#) ([orsopy.fileio.data_source.Sample](#) attribute), 9
[save\(\)](#) ([orsopy.fileio.orso.OrsoDataset](#) method), 15
[save_orso\(\)](#) (in module [orsopy.fileio.orso](#)), 15
[scheme](#) ([orsopy.fileio.data_source.Measurement](#) attribute), 11
[script](#) ([orsopy.fileio.reduction.Reduction](#) attribute), 13
[sigma](#) ([orsopy.fileio.base.ErrorValue](#) property), 5
[sigma](#) ([orsopy.fileio.data_source.Polarization](#) attribute), 10
[sigma_pi](#) ([orsopy.fileio.data_source.Polarization](#) attribute), 10
[sigma_sigma](#) ([orsopy.fileio.data_source.Polarization](#) attribute), 10
[size](#) ([orsopy.fileio.data_source.Sample](#) attribute), 9
[Software](#) (class in [orsopy.fileio.reduction](#)), 12
[software](#) ([orsopy.fileio.reduction.Reduction](#) attribute), 12
[start_date](#) ([orsopy.fileio.data_source.Experiment](#) attribute), 9

T

[timestamp](#) ([orsopy.fileio.base.File](#) attribute), 8
[timestamp](#) ([orsopy.fileio.reduction.Reduction](#) attribute), 13
[title](#) ([orsopy.fileio.data_source.Experiment](#) attribute), 9
[to_dict\(\)](#) ([orsopy.fileio.base.Header](#) method), 4
[to_dict\(\)](#) ([orsopy.fileio.orso.Orso](#) method), 14
[to_difference\(\)](#) ([orsopy.fileio.orso.Orso](#) method), 14
[to_sigma](#) ([orsopy.fileio.base.ErrorColumn](#) property), 8
[to_yaml\(\)](#) ([orsopy.fileio.base.Header](#) method), 5

U

[unit](#) ([orsopy.fileio.base.Column](#) attribute), 7
[unit](#) ([orsopy.fileio.base.ComplexValue](#) attribute), 6
[unit](#) ([orsopy.fileio.base.Value](#) attribute), 5
[unit](#) ([orsopy.fileio.base.ValueRange](#) attribute), 6
[unit](#) ([orsopy.fileio.base.ValueVector](#) attribute), 7
[unpolarized](#) ([orsopy.fileio.data_source.Polarization](#) attribute), 10
[user_data](#) ([orsopy.fileio.base.Header](#) property), 4
[user_data](#) ([orsopy.fileio.orso.Orso](#) property), 13

V

[Value](#) (class in [orsopy.fileio.base](#)), 5
[value_is](#) ([orsopy.fileio.base.ErrorColumn](#) attribute), 7
[value_is](#) ([orsopy.fileio.base.ErrorValue](#) attribute), 5
[ValueRange](#) (class in [orsopy.fileio.base](#)), 6
[ValueVector](#) (class in [orsopy.fileio.base](#)), 6

[version](#) ([orsopy.fileio.reduction.Software](#) attribute), 12

W

[wavelength](#) ([orsopy.fileio.data_source.InstrumentSettings](#) attribute), 11

X

[x](#) ([orsopy.fileio.base.ValueVector](#) attribute), 7

Y

[y](#) ([orsopy.fileio.base.ValueVector](#) attribute), 7
[yaml_representer\(\)](#) ([orsopy.fileio.base.Column](#) method), 7
[yaml_representer\(\)](#) ([orsopy.fileio.base.ComplexValue](#) method), 6
[yaml_representer\(\)](#) ([orsopy.fileio.base.ErrorColumn](#) method), 8
[yaml_representer\(\)](#) ([orsopy.fileio.base.ErrorValue](#) method), 5
[yaml_representer\(\)](#) ([orsopy.fileio.base.Header](#) method), 5
[yaml_representer\(\)](#) ([orsopy.fileio.base.Value](#) method), 6
[yaml_representer\(\)](#) ([orsopy.fileio.base.ValueRange](#) method), 6
[yaml_representer\(\)](#) ([orsopy.fileio.base.ValueVector](#) method), 7
[yaml_representer\(\)](#) ([orsopy.fileio.data_source.Polarization](#) method), 10
[yaml_representer\(\)](#) ([orsopy.fileio.reduction.Software](#) method), 12
[yaml_representer_compact\(\)](#) ([orsopy.fileio.base.Header](#) method), 5

Z

[z](#) ([orsopy.fileio.base.ValueVector](#) attribute), 7