

---

# Sausage Documentation

*Release 0.1.0*

**Bas Hoonhout**

**Nov 15, 2017**



---

## Contents:

---

|          |                               |
|----------|-------------------------------|
| <b>1</b> | <b>Command Line Interface</b> |
|----------|-------------------------------|

|          |
|----------|
| <b>3</b> |
|----------|



Sausage facilitates the conversion of input and output files for geophysical models. Geophysical models typically describe a spatiotemporal varying model state (output) that is determined by a set of initial and boundary conditions (input). Conversion between model input and output from and to different models and data sources allows you to:

- Quickly setup model schematization from measurement data
- Quickly compare model output with measurement data (e.g. calibration and validation)
- Offline coupling of models (e.g. nesting)

Sausage contains a set of model-specific classes that do the heavy lifting. We distinguish three types of classes:

- Readers
- Writers
- Compilers

Readers can read model-specific input and output files into an `xarray.Dataset` object. Readers are clustered by model (format), but one format can contain multiple readers (categories). For example, a separate reader for water level and wave height files. Readers at least implement a `read` method. Readers have a `dataset` property that can be fed to a Writer.

Writers can write model-specific input and output files from an `xarray.Dataset` object. Writers are also clustered by model (format), but one format can again contain multiple writers (categories). Writers register written files based on their format and category in order to be used by a Compiler. Writers at least implement a `write` method.

Compilers use written (and registered) model-specific input and output files to compile a fully described model schematization. Typically, a Compiler writes a model-specific configuration or parameters file that links to the individual initial and boundary condition files previously written. Optionally, a compiler can also execute the model on the newly written model schematization. Compilers at least implement a `compile` and `run` method.



---

## Command Line Interface

---

Sausage can be invoked using two commands:

- `sausage-convert`
- `sausage-compile`

All commands act on the current working directory, which will contain the generated model schematization. Run the command with the `-h` option to display usage instructions.

*sausage-convert* handles the conversion from a single file of arbitrary format to another file of arbitrary format. It first invokes a Reader and subsequently a Writer object. For example:

```
sausage-convert A1.SEQ aukepc:WHM01 xbeach:waterlevel
sausage-convert SA1.SEQ aukepc:WHM01 xbeach:wavespectrum
```

*sausage-compile* generates a fully described model schematization in a given format, using the generated files registered in the current working directory. For example:

```
sausage-compile xbeach
```

Optionally, the model can be executed on the newly created model schematization as well:

```
sausage-compile xbeach --run
```