



EUROPEAN  
SPALLATION  
SOURCE



# Taking the Pain out of Generating Complete and Compliant NeXus Files for Dynamic Setups

Kenan Murić, Tobias Richter  
European Spallation Source

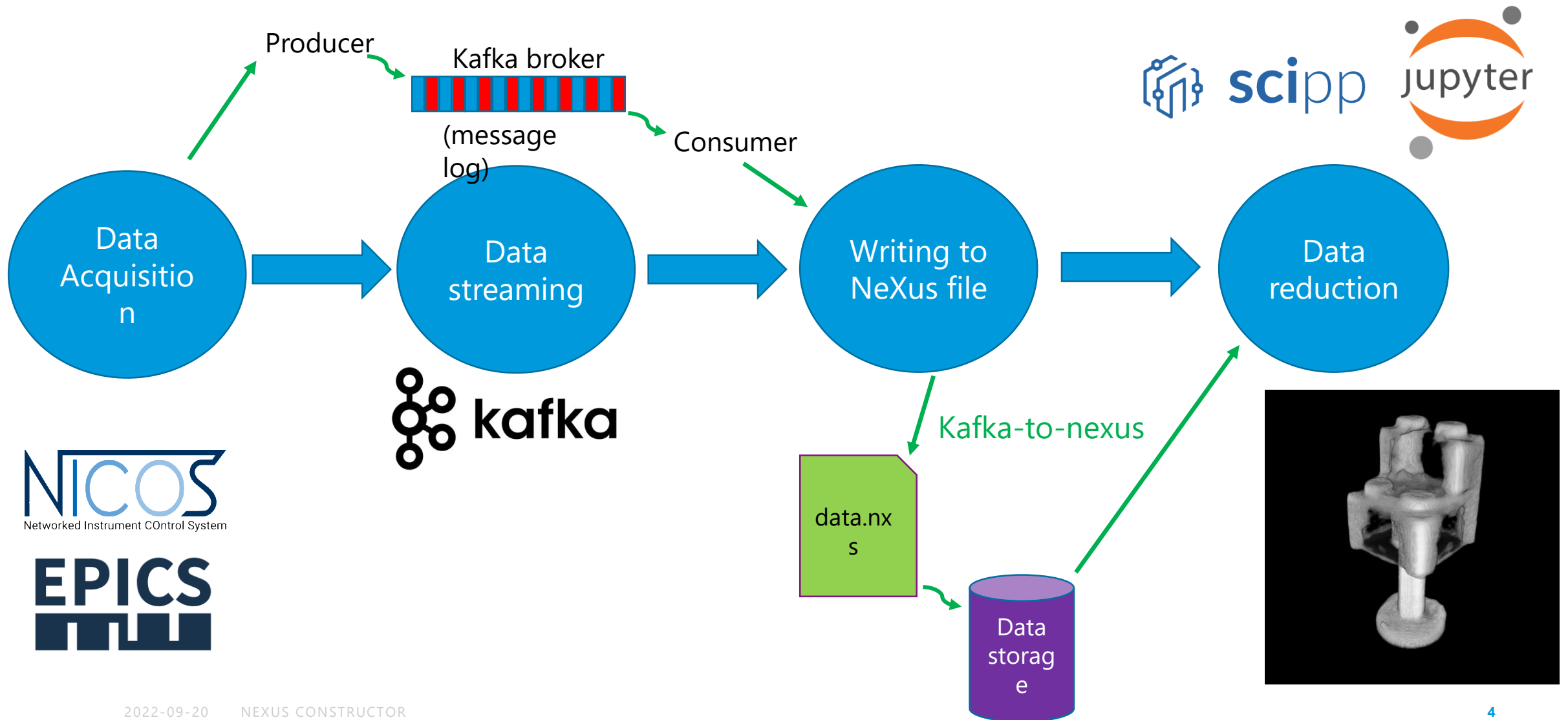
2022-09-20

# Agenda

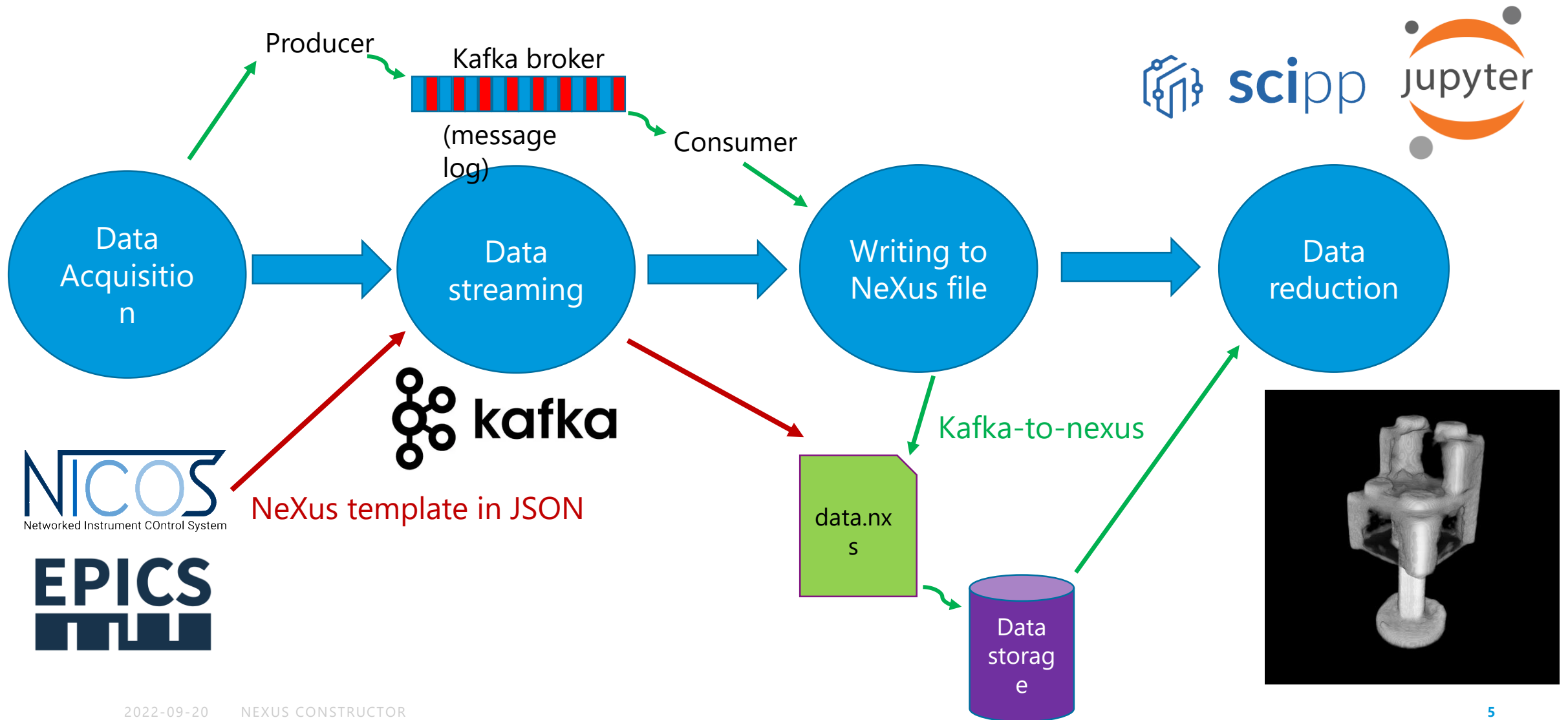


- 1 Motivation
- 2 NeXus Constructor
- 3 Creating a NeXus group
- 4 Components and component geometries
- 5 File writer modules
- 6 Summary

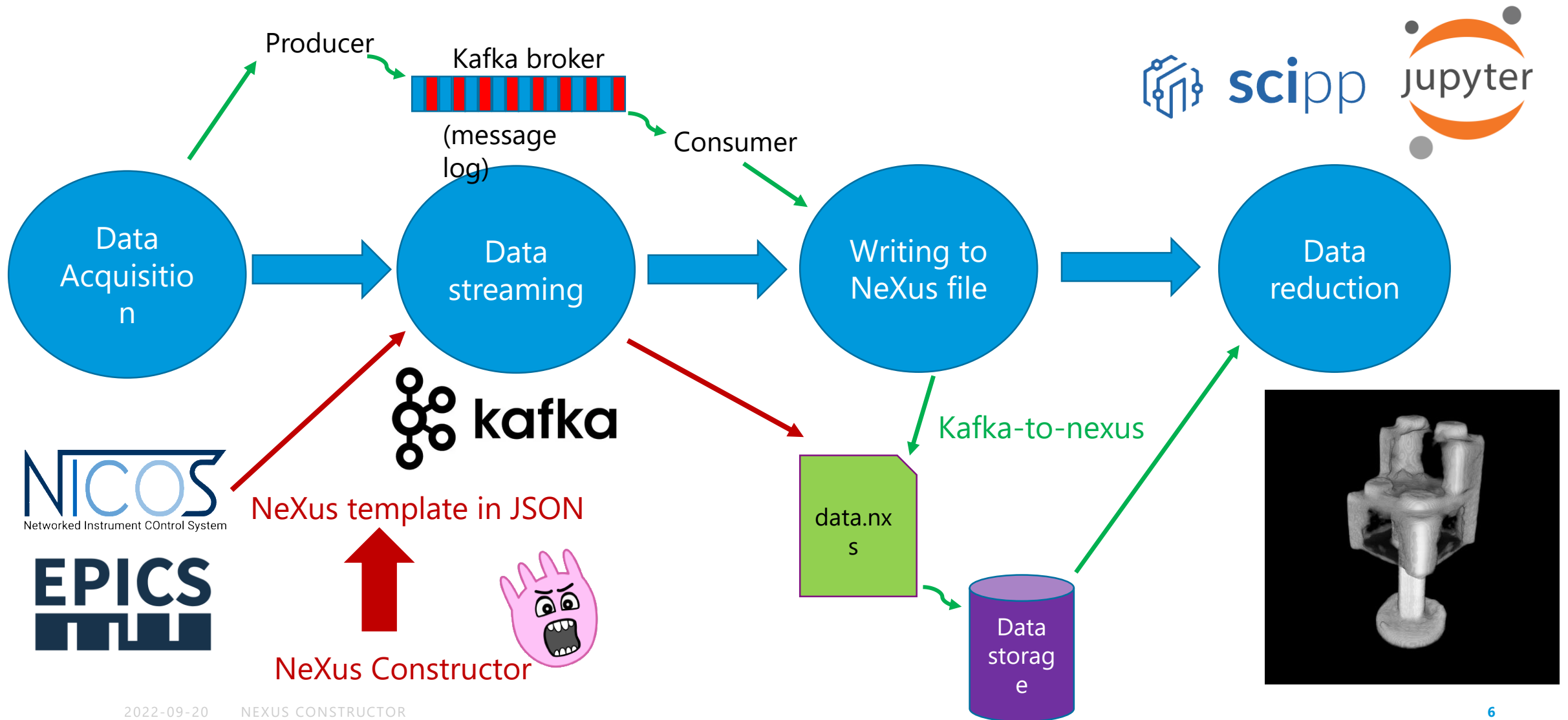
# Motivation



# Motivation



# Motivation



# Motivation



The **NeXus** file will contain:

- Groups with corresponding NeXus classes.
- Static data (e.g. experiment title, users etc.).
- Dynamic data.
- Alarm information from instrument devices.
- Transformations (dynamic and static) that define instrument component locations and rotations.

*At ESS, our data structures will follow the NeXus standard specification as much as possible!*

| Name                   | Description                                | Type              | Shape           |
|------------------------|--|-------------------|-----------------|
| 964281_00003931.hdf    |  |                   |                 |
| entry                  | ["Unassigned experiments"]                 | NXentry           |                 |
| data                   |  | NXdata            |                 |
| definition             | ["NXtomo"]                                 | string            | 1               |
| experiment_identifier  | ["964281"]                                 | string            | 1               |
| instrument             |  | NXinstrument      |                 |
| flir_camera            |  | NXdetector        |                 |
| data                   |  | NXlog             |                 |
| cue_index              | []   | uint32            | 0               |
| cue_timestamp_zero     | []   | uint64            | 0               |
| time                   | 1D data                                    | uint64            | 416             |
| value                  | 3D data                                    | float64           | 416 × 480 × 830 |
| image_key              |  | NXlog             |                 |
| alarm_severity         | 1D data                                    | string            | 7               |
| alarm_status           | 1D data                                    | string            | 7               |
| alarm_time             | 1D data                                    | uint64            | 7               |
| average_value          | [1.71429]                                  | float64           | 1               |
| connection_status      | []   | string            | 0               |
| connection_status_time | []   | uint64            | 0               |
| cue_index              | []   | uint32            | 0               |
| cue_timestamp_zero     | []   | uint64            | 0               |
| maximum_value          | [3]  | float64           | 1               |
| minimum_value          | [0]  | float64           | 1               |
| time                   | 1D data                                    | uint64            | 7               |
| value                  | 2D data                                    | int16             | 7 × 1           |
| laser_monitor          |  | NXmonitor         |                 |
| laser_source           |  | NXsource          |                 |
| light_source           |  | NXsource          |                 |
| probe                  | ["visible light"]                          | string            | 1               |
| mini_chopper           |  | NXdisk_chopper    |                 |
| sample                 | ["lego"]                                   | NXsample          |                 |
| depends_on             | ["/entry/sample/transformations/rotation"] | string            | 1               |
| name                   | ["lego"]                                   | string            | 1               |
| transformations        |  | NXtransformations |                 |
| title                  | ["Unassigned experiments"]                 | string            | 1               |

# Motivation



Configuration file the ESS file writer uses to write NeXus files

```

63     "name": "probe",
64     "values": "visible light",
65     "type": "string"
66   }
67 }
68 }
69 },
70 {
71   "name": "flir_camera",
72   "type": "group",
73   "attributes": [
74     {
75       "name": "NX_class",
76       "dtype": "string",
77       "values": "NXdetector"
78     }
79   ],
80   "children": [
81     {
82       "name": "image_key",
83       "type": "group",
84       "attributes": [
85         {
86           "name": "NX_class",
87           "dtype": "string",
88           "values": "NXlog"
89         }
90       ],
91       "children": [
92         {
93           "module": "f142",
94           "config": {
95             "source": "flir_image_key",
96             "topic": "ymir_nicos_devices",
97             "dtype": "int16"
98           }
99         }
100       ]
101     },
102     {
103       "name": "data",
104       "type": "group",
105       "attributes": [
106         {
107           "name": "NX_class",
108           "dtype": "string",
109           "values": "NXlog"
110         }
111       ],
112       "children": [
113         {
114           "module": "ADAr",
115           "config": {
116             "source": "some_source",
117             "topic": "ymir_camera",
118             "array_size": "$AREADETS"
119           }
120         }
121       ]
122     }
123   ]
124 }

```

File writing



| Name                   | Description                                | Type              | Shape           |
|------------------------|--|-------------------|-----------------|
| 964281_00003931.hdf    |  |                   |                 |
| entry                  | ["Unassigned experiments"]                 | NXentry           |                 |
| data                   |  | NXdata            |                 |
| definition             | ["NXtomo"]                                 | string            | 1               |
| experiment_identifier  | ["964281"]                                 | string            | 1               |
| instrument             |  | NXinstrument      |                 |
| flir_camera            |  | NXdetector        |                 |
| data                   |  | NXlog             |                 |
| cue_index              |  | uint32            | 0               |
| cue_timestamp_zero     |  | uint64            | 0               |
| time                   | 1D data                                    | uint64            | 416             |
| value                  | 3D data                                    | float64           | 416 x 480 x 830 |
| image_key              |  | NXlog             |                 |
| alarm_severity         | 1D data                                    | string            | 7               |
| alarm_status           | 1D data                                    | string            | 7               |
| alarm_time             | 1D data                                    | uint64            | 7               |
| average_value          | [1.71429]                                  | float64           | 1               |
| connection_status      |  | string            | 0               |
| connection_status_time |  | uint64            | 0               |
| cue_index              |  | uint32            | 0               |
| cue_timestamp_zero     |  | uint64            | 0               |
| maximum_value          | [3]  | float64           | 1               |
| minimum_value          | [0]  | float64           | 1               |
| time                   | 1D data                                    | uint64            | 7               |
| value                  | 2D data                                    | int16             | 7 x 1           |
| laser_monitor          |  | NXmonitor         |                 |
| laser_source           |  | NXsource          |                 |
| light_source           |  | NXsource          |                 |
| probe                  | ["visible light"]                          | string            | 1               |
| mini_chopper           |  | NXdisk_chopper    |                 |
| sample                 | ["lego"]                                   | NXsample          |                 |
| depends_on             | ["/entry/sample/transformations/rotation"] | string            | 1               |
| name                   | ["lego"]                                   | string            | 1               |
| transformations        |  | NXtransformations |                 |
| title                  | ["Unassigned experiments"]                 | string            | 1               |



# Motivation

## Some problems to solve...



- Generating template files is a tedious work.

### **Why?**

- Even the simplest cases result in files that are 1000s of lines.
- It is hard to verify the validity of structures with respect to the NeXus standard specification.  
<https://manual.nexusformat.org/classes/index.html>
- Verifying instrument component layouts requires writing your own plotting tools.
- People use different libraries to create template files. Scripts used are usually not well-documented.
- Sometimes manual editing of the JSON file is needed.

# NeXus Constructor



## Seamlessly building configuration files

- NeXus Constructor comes with a proper and rather simple graphical user interface.
- Developed in *Python 3.7* using *PySide2* (upgrading to *PySide6*) for graphical components.
- You are able to see both the NeXus configuration structure and a 3D rendering of NeXus groups with geometric properties (e.g. choppers, detectors, monitors and slits).
- It is easy to load an existing configuration file and edit it in the NeXus constructor.
- Definitions and documentation of NeXus base classes is integrated in the application.
- NeXus constructor suggests group/field names, does verifications of inputs, suggests units etc.
- Works on macOS, Linux and Windows.

**NeXus Constructor is open-source and can be found**

<https://github.com/ess-dmsc/nexus-creator>



# NeXus Constructor



NeXus Constructor

Group

Translation

Rotation

Link

Edit

Zoom

Delete

Proposal ID:

☐ Use placeholder

Title:

☐ Use placeholder

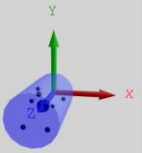
Configure users

☐ Use placeholder

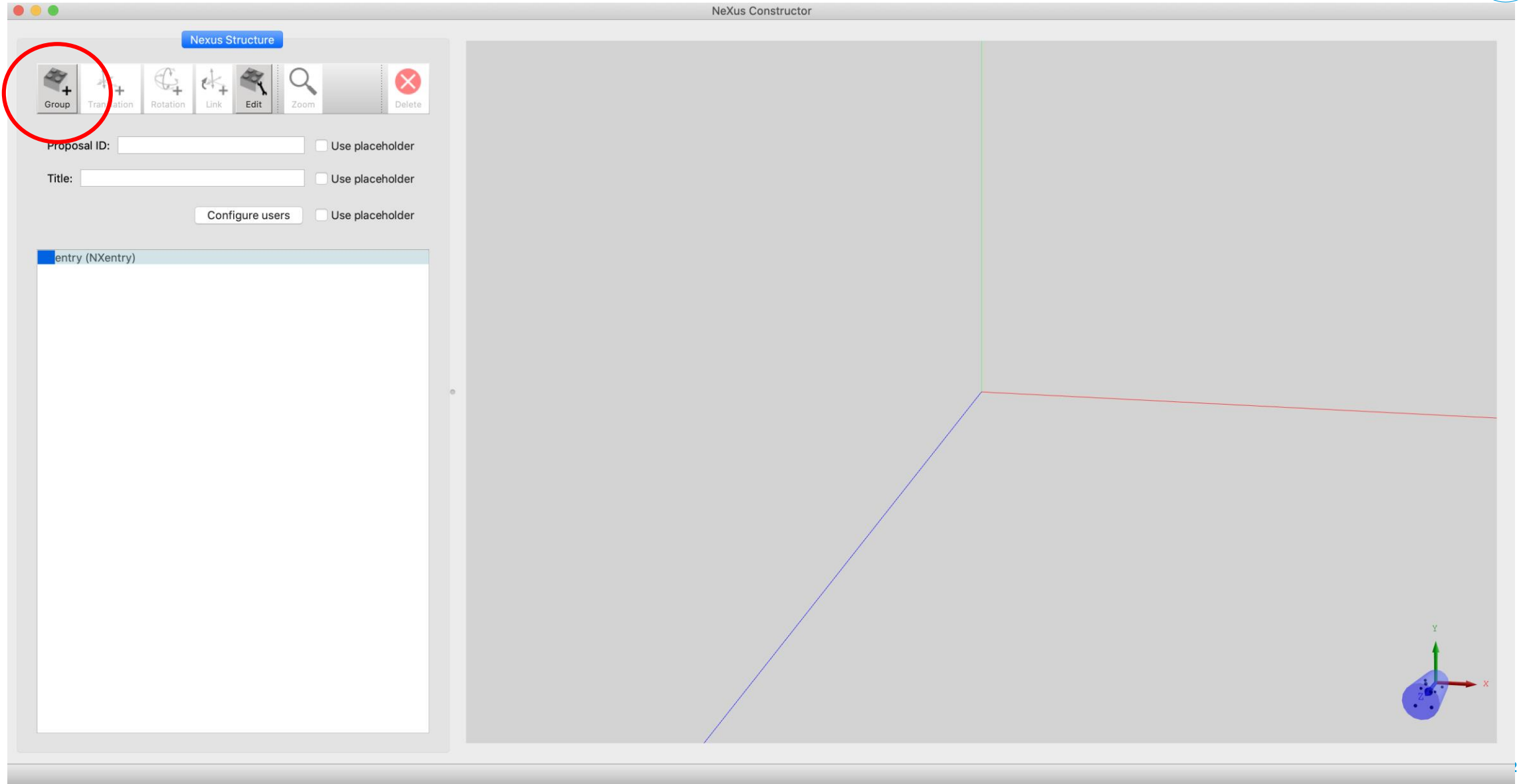
entry (NXentry)

NeXus Tree

Component 3D viewer



# Creating a NeXus group



# Creating a NeXus group

A screenshot of a web-based interface for creating a NeXus group. On the left, there is a form with fields for 'Name' (containing 'sample'), 'Description', 'Group type', and 'Shape type' (with radio buttons for 'No Shape' and 'Box'). Below these is a 'Fields' section with an 'Add field' button. On the right, a scrollable list of NeXus classes is displayed. The list includes various components like NXattenuator, NXbeam\_stop, and NXdetector, as well as groups like NXsample and NXsample\_component. The 'NXsample' class is currently selected and highlighted in blue.

- Choose the desired NeXus class from the list.
- NeXus classes in the list are split in two categories: components and groups.
- A component is something physical on the beam line, with geometric properties (dimensions, shape and location).
- Suggested NeXus classes are based on the NeXus standard specification.

# Creating a NeXus group



Name:

Description:

Group type: ☐ Use placeholder

Shape type: ☒ Auto ☐ Box ☐ Mesh ☐ Cylinder

Fields

Add field

Remove field

nexus v2020.10 documentation » 3. NeXus: Reference Documentation » previous | next

3.3. NeXus Class Definitions » 3.3.1. Base Class Definitions » 3.3.1.46. NXsample

### 3.3.1.46. NXsample

**Status:**

base class, extends [NXObject](#)

**Description:**

Any information on the sample.

This could include scanned variables that are associated with one of the data dimensions, e.g. the magnetic field, or logged data, e.g. monitored temperature vs elapsed time.

**Symbols:**

symbolic array lengths to be coordinated between various fields

**n\_comp:** number of compositions

**n\_Temp:** number of temperatures

**n\_eField:** number of values in applied electric field

**n\_mField:** number of values in applied magnetic field

**n\_pField:** number of values in applied pressure field

**n\_sField:** number of values in applied stress field

**Groups cited:**

[NXbeam](#), [NXdata](#), [NXenvironment](#), [NXgeometry](#), [NXlog](#), [NXpositioner](#), [NXsample\\_component](#), [NXtransformations](#)

**Structure:**

**@default:** (optional) [NX\\_CHAR](#)

Declares which child group contains a path leading to a [NXdata](#) group.

Table of Contents

3.3.1.46. NXsample

- Hypertext Anchors

Previous topic

3.3.1.45. NXroot

Next topic

3.3.1.47. NXsample\_component

This Page

Show Source

Quick search

Google search

☐ global ☒ NeXus manual

Add group

Cancel



# Creating a NeXus group

Creation of a sample using the NXsample class

Name:

Description:

Group type: ☐ Use placeholder NXsample

Shape type:

☒ Auto ☐ Box ☐ Mesh ☐ Cylinder

Fields

Add field

Remove field

d

changer\_position  
chemical\_formula  
component  
concentration

Scalar dataset

value

double

units

Attrs

# Creating a NeXus group

## Creation of a sample using the NXsample class

Name:

Description:

Group type: ☐ Use placeholder

Shape type:

☐ Auto ☒ Box ☐ Mesh ☐ Cylinder

Shape options:

Units

Length:

Width:

Height:

Fields

| Field name       | Dataset type   | Value | Units  | Attrs |
|------------------|----------------|-------|--------|-------|
| chemical_formula | Scalar dataset | SiO2  | string |       |
| Kafka stream     |                |       |        |       |

nexus v2020.10 documentation » 3. NeXus: Reference Documentation » previous | next |

3.3. NeXus Class Definitions » 3.3.1. Base Class Definitions » 3.3.1.46. NXsample

### 3.3.1.46. NXsample

#### Status:

base class, extends [NXObject](#)

#### Description:

Any information on the sample.

This could include scanned variables that are associated with one of the data dimensions, e.g. the magnetic field, or logged data, e.g. monitored temperature vs elapsed time.

#### Symbols:

symbolic array lengths to be coordinated between various fields

**n\_comp**: number of compositions

**n\_Temp**: number of temperatures

**n\_eField**: number of values in applied electric field

**n\_mField**: number of values in applied magnetic field

**n\_pField**: number of values in applied pressure field

**n\_sField**: number of values in applied stress field

#### Groups cited:

[NXbeam](#), [NXdata](#), [NXenvironment](#), [NXgeometry](#), [NXlog](#), [NXpositioner](#), [NXsample\\_component](#), [NXtransformations](#)

#### Structure:

@default: (optional) [NX\\_CHAR](#)

Declares which child group contains a path leading to a [NXdata](#) group.

#### Table of Contents

3.3.1.46. NXsample

- Hypertext Anchors

#### Previous topic

3.3.1.45. NXroot

#### Next topic

3.3.1.47. NXsample\_component

#### This Page

Show Source

#### Quick search

#### Google search

☐ global ☒ NeXus manual



# Creating a NeXus group

## Creation of a sample using the NXsample class

File View

Nexus Structure

Group Translation Rotation Link Edit Zoom Delete

Proposal ID:  ☐ Use placeholder

Title:  ☐ Use placeholder

☐ Use placeholder

entry (NXentry)

▼ sample (NXsample)

dataset

...

f142

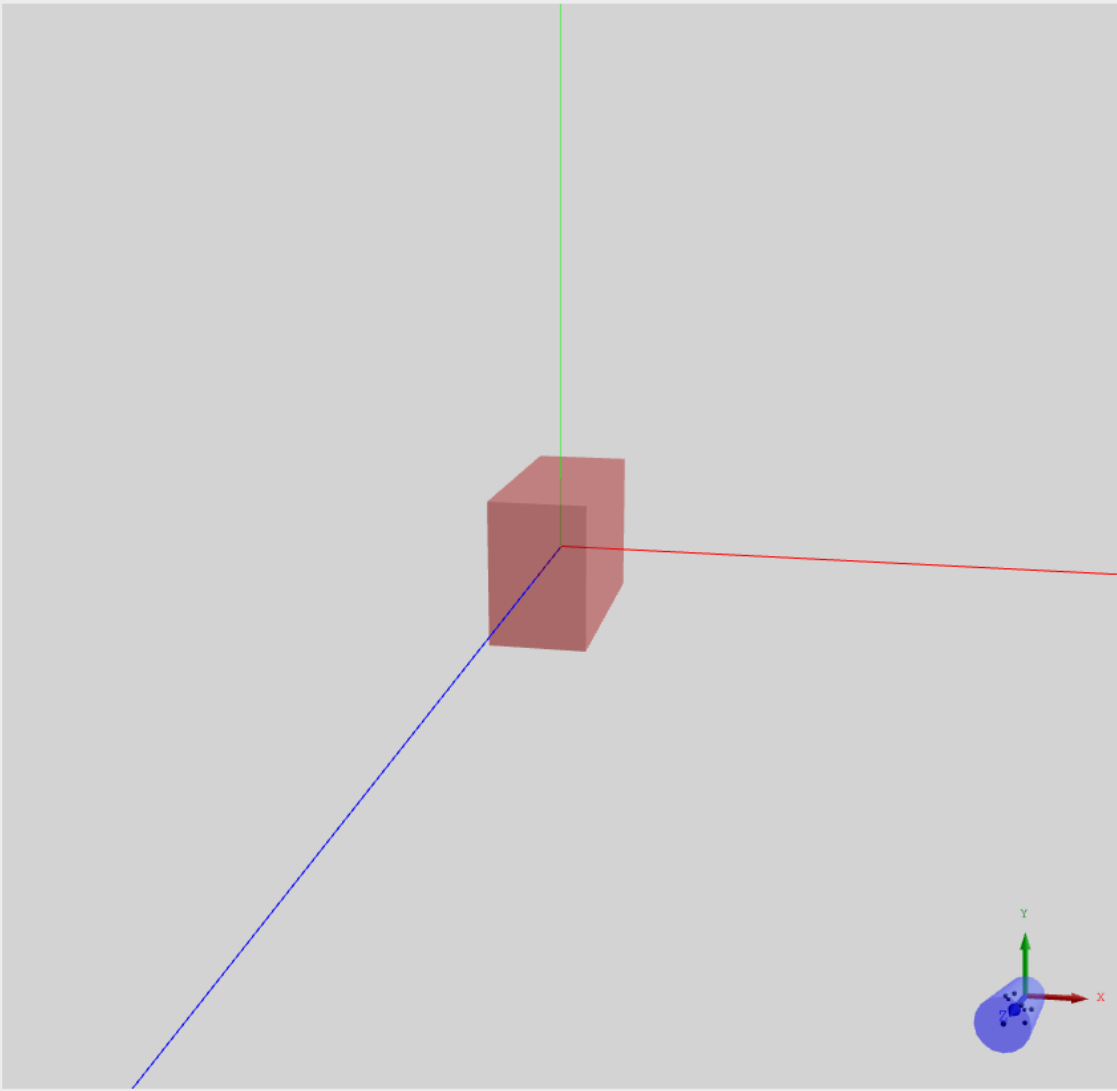
▼ geometry (NXgeometry)

▼ shape (NXshape)

dataset

dataset

...



# Components

## Aspects of components in NeXus Constructor

1. A component has dimensions, shape and location.
2. Shape type is at the moment limited to following set of shapes:  
**Auto**, **box**, **mesh** and **cylinder**.
3. There are components with customized shapes, such as **NXslit**, **NXdetector** and **NXdisk\_chopper**.
4. More shapes and customized components could be added in the future.

Note: the sample is a component!

Shape type: ☐ No Shape ☒ Box ☐ Mesh ☐ Cylinder

Shape options:

Units: m

Length: 0,00  
Width: 0,00  
Height: 0,00

Shape type: ☐ No Shape ☐ Box ☒ Mesh ☐ Cylinder

Shape options:

Units: m

CAD file: /Users/kenanmuric/Desktop/OFF\_files/cube\_color.off Browse...

Fields: Add field Remove field

Shape type: ☐ No Shape ☐ Box ☐ Mesh ☒ Cylinder

Shape options:

Units: m

Cylinder options:

Height: 0,00 Radius: 0,00  
X: 0,00 Y: 0,00 Z: 1,00  
Cylinder Count: 1

# Components

## Defining sample (or other component) position and rotation

The NeXus standard uses the NeXus class **NXtransformations** to define translations and rotations for a given component. Link is used to link transformations to other components.

Nexus Structure

Group + Translation + Rotation + Link + Edit + Zoom + Delete

Proposal ID:  ☐ Use placeholder

Title:  ☐ Use placeholder

☐ Use placeholder

▼ entry (NXentry)

▼ sample (NXsample)

▼ transformations (NXtransformations)

rotation

Name:

Rotation Axis

x:  y:  z:

Angle (°)

Scalar dataset:  double degrees Attrs

Value to use in 3D view:

translation

Name:

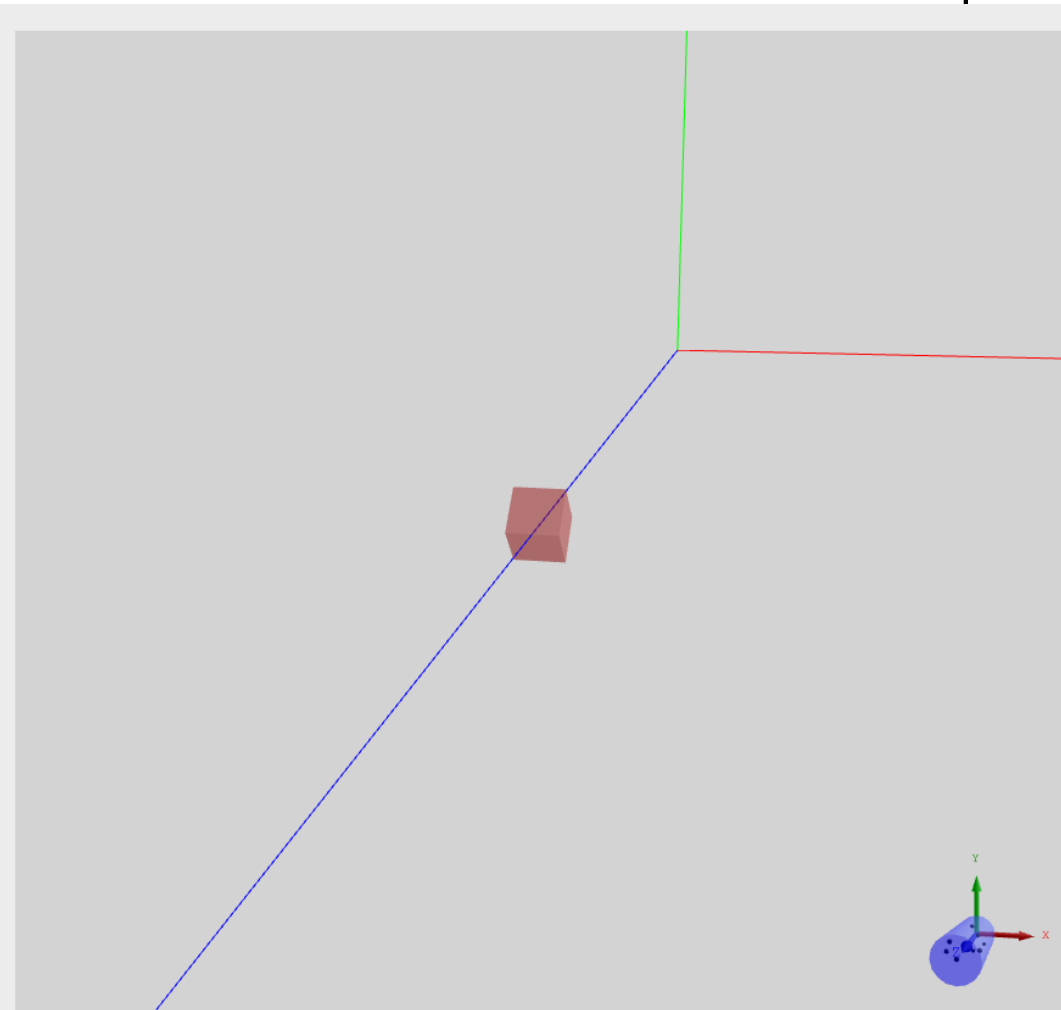
Direction

x:  y:  z:

Distance (m)

Scalar dataset:  double m Attrs

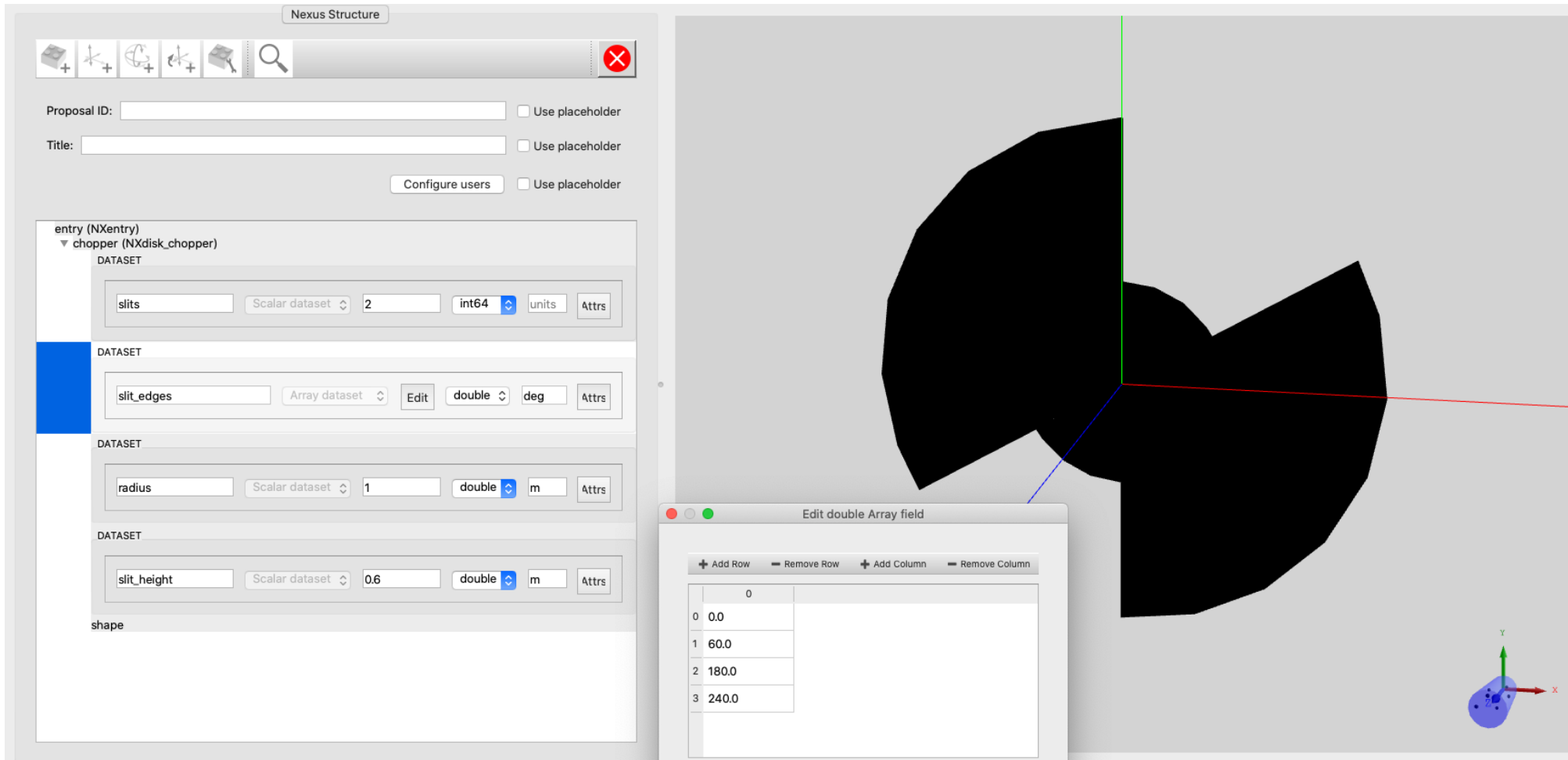
Value to use in 3D view:



# Components

## Defining a chopper using NXdisk\_chopper

You need to provide radius, slits, slit\_edges and slit\_height to get a proper disk chopper.



The screenshot displays the Nexus Structure interface for defining a chopper using the `NXdisk_chopper` component. The interface includes a top toolbar with icons for adding, deleting, and searching components. Below the toolbar, there are input fields for 'Proposal ID' and 'Title', each with a 'Use placeholder' checkbox. A 'Configure users' button is also present.

The main configuration area is titled 'entry (NXentry)' and contains a sub-entry 'chopper (NXdisk\_chopper)'. This sub-entry is expanded, showing four dataset fields:

- slits**: A scalar dataset with a value of 2, type 'int64', and units 'units'.
- slit\_edges**: An array dataset with a value of 0, type 'double', and units 'deg'. This field is highlighted with a blue bar.
- radius**: A scalar dataset with a value of 1, type 'double', and units 'm'.
- slit\_height**: A scalar dataset with a value of 0.6, type 'double', and units 'm'.

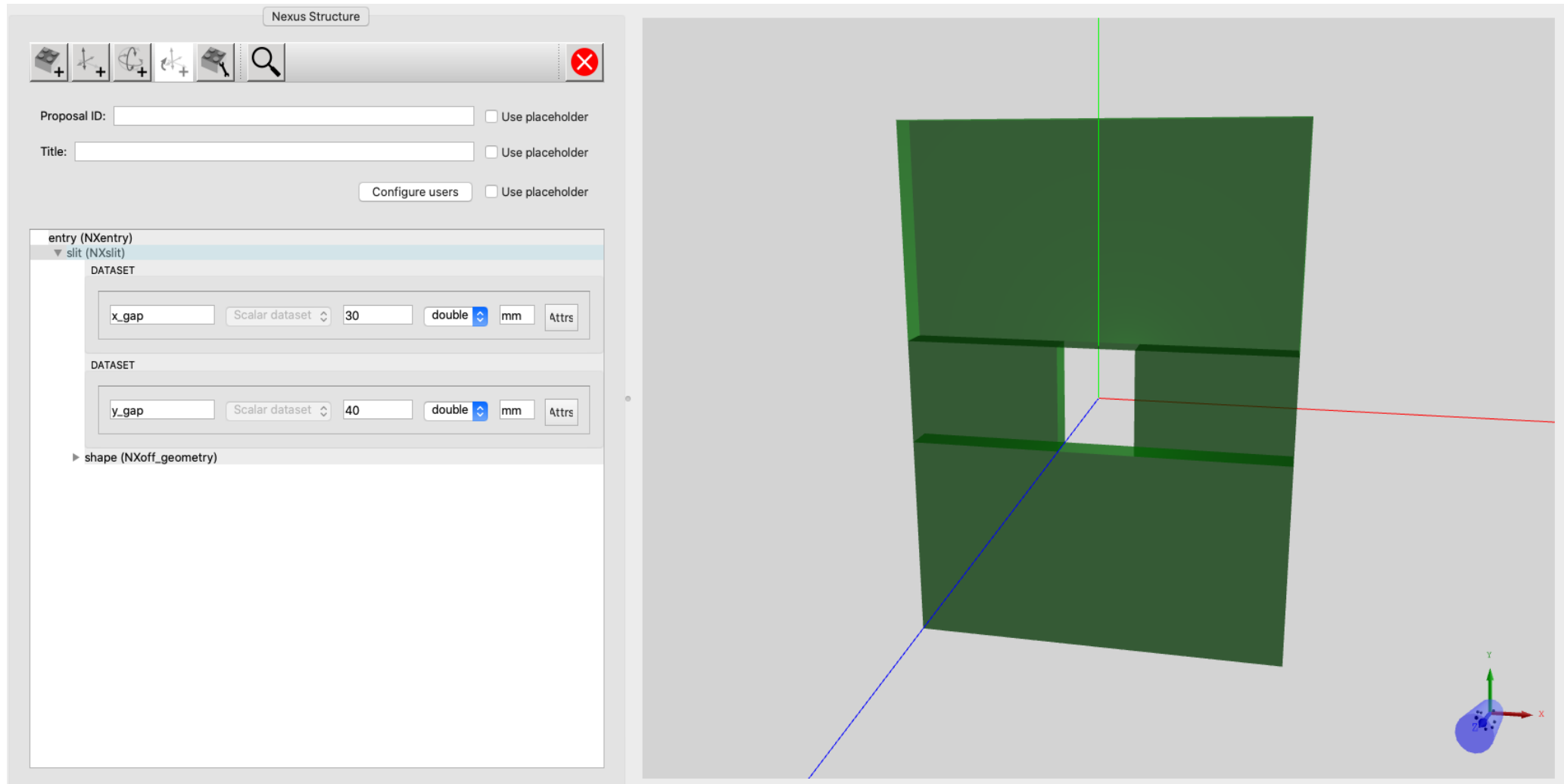
Below the dataset fields, there is a 'shape' field. To the right of the configuration area, a 3D visualization shows a black disk chopper with a green vertical line and a red horizontal line. A small inset window titled 'Edit double Array field' shows a table with values for the 'slit\_edges' dataset:

|   | 0     |
|---|-------|
| 0 | 0.0   |
| 1 | 60.0  |
| 2 | 180.0 |
| 3 | 240.0 |

# Components

## Defining a slit using NXslit

Provide x\_gap and y\_gap to render a proper slit.



The screenshot displays the Nexus Structure software interface. On the left, the 'entry (NXentry)' tree shows a selected 'slit (NXslit)' component. The configuration panel for this component includes two 'DATASET' sections. The first section is for 'x\_gap', with a value of 30, unit 'mm', and a 'double' data type. The second section is for 'y\_gap', with a value of 40, unit 'mm', and a 'double' data type. Below these, there is a 'shape (NXoff\_geometry)' section. On the right, a 3D visualization shows a dark green, rectangular slit component within a coordinate system. The slit is defined by a central opening. A red line and a blue line intersect at the center of the slit, indicating the coordinate system. A small 3D coordinate system icon is visible in the bottom right corner of the visualization area.

# Components

## Defining a detector using NXdetector and cylinder shapes

Name:

Description:

Group type:

Shape type:

☐ Auto ☐ Box ☐ Mesh ☒ Cylinder

Shape options:

Units

Cylinder options

Height:  Radius:

X:  Y:  Z:

Cylinder Count:

Pixel options

Pixel layout:

☒ Repeated Single Pixel Shape ☐ Entire Shape ☐ No Pixels

Pixel grid:

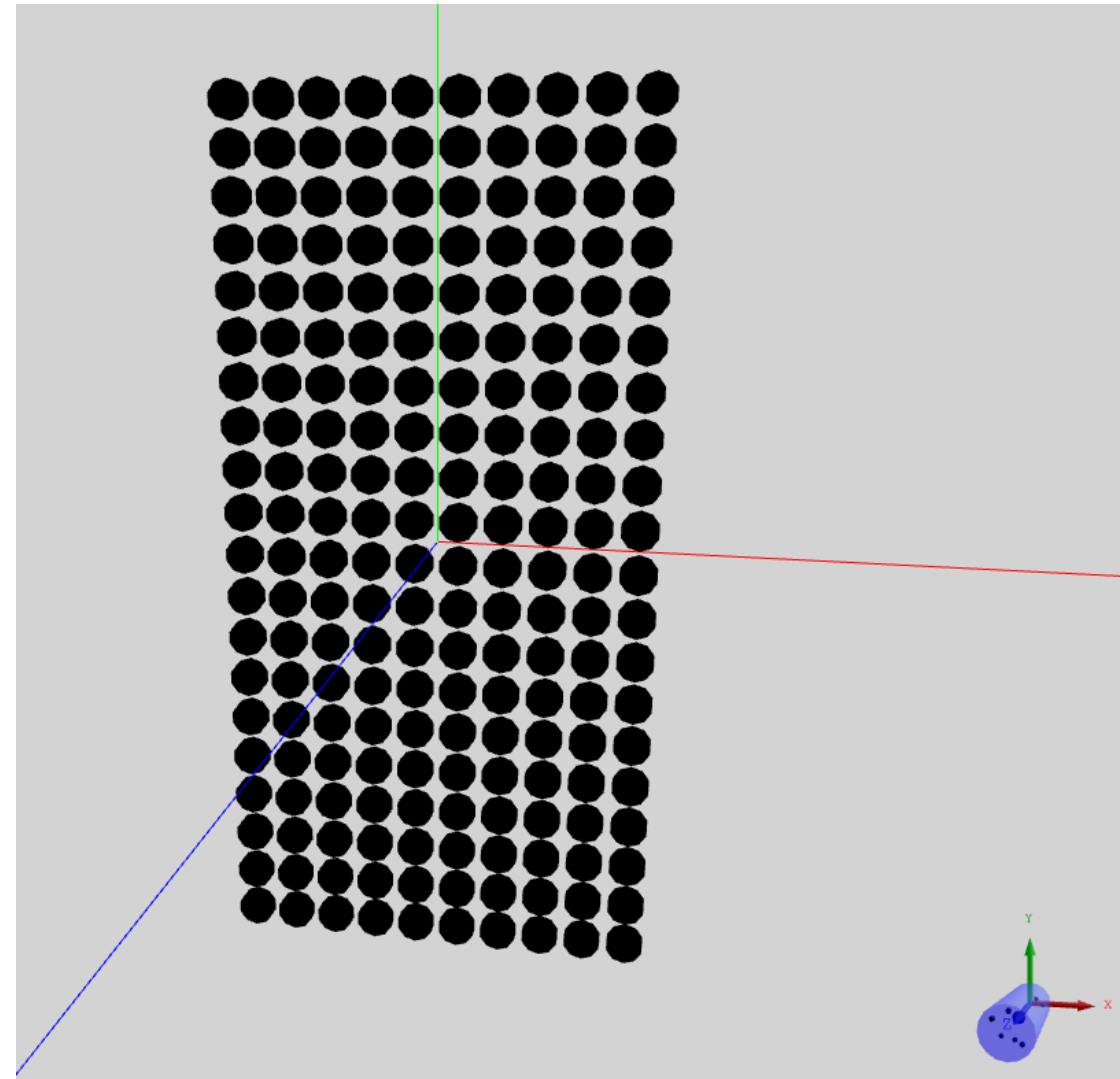
Row:  Row height:

Columns:  Column width:

First ID:

Start counting from:

Fields

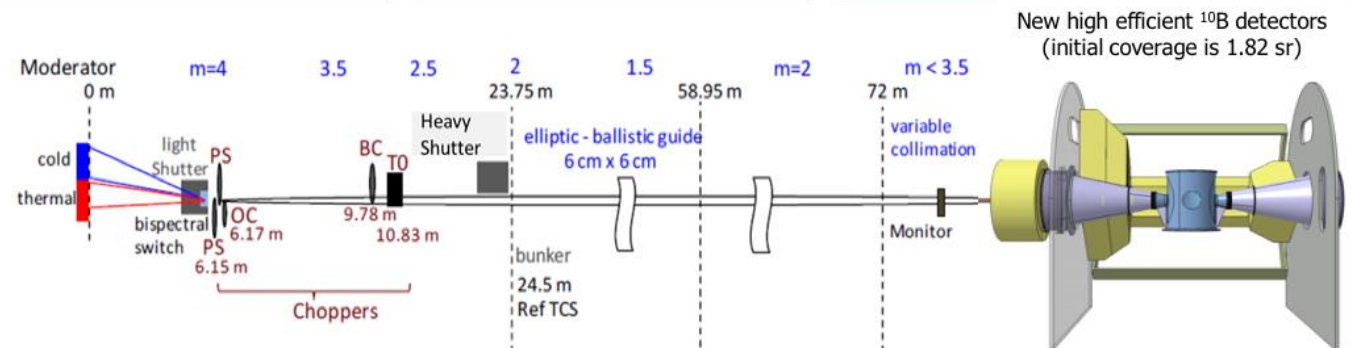
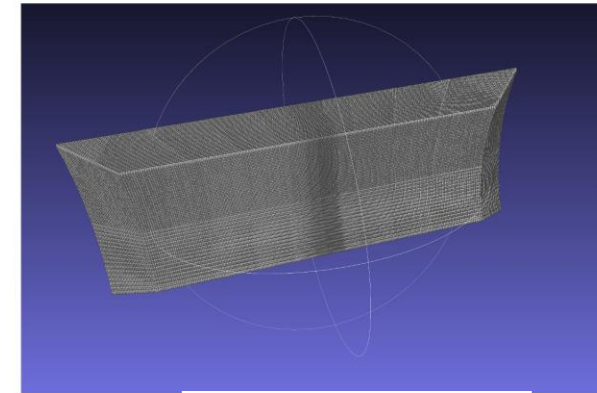
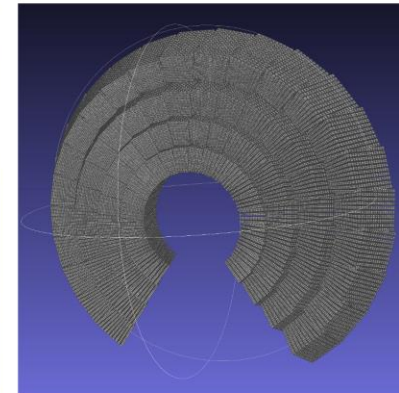
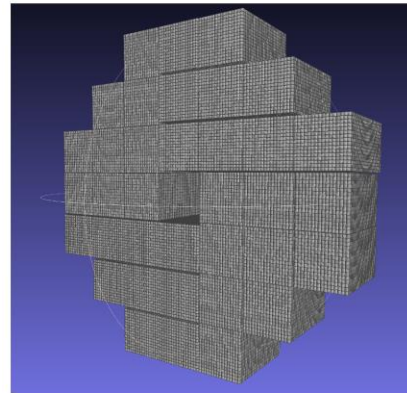
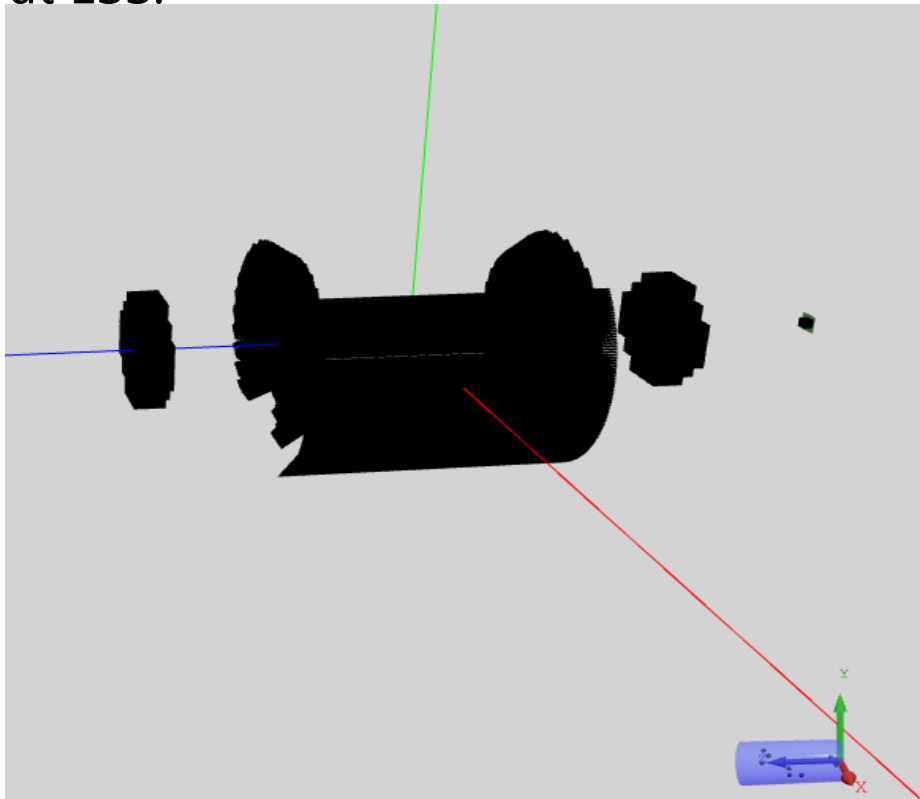


# Components

## Defining a component with complicated geometry using an OFF file

Geometry of the detector used in the different instruments can be defined using an OFF file if the geometry does not conform to the shapes in NeXus Constructor.

Example of such a case is the DREAM (Bi-spectral Powder Diffractometer) instrument at ESS.





# File writer modules

## Different fields to add in a group

1. When creating the sample, we added a static dataset containing the chemical formula of the sample.
2. This dataset is added as a field to the instance of the NXsample called sample.
3. However, there are in total 3 types of fields that can be added to a group (or component).
4. These are: datasets, links and streams.

### Datasets:

Datasets contain static data that we know the value of beforehand. Added directly to NeXus structure.

### Streams:

Streams define how dynamic data is written by the file writer using predefined flatbuffer schemas:

*Event data* would use **ev42** schema.

*Chopper rotation speed* would use **f142** schema.

### Links:

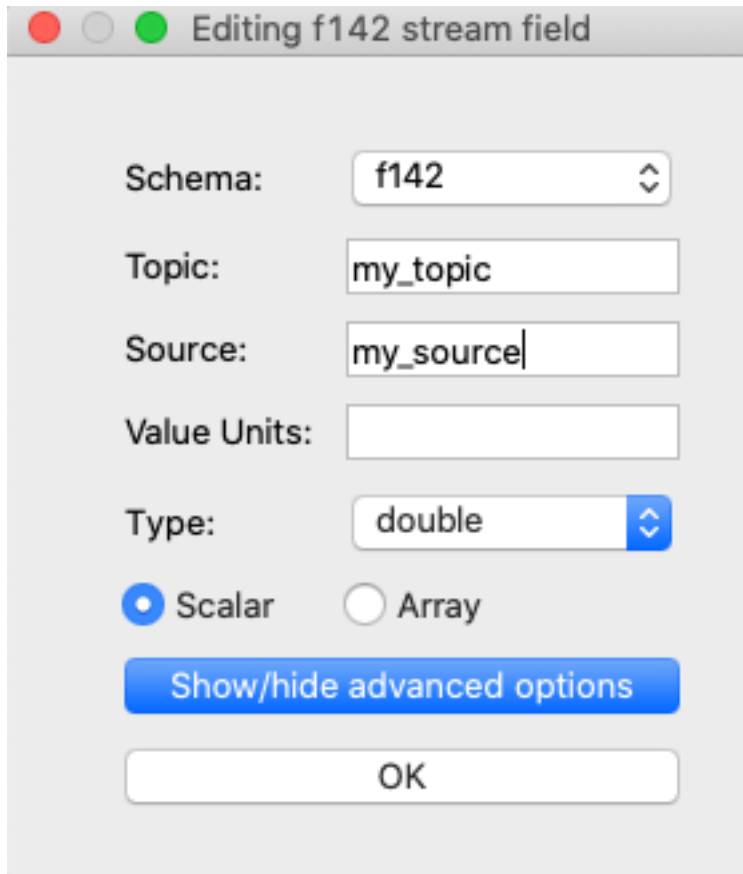
These are “shortcuts” to other parts of the NeXus file.



# File writer modules

Different fields to add in a group

## Configuring a f142 stream



Editing f142 stream field

Schema: f142

Topic: my\_topic

Source: my\_source

Value Units:

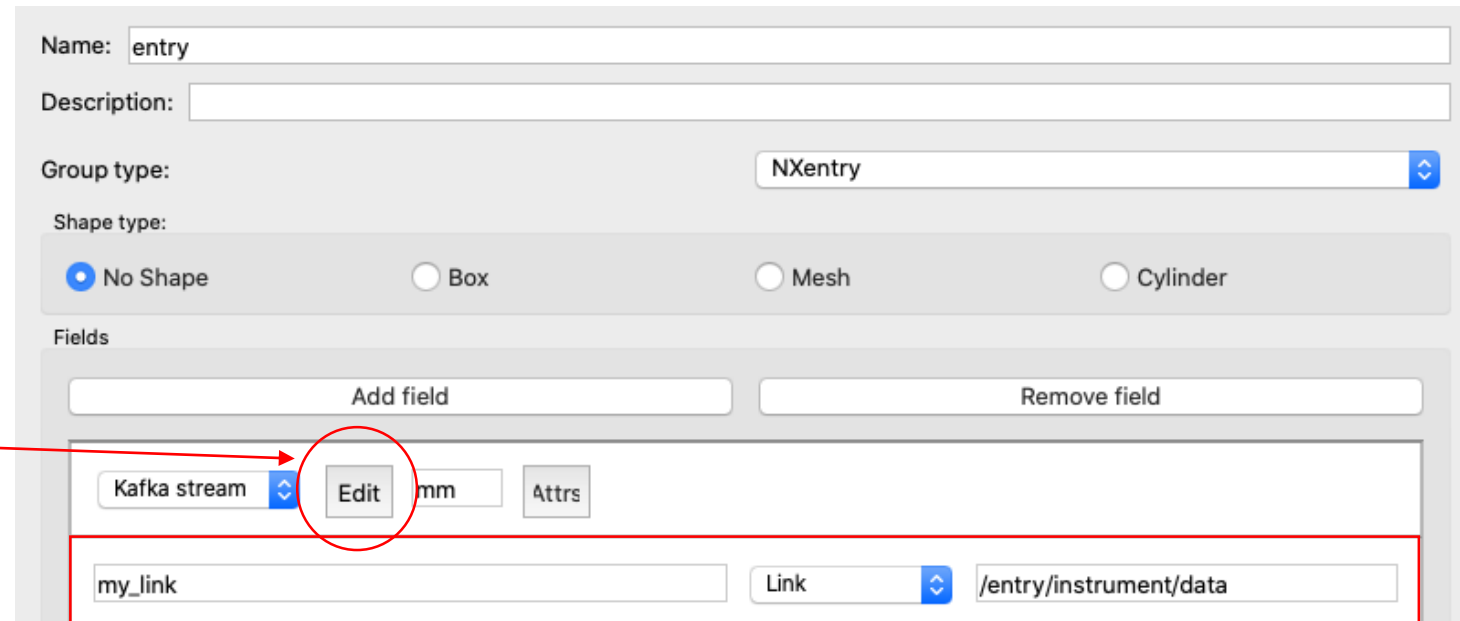
Type: double

☒ Scalar ☐ Array

Show/hide advanced options

OK

## Configuring a link



Name: entry

Description:

Group type: NXentry

Shape type:

☒ No Shape ☐ Box ☐ Mesh ☐ Cylinder

Fields

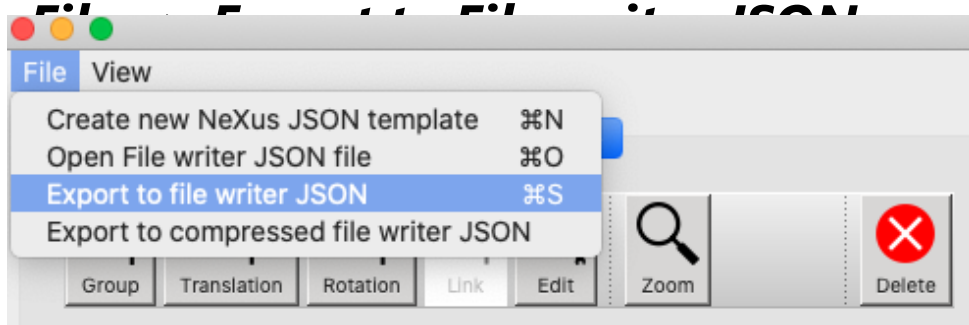
Add field Remove field

Kafka stream Edit mm Attrs

my\_link Link /entry/instrument/data

# Saving the NeXus template file is easy

1. Once the desired NeXus structure has been produced, it can export it to a JSON file.



2. The saved JSON file can be provided to the file writer to write a NeXus data file.


3. You can also load an existing and valid JSON file and edit it by:

**File → Open File writer JSON file**

```

63         "name": "probe",
64         "values": "visible light",
65         "type": "string"
66     }
67 }
68 ],
69 {
70     "name": "flir_camera",
71     "type": "group",
72     "attributes": [
73         {
74             "name": "NX_class",
75             "dtype": "string",
76             "values": "NXdetector"
77         }
78     ],
79     "children": [
80         {
81             "name": "image_key",
82             "type": "group",
83             "attributes": [
84                 {
85                     "name": "NX_class",
86                     "dtype": "string",
87                     "values": "NXlog"
88                 }
89             ],
90             "children": [
91                 {
92                     "module": "f142",
93                     "config": {
94                         "source": "flir_image_key",
95                         "topic": "ymir_nicos_devices",
96                         "dtype": "int16"
97                     }
98                 }
99             ]
100         }
101     ],
102     {
103         "name": "data",
104         "type": "group",
105         "attributes": [
106             {
107                 "name": "NX_class",
108                 "dtype": "string",
109                 "values": "NXlog"
110             }
111         ],
112         "children": [
113             {
114                 "module": "ADAr",
115                 "config": {
116                     "source": "some_source",
117                     "topic": "ymir_camera",
118                     "array_size": "$AREADET$"
119                 }
120             }
121         ]
122     }
123 }

```



| Name                   | Description                                |
|------------------------|--|
| 964281_00003931.hdf    | ["Unassigned experiments"]                 |
| entry                  | ["NXtomo"]                                 |
| data                   | ["964281"]                                 |
| definition             |  |
| experiment_identifier  |  |
| instrument             |  |
| flir_camera            |  |
| data                   |  |
| cue_index              | 1D data                                    |
| cue_timestamp_zero     | 1D data                                    |
| time                   | 1D data                                    |
| value                  | 3D data                                    |
| image_key              |  |
| alarm_severity         | 1D data                                    |
| alarm_status           | 1D data                                    |
| alarm_time             | 1D data                                    |
| average_value          | [1.71429]                                  |
| connection_status      |  |
| connection_status_time |  |
| cue_index              |  |
| cue_timestamp_zero     |  |
| maximum_value          | [3]  |
| minimum_value          | [0]  |
| time                   | 1D data                                    |
| value                  | 2D data                                    |
| laser_monitor          |  |
| laser_source           |  |
| light_source           |  |
| probe                  | ["visible light"]                          |
| mini_chopper           |  |
| sample                 |  |
| depends_on             | ["/entry/sample/transformations/rotation"] |
| name                   | ["lego"]                                   |
| transformations        | ["lego"]                                   |
| title                  | ["Unassigned experiments"]                 |

# Summary

## NeXus Constructor

1. As can be seen, the graphical user interface is simple.
2. The 3D component viewer enables the user to verify component geometries and locations while building the NeXus structure.
3. Complicated geometries can be imported as OFF files.
4. It is easy to save, load and share templates between people.
5. NeXus class documentation in the UI, field name suggestions and error checking helps the user to avoid the most obvious mistakes.

**NeXus Constructor can be found on:**

<https://github.com/ess-dmsc/nexus-constructor>





# Thank you for your attention!

## Questions?

Kenan Murić, Data Acquisition Scientist, ESS ERIC  
Contact: [kenan.muric@ess.eu](mailto:kenan.muric@ess.eu)