

# Locan Documentation

*Release 0.6.0.dev1+gbdfbc41*

**napari-locan Developers**

**Dec 12, 2023**

# GETTING STARTED

<b>1</b>	<b>Installation</b>	<b>2</b>
1.1	Dependencies . . . . .	2
1.2	Install from PyPI . . . . .	2
1.3	Install from conda-forge . . . . .	2
1.4	Install from distribution or sources . . . . .	3
1.5	Run tests . . . . .	3
<b>2</b>	<b>First steps</b>	<b>4</b>
2.1	SMLM data . . . . .	4
2.2	Sample data . . . . .	4
2.3	SMLM data procedures . . . . .	4
<b>3</b>	<b>Widgets</b>	<b>5</b>
<b>4</b>	<b>SmlmData and collections</b>	<b>6</b>
4.1	Localization data . . . . .	6
4.2	Collections . . . . .	6
<b>5</b>	<b>Region and Region of interest</b>	<b>7</b>
5.1	Regions . . . . .	7
5.2	ROIs . . . . .	7
<b>6</b>	<b>API Reference</b>	<b>8</b>
6.1	napari_locan.data_model . . . . .	8
6.1.1	Submodules: . . . . .	8
6.2	napari_locan.sample_data . . . . .	30
6.2.1	Submodules: . . . . .	30
6.3	napari_locan.scripts . . . . .	32
6.4	napari_locan.widgets . . . . .	32
6.4.1	Submodules: . . . . .	32
<b>7</b>	<b>Changelog</b>	<b>324</b>
7.1	0.6 - dev . . . . .	324
7.1.1	New Features . . . . .	324
7.1.2	API Changes . . . . .	324
7.1.3	Bug Fixes . . . . .	324
7.1.4	Other Changes and Additions . . . . .	324
7.2	0.5 - 2023-12-07 . . . . .	324
7.2.1	Bug Fixes . . . . .	324
7.2.2	Other Changes and Additions . . . . .	325

7.3	0.4.0 - 2023-11-08 . . . . .	325
	7.3.1 New Features . . . . .	325
	7.3.2 Other Changes and Additions . . . . .	325
7.4	0.3.0 - 2023-11-01 . . . . .	325
	7.4.1 Bug Fixes . . . . .	325
	7.4.2 Other Changes and Additions . . . . .	325
7.5	0.2.0 - 2023-10-31 . . . . .	325
	7.5.1 Bug Fixes . . . . .	325
	7.5.2 Other Changes and Additions . . . . .	326
7.6	0.1.0 - 2023-10-29 . . . . .	326
	7.6.1 New Features . . . . .	326
<b>8</b>	<b>License</b>	<b>327</b>
<b>9</b>	<b>Development</b>	<b>328</b>
9.1	Install . . . . .	328
9.2	Import Conventions . . . . .	328
9.3	Unit tests . . . . .	328
9.4	Coverage . . . . .	329
9.5	Code checks . . . . .	329
9.6	Versioning . . . . .	329
9.7	Documentation . . . . .	329
9.8	Type hints . . . . .	329
9.9	To remember . . . . .	329
<b>10</b>	<b>Indices and tables</b>	<b>330</b>
	<b>Python Module Index</b>	<b>331</b>
	<b>Index</b>	<b>332</b>

»

napari-locan is a plugin for working with single-molecule localization microscopy (SMLM) data within napari. Such data is typically generated in fluorescence-based super-resolution microscopy methods. SMLM techniques rely on finding the position of single-molecule emitters in time and space and reconstructing a super-resolved image or movie. The generated localizations are analyzed point-by-point for statistical and structural insight.

The plugin implements a subset of methods from [Locan](#), a python-based library with code for analyzing SMLM data. Locan provides extended functionality that is better suited for script- or notebook-based analysis procedures. napari-locan is well suited for exploratory data analysis.

## INSTALLATION

### 1.1 Dependencies

- python 3
- napari, locan and its dependencies on standard scipy and other open source libraries

A list with all hard and optional dependencies is given in *pyproject.toml*.

napari, locan and napari-locan require a Qt library like pyqt5 or pyside2, which is not specified as hard requirement. Make sure to have one (and only one) installed directly or through `napari[pyqt5]` or `locan[pyqt5]`.

### 1.2 Install from PyPI

Install napari-locan directly from the Python Package Index:

```
pip install napari-locan
```

Extra dependencies can be included:

```
pip install napari-locan[test,dev,docs]
```

### 1.3 Install from conda-forge

Install locan with the conda package manager (use mamba for better performance):

```
mamba install -c conda-forge napari-locan
```

## 1.4 Install from distribution or sources

In order to get the latest changes install from the GitHub repository main branch:

```
pip install git+https://github.com/super-resolution/napari-locan.git@main
```

or download distribution or wheel archive and install with pip:

```
pip install <distribution_file>
```

Install from local sources:

```
pip install <napari-locan_directory>
```

## 1.5 Run tests

Use pytest to run the tests from the source directory:

```
pytest
```

## FIRST STEPS

If napari is up and running with napari-locan installed you can process SMLM data by either opening sample data or loading SMLM data from a file.

### 2.1 SMLM data

When working with napari-locan you deal with SMLM data containing localizations with coordinates and other localization properties.

These datasets are stored as SMLM data model that can be accessed through the SMLM data widget.

It is important to note that the SMLM data model is completely independent from napari layers.

### 2.2 Sample data

When opening the sample data, two things happen: (i) Localization data is loaded and stored as a SMLM dataset. (ii) The SMLM dataset is rendered as point cloud or image.

The SMLM dataset can be accessed through the SMLM data widget and further processed through other widgets.

### 2.3 SMLM data procedures

Some widgets process SMLM datasets without modifying the SMLM data storage. For instance, rendering SMLM data as image creates a new napari layer that is then independent from the SMLM data.

Other widgets compute new SMLM datasets, e.g. by selecting data or computing clusters, that are stored as addition to the SMLM data. The new SMLM dataset is then accessible through the SMLM data widget.

Typically, widgets process the current selection in the SMLM data widget.

## WIDGETS

The plugin napari-locan contains the following widgets:

- 1) Inspecting SMLM data
  - SMLM data
  - Show metadata
  - Show properties
  - Show localization data
  - Show localization property distributions
- 2) Providing SMLM data
  - Load
  - Filter specifications
  - Select
  - Region of interest
- 3) Render SMLM data
  - Render points 2D / 3D
  - Render image 2D / 3D
  - Render features of a SMLM dataset
- 4) Processing SMLM data
  - Cluster
- 5) Collections of SMLM datasets
  - Render collection as 2D / 3D point cloud
  - Render collection features
- 6) Miscellaneous
  - Run script

Look up the API documentation for further details.



## SMLMDATA AND COLLECTIONS

**Localization data** in napari-locan is made of point-clouds with attributes.

All localization data is kept in the **napari\_locan.SmlmData model** as **locan.LocData objects**.

LocData objects can also contain a **collection** of several localization datasets.

### 4.1 Localization data

Localization data consists of a list of localizations with various localization properties. The dataset is represented by a dataframe together with general properties, metadata and other attributes. In napari-locan a dataset is kept as *locan.LocData* object that is inserted in the *napari\_locan.SmlmData* model and can be accessed by the attribute *smlm\_data.locdata*.

For details on *locan.LocData* and data structures in locan please see the [locan documentation on data structures](#)

### 4.2 Collections

Collections contain the individual LocData objects together with aggregated properties that make up a new LocData object. Think of it as localization clusters where each cluster can be represented as a single “localization” with a center position and other localization attributes.

In napari-locan, widgets typically access the current selection of the SmlmData model. Some widgets deal with collections. In that case, the current selection of the SmlmData model must be a collection.

## REGION AND REGION OF INTEREST

There are regions, regions of interest (ROI) and SMLM datasets.

Regions are geometrical objects like Rectangle, Ellipse or Polygon.

A ROI connects a region with a selected SMLM dataset for selected coordinates (or, more general, for localization properties).

ROI specifications can be saved as yaml file.

### 5.1 Regions

The napari shapes layer provides geometrical objects we call regions (e.g. Rectangle). These shapes can be transformed into corresponding locan regions (e.g. locan.Rectangle). They are independent of any image or SMLM dataset.

To get regions for use in napari-locan:

- 1) Select a napari shapes layer with shapes.
- 2) Transform napari shapes into regions by *Get regions*.
- 3) Make sure the scale parameter of that shapes layer is 1. Depending on layer creation history it might be set to the shapes value of another layer. If so, press *Reset scale* and repeat (2).

### 5.2 ROIs

When dealing with ROIs we have to distinguish two workflows:

- 1) Select localizations within a region: for this, define a ROI with reference to a SMLM dataset and press *Apply* to create a new SmlmData with the selection.
- 2) Save ROI specifications: for this, define a ROI with reference to a localization file and press *Save* to save ROI specifications.

Every ROI specification contains a reference to data, a region definition and a selection of localization properties. It is kept as instance of *locan.Roi(reference, region, loc\_properties)*. Before creating a ROI in the Roi widget you have to specify the kind of reference. For *SmlmData* the currently selected SMLM dataset is taken. For *File reference*, a file path and type is extracted from the currently selected SMLM dataset. For *Open dialog* an existing localization file can be selected.

## API REFERENCE

napari-locan should be used as napari plugin. Therefore, there are not many public functions accessible. For information on backend modules have a look at [locan](#) and its [documentation](#). napari-locan consists of the following modules:

<i>data_model</i>	Data models for napari-locan.
<i>sample_data</i>	Sample data for napari
<i>scripts</i>	Example scripts for napari-locan.
<i>widgets</i>	QWidgets for napari-locan.

### 6.1 napari\_locan.data\_model

Data models for napari-locan.

#### 6.1.1 Submodules:

<i>data_model_base</i>	Abstract base class for a data model.
<i>filter_specifications</i>	The data model for filter specifications.
<i>region_specifications</i>	The data model for region specifications.
<i>roi_specifications</i>	The data model for region of interest (ROI) specifications.
<i>smlm_data</i>	The data model for localization data

#### napari\_locan.data\_model.data\_model\_base

Abstract base class for a data model.

A data model holds either `smlm_data`, `filter_specifications`, `regions` or other data structures.

The interface provides methods to manipulate the data contents.

## Classes

<code>DataModel</code> ([datasets, names])	Abstract base class for container classes holding various data structures.
<code>QABCMeta</code> (name, bases, namespace, **kwargs)	

---

### napari\_locan.data\_model.data\_model\_base.DataModel

**class** napari\_locan.data\_model.data\_model\_base.**DataModel** (*datasets=None, names=None*)

Bases: PyQt5.QtCore.QObject, abc.ABC

Abstract base class for container classes holding various data structures.

#### Variables

- **count** (*int*) – Monotonically increasing integer counting the overall created datasets.
- **datasets\_changed\_signal** (*PyQt5.QtCore.pyqtSignal*) – A Qt signal for index
- **names\_changed\_signal** (*PyQt5.QtCore.pyqtSignal*) – A Qt signal for names
- **index\_changed\_signal** (*PyQt5.QtCore.pyqtSignal*) – A Qt signal for index
- **datasets** – Data structures
- **names** – Data structure string identifier
- **index** – Current selection of data structure
- **dataset** – The selected data object
- **name** – The selected data identifier

#### Methods

<code>__init__</code> ([datasets, names])	
<code>append_item</code> (dataset[, name, set_index])	Append a new item to the end of datasets and point index to new dataset if set_index is true.
<code>blockSignals</code> (self, b)	
<code>childEvent</code> (self, a0)	
<code>children</code> (self)	

---

continues on next page

Table 1 – continued from previous page

<code>connectNotify(self, signal)</code>	
<code>customEvent(self, a0)</code>	
<code>deleteLater(self)</code>	
<code><i>delete_all()</i></code>	Delete all datasets and set index to -1.
<code><i>delete_item()</i></code>	Delete current dataset and set index to the previous dataset.
<code>disconnect(-&gt; bool)</code>	
<code>disconnectNotify(self, signal)</code>	
<code>dumpObjectInfo(self)</code>	
<code>dumpObjectTree(self)</code>	
<code>dynamicPropertyNames(self)</code>	
<code>event(self, a0)</code>	
<code>eventFilter(self, a0, a1)</code>	
<code>findChild(-&gt; QObjectT)</code>	
<code>findChildren(...)</code>	
<code>inherits(self, classname)</code>	
<code>installEventFilter(self, a0)</code>	
<code>isSignalConnected(self, signal)</code>	
<code>isWidgetType(self)</code>	
<code>isWindowType(self)</code>	
<code>killTimer(self, id)</code>	
<code>metaObject(self)</code>	
<code>moveToThread(self, thread)</code>	
<code>objectName(self)</code>	
<code>parent(self)</code>	
<code>property(self, name)</code>	

continues on next page

Table 1 – continued from previous page

<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>receivers(self, signal)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>sender(self)</code>	
<code>senderSignalIndex(self)</code>	
<code>setObjectName(self, name)</code>	
<code>setParent(self, a0)</code>	
<code>setProperty(self, name, value)</code>	
<code><i>set_datasets_and_names</i>([datasets, names])</code>	Set datasets and names to the given values and point index to the last item.
<code><i>set_index_slot</i>(value)</code>	QT slot for property <code>self.index</code> .
<code>signalsBlocked(self)</code>	
<code>startTimer(self, interval[, timerType])</code>	
<code>thread(self)</code>	
<code>timerEvent(self, a0)</code>	
<code>tr(self, sourceText[, disambiguation, n])</code>	

## Attributes

<i>count</i>	
<i>dataset</i>	<b>rtype</b> Optional[Any]
<i>datasets</i>	<b>rtype</b> list[Any]
<i>datasets_changed_signal</i>	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
<i>destroyed</i>	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
<i>index</i>	<b>rtype</b> int
<i>index_changed_signal</i>	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
<i>name</i>	<b>rtype</b> str
<i>names</i>	<b>rtype</b> list[str]
<i>names_changed_signal</i>	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
<i>objectNameChanged</i>	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
<i>staticMetaObject</i>	

**append\_item**(*dataset*, *name=None*, *set\_index=True*)

Append a new item to the end of datasets and point index to new dataset if *set\_index* is true.

**Return type** None

**count:** int = 0

**property dataset:** Optional[Any]

**Return type** Optional[Any]

**property datasets:** list[Any]

**Return type** list[Any]

**datasets\_changed\_signal:** PyQt5.QtCore.pyqtSignal

int = ..., arguments: Sequence = ...) -> PYQT\_SIGNAL

types is normally a sequence of individual types. Each type is either a type object or a string that is the name of a C++ type. Alternatively each type could itself be a sequence of types each describing a different overloaded signal. *name* is the optional C++ name of the signal.

If it is not specified then the name of the class attribute that is bound to the signal is used. revision is the optional revision of the signal that is exported to QML. If it is not specified then 0 is used. arguments is the optional sequence of the names of the signal's arguments.

**Type** `pyqtSignal(*types, name`

**Type** `str = ..., revision`

**delete\_all()**

Delete all datasets and set index to -1.

**Return type** `None`

**delete\_item()**

Delete current dataset and set index to the previous dataset.

**Return type** `None`

**property index:** `int`

**Return type** `int`

**index\_changed\_signal:** `PyQt5.QtCore.pyqtSignal`

`int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL`

types is normally a sequence of individual types. Each type is either a type object or a string that is the name of a C++ type. Alternatively each type could itself be a sequence of types each describing a different overloaded signal. name is the optional C++ name of the signal. If it is not specified then the name of the class attribute that is bound to the signal is used. revision is the optional revision of the signal that is exported to QML. If it is not specified then 0 is used. arguments is the optional sequence of the names of the signal's arguments.

**Type** `pyqtSignal(*types, name`

**Type** `str = ..., revision`

**property name:** `str`

**Return type** `str`

**property names:** `list[str]`

**Return type** `list[str]`

**names\_changed\_signal:** `PyQt5.QtCore.pyqtSignal`

`int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL`

types is normally a sequence of individual types. Each type is either a type object or a string that is the name of a C++ type. Alternatively each type could itself be a sequence of types each describing a different overloaded signal. name is the optional C++ name of the signal. If it is not specified then the name of the class attribute that is bound to the signal is used. revision is the optional revision of the signal that is exported to QML. If it is not specified then 0 is used. arguments is the optional sequence of the names of the signal's arguments.

**Type** `pyqtSignal(*types, name`

**Type** `str = ..., revision`



**set\_datasets\_and\_names**(*datasets=None, names=None*)

Set datasets and names to the given values and point index to the last item.

**Return type** None

**set\_index\_slot**(*value*)

QT slot for property self.index.

**Return type** None

## napari\_locan.data\_model.data\_model\_base.QABCMeta

**class** napari\_locan.data\_model.data\_model\_base.QABCMeta(*name, bases, namespace, \*\*kwargs*)

Bases: sip.wrappertype, abc.ABCMeta

### Methods

<code>__init__(*args, **kwargs)</code>	
<code>mro()</code>	Return a type's method resolution order.
<code>register(subclass)</code>	Register a virtual subclass of an ABC.

## napari\_locan.data\_model.filter\_specifications

The data model for filter specifications.

This module contains a data model to serve as container for filter specifications to select localization property values.

The data model is used by other napari-locan widgets to process localization data and yield new SMLM datasets. It is entirely independent of napari layers.

### Classes

<code>FilterSpecifications([datasets, names])</code>	Container for one or more filter specifications.
--	--

## napari\_locan.data\_model.filter\_specifications.FilterSpecifications

**class** napari\_locan.data\_model.filter\_specifications.FilterSpecifications(*datasets=None, names=None*)

Bases: *napari\_locan.data\_model.data\_model\_base.DataModel*

Container for one or more filter specifications.

### Variables

- **datasets\_changed\_signal** (*PyQt5.QtCore.pyqtSignal*) – A Qt signal for index

- **names\_changed\_signal** (*PyQt5.QtCore.pyqtSignal*) – A Qt signal for names
- **index\_changed\_signal** (*PyQt5.QtCore.pyqtSignal*) – A Qt signal for index
- **datasets** – Data structures
- **names** – Data structure string identifier
- **index** – Current selection of data structure
- **dataset** – The selected data object
- **name** – The selected data identifier

## Methods

---

<code>__init__([datasets, names])</code>	
<code>append_item(dataset[, name, set_index])</code>	Append a new item to the end of datasets and point index to new dataset if <code>set_index</code> is true.
<code>blockSignals(self, b)</code>	
<code>childEvent(self, a0)</code>	
<code>children(self)</code>	
<code>connectNotify(self, signal)</code>	
<code>customEvent(self, a0)</code>	
<code>deleteLater(self)</code>	
<code>delete_all()</code>	Delete all datasets and set index to -1.
<code>delete_item()</code>	Delete current dataset and set index to the previous dataset.
<code>disconnect(-&gt; bool)</code>	
<code>disconnectNotify(self, signal)</code>	
<code>dumpObjectInfo(self)</code>	
<code>dumpObjectTree(self)</code>	
<code>dynamicPropertyNames(self)</code>	
<code>event(self, a0)</code>	
<code>eventFilter(self, a0, a1)</code>	

---

continues on next page

Table 2 – continued from previous page

<code>findChild(-&gt; QObjectT)</code>	
<code>findChildren(...)</code>	
<code>inherits(self, classname)</code>	
<code>installEventFilter(self, a0)</code>	
<code>isSignalConnected(self, signal)</code>	
<code>isWidgetType(self)</code>	
<code>isWindowType(self)</code>	
<code>killTimer(self, id)</code>	
<code>metaObject(self)</code>	
<code>moveToThread(self, thread)</code>	
<code>objectName(self)</code>	
<code>parent(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>receivers(self, signal)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>sender(self)</code>	
<code>senderSignalIndex(self)</code>	
<code>setObjectName(self, name)</code>	
<code>setParent(self, a0)</code>	
<code>setProperty(self, name, value)</code>	
<code>set_datasets_and_names([datasets, names])</code>	Set datasets and names to the given values and point index to the last item.
<code>set_index_slot(value)</code>	QT slot for property <code>self.index</code> .
<code>signalsBlocked(self)</code>	
<code>startTimer(self, interval[, timerType])</code>	

continues on next page

Table 2 – continued from previous page

thread(self)
timerEvent(self, a0)
tr(self, sourceText[, disambiguation, n])

## Attributes

count	
dataset	<b>rtype</b> Optional[Any]
datasets	<b>rtype</b> list[Any]
datasets_changed_signal	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
<i>filter_condition</i>	<b>rtype</b> str
index	<b>rtype</b> int
index_changed_signal	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
name	<b>rtype</b> str
names	<b>rtype</b> list[str]
names_changed_signal	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	

**property** *filter\_condition*: str

**Return type** str

## napari\_locan.data\_model.region\_specifications

The data model for region specifications.

This module contains a data model to serve as container for region specifications.

The data model is used by other napari-locan widgets to process localization data. It is entirely independent of napari layers.

### Classes

---

<code>RegionSpecifications([datasets, names])</code>	Container for one or more region specifications.
--	--

---

## napari\_locan.data\_model.region\_specifications.RegionSpecifications

**class** `napari_locan.data_model.region_specifications.RegionSpecifications`(*datasets=None, names=None*)

Bases: `napari_locan.data_model.data_model_base.DataModel`

Container for one or more region specifications.

### Variables

- **datasets\_changed\_signal** (`PyQt5.QtCore.pyqtSignal`) – A Qt signal for index
- **names\_changed\_signal** (`PyQt5.QtCore.pyqtSignal`) – A Qt signal for names
- **index\_changed\_signal** (`PyQt5.QtCore.pyqtSignal`) – A Qt signal for index
- **datasets** – Data structures
- **names** – Data structure string identifier
- **index** – Current selection of data structure
- **dataset** – The selected data object
- **name** – The selected data identifier

### Methods

---

`__init__([datasets, names])`

---

<code>append_item(dataset[, name, set_index])</code>	Append a new item to the end of datasets and point index to new dataset if <code>set_index</code> is true.
--	--

---

`blockSignals(self, b)`

---

`childEvent(self, a0)`

---

continues on next page

Table 3 – continued from previous page

<code>children(self)</code>	
<code>connectNotify(self, signal)</code>	
<code>customEvent(self, a0)</code>	
<code>deleteLater(self)</code>	
<code>delete_all()</code>	Delete all datasets and set index to -1.
<code>delete_item()</code>	Delete current dataset and set index to the previous dataset.
<code>disconnect(-&gt; bool)</code>	
<code>disconnectNotify(self, signal)</code>	
<code>dumpObjectInfo(self)</code>	
<code>dumpObjectTree(self)</code>	
<code>dynamicPropertyNames(self)</code>	
<code>event(self, a0)</code>	
<code>eventFilter(self, a0, a1)</code>	
<code>findChild(-&gt; QObjectT)</code>	
<code>findChildren(...)</code>	
<code>inherits(self, classname)</code>	
<code>installEventFilter(self, a0)</code>	
<code>isSignalConnected(self, signal)</code>	
<code>isWidgetType(self)</code>	
<code>isWindowType(self)</code>	
<code>killTimer(self, id)</code>	
<code>metaObject(self)</code>	
<code>moveToThread(self, thread)</code>	
<code>objectName(self)</code>	
<code>parent(self)</code>	

continues on next page

Table 3 – continued from previous page

<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>receivers(self, signal)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>sender(self)</code>	
<code>senderSignalIndex(self)</code>	
<code>setObjectName(self, name)</code>	
<code>setParent(self, a0)</code>	
<code>setProperty(self, name, value)</code>	
<code>set_datasets_and_names([datasets, names])</code>	Set datasets and names to the given values and point index to the last item.
<code>set_index_slot(value)</code>	QT slot for property <code>self.index</code> .
<code>signalsBlocked(self)</code>	
<code>startTimer(self, interval[, timerType])</code>	
<code>thread(self)</code>	
<code>timerEvent(self, a0)</code>	
<code>tr(self, sourceText[, disambiguation, n])</code>	

## Attributes

count	
dataset	<b>rtype</b> Optional[Any]
datasets	<b>rtype</b> list[Any]
datasets_changed_signal	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
index	<b>rtype</b> int
index_changed_signal	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
name	<b>rtype</b> str
names	<b>rtype</b> list[str]
names_changed_signal	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	

## napari\_locan.data\_model.roi\_specifications

The data model for region of interest (ROI) specifications.

This module contains a data model to serve as container for ROI specifications.

The data model is used by other napari-locan widgets to process localization data. It is entirely independent of napari layers.



## Classes

---

*RoiSpecifications*([datasets, names])      Container for one or more ROI specifications.

---

### napari\_locan.data\_model.roi\_specifications.RoiSpecifications

**class** napari\_locan.data\_model.roi\_specifications.**RoiSpecifications**(*datasets=None, names=None*)

Bases: *napari\_locan.data\_model.data\_model\_base.DataModel*

Container for one or more ROI specifications.

#### Variables

- **datasets\_changed\_signal** (*PyQt5.QtCore.pyqtSignal*) – A Qt signal for index
- **names\_changed\_signal** (*PyQt5.QtCore.pyqtSignal*) – A Qt signal for names
- **index\_changed\_signal** (*PyQt5.QtCore.pyqtSignal*) – A Qt signal for index
- **datasets** – Data structures
- **names** – Data structure string identifier
- **index** – Current selection of data structure
- **dataset** – The selected data object
- **name** – The selected data identifier

#### Methods

---

`__init__`([datasets, names])

---

`append_item`(dataset[, name, set\_index])      Append a new item to the end of datasets and point index to new dataset if set\_index is true.

---

`blockSignals`(self, b)

---

`childEvent`(self, a0)

---

`children`(self)

---

`connectNotify`(self, signal)

---

`customEvent`(self, a0)

---

`deleteLater`(self)

---

continues on next page

Table 4 – continued from previous page

<code>delete_all()</code>	Delete all datasets and set index to -1.
<code>delete_item()</code>	Delete current dataset and set index to the previous dataset.
<code>disconnect(-&gt; bool)</code>	
<code>disconnectNotify(self, signal)</code>	
<code>dumpObjectInfo(self)</code>	
<code>dumpObjectTree(self)</code>	
<code>dynamicPropertyNames(self)</code>	
<code>event(self, a0)</code>	
<code>eventFilter(self, a0, a1)</code>	
<code>findChild(-&gt; QObjectT)</code>	
<code>findChildren(...)</code>	
<code>inherits(self, classname)</code>	
<code>installEventFilter(self, a0)</code>	
<code>isSignalConnected(self, signal)</code>	
<code>isWidgetType(self)</code>	
<code>isWindowType(self)</code>	
<code>killTimer(self, id)</code>	
<code>metaObject(self)</code>	
<code>moveToThread(self, thread)</code>	
<code>objectName(self)</code>	
<code>parent(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>receivers(self, signal)</code>	
<code>removeEventFilter(self, a0)</code>	

continues on next page

Table 4 – continued from previous page

sender(self)	
senderSignalIndex(self)	
setObjectName(self, name)	
setParent(self, a0)	
setProperty(self, name, value)	
set_datasets_and_names([datasets, names])	Set datasets and names to the given values and point index to the last item.
set_index_slot(value)	QT slot for property self.index.
signalsBlocked(self)	
startTimer(self, interval[, timerType])	
thread(self)	
timerEvent(self, a0)	
tr(self, sourceText[, disambiguation, n])	

## Attributes

count	
dataset	<b>rtype</b> Optional[Any]
datasets	<b>rtype</b> list[Any]
datasets_changed_signal	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
index	<b>rtype</b> int
index_changed_signal	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
name	<b>rtype</b> str
names	<b>rtype</b> list[str]
names_changed_signal	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	

## napari\_locan.data\_model.slm\_data

The data model for localization data

This module contains a data model to serve as container for SMLM data. The individual SMLM datasets are provided as `locan.LocData` instances.

SMLM data serves as data model for other napari-locan widgets to process or render the localization data. It is entirely independent of napari layers. Upon rendering a SMLM dataset a new image is created in a new napari layer.

## Classes

---

*SmlmData*([locdatas, locdata\_names])      Container for one or more LocData objects.

---

### napari\_locan.data\_model.smlm\_data.SmlmData

**class** napari\_locan.data\_model.smlm\_data.**SmlmData**(locdatas=None,  
locdata\_names=None)

Bases: PyQt5.QtCore.QObject

Container for one or more LocData objects.

#### Variables

- **index\_changed\_signal** (*PyQt5.QtCore.pyqtSignal*) – A Qt signal for index
- **locdata\_names\_changed\_signal** (*PyQt5.QtCore.pyqtSignal*) – A Qt signal for locdata\_names
- **locdatas** – Localization datasets
- **locdata\_names** – Localization string identifier
- **index** – Current selection of locdatas
- **locdata** – The selected LocData object
- **locdata\_name** – The selected LocData identifier

#### Methods

---

`__init__`([locdatas, locdata\_names])

---

`append_item`(locdata[, locdata\_name,  
set\_index])      **rtype** None

---

`blockSignals`(self, b)

---

`childEvent`(self, a0)

---

`children`(self)

---

`connectNotify`(self, signal)

---

`customEvent`(self, a0)

---

`deleteLater`(self)

---

continues on next page

Table 5 – continued from previous page

<code>delete_all()</code>	<b>rtype</b> None
<code>delete_item()</code>	<b>rtype</b> None
<code>disconnect(-&gt; bool)</code>	
<code>disconnectNotify(self, signal)</code>	
<code>dumpObjectInfo(self)</code>	
<code>dumpObjectTree(self)</code>	
<code>dynamicPropertyNames(self)</code>	
<code>event(self, a0)</code>	
<code>eventFilter(self, a0, a1)</code>	
<code>findChild(-&gt; QObjectT)</code>	
<code>findChildren(...)</code>	
<code>inherits(self, classname)</code>	
<code>installEventFilter(self, a0)</code>	
<code>isSignalConnected(self, signal)</code>	
<code>isWidgetType(self)</code>	
<code>isWindowType(self)</code>	
<code>killTimer(self, id)</code>	
<code>metaObject(self)</code>	
<code>moveToThread(self, thread)</code>	
<code>objectName(self)</code>	
<code>parent(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.

continues on next page

Table 5 – continued from previous page

<code>receivers(self, signal)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>sender(self)</code>	
<code>senderSignalIndex(self)</code>	
<code>setObjectName(self, name)</code>	
<code>setParent(self, a0)</code>	
<code>setProperty(self, name, value)</code>	
<code><i>set_index_slot</i>(value)</code>	QT slot for property self.index.
<code>signalsBlocked(self)</code>	
<code>startTimer(self, interval[, timerType])</code>	
<code>thread(self)</code>	
<code>timerEvent(self, a0)</code>	
<code>tr(self, sourceText[, disambiguation, n])</code>	

## Attributes

<code>destroyed</code>	<code>int = ..., arguments: Sequence = ...) -&gt; PYQT_SIGNAL</code>
<code>index</code>	<code>rtype int</code>
<code>index_changed_signal</code>	<code>int = ..., arguments: Sequence = ...) -&gt; PYQT_SIGNAL</code>
<code>locdata</code>	<code>rtype Optional[LocData]</code>
<code>locdata_name</code>	<code>rtype str</code>
<code>locdata_names</code>	<code>rtype list[str]</code>
<code>locdata_names_changed_signal</code>	<code>int = ..., arguments: Sequence = ...) -&gt; PYQT_SIGNAL</code>
<code>locdatas</code>	<code>rtype list[LocData]</code>
<code>objectNameChanged</code>	<code>int = ..., arguments: Sequence = ...) -&gt; PYQT_SIGNAL</code>
<code>staticMetaObject</code>	

**append\_item**(*locdata*, *locdata\_name*=None, *set\_index*=True)

**Return type** None

**delete\_all**()

**Return type** None

**delete\_item**()

**Return type** None

**property index:** `int`

**Return type** `int`

**index\_changed\_signal:** `PyQt5.QtCore.pyqtSignal`

`int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL`

types is normally a sequence of individual types. Each type is either a type object or a string that is the name of a C++ type. Alternatively each type could itself be a sequence of types each describing a different overloaded signal. *name* is the optional C++ name of the signal. If it is not specified then the name of the class attribute that is bound to the signal is used. *revision* is the optional revision of the signal that is exported to QML. If it is not specified then 0 is used. *arguments* is the optional sequence of the names of the signal's arguments.



**Type** pyqtSignal(\*types, name

**Type** str = ..., revision

**property locdata:** locan.data.locdata.LocData | None

**Return type** Optional[LocData]

**property locdata\_name:** str

**Return type** str

**property locdata\_names:** list[str]

**Return type** list[str]

**locdata\_names\_changed\_signal:** PyQt5.QtCore.pyqtSignal

int = ..., arguments: Sequence = ...) -> PYQT\_SIGNAL

types is normally a sequence of individual types. Each type is either a type object or a string that is the name of a C++ type. Alternatively each type could itself be a sequence of types each describing a different overloaded signal. name is the optional C++ name of the signal. If it is not specified then the name of the class attribute that is bound to the signal is used. revision is the optional revision of the signal that is exported to QML. If it is not specified then 0 is used. arguments is the optional sequence of the names of the signal's arguments.

**Type** pyqtSignal(\*types, name

**Type** str = ..., revision

**property locdatas:** list[locan.data.locdata.LocData]

**Return type** list[LocData]

**set\_index\_slot**(value)

QT slot for property self.index.

**Return type** None

## 6.2 napari\_locan.sample\_data

Sample data for napari

### 6.2.1 Submodules:

---

*sample\_data*

SMLM sample data

---

## napari\_locan.sample\_data.sample\_data

SMLM sample data

This module provides SMLM sample data as could be generated with napari-locan.

It implements the “sample data” specification. see: [“https://napari.org/stable/plugins/guides.html/?/#sample-data”](https://napari.org/stable/plugins/guides.html/?/#sample-data)

### Functions

<code>make_image_npc([smlm_data])</code>	Generate a sample image from <i>locan.datasets.load_npc</i> .
<code>make_image_tubulin([smlm_data])</code>	Generate a sample image from <i>locan.datasets.load_tubulin</i> .
<code>make_points_npc([smlm_data])</code>	Generate localizations from <i>locan.datasets.load_npc</i> .
<code>make_points_tubulin([smlm_data])</code>	Generate localizations from <i>locan.datasets.load_tubulin</i> .

### napari\_locan.sample\_data.sample\_data.make\_image\_npc

`napari_locan.sample_data.sample_data.make_image_npc(smlm_data=<napari_locan.data_model.smlm_data_model object>)`

Generate a sample image from *locan.datasets.load\_npc*.

**Return type** list[Union[Tuple[Any], Tuple[Any, Dict], Tuple[Any, Dict, Literal[‘image’, ‘labels’, ‘points’, ‘shapes’, ‘surface’, ‘tracks’, ‘vectors’]]]]

### napari\_locan.sample\_data.sample\_data.make\_image\_tubulin

`napari_locan.sample_data.sample_data.make_image_tubulin(smlm_data=<napari_locan.data_model.smlm_data_model object>)`

Generate a sample image from *locan.datasets.load\_tubulin*.

**Return type** list[Union[Tuple[Any], Tuple[Any, Dict], Tuple[Any, Dict, Literal[‘image’, ‘labels’, ‘points’, ‘shapes’, ‘surface’, ‘tracks’, ‘vectors’]]]]

### napari\_locan.sample\_data.sample\_data.make\_points\_npc

`napari_locan.sample_data.sample_data.make_points_npc(smlm_data=<napari_locan.data_model.smlm_data_model object>)`

Generate localizations from *locan.datasets.load\_npc*.

**Return type** list[Union[Tuple[Any], Tuple[Any, Dict], Tuple[Any, Dict, Literal[‘image’, ‘labels’, ‘points’, ‘shapes’, ‘surface’, ‘tracks’, ‘vectors’]]]]

## napari\_locan.sample\_data.sample\_data.make\_points\_tubulin

napari\_locan.sample\_data.sample\_data.**make\_points\_tubulin**(*smlm\_data*=<napari\_locan.data\_model.smlm\_data\_model>  
*object*>)

Generate localizations from *locan.datasets.load\_tubulin*.

**Return type** list[Union[Tuple[Any], Tuple[Any, Dict], Tuple[Any, Dict, Literal['image', 'labels', 'points', 'shapes', 'surface', 'tracks', 'vectors']]]

## 6.3 napari\_locan.scripts

Example scripts for napari-locan.

This module contains python scripts to be run from within the plugin.

## 6.4 napari\_locan.widgets

QWidgets for napari-locan.

### 6.4.1 Submodules:

<i>widget_clustering</i>	Compute localization clusters.
<i>widget_filter_specifications</i>	The data model for filter specifications.
<i>widget_load</i>	Load SMLM data files.
<i>widget_napari_locan_project</i>	Save and load the current state of napari-locan.
<i>widget_property_distribution</i>	Show localization property distribution.
<i>widget_render_collection_2d</i>	Render collection of SMLM data as 2d point clouds.
<i>widget_render_collection_features</i>	Render selected features of all SMLM datasets in a collection.
<i>widget_render_features</i>	Render selected features of a SMLM dataset.
<i>widget_render_image_2d</i>	Render 2d image.
<i>widget_render_image_3d</i>	Render 3d image.
<i>widget_render_points_2d</i>	Render 2d point cloud.
<i>widget_render_points_3d</i>	Render 3d point cloud.
<i>widget_roi</i>	Create regions of interest.
<i>widget_run_script</i>	Run python script.
<i>widget_select</i>	Select localizations from SMLM dataset.
<i>widget_show_data</i>	Show data statistics for a SMLM dataset.
<i>widget_show_metadata</i>	Show metadata for a SMLM dataset.
<i>widget_show_properties</i>	Show locdata properties for a SMLM dataset.
<i>widget_smlm_data</i>	The data model for localization-based SMLM data.

## napari\_locan.widgets.widget\_clustering

Compute localization clusters.

QWidget plugin for clustering SMLM data. More advanced clustering routines are available through locan-based scripts.

### Classes

---

*ClusteringQWidget*(napari\_viewer[,  
smlm\_data])

---

## napari\_locan.widgets.widget\_clustering.ClusteringQWidget

**class** napari\_locan.widgets.widget\_clustering.ClusteringQWidget(*napari\_viewer*,  
*smlm\_data*=<napari\_locan.data\_mo  
object>)

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

`__init__(napari_viewer[, smlm_data])`

---

`acceptDrops(self)`

---

`accessibleDescription(self)`

---

`accessibleName(self)`

---

`actionEvent(self, a0)`

---

`actions(self)`

---

`activateWindow(self)`

---

`addAction(self, action)`

---

`addActions(self, actions)`

---

`adjustSize(self)`

---

`autoFillBackground(self)`

---

`backgroundRole(self)`

---

continues on next page

Table 6 – continued from previous page

---

<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>
<code>depth(self)</code>

---

continues on next page

Table 6 – continued from previous page

---

<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>

---

continues on next page

Table 6 – continued from previous page

---

`focusNextPrevChild(self, next)`

---

`focusOutEvent(self, a0)`

---

`focusPolicy(self)`

---

`focusPreviousChild(self)`

---

`focusProxy(self)`

---

`focusWidget(self)`

---

`font(self)`

---

`fontInfo(self)`

---

`fontMetrics(self)`

---

`foregroundRole(self)`

---

`frameGeometry(self)`

---

`frameSize(self)`

---

`geometry(self)`

---

`getContentsMargins(self)`

---

`grab(self[, rectangle])`

---

`grabGesture(self, type[, flags])`

---

`grabKeyboard(self)`

---

`grabMouse()`

---

`grabShortcut(self, key[, context])`

---

`graphicsEffect(self)`

---

`graphicsProxyWidget(self)`

---

`hasFocus(self)`

---

`hasHeightForWidth(self)`

---

`hasMouseTracking(self)`

---

continues on next page

Table 6 – continued from previous page

---

hasTabletTracking(self)

---

height(self)

---

heightForWidth(self, a0)

---

heightMM(self)

---

hide(self)

---

hideEvent(self, a0)

---

inherits(self, classname)

---

initPainter(self, painter)

---

inputMethodEvent(self, a0)

---

inputMethodHints(self)

---

inputMethodQuery(self, a0)

---

insertAction(self, before, action)

---

insertActions(self, before, actions)

---

installEventFilter(self, a0)

---

isActiveWindow(self)

---

isAncestorOf(self, child)

---

isEnabled(self)

---

isEnabledTo(self, a0)

---

isFullScreen(self)

---

isHidden(self)

---

isLeftToRight(self)

---

isMaximized(self)

---

isMinimized(self)

---

isModal(self)

---

continues on next page



Table 6 – continued from previous page

---

`isRightToLeft(self)`

---

`isSignalConnected(self, signal)`

---

`isVisible(self)`

---

`isVisibleTo(self, a0)`

---

`isWidgetType(self)`

---

`isWindow(self)`

---

`isWindowModified(self)`

---

`isWindowType(self)`

---

`keyPressEvent(self, a0)`

---

`keyReleaseEvent(self, a0)`

---

`keyboardGrabber()`

---

`killTimer(self, id)`

---

`layout(self)`

---

`layoutDirection(self)`

---

`leaveEvent(self, a0)`

---

`locale(self)`

---

`logicalDpiX(self)`

---

`logicalDpiY(self)`

---

`lower(self)`

---

`mapFrom(self, a0, a1)`

---

`mapFromGlobal(self, a0)`

---

`mapFromParent(self, a0)`

---

`mapTo(self, a0, a1)`

---

`mapToGlobal(self, a0)`

---

continues on next page

Table 6 – continued from previous page

---

<code>mapToParent(self, a0)</code>
<code>mask(self)</code>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumWidth(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>mouseDoubleClickEvent(self, a0)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, a0)</code>
<code>mousePressEvent(self, a0)</code>
<code>mouseReleaseEvent(self, a0)</code>
<code>move()</code>
<code>moveEvent(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>
<code>nextInFocusChain(self)</code>
<code>normalGeometry(self)</code>
<code>objectName(self)</code>

---

continues on next page

Table 6 – continued from previous page

<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	

continues on next page

Table 6 – continued from previous page

---

`resize()`

---

`resizeEvent(self, a0)`

---

`restoreGeometry(self, geometry)`

---

`saveGeometry(self)`

---

`screen(self)`

---

`scroll()`

---

`sender(self)`

---

`senderSignalIndex(self)`

---

`setAcceptDrops(self, on)`

---

`setAccessibleDescription(self, descrip-  
tion)`

---

`setAccessibleName(self, name)`

---

`setAttribute(self, attribute[, on])`

---

`setAutoFillBackground(self, enabled)`

---

`setBackgroundRole(self, a0)`

---

`setBaseSize()`

---

`setContentsMargins()`

---

`setContextMenuPolicy(self, policy)`

---

`setCursor(self, a0)`

---

`setDisabled(self, a0)`

---

`setEnabled(self, a0)`

---

`setFixedHeight(self, h)`

---

`setFixedSize()`

---

`setFixedWidth(self, w)`

---

`setFocus()`

---

continues on next page

Table 6 – continued from previous page

---

`setFocusPolicy(self, policy)`

---

`setFocusProxy(self, a0)`

---

`setFont(self, a0)`

---

`setForegroundRole(self, a0)`

---

`setGeometry()`

---

`setGraphicsEffect(self, effect)`

---

`setHidden(self, hidden)`

---

`setInputMethodHints(self, hints)`

---

`setLayout(self, a0)`

---

`setLayoutDirection(self, direction)`

---

`setLocale(self, locale)`

---

`setMask()`

---

`setMaximumHeight(self, maxh)`

---

`setMaximumSize()`

---

`setMaximumWidth(self, maxw)`

---

`setMinimumHeight(self, minh)`

---

`setMinimumSize()`

---

`setMinimumWidth(self, minw)`

---

`setMouseTracking(self, enable)`

---

`setObjectName(self, name)`

---

`setPalette(self, a0)`

---

`setParent()`

---

`setProperty(self, name, value)`

---

`setShortcutAutoRepeat(self, id[, en-  
abled])`

---

continues on next page

Table 6 – continued from previous page

---

<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>

---

continues on next page

Table 6 – continued from previous page

---

<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>

---

continues on next page

Table 6 – continued from previous page

---

`underMouse(self)`

---

`ungrabGesture(self, type)`

---

`unsetCursor(self)`

---

`unsetLayoutDirection(self)`

---

`unsetLocale(self)`

---

`update(-> None -> None)`

---

`updateGeometry(self)`

---

`updateMicroFocus(self)`

---

`updatesEnabled(self)`

---

`visibleRegion(self)`

---

`whatsThis(self)`

---

`wheelEvent(self, a0)`

---

`width(self)`

---

`widthMM(self)`

---

`winId(self)`

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

`windowRole(self)`

---

continues on next page



Table 6 – continued from previous page

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

## napari\_locan.widgets.widget\_filter\_specifications

The data model for filter specifications.

A QWidget plugin to list filter specifications that can be applied to select localizations from a SMLM dataset.

### Classes

---

*FilterSpecificationsQWidget*(napari\_viewer[, ...])

---

## napari\_locan.widgets.widget\_filter\_specifications.FilterSpecificationsQWidget

**class** napari\_locan.widgets.widget\_filter\_specifications.**FilterSpecificationsQWidget**(*napari\_viewer*, *smlm\_data*, *object*>, *filter\_specification*, *object*>)

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

`__init__(napari_viewer[, smlm_data, ...])`

---

`acceptDrops(self)`

---

`accessibleDescription(self)`

---

`accessibleName(self)`

---

`actionEvent(self, a0)`

---

`actions(self)`

---

`activateWindow(self)`

---

`addAction(self, action)`

---

`addActions(self, actions)`

---

`adjustSize(self)`

---

continues on next page

Table 7 – continued from previous page

---

<code>autoFillBackground(self)</code>
<code>backgroundRole(self)</code>
<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>

---

continues on next page

Table 7 – continued from previous page

---

<code>deleteLater(self)</code>
<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>

---

continues on next page

Table 7 – continued from previous page

---

`focusInEvent(self, a0)`

---

`focusNextChild(self)`

---

`focusNextPrevChild(self, next)`

---

`focusOutEvent(self, a0)`

---

`focusPolicy(self)`

---

`focusPreviousChild(self)`

---

`focusProxy(self)`

---

`focusWidget(self)`

---

`font(self)`

---

`fontInfo(self)`

---

`fontMetrics(self)`

---

`foregroundRole(self)`

---

`frameGeometry(self)`

---

`frameSize(self)`

---

`geometry(self)`

---

`getContentsMargins(self)`

---

`grab(self[, rectangle])`

---

`grabGesture(self, type[, flags])`

---

`grabKeyboard(self)`

---

`grabMouse()`

---

`grabShortcut(self, key[, context])`

---

`graphicsEffect(self)`

---

`graphicsProxyWidget(self)`

---

`hasFocus(self)`

---

continues on next page

Table 7 – continued from previous page

---

hasHeightForWidth(self)

---

hasMouseTracking(self)

---

hasTabletTracking(self)

---

height(self)

---

heightForWidth(self, a0)

---

heightMM(self)

---

hide(self)

---

hideEvent(self, a0)

---

inherits(self, classname)

---

initPainter(self, painter)

---

inputMethodEvent(self, a0)

---

inputMethodHints(self)

---

inputMethodQuery(self, a0)

---

insertAction(self, before, action)

---

insertActions(self, before, actions)

---

installEventFilter(self, a0)

---

isActiveWindow(self)

---

isAncestorOf(self, child)

---

isEnabled(self)

---

isEnabledTo(self, a0)

---

isFullScreen(self)

---

isHidden(self)

---

isLeftToRight(self)

---

isMaximized(self)

---

continues on next page

Table 7 – continued from previous page

<code>isMinimized(self)</code>
<code>isModal(self)</code>
<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, signal)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, a0)</code>
<code>isWidgetType(self)</code>
<code>isWindow(self)</code>
<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>keyPressEvent(self, a0)</code>
<code>keyReleaseEvent(self, a0)</code>
<code>keyboardGrabber()</code>
<code>killTimer(self, id)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code>leaveEvent(self, a0)</code>
<code>locale(self)</code>
<code>logicalDpiX(self)</code>
<code>logicalDpiY(self)</code>
<code>lower(self)</code>
<code>mapFrom(self, a0, a1)</code>
<code>mapFromGlobal(self, a0)</code>
<code>mapFromParent(self, a0)</code>

continues on next page



Table 7 – continued from previous page

---

<code>mapTo(self, a0, a1)</code>
<code>mapToGlobal(self, a0)</code>
<code>mapToParent(self, a0)</code>
<code>mask(self)</code>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumWidth(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>mouseDoubleClickEvent(self, a0)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, a0)</code>
<code>mousePressEvent(self, a0)</code>
<code>mouseReleaseEvent(self, a0)</code>
<code>move()</code>
<code>moveEvent(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>
<code>nextInFocusChain(self)</code>

---

continues on next page

Table 7 – continued from previous page

<code>normalGeometry(self)</code>	
<code>objectName(self)</code>	
<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	

continues on next page

Table 7 – continued from previous page

---

`render(, sourceRegion, flags, ...)`

---

`repaint(-> None -> None)`

---

`resize()`

---

`resizeEvent(self, a0)`

---

`restoreGeometry(self, geometry)`

---

`saveGeometry(self)`

---

`screen(self)`

---

`scroll()`

---

`sender(self)`

---

`senderSignalIndex(self)`

---

`setAcceptDrops(self, on)`

---

`setAccessibleDescription(self, description)`

---

`setAccessibleName(self, name)`

---

`setAttribute(self, attribute[, on])`

---

`setAutoFillBackground(self, enabled)`

---

`setBackgroundRole(self, a0)`

---

`setBaseSize()`

---

`setContentsMargins()`

---

`setContextMenuPolicy(self, policy)`

---

`setCursor(self, a0)`

---

`setDisabled(self, a0)`

---

`setEnabled(self, a0)`

---

`setFixedHeight(self, h)`

---

`setFixedSize()`

---

continues on next page

Table 7 – continued from previous page

---

`setFixedWidth(self, w)`

---

`setFocus()`

---

`setFocusPolicy(self, policy)`

---

`setFocusProxy(self, a0)`

---

`setFont(self, a0)`

---

`setForegroundRole(self, a0)`

---

`setGeometry()`

---

`setGraphicsEffect(self, effect)`

---

`setHidden(self, hidden)`

---

`setInputMethodHints(self, hints)`

---

`setLayout(self, a0)`

---

`setLayoutDirection(self, direction)`

---

`setLocale(self, locale)`

---

`setMask()`

---

`setMaximumHeight(self, maxh)`

---

`setMaximumSize()`

---

`setMaximumWidth(self, maxw)`

---

`setMinimumHeight(self, minh)`

---

`setMinimumSize()`

---

`setMinimumWidth(self, minw)`

---

`setMouseTracking(self, enable)`

---

`setObjectName(self, name)`

---

`setPalette(self, a0)`

---

`setParent()`

---

continues on next page

Table 7 – continued from previous page

---

<code>setProperty(self, name, value)</code>
<code>setShortcutAutoRepeat(self, id[, enabled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>

---

continues on next page

Table 7 – continued from previous page

---

<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>
<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>

---

continues on next page

Table 7 – continued from previous page

---

<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>
<code>underMouse(self)</code>
<code>ungrabGesture(self, type)</code>
<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>
<code>update(-&gt; None -&gt; None)</code>
<code>updateGeometry(self)</code>
<code>updateMicroFocus(self)</code>
<code>updatesEnabled(self)</code>
<code>visibleRegion(self)</code>
<code>whatsThis(self)</code>
<code>wheelEvent(self, a0)</code>
<code>width(self)</code>
<code>widthMM(self)</code>
<code>winId(self)</code>
<code>window(self)</code>
<code>windowFilePath(self)</code>
<code>windowFlags(self)</code>
<code>windowHandle(self)</code>
<code>windowIcon(self)</code>
<code>windowIconText(self)</code>
<code>windowModality(self)</code>

---

continues on next page

Table 7 – continued from previous page

---

<code>windowOpacity(self)</code>
<code>windowRole(self)</code>
<code>windowState(self)</code>
<code>windowTitle(self)</code>
<code>windowType(self)</code>
<code>x(self)</code>
<code>y(self)</code>

---



## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

## napari\_locan.widgets.widget\_load

Load SMLM data files.

A QWidget plugin to load SMLM data files into the SMLM data model. A new SMLM dataset will be created.

### Classes

---

*LoadQWidget*(napari\_viewer[, smlm\_data])

---

## napari\_locan.widgets.widget\_load.LoadQWidget

**class** napari\_locan.widgets.widget\_load.**LoadQWidget**(*napari\_viewer*,  
*smlm\_data*=<napari\_locan.data\_model.smlm\_data.  
*object*>)

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

`__init__(napari_viewer[, smlm_data])`

---

`acceptDrops(self)`

---

`accessibleDescription(self)`

---

`accessibleName(self)`

---

`actionEvent(self, a0)`

---

`actions(self)`

---

`activateWindow(self)`

---

`addAction(self, action)`

---

`addActions(self, actions)`

---

`adjustSize(self)`

---

`autoFillBackground(self)`

---

`backgroundRole(self)`

---

continues on next page

Table 8 – continued from previous page

---

<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>
<code>depth(self)</code>

---

continues on next page

Table 8 – continued from previous page

---

<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>

---

continues on next page

Table 8 – continued from previous page

---

`focusNextPrevChild(self, next)`

---

`focusOutEvent(self, a0)`

---

`focusPolicy(self)`

---

`focusPreviousChild(self)`

---

`focusProxy(self)`

---

`focusWidget(self)`

---

`font(self)`

---

`fontInfo(self)`

---

`fontMetrics(self)`

---

`foregroundRole(self)`

---

`frameGeometry(self)`

---

`frameSize(self)`

---

`geometry(self)`

---

`getContentsMargins(self)`

---

`grab(self[, rectangle])`

---

`grabGesture(self, type[, flags])`

---

`grabKeyboard(self)`

---

`grabMouse()`

---

`grabShortcut(self, key[, context])`

---

`graphicsEffect(self)`

---

`graphicsProxyWidget(self)`

---

`hasFocus(self)`

---

`hasHeightForWidth(self)`

---

`hasMouseTracking(self)`

---

continues on next page

Table 8 – continued from previous page

---

`hasTabletTracking(self)`

---

`height(self)`

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

`isAncestorOf(self, child)`

---

`isEnabled(self)`

---

`isEnabledTo(self, a0)`

---

`isFullScreen(self)`

---

`isHidden(self)`

---

`isLeftToRight(self)`

---

`isMaximized(self)`

---

`isMinimized(self)`

---

`isModal(self)`

---

continues on next page

Table 8 – continued from previous page

---

<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, signal)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, a0)</code>
<code>isWidgetType(self)</code>
<code>isWindow(self)</code>
<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>keyPressEvent(self, a0)</code>
<code>keyReleaseEvent(self, a0)</code>
<code>keyboardGrabber()</code>
<code>killTimer(self, id)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code>leaveEvent(self, a0)</code>
<code>locale(self)</code>
<code>logicalDpiX(self)</code>
<code>logicalDpiY(self)</code>
<code>lower(self)</code>
<code>mapFrom(self, a0, a1)</code>
<code>mapFromGlobal(self, a0)</code>
<code>mapFromParent(self, a0)</code>
<code>mapTo(self, a0, a1)</code>
<code>mapToGlobal(self, a0)</code>

---

continues on next page

Table 8 – continued from previous page

---

`mapToParent(self, a0)`

---

`mask(self)`

---

`maximumHeight(self)`

---

`maximumSize(self)`

---

`maximumWidth(self)`

---

`metaObject(self)`

---

`metric(self, a0)`

---

`minimumHeight(self)`

---

`minimumSize(self)`

---

`minimumSizeHint(self)`

---

`minimumWidth(self)`

---

`mouseDoubleClickEvent(self, a0)`

---

`mouseGrabber()`

---

`mouseMoveEvent(self, a0)`

---

`mousePressEvent(self, a0)`

---

`mouseReleaseEvent(self, a0)`

---

`move()`

---

`moveEvent(self, a0)`

---

`moveToThread(self, thread)`

---

`nativeEvent(self, eventType, message)`

---

`nativeParentWidget(self)`

---

`nextInFocusChain(self)`

---

`normalGeometry(self)`

---

`objectName(self)`

---

continues on next page



Table 8 – continued from previous page

<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	

continues on next page

Table 8 – continued from previous page

---

`resize()`

---

`resizeEvent(self, a0)`

---

`restoreGeometry(self, geometry)`

---

`saveGeometry(self)`

---

`screen(self)`

---

`scroll()`

---

`sender(self)`

---

`senderSignalIndex(self)`

---

`setAcceptDrops(self, on)`

---

`setAccessibleDescription(self, descrip-  
tion)`

---

`setAccessibleName(self, name)`

---

`setAttribute(self, attribute[, on])`

---

`setAutoFillBackground(self, enabled)`

---

`setBackgroundRole(self, a0)`

---

`setBaseSize()`

---

`setContentsMargins()`

---

`setContextMenuPolicy(self, policy)`

---

`setCursor(self, a0)`

---

`setDisabled(self, a0)`

---

`setEnabled(self, a0)`

---

`setFixedHeight(self, h)`

---

`setFixedSize()`

---

`setFixedWidth(self, w)`

---

`setFocus()`

---

continues on next page

Table 8 – continued from previous page

---

`setFocusPolicy(self, policy)`

---

`setFocusProxy(self, a0)`

---

`setFont(self, a0)`

---

`setForegroundRole(self, a0)`

---

`setGeometry()`

---

`setGraphicsEffect(self, effect)`

---

`setHidden(self, hidden)`

---

`setInputMethodHints(self, hints)`

---

`setLayout(self, a0)`

---

`setLayoutDirection(self, direction)`

---

`setLocale(self, locale)`

---

`setMask()`

---

`setMaximumHeight(self, maxh)`

---

`setMaximumSize()`

---

`setMaximumWidth(self, maxw)`

---

`setMinimumHeight(self, minh)`

---

`setMinimumSize()`

---

`setMinimumWidth(self, minw)`

---

`setMouseTracking(self, enable)`

---

`setObjectName(self, name)`

---

`setPalette(self, a0)`

---

`setParent()`

---

`setProperty(self, name, value)`

---

`setShortcutAutoRepeat(self, id[, en-  
abled])`

---

continues on next page

Table 8 – continued from previous page

---

<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>

---

continues on next page

Table 8 – continued from previous page

---

<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>

---

continues on next page

Table 8 – continued from previous page

---

<code>underMouse(self)</code>
<code>ungrabGesture(self, type)</code>
<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>
<code>update(-&gt; None -&gt; None)</code>
<code>updateGeometry(self)</code>
<code>updateMicroFocus(self)</code>
<code>updatesEnabled(self)</code>
<code>visibleRegion(self)</code>
<code>whatsThis(self)</code>
<code>wheelEvent(self, a0)</code>
<code>width(self)</code>
<code>widthMM(self)</code>
<code>winId(self)</code>
<code>window(self)</code>
<code>windowFilePath(self)</code>
<code>windowFlags(self)</code>
<code>windowHandle(self)</code>
<code>windowIcon(self)</code>
<code>windowIconText(self)</code>
<code>windowModality(self)</code>
<code>windowOpacity(self)</code>
<code>windowRole(self)</code>

---

continues on next page

Table 8 – continued from previous page

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL



## napari\_locan.widgets.widget\_napari\_locan\_project

Save and load the current state of napari-locan.

QWidget plugin to save and load the napari-locan state, which currently includes the following data models

- 1) filter\_specifications
- 2) region\_specifications
- 3) roi\_specifications
- 4) smlm\_data

The data is serialized by the pickle module using protocol 5.

### Classes

---

*NapariLocanProjectQWidget*(napari\_viewer[, ...])

---

## napari\_locan.widgets.widget\_napari\_locan\_project.NapariLocanProjectQWidget

```
class napari_locan.widgets.widget_napari_locan_project.NapariLocanProjectQWidget(napari_viewer,
    filter_specifications,
    region_specifications,
    roi_specifications,
    smlm_data)
```

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

`__init__(napari_viewer[, ...])`

---

`acceptDrops(self)`

---

`accessibleDescription(self)`

---

continues on next page

Table 9 – continued from previous page

---

<code>accessibleName(self)</code>
<code>actionEvent(self, a0)</code>
<code>actions(self)</code>
<code>activateWindow(self)</code>
<code>addAction(self, action)</code>
<code>addActions(self, actions)</code>
<code>adjustSize(self)</code>
<code>autoFillBackground(self)</code>
<code>backgroundRole(self)</code>
<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>

---

continues on next page

Table 9 – continued from previous page

<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>
<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySubWindows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>

continues on next page

Table 9 – continued from previous page

<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>
<code>focusNextPrevChild(self, next)</code>
<code>focusOutEvent(self, a0)</code>
<code>focusPolicy(self)</code>
<code>focusPreviousChild(self)</code>
<code>focusProxy(self)</code>
<code>focusWidget(self)</code>
<code>font(self)</code>
<code>fontInfo(self)</code>
<code>fontMetrics(self)</code>
<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>
<code>frameSize(self)</code>
<code>geometry(self)</code>
<code>getContentsMargins(self)</code>
<code>grab(self[, rectangle])</code>

continues on next page

Table 9 – continued from previous page

---

`grabGesture(self, type[, flags])`

---

`grabKeyboard(self)`

---

`grabMouse()`

---

`grabShortcut(self, key[, context])`

---

`graphicsEffect(self)`

---

`graphicsProxyWidget(self)`

---

`hasFocus(self)`

---

`hasHeightForWidth(self)`

---

`hasMouseTracking(self)`

---

`hasTabletTracking(self)`

---

`height(self)`

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

continues on next page

Table 9 – continued from previous page

<code>isAncestorOf(self, child)</code>
<code>isEnabled(self)</code>
<code>isEnabledTo(self, a0)</code>
<code>isFullScreen(self)</code>
<code>isHidden(self)</code>
<code>isLeftToRight(self)</code>
<code>isMaximized(self)</code>
<code>isMinimized(self)</code>
<code>isModal(self)</code>
<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, signal)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, a0)</code>
<code>isWidgetType(self)</code>
<code>isWindow(self)</code>
<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>keyPressEvent(self, a0)</code>
<code>keyReleaseEvent(self, a0)</code>
<code>keyboardGrabber()</code>
<code>killTimer(self, id)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code>leaveEvent(self, a0)</code>

continues on next page

Table 9 – continued from previous page

---

`locale(self)`

---

`logicalDpiX(self)`

---

`logicalDpiY(self)`

---

`lower(self)`

---

`mapFrom(self, a0, a1)`

---

`mapFromGlobal(self, a0)`

---

`mapFromParent(self, a0)`

---

`mapTo(self, a0, a1)`

---

`mapToGlobal(self, a0)`

---

`mapToParent(self, a0)`

---

`mask(self)`

---

`maximumHeight(self)`

---

`maximumSize(self)`

---

`maximumWidth(self)`

---

`metaObject(self)`

---

`metric(self, a0)`

---

`minimumHeight(self)`

---

`minimumSize(self)`

---

`minimumSizeHint(self)`

---

`minimumWidth(self)`

---

`mouseDoubleClickEvent(self, a0)`

---

`mouseGrabber()`

---

`mouseMoveEvent(self, a0)`

---

`mousePressEvent(self, a0)`

---

continues on next page

Table 9 – continued from previous page

<code>mousePressEvent(self, a0)</code>	
<code>move()</code>	
<code>moveEvent(self, a0)</code>	
<code>moveToThread(self, thread)</code>	
<code>nativeEvent(self, eventType, message)</code>	
<code>nativeParentWidget(self)</code>	
<code>nextInFocusChain(self)</code>	
<code>normalGeometry(self)</code>	
<code>objectName(self)</code>	
<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	

continues on next page



Table 9 – continued from previous page

---

<code>receivers(self, signal)</code>
<code>rect(self)</code>
<code>releaseKeyboard(self)</code>
<code>releaseMouse(self)</code>
<code>releaseShortcut(self, id)</code>
<code>removeAction(self, action)</code>
<code>removeEventFilter(self, a0)</code>
<code>render(, sourceRegion, flags, ...)</code>
<code>repaint(-&gt; None -&gt; None)</code>
<code>resize()</code>
<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>sender(self)</code>
<code>senderSignalIndex(self)</code>
<code>setAcceptDrops(self, on)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>

---

continues on next page

Table 9 – continued from previous page

---

`setContentMargins()`

---

`setContextMenuPolicy(self, policy)`

---

`setCursor(self, a0)`

---

`setDisabled(self, a0)`

---

`setEnabled(self, a0)`

---

`setFixedHeight(self, h)`

---

`setFixedSize()`

---

`setFixedWidth(self, w)`

---

`setFocus()`

---

`setFocusPolicy(self, policy)`

---

`setFocusProxy(self, a0)`

---

`setFont(self, a0)`

---

`setForegroundRole(self, a0)`

---

`setGeometry()`

---

`setGraphicsEffect(self, effect)`

---

`setHidden(self, hidden)`

---

`setInputMethodHints(self, hints)`

---

`setLayout(self, a0)`

---

`setLayoutDirection(self, direction)`

---

`setLocale(self, locale)`

---

`setMask()`

---

`setMaximumHeight(self, maxh)`

---

`setMaximumSize()`

---

`setMaximumWidth(self, maxw)`

---

continues on next page

Table 9 – continued from previous page

---

<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setMouseTracking(self, enable)</code>
<code>setObjectName(self, name)</code>
<code>setPalette(self, a0)</code>
<code>setParent()</code>
<code>setProperty(self, name, value)</code>
<code>setShortcutAutoRepeat(self, id[, en- abled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>

---

continues on next page

Table 9 – continued from previous page

<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>
<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>

continues on next page

Table 9 – continued from previous page

---

<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>
<code>underMouse(self)</code>
<code>ungrabGesture(self, type)</code>
<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>
<code>update(-&gt; None -&gt; None)</code>
<code>updateGeometry(self)</code>
<code>updateMicroFocus(self)</code>
<code>updatesEnabled(self)</code>
<code>visibleRegion(self)</code>
<code>whatsThis(self)</code>
<code>wheelEvent(self, a0)</code>
<code>width(self)</code>
<code>widthMM(self)</code>
<code>winId(self)</code>

---

continues on next page

Table 9 – continued from previous page

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

`windowRole(self)`

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

---

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

---

## napari\_locan.widgets.widget\_property\_distribution

Show localization property distribution.

A QWidget plugin to show localization property distributions

### Classes

---

*PropertyDistributionQWidget*(napari\_viewer[,  
...])

---

## napari\_locan.widgets.widget\_property\_distribution.PropertyDistributionQWidget

**class** napari\_locan.widgets.widget\_property\_distribution.**PropertyDistributionQWidget**(*napari\_viewer*,  
*smlm\_data*,  
*object*>,  
*parent*=None)

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

`__init__(napari_viewer[, smlm_data, parent])`

---

`acceptDrops(self)`

---

`accessibleDescription(self)`

---

`accessibleName(self)`

---

`actionEvent(self, a0)`

---

`actions(self)`

---

`activateWindow(self)`

---

`addAction(self, action)`

---

`addActions(self, actions)`

---

`adjustSize(self)`

---

`autoFillBackground(self)`

---

continues on next page



Table 10 – continued from previous page

---

<code>backgroundRole(self)</code>
<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>

---

continues on next page

Table 10 – continued from previous page

---

<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>

---

continues on next page

Table 10 – continued from previous page

---

`focusNextChild(self)`

---

`focusNextPrevChild(self, next)`

---

`focusOutEvent(self, a0)`

---

`focusPolicy(self)`

---

`focusPreviousChild(self)`

---

`focusProxy(self)`

---

`focusWidget(self)`

---

`font(self)`

---

`fontInfo(self)`

---

`fontMetrics(self)`

---

`foregroundRole(self)`

---

`frameGeometry(self)`

---

`frameSize(self)`

---

`geometry(self)`

---

`getContentsMargins(self)`

---

`grab(self[, rectangle])`

---

`grabGesture(self, type[, flags])`

---

`grabKeyboard(self)`

---

`grabMouse()`

---

`grabShortcut(self, key[, context])`

---

`graphicsEffect(self)`

---

`graphicsProxyWidget(self)`

---

`hasFocus(self)`

---

`hasHeightForWidth(self)`

---

continues on next page

Table 10 – continued from previous page

---

`hasMouseTracking(self)`

---

`hasTabletTracking(self)`

---

`height(self)`

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

`isAncestorOf(self, child)`

---

`isEnabled(self)`

---

`isEnabledTo(self, a0)`

---

`isFullScreen(self)`

---

`isHidden(self)`

---

`isLeftToRight(self)`

---

`isMaximized(self)`

---

`isMinimized(self)`

---

continues on next page

Table 10 – continued from previous page

---

`isModal(self)`

---

`isRightToLeft(self)`

---

`isSignalConnected(self, signal)`

---

`isVisible(self)`

---

`isVisibleTo(self, a0)`

---

`isWidgetType(self)`

---

`isWindow(self)`

---

`isWindowModified(self)`

---

`isWindowType(self)`

---

`keyPressEvent(self, a0)`

---

`keyReleaseEvent(self, a0)`

---

`keyboardGrabber()`

---

`killTimer(self, id)`

---

`layout(self)`

---

`layoutDirection(self)`

---

`leaveEvent(self, a0)`

---

`locale(self)`

---

`logicalDpiX(self)`

---

`logicalDpiY(self)`

---

`lower(self)`

---

`mapFrom(self, a0, a1)`

---

`mapFromGlobal(self, a0)`

---

`mapFromParent(self, a0)`

---

`mapTo(self, a0, a1)`

---

continues on next page

Table 10 – continued from previous page

---

<code>mapToGlobal(self, a0)</code>
<code>mapToParent(self, a0)</code>
<code>mask(self)</code>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumWidth(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>mouseDoubleClickEvent(self, a0)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, a0)</code>
<code>mousePressEvent(self, a0)</code>
<code>mouseReleaseEvent(self, a0)</code>
<code>move()</code>
<code>moveEvent(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>
<code>nextInFocusChain(self)</code>
<code>normalGeometry(self)</code>

---

continues on next page

Table 10 – continued from previous page

<code>objectName(self)</code>	
<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	

continues on next page

Table 10 – continued from previous page

<code>repaint(-&gt; None -&gt; None)</code>
<code>resize()</code>
<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>sender(self)</code>
<code>senderSignalIndex(self)</code>
<code>setAcceptDrops(self, on)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setEnabled(self, a0)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>

continues on next page



Table 10 – continued from previous page

---

`setFocus()`

---

`setFocusPolicy(self, policy)`

---

`setFocusProxy(self, a0)`

---

`setFont(self, a0)`

---

`setForegroundRole(self, a0)`

---

`setGeometry()`

---

`setGraphicsEffect(self, effect)`

---

`setHidden(self, hidden)`

---

`setInputMethodHints(self, hints)`

---

`setLayout(self, a0)`

---

`setLayoutDirection(self, direction)`

---

`setLocale(self, locale)`

---

`setMask()`

---

`setMaximumHeight(self, maxh)`

---

`setMaximumSize()`

---

`setMaximumWidth(self, maxw)`

---

`setMinimumHeight(self, minh)`

---

`setMinimumSize()`

---

`setMinimumWidth(self, minw)`

---

`setMouseTracking(self, enable)`

---

`setObjectName(self, name)`

---

`setPalette(self, a0)`

---

`setParent()`

---

`setProperty(self, name, value)`

---

continues on next page

Table 10 – continued from previous page

---

<code>setShortcutAutoRepeat(self, id[, enabled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>

---

continues on next page

Table 10 – continued from previous page

---

<code>setWindowTitle(self, a0)</code>
<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>

---

continues on next page

Table 10 – continued from previous page

<code>tr(self, sourceText[, disambiguation, n])</code>
<code>underMouse(self)</code>
<code>ungrabGesture(self, type)</code>
<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>
<code>update(-&gt; None -&gt; None)</code>
<code>updateGeometry(self)</code>
<code>updateMicroFocus(self)</code>
<code>updatesEnabled(self)</code>
<code>visibleRegion(self)</code>
<code>whatIsThis(self)</code>
<code>wheelEvent(self, a0)</code>
<code>width(self)</code>
<code>widthMM(self)</code>
<code>winId(self)</code>
<code>window(self)</code>
<code>windowFilePath(self)</code>
<code>windowFlags(self)</code>
<code>windowHandle(self)</code>
<code>windowIcon(self)</code>
<code>windowIconText(self)</code>
<code>windowModality(self)</code>
<code>windowOpacity(self)</code>

continues on next page

Table 10 – continued from previous page

---

`windowRole(self)`

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

---

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

---

## napari\_locan.widgets.widget\_render\_collection\_2d

Render collection of SMLM data as 2d point clouds.

A QWidget plugin to render a collection of SMLM data (e.g. a cluster collection) as series of point clouds in 2d.

### Classes

---

*RenderCollection2dQWidget*(napari\_viewer[,  
...])

---

## napari\_locan.widgets.widget\_render\_collection\_2d.RenderCollection2dQWidget

**class** napari\_locan.widgets.widget\_render\_collection\_2d.RenderCollection2dQWidget(*napari\_viewer*, *smlm\_data*=None) *object*

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

`__init__(napari_viewer[, smlm_data])`

---

`acceptDrops(self)`

---

`accessibleDescription(self)`

---

`accessibleName(self)`

---

`actionEvent(self, a0)`

---

`actions(self)`

---

`activateWindow(self)`

---

`addAction(self, action)`

---

`addActions(self, actions)`

---

`adjustSize(self)`

---

`autoFillBackground(self)`

---

`backgroundRole(self)`

---

continues on next page

Table 11 – continued from previous page

---

`baseSize(self)`

---

`blockSignals(self, b)`

---

`changeEvent(self, a0)`

---

`childAt(-> Optional[QWidget])`

---

`childEvent(self, a0)`

---

`children(self)`

---

`childrenRect(self)`

---

`childrenRegion(self)`

---

`clearFocus(self)`

---

`clearMask(self)`

---

`close(self)`

---

`closeEvent(self, a0)`

---

`colorCount(self)`

---

`connectNotify(self, signal)`

---

`contentsMargins(self)`

---

`contentsRect(self)`

---

`contextMenuEvent(self, a0)`

---

`contextMenuPolicy(self)`

---

`create(self[, window, initializeWindow, ...])`

---

`createWindowContainer(window[, parent,  
flags])`

---

`cursor(self)`

---

`customEvent(self, a0)`

---

`deleteLater(self)`

---

`depth(self)`

---

continues on next page



Table 11 – continued from previous page

---

<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>

---

continues on next page

Table 11 – continued from previous page

---

`focusNextPrevChild(self, next)`

---

`focusOutEvent(self, a0)`

---

`focusPolicy(self)`

---

`focusPreviousChild(self)`

---

`focusProxy(self)`

---

`focusWidget(self)`

---

`font(self)`

---

`fontInfo(self)`

---

`fontMetrics(self)`

---

`foregroundRole(self)`

---

`frameGeometry(self)`

---

`frameSize(self)`

---

`geometry(self)`

---

`getContentsMargins(self)`

---

`grab(self[, rectangle])`

---

`grabGesture(self, type[, flags])`

---

`grabKeyboard(self)`

---

`grabMouse()`

---

`grabShortcut(self, key[, context])`

---

`graphicsEffect(self)`

---

`graphicsProxyWidget(self)`

---

`hasFocus(self)`

---

`hasHeightForWidth(self)`

---

`hasMouseTracking(self)`

---

continues on next page

Table 11 – continued from previous page

---

`hasTabletTracking(self)`

---

`height(self)`

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

`isAncestorOf(self, child)`

---

`isEnabled(self)`

---

`isEnabledTo(self, a0)`

---

`isFullScreen(self)`

---

`isHidden(self)`

---

`isLeftToRight(self)`

---

`isMaximized(self)`

---

`isMinimized(self)`

---

`isModal(self)`

---

continues on next page

Table 11 – continued from previous page

---

`isRightToLeft(self)`

---

`isSignalConnected(self, signal)`

---

`isVisible(self)`

---

`isVisibleTo(self, a0)`

---

`isWidgetType(self)`

---

`isWindow(self)`

---

`isWindowModified(self)`

---

`isWindowType(self)`

---

`keyPressEvent(self, a0)`

---

`keyReleaseEvent(self, a0)`

---

`keyboardGrabber()`

---

`killTimer(self, id)`

---

`layout(self)`

---

`layoutDirection(self)`

---

`leaveEvent(self, a0)`

---

`locale(self)`

---

`logicalDpiX(self)`

---

`logicalDpiY(self)`

---

`lower(self)`

---

`mapFrom(self, a0, a1)`

---

`mapFromGlobal(self, a0)`

---

`mapFromParent(self, a0)`

---

`mapTo(self, a0, a1)`

---

`mapToGlobal(self, a0)`

---

continues on next page

Table 11 – continued from previous page

<code>mapToParent(self, a0)</code>
<code>mask(self)</code>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumWidth(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>mouseDoubleClickEvent(self, a0)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, a0)</code>
<code>mousePressEvent(self, a0)</code>
<code>mouseReleaseEvent(self, a0)</code>
<code>move()</code>
<code>moveEvent(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>
<code>nextInFocusChain(self)</code>
<code>normalGeometry(self)</code>
<code>objectName(self)</code>

continues on next page

Table 11 – continued from previous page

<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	

continues on next page

Table 11 – continued from previous page

---

<code>resize()</code>
<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>sender(self)</code>
<code>senderSignalIndex(self)</code>
<code>setAcceptDrops(self, on)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setEnabled(self, a0)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>
<code>setFocus()</code>

---

continues on next page

Table 11 – continued from previous page

---

<code>setFocusPolicy(self, policy)</code>
<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setForegroundRole(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setHidden(self, hidden)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLocale(self, locale)</code>
<code>setMask()</code>
<code>setMaximumHeight(self, maxh)</code>
<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setMouseTracking(self, enable)</code>
<code>setObjectName(self, name)</code>
<code>setPalette(self, a0)</code>
<code>setParent()</code>
<code>setProperty(self, name, value)</code>
<code>setShortcutAutoRepeat(self, id[, en- abled])</code>

---

continues on next page



Table 11 – continued from previous page

---

<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>

---

continues on next page

Table 11 – continued from previous page

---

<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>

---

continues on next page

Table 11 – continued from previous page

---

`underMouse(self)`

---

`ungrabGesture(self, type)`

---

`unsetCursor(self)`

---

`unsetLayoutDirection(self)`

---

`unsetLocale(self)`

---

`update(-> None -> None)`

---

`updateGeometry(self)`

---

`updateMicroFocus(self)`

---

`updatesEnabled(self)`

---

`visibleRegion(self)`

---

`whatsThis(self)`

---

`wheelEvent(self, a0)`

---

`width(self)`

---

`widthMM(self)`

---

`winId(self)`

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

`windowRole(self)`

---

continues on next page

Table 11 – continued from previous page

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

## napari\_locan.widgets.widget\_render\_collection\_features

Render selected features of all SMLM datasets in a collection.

A QWidget plugin to represent collection features including centroid, bounding box, oriented bounding box, convex hull and alpha shape. The SMLM datasets must be kept in a Locdata collection (locdata.references).

### Classes

---

*RenderCollectionFeaturesQWidget*(napari\_viewer)

---

## napari\_locan.widgets.widget\_render\_collection\_features.RenderCollectionFeaturesQWidget

**class** napari\_locan.widgets.widget\_render\_collection\_features.RenderCollectionFeaturesQWidget

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

`__init__(napari_viewer[, smlm_data])`

---

`acceptDrops(self)`

---

`accessibleDescription(self)`

---

`accessibleName(self)`

---

`actionEvent(self, a0)`

---

`actions(self)`

---

`activateWindow(self)`

---

`addAction(self, action)`

---

`addActions(self, actions)`

---

`adjustSize(self)`

---

`autoFillBackground(self)`

---

continues on next page

Table 12 – continued from previous page

---

<code>backgroundRole(self)</code>
<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>

---

continues on next page

Table 12 – continued from previous page

---

<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>

---

continues on next page



Table 12 – continued from previous page

---

<code>focusNextChild(self)</code>
<code>focusNextPrevChild(self, next)</code>
<code>focusOutEvent(self, a0)</code>
<code>focusPolicy(self)</code>
<code>focusPreviousChild(self)</code>
<code>focusProxy(self)</code>
<code>focusWidget(self)</code>
<code>font(self)</code>
<code>fontInfo(self)</code>
<code>fontMetrics(self)</code>
<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>
<code>frameSize(self)</code>
<code>geometry(self)</code>
<code>getContentsMargins(self)</code>
<code>grab(self[, rectangle])</code>
<code>grabGesture(self, type[, flags])</code>
<code>grabKeyboard(self)</code>
<code>grabMouse()</code>
<code>grabShortcut(self, key[, context])</code>
<code>graphicsEffect(self)</code>
<code>graphicsProxyWidget(self)</code>
<code>hasFocus(self)</code>
<code>hasHeightForWidth(self)</code>

---

continues on next page

Table 12 – continued from previous page

---

`hasMouseTracking(self)`

---

`hasTabletTracking(self)`

---

`height(self)`

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

`isAncestorOf(self, child)`

---

`isEnabled(self)`

---

`isEnabledTo(self, a0)`

---

`isFullScreen(self)`

---

`isHidden(self)`

---

`isLeftToRight(self)`

---

`isMaximized(self)`

---

`isMinimized(self)`

---

continues on next page

Table 12 – continued from previous page

---

`isModal(self)`

---

`isRightToLeft(self)`

---

`isSignalConnected(self, signal)`

---

`isVisible(self)`

---

`isVisibleTo(self, a0)`

---

`isWidgetType(self)`

---

`isWindow(self)`

---

`isWindowModified(self)`

---

`isWindowType(self)`

---

`keyPressEvent(self, a0)`

---

`keyReleaseEvent(self, a0)`

---

`keyboardGrabber()`

---

`killTimer(self, id)`

---

`layout(self)`

---

`layoutDirection(self)`

---

`leaveEvent(self, a0)`

---

`locale(self)`

---

`logicalDpiX(self)`

---

`logicalDpiY(self)`

---

`lower(self)`

---

`mapFrom(self, a0, a1)`

---

`mapFromGlobal(self, a0)`

---

`mapFromParent(self, a0)`

---

`mapTo(self, a0, a1)`

---

continues on next page

Table 12 – continued from previous page

<code>mapToGlobal(self, a0)</code>
<code>mapToParent(self, a0)</code>
<code>mask(self)</code>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumWidth(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>mouseDoubleClickEvent(self, a0)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, a0)</code>
<code>mousePressEvent(self, a0)</code>
<code>mouseReleaseEvent(self, a0)</code>
<code>move()</code>
<code>moveEvent(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>
<code>nextInFocusChain(self)</code>
<code>normalGeometry(self)</code>

continues on next page

Table 12 – continued from previous page

<code>objectName(self)</code>	
<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	

continues on next page

Table 12 – continued from previous page

---

`repaint(-> None -> None)`

---

`resize()`

---

`resizeEvent(self, a0)`

---

`restoreGeometry(self, geometry)`

---

`saveGeometry(self)`

---

`screen(self)`

---

`scroll()`

---

`sender(self)`

---

`senderSignalIndex(self)`

---

`setAcceptDrops(self, on)`

---

`setAccessibleDescription(self, description)`

---

`setAccessibleName(self, name)`

---

`setAttribute(self, attribute[, on])`

---

`setAutoFillBackground(self, enabled)`

---

`setBackgroundRole(self, a0)`

---

`setBaseSize()`

---

`setContentsMargins()`

---

`setContextMenuPolicy(self, policy)`

---

`setCursor(self, a0)`

---

`setDisabled(self, a0)`

---

`setEnabled(self, a0)`

---

`setFixedHeight(self, h)`

---

`setFixedSize()`

---

`setFixedWidth(self, w)`

---

continues on next page

Table 12 – continued from previous page

---

`setFocus()`

---

`setFocusPolicy(self, policy)`

---

`setFocusProxy(self, a0)`

---

`setFont(self, a0)`

---

`setForegroundRole(self, a0)`

---

`setGeometry()`

---

`setGraphicsEffect(self, effect)`

---

`setHidden(self, hidden)`

---

`setInputMethodHints(self, hints)`

---

`setLayout(self, a0)`

---

`setLayoutDirection(self, direction)`

---

`setLocale(self, locale)`

---

`setMask()`

---

`setMaximumHeight(self, maxh)`

---

`setMaximumSize()`

---

`setMaximumWidth(self, maxw)`

---

`setMinimumHeight(self, minh)`

---

`setMinimumSize()`

---

`setMinimumWidth(self, minw)`

---

`setMouseTracking(self, enable)`

---

`setObjectName(self, name)`

---

`setPalette(self, a0)`

---

`setParent()`

---

`setProperty(self, name, value)`

---

continues on next page

Table 12 – continued from previous page

---

<code>setShortcutAutoRepeat(self, id[, enabled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>

---

continues on next page



Table 12 – continued from previous page

---

<code>setWindowTitle(self, a0)</code>
<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>

---

continues on next page

Table 12 – continued from previous page

---

`tr(self, sourceText[, disambiguation, n])`

---

`underMouse(self)`

---

`ungrabGesture(self, type)`

---

`unsetCursor(self)`

---

`unsetLayoutDirection(self)`

---

`unsetLocale(self)`

---

`update(-> None -> None)`

---

`updateGeometry(self)`

---

`updateMicroFocus(self)`

---

`updatesEnabled(self)`

---

`visibleRegion(self)`

---

`whatIsThis(self)`

---

`wheelEvent(self, a0)`

---

`width(self)`

---

`widthMM(self)`

---

`winId(self)`

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

continues on next page

Table 12 – continued from previous page

---

`windowRole(self)`

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

## napari\_locan.widgets.widget\_render\_features

Render selected features of a SMLM dataset.

A QWidget plugin to represent locdata features including centroid, bounding box, oriented bounding box, convex hull and alpha shape.

### Classes

---

*RenderFeaturesQWidget*(napari\_viewer[,  
smlm\_data])

---

## napari\_locan.widgets.widget\_render\_features.RenderFeaturesQWidget

**class** napari\_locan.widgets.widget\_render\_features.**RenderFeaturesQWidget**(*napari\_viewer*,  
*smlm\_data*=<napari\_locan.widgets.widget\_render\_features.RenderFeaturesQWidget object>)

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

`__init__(napari_viewer[, smlm_data])`

---

`acceptDrops(self)`

---

`accessibleDescription(self)`

---

`accessibleName(self)`

---

`actionEvent(self, a0)`

---

`actions(self)`

---

`activateWindow(self)`

---

`addAction(self, action)`

---

`addActions(self, actions)`

---

`adjustSize(self)`

---

`autoFillBackground(self)`

---

`backgroundRole(self)`

---

continues on next page

Table 13 – continued from previous page

---

`baseSize(self)`

---

`blockSignals(self, b)`

---

`changeEvent(self, a0)`

---

`childAt(-> Optional[QWidget])`

---

`childEvent(self, a0)`

---

`children(self)`

---

`childrenRect(self)`

---

`childrenRegion(self)`

---

`clearFocus(self)`

---

`clearMask(self)`

---

`close(self)`

---

`closeEvent(self, a0)`

---

`colorCount(self)`

---

`connectNotify(self, signal)`

---

`contentsMargins(self)`

---

`contentsRect(self)`

---

`contextMenuEvent(self, a0)`

---

`contextMenuPolicy(self)`

---

`create(self[, window, initializeWindow, ...])`

---

`createWindowContainer(window[, parent,  
flags])`

---

`cursor(self)`

---

`customEvent(self, a0)`

---

`deleteLater(self)`

---

`depth(self)`

---

continues on next page

Table 13 – continued from previous page

---

<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>

---

continues on next page

Table 13 – continued from previous page

---

`focusNextPrevChild(self, next)`

---

`focusOutEvent(self, a0)`

---

`focusPolicy(self)`

---

`focusPreviousChild(self)`

---

`focusProxy(self)`

---

`focusWidget(self)`

---

`font(self)`

---

`fontInfo(self)`

---

`fontMetrics(self)`

---

`foregroundRole(self)`

---

`frameGeometry(self)`

---

`frameSize(self)`

---

`geometry(self)`

---

`getContentsMargins(self)`

---

`grab(self[, rectangle])`

---

`grabGesture(self, type[, flags])`

---

`grabKeyboard(self)`

---

`grabMouse()`

---

`grabShortcut(self, key[, context])`

---

`graphicsEffect(self)`

---

`graphicsProxyWidget(self)`

---

`hasFocus(self)`

---

`hasHeightForWidth(self)`

---

`hasMouseTracking(self)`

---

continues on next page



Table 13 – continued from previous page

---

`hasTabletTracking(self)`

---

`height(self)`

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

`isAncestorOf(self, child)`

---

`isEnabled(self)`

---

`isEnabledTo(self, a0)`

---

`isFullScreen(self)`

---

`isHidden(self)`

---

`isLeftToRight(self)`

---

`isMaximized(self)`

---

`isMinimized(self)`

---

`isModal(self)`

---

continues on next page

Table 13 – continued from previous page

---

`isRightToLeft(self)`

---

`isSignalConnected(self, signal)`

---

`isVisible(self)`

---

`isVisibleTo(self, a0)`

---

`isWidgetType(self)`

---

`isWindow(self)`

---

`isWindowModified(self)`

---

`isWindowType(self)`

---

`keyPressEvent(self, a0)`

---

`keyReleaseEvent(self, a0)`

---

`keyboardGrabber()`

---

`killTimer(self, id)`

---

`layout(self)`

---

`layoutDirection(self)`

---

`leaveEvent(self, a0)`

---

`locale(self)`

---

`logicalDpiX(self)`

---

`logicalDpiY(self)`

---

`lower(self)`

---

`mapFrom(self, a0, a1)`

---

`mapFromGlobal(self, a0)`

---

`mapFromParent(self, a0)`

---

`mapTo(self, a0, a1)`

---

`mapToGlobal(self, a0)`

---

continues on next page

Table 13 – continued from previous page

---

`mapToParent(self, a0)`

---

`mask(self)`

---

`maximumHeight(self)`

---

`maximumSize(self)`

---

`maximumWidth(self)`

---

`metaObject(self)`

---

`metric(self, a0)`

---

`minimumHeight(self)`

---

`minimumSize(self)`

---

`minimumSizeHint(self)`

---

`minimumWidth(self)`

---

`mouseDoubleClickEvent(self, a0)`

---

`mouseGrabber()`

---

`mouseMoveEvent(self, a0)`

---

`mousePressEvent(self, a0)`

---

`mouseReleaseEvent(self, a0)`

---

`move()`

---

`moveEvent(self, a0)`

---

`moveToThread(self, thread)`

---

`nativeEvent(self, eventType, message)`

---

`nativeParentWidget(self)`

---

`nextInFocusChain(self)`

---

`normalGeometry(self)`

---

`objectName(self)`

---

continues on next page

Table 13 – continued from previous page

<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	

continues on next page

Table 13 – continued from previous page

---

<code>resize()</code>
<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>sender(self)</code>
<code>senderSignalIndex(self)</code>
<code>setAcceptDrops(self, on)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setEnabled(self, a0)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>
<code>setFocus()</code>

---

continues on next page

Table 13 – continued from previous page

---

<code>setFocusPolicy(self, policy)</code>
<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setForegroundRole(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setHidden(self, hidden)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLocale(self, locale)</code>
<code>setMask()</code>
<code>setMaximumHeight(self, maxh)</code>
<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setMouseTracking(self, enable)</code>
<code>setObjectName(self, name)</code>
<code>setPalette(self, a0)</code>
<code>setParent()</code>
<code>setProperty(self, name, value)</code>
<code>setShortcutAutoRepeat(self, id[, en- abled])</code>

---

continues on next page

Table 13 – continued from previous page

---

`setShortcutEnabled(self, id[, enabled])`

---

`setSizeIncrement()`

---

`setSizePolicy()`

---

`setStatusTip(self, a0)`

---

`setStyle(self, a0)`

---

`setStyleSheet(self, styleSheet)`

---

`setTabOrder(a0, a1)`

---

`setTabletTracking(self, enable)`

---

`setToolTip(self, a0)`

---

`setToolTipDuration(self, msec)`

---

`setUpdatesEnabled(self, enable)`

---

`setVisible(self, visible)`

---

`setWhatsThis(self, a0)`

---

`setWindowFilePath(self, filePath)`

---

`setWindowFlag(self, a0[, on])`

---

`setWindowFlags(self, type)`

---

`setWindowIcon(self, icon)`

---

`setWindowIconText(self, a0)`

---

`setWindowModality(self, windowModal-  
ity)`

---

`setWindowModified(self, a0)`

---

`setWindowOpacity(self, level)`

---

`setWindowRole(self, a0)`

---

`setWindowState(self, state)`

---

`setWindowTitle(self, a0)`

---

continues on next page

Table 13 – continued from previous page

<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>

continues on next page



Table 13 – continued from previous page

---

`underMouse(self)`

---

`ungrabGesture(self, type)`

---

`unsetCursor(self)`

---

`unsetLayoutDirection(self)`

---

`unsetLocale(self)`

---

`update(-> None -> None)`

---

`updateGeometry(self)`

---

`updateMicroFocus(self)`

---

`updatesEnabled(self)`

---

`visibleRegion(self)`

---

`whatsThis(self)`

---

`wheelEvent(self, a0)`

---

`width(self)`

---

`widthMM(self)`

---

`winId(self)`

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

`windowRole(self)`

---

continues on next page

Table 13 – continued from previous page

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

---

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

---

## napari\_locan.widgets.widget\_render\_image\_2d

Render 2d image.

A QWidget plugin to render SMLM data as image by binning localization properties into 2d pixels.

### Classes

---

```
RenderImage2dQWidget(napari_viewer[,  
smlm_data])
```

---

## napari\_locan.widgets.widget\_render\_image\_2d.RenderImage2dQWidget

```
class napari_locan.widgets.widget_render_image_2d.RenderImage2dQWidget(napari_viewer,  
                                                                    smlm_data=<napari_locan.  
                                                                    object>)
```

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

```
__init__(napari_viewer[, smlm_data])
```

---

```
acceptDrops(self)
```

---

```
accessibleDescription(self)
```

---

```
accessibleName(self)
```

---

```
actionEvent(self, a0)
```

---

```
actions(self)
```

---

```
activateWindow(self)
```

---

```
addAction(self, action)
```

---

```
addActions(self, actions)
```

---

```
adjustSize(self)
```

---

```
autoFillBackground(self)
```

---

```
backgroundRole(self)
```

---

```
baseSize(self)
```

---

continues on next page

Table 14 – continued from previous page

---

`blockSignals(self, b)`

---

`changeEvent(self, a0)`

---

`childAt(-> Optional[QWidget])`

---

`childEvent(self, a0)`

---

`children(self)`

---

`childrenRect(self)`

---

`childrenRegion(self)`

---

`clearFocus(self)`

---

`clearMask(self)`

---

`close(self)`

---

`closeEvent(self, a0)`

---

`colorCount(self)`

---

`connectNotify(self, signal)`

---

`contentsMargins(self)`

---

`contentsRect(self)`

---

`contextMenuEvent(self, a0)`

---

`contextMenuPolicy(self)`

---

`create(self[, window, initializeWindow, ...])`

---

`createWindowContainer(window[, parent,  
flags])`

---

`cursor(self)`

---

`customEvent(self, a0)`

---

`deleteLater(self)`

---

`depth(self)`

---

`destroy(self[, destroyWindow, destroySub-  
Windows])`

---

continues on next page

Table 14 – continued from previous page

---

<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>
<code>focusNextPrevChild(self, next)</code>

---

continues on next page

Table 14 – continued from previous page

---

<code>focusOutEvent(self, a0)</code>
<code>focusPolicy(self)</code>
<code>focusPreviousChild(self)</code>
<code>focusProxy(self)</code>
<code>focusWidget(self)</code>
<code>font(self)</code>
<code>fontInfo(self)</code>
<code>fontMetrics(self)</code>
<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>
<code>frameSize(self)</code>
<code>geometry(self)</code>
<code>getContentsMargins(self)</code>
<code>grab(self[, rectangle])</code>
<code>grabGesture(self, type[, flags])</code>
<code>grabKeyboard(self)</code>
<code>grabMouse()</code>
<code>grabShortcut(self, key[, context])</code>
<code>graphicsEffect(self)</code>
<code>graphicsProxyWidget(self)</code>
<code>hasFocus(self)</code>
<code>hasHeightForWidth(self)</code>
<code>hasMouseTracking(self)</code>
<code>hasTabletTracking(self)</code>

---

continues on next page

Table 14 – continued from previous page

---

<code>height(self)</code>
<code>heightForWidth(self, a0)</code>
<code>heightMM(self)</code>
<code>hide(self)</code>
<code>hideEvent(self, a0)</code>
<code>inherits(self, classname)</code>
<code>initPainter(self, painter)</code>
<code>inputMethodEvent(self, a0)</code>
<code>inputMethodHints(self)</code>
<code>inputMethodQuery(self, a0)</code>
<code>insertAction(self, before, action)</code>
<code>insertActions(self, before, actions)</code>
<code>installEventFilter(self, a0)</code>
<code>isActiveWindow(self)</code>
<code>isAncestorOf(self, child)</code>
<code>isEnabled(self)</code>
<code>isEnabledTo(self, a0)</code>
<code>isFullScreen(self)</code>
<code>isHidden(self)</code>
<code>isLeftToRight(self)</code>
<code>isMaximized(self)</code>
<code>isMinimized(self)</code>
<code>isModal(self)</code>
<code>isRightToLeft(self)</code>

---

continues on next page



Table 14 – continued from previous page

---

isSignalConnected(self, signal)

---

isVisible(self)

---

isVisibleTo(self, a0)

---

isWidgetType(self)

---

isWindow(self)

---

isWindowModified(self)

---

isWindowType(self)

---

keyPressEvent(self, a0)

---

keyReleaseEvent(self, a0)

---

keyboardGrabber()

---

killTimer(self, id)

---

layout(self)

---

layoutDirection(self)

---

leaveEvent(self, a0)

---

locale(self)

---

logicalDpiX(self)

---

logicalDpiY(self)

---

lower(self)

---

mapFrom(self, a0, a1)

---

mapFromGlobal(self, a0)

---

mapFromParent(self, a0)

---

mapTo(self, a0, a1)

---

mapToGlobal(self, a0)

---

mapToParent(self, a0)

---

continues on next page

Table 14 – continued from previous page

---

<code>mask(self)</code>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumWidth(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>mouseDoubleClickEvent(self, a0)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, a0)</code>
<code>mousePressEvent(self, a0)</code>
<code>mouseReleaseEvent(self, a0)</code>
<code>move()</code>
<code>moveEvent(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>
<code>nextInFocusChain(self)</code>
<code>normalGeometry(self)</code>
<code>objectName(self)</code>
<code>overrideWindowFlags(self, type)</code>

---

continues on next page

Table 14 – continued from previous page

<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	
<code>resize()</code>	

continues on next page

Table 14 – continued from previous page

---

<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>sender(self)</code>
<code>senderSignalIndex(self)</code>
<code>setAcceptDrops(self, on)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setEnabled(self, a0)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>
<code>setFocus()</code>
<code>setFocusPolicy(self, policy)</code>

---

continues on next page

Table 14 – continued from previous page

---

<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setForegroundRole(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setHidden(self, hidden)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLocale(self, locale)</code>
<code>setMask()</code>
<code>setMaximumHeight(self, maxh)</code>
<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setMouseTracking(self, enable)</code>
<code>setObjectName(self, name)</code>
<code>setPalette(self, a0)</code>
<code>setParent()</code>
<code>setProperty(self, name, value)</code>
<code>setShortcutAutoRepeat(self, id[, en- abled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>

---

continues on next page

Table 14 – continued from previous page

---

<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModal- ity)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>
<code>sharedPainter(self)</code>

---

continues on next page

Table 14 – continued from previous page

---

<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>
<code>underMouse(self)</code>

---

continues on next page

Table 14 – continued from previous page

---

`ungrabGesture(self, type)`

---

`unsetCursor(self)`

---

`unsetLayoutDirection(self)`

---

`unsetLocale(self)`

---

`update(-> None -> None)`

---

`updateGeometry(self)`

---

`updateMicroFocus(self)`

---

`updatesEnabled(self)`

---

`visibleRegion(self)`

---

`whatsThis(self)`

---

`wheelEvent(self, a0)`

---

`width(self)`

---

`widthMM(self)`

---

`winId(self)`

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

`windowRole(self)`

---

`windowState(self)`

---

continues on next page



Table 14 – continued from previous page

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

## napari\_locan.widgets.widget\_render\_image\_3d

Render 3d image.

A QWidget plugin to render SMLM data as image by binning localization properties into 3d pixels.

### Classes

---

```
RenderImage3dQWidget(napari_viewer[,  
smlm_data])
```

---

## napari\_locan.widgets.widget\_render\_image\_3d.RenderImage3dQWidget

```
class napari_locan.widgets.widget_render_image_3d.RenderImage3dQWidget(napari_viewer,  
                                                                    smlm_data=<napari_locan.  
                                                                    object>)
```

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

```
__init__(napari_viewer[, smlm_data])
```

---

```
acceptDrops(self)
```

---

```
accessibleDescription(self)
```

---

```
accessibleName(self)
```

---

```
actionEvent(self, a0)
```

---

```
actions(self)
```

---

```
activateWindow(self)
```

---

```
addAction(self, action)
```

---

```
addActions(self, actions)
```

---

```
adjustSize(self)
```

---

```
autoFillBackground(self)
```

---

```
backgroundRole(self)
```

---

```
baseSize(self)
```

---

continues on next page

Table 15 – continued from previous page

---

<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>
<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySub- Windows])</code>

---

continues on next page

Table 15 – continued from previous page

---

<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>
<code>focusNextPrevChild(self, next)</code>

---

continues on next page

Table 15 – continued from previous page

---

<code>focusOutEvent(self, a0)</code>
<code>focusPolicy(self)</code>
<code>focusPreviousChild(self)</code>
<code>focusProxy(self)</code>
<code>focusWidget(self)</code>
<code>font(self)</code>
<code>fontInfo(self)</code>
<code>fontMetrics(self)</code>
<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>
<code>frameSize(self)</code>
<code>geometry(self)</code>
<code>getContentsMargins(self)</code>
<code>grab(self[, rectangle])</code>
<code>grabGesture(self, type[, flags])</code>
<code>grabKeyboard(self)</code>
<code>grabMouse()</code>
<code>grabShortcut(self, key[, context])</code>
<code>graphicsEffect(self)</code>
<code>graphicsProxyWidget(self)</code>
<code>hasFocus(self)</code>
<code>hasHeightForWidth(self)</code>
<code>hasMouseTracking(self)</code>
<code>hasTabletTracking(self)</code>

---

continues on next page

Table 15 – continued from previous page

---

`height(self)`

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

`isAncestorOf(self, child)`

---

`isEnabled(self)`

---

`isEnabledTo(self, a0)`

---

`isFullScreen(self)`

---

`isHidden(self)`

---

`isLeftToRight(self)`

---

`isMaximized(self)`

---

`isMinimized(self)`

---

`isModal(self)`

---

`isRightToLeft(self)`

---

continues on next page

Table 15 – continued from previous page

---

`isSignalConnected(self, signal)`

---

`isVisible(self)`

---

`isVisibleTo(self, a0)`

---

`isWidgetType(self)`

---

`isWindow(self)`

---

`isWindowModified(self)`

---

`isWindowType(self)`

---

`keyPressEvent(self, a0)`

---

`keyReleaseEvent(self, a0)`

---

`keyboardGrabber()`

---

`killTimer(self, id)`

---

`layout(self)`

---

`layoutDirection(self)`

---

`leaveEvent(self, a0)`

---

`locale(self)`

---

`logicalDpiX(self)`

---

`logicalDpiY(self)`

---

`lower(self)`

---

`mapFrom(self, a0, a1)`

---

`mapFromGlobal(self, a0)`

---

`mapFromParent(self, a0)`

---

`mapTo(self, a0, a1)`

---

`mapToGlobal(self, a0)`

---

`mapToParent(self, a0)`

---

continues on next page



Table 15 – continued from previous page

---

<code>mask(self)</code>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumWidth(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>mouseDoubleClickEvent(self, a0)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, a0)</code>
<code>mousePressEvent(self, a0)</code>
<code>mouseReleaseEvent(self, a0)</code>
<code>move()</code>
<code>moveEvent(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>
<code>nextInFocusChain(self)</code>
<code>normalGeometry(self)</code>
<code>objectName(self)</code>
<code>overrideWindowFlags(self, type)</code>

---

continues on next page

Table 15 – continued from previous page

<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	
<code>resize()</code>	

continues on next page

Table 15 – continued from previous page

---

`resizeEvent(self, a0)`

---

`restoreGeometry(self, geometry)`

---

`saveGeometry(self)`

---

`screen(self)`

---

`scroll()`

---

`sender(self)`

---

`senderSignalIndex(self)`

---

`setAcceptDrops(self, on)`

---

`setAccessibleDescription(self, description)`

---

`setAccessibleName(self, name)`

---

`setAttribute(self, attribute[, on])`

---

`setAutoFillBackground(self, enabled)`

---

`setBackgroundRole(self, a0)`

---

`setBaseSize()`

---

`setContentsMargins()`

---

`setContextMenuPolicy(self, policy)`

---

`setCursor(self, a0)`

---

`setDisabled(self, a0)`

---

`setEnabled(self, a0)`

---

`setFixedHeight(self, h)`

---

`setFixedSize()`

---

`setFixedWidth(self, w)`

---

`setFocus()`

---

`setFocusPolicy(self, policy)`

---

continues on next page

Table 15 – continued from previous page

---

`setFocusProxy(self, a0)`

---

`setFont(self, a0)`

---

`setForegroundRole(self, a0)`

---

`setGeometry()`

---

`setGraphicsEffect(self, effect)`

---

`setHidden(self, hidden)`

---

`setInputMethodHints(self, hints)`

---

`setLayout(self, a0)`

---

`setLayoutDirection(self, direction)`

---

`setLocale(self, locale)`

---

`setMask()`

---

`setMaximumHeight(self, maxh)`

---

`setMaximumSize()`

---

`setMaximumWidth(self, maxw)`

---

`setMinimumHeight(self, minh)`

---

`setMinimumSize()`

---

`setMinimumWidth(self, minw)`

---

`setMouseTracking(self, enable)`

---

`setObjectName(self, name)`

---

`setPalette(self, a0)`

---

`setParent()`

---

`setProperty(self, name, value)`

---

`setShortcutAutoRepeat(self, id[, en-  
abled])`

---

`setShortcutEnabled(self, id[, enabled])`

---

continues on next page

Table 15 – continued from previous page

---

<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModal- ity)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>
<code>sharedPainter(self)</code>

---

continues on next page

Table 15 – continued from previous page

---

<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>
<code>underMouse(self)</code>

---

continues on next page

Table 15 – continued from previous page

---

`ungrabGesture(self, type)`

---

`unsetCursor(self)`

---

`unsetLayoutDirection(self)`

---

`unsetLocale(self)`

---

`update(-> None -> None)`

---

`updateGeometry(self)`

---

`updateMicroFocus(self)`

---

`updatesEnabled(self)`

---

`visibleRegion(self)`

---

`whatsThis(self)`

---

`wheelEvent(self, a0)`

---

`width(self)`

---

`widthMM(self)`

---

`winId(self)`

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

`windowRole(self)`

---

`windowState(self)`

---

continues on next page

Table 15 – continued from previous page

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---



## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

## napari\_locan.widgets.widget\_render\_points\_2d

Render 2d point cloud.

A QWidget plugin to render SMLM data in 2d.

### Classes

---

```
RenderPoints2dQWidget(napari_viewer[,
smlm_data])
```

---

## napari\_locan.widgets.widget\_render\_points\_2d.RenderPoints2dQWidget

```
class napari_locan.widgets.widget_render_points_2d.RenderPoints2dQWidget(napari_viewer,
                                                                    smlm_data=<napari_lo
                                                                    ob-
                                                                    ject>)
```

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

```
__init__(napari_viewer[, smlm_data])
```

---

```
acceptDrops(self)
```

---

```
accessibleDescription(self)
```

---

```
accessibleName(self)
```

---

```
actionEvent(self, a0)
```

---

```
actions(self)
```

---

```
activateWindow(self)
```

---

```
addAction(self, action)
```

---

```
addActions(self, actions)
```

---

```
adjustSize(self)
```

---

```
autoFillBackground(self)
```

---

```
backgroundRole(self)
```

---

continues on next page

Table 16 – continued from previous page

---

`baseSize(self)`

---

`blockSignals(self, b)`

---

`changeEvent(self, a0)`

---

`childAt(-> Optional[QWidget])`

---

`childEvent(self, a0)`

---

`children(self)`

---

`childrenRect(self)`

---

`childrenRegion(self)`

---

`clearFocus(self)`

---

`clearMask(self)`

---

`close(self)`

---

`closeEvent(self, a0)`

---

`colorCount(self)`

---

`connectNotify(self, signal)`

---

`contentsMargins(self)`

---

`contentsRect(self)`

---

`contextMenuEvent(self, a0)`

---

`contextMenuPolicy(self)`

---

`create(self[, window, initializeWindow, ...])`

---

`createWindowContainer(window[, parent,  
flags])`

---

`cursor(self)`

---

`customEvent(self, a0)`

---

`deleteLater(self)`

---

`depth(self)`

---

continues on next page

Table 16 – continued from previous page

<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>

continues on next page

Table 16 – continued from previous page

---

<code>focusNextPrevChild(self, next)</code>
<code>focusOutEvent(self, a0)</code>
<code>focusPolicy(self)</code>
<code>focusPreviousChild(self)</code>
<code>focusProxy(self)</code>
<code>focusWidget(self)</code>
<code>font(self)</code>
<code>fontInfo(self)</code>
<code>fontMetrics(self)</code>
<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>
<code>frameSize(self)</code>
<code>geometry(self)</code>
<code>getContentsMargins(self)</code>
<code>grab(self[, rectangle])</code>
<code>grabGesture(self, type[, flags])</code>
<code>grabKeyboard(self)</code>
<code>grabMouse()</code>
<code>grabShortcut(self, key[, context])</code>
<code>graphicsEffect(self)</code>
<code>graphicsProxyWidget(self)</code>
<code>hasFocus(self)</code>
<code>hasHeightForWidth(self)</code>
<code>hasMouseTracking(self)</code>

---

continues on next page

Table 16 – continued from previous page

---

`hasTabletTracking(self)`

---

`height(self)`

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

`isAncestorOf(self, child)`

---

`isEnabled(self)`

---

`isEnabledTo(self, a0)`

---

`isFullScreen(self)`

---

`isHidden(self)`

---

`isLeftToRight(self)`

---

`isMaximized(self)`

---

`isMinimized(self)`

---

`isModal(self)`

---

continues on next page

Table 16 – continued from previous page

---

<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, signal)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, a0)</code>
<code>isWidgetType(self)</code>
<code>isWindow(self)</code>
<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>keyPressEvent(self, a0)</code>
<code>keyReleaseEvent(self, a0)</code>
<code>keyboardGrabber()</code>
<code>killTimer(self, id)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code>leaveEvent(self, a0)</code>
<code>locale(self)</code>
<code>logicalDpiX(self)</code>
<code>logicalDpiY(self)</code>
<code>lower(self)</code>
<code>mapFrom(self, a0, a1)</code>
<code>mapFromGlobal(self, a0)</code>
<code>mapFromParent(self, a0)</code>
<code>mapTo(self, a0, a1)</code>
<code>mapToGlobal(self, a0)</code>

---

continues on next page

Table 16 – continued from previous page

---

<code>mapToParent(self, a0)</code>
<code>mask(self)</code>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumWidth(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>mouseDoubleClickEvent(self, a0)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, a0)</code>
<code>mousePressEvent(self, a0)</code>
<code>mouseReleaseEvent(self, a0)</code>
<code>move()</code>
<code>moveEvent(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>
<code>nextInFocusChain(self)</code>
<code>normalGeometry(self)</code>
<code>objectName(self)</code>

---

continues on next page



Table 16 – continued from previous page

<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	

continues on next page

Table 16 – continued from previous page

---

<code>resize()</code>
<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>sender(self)</code>
<code>senderSignalIndex(self)</code>
<code>setAcceptDrops(self, on)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setEnabled(self, a0)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>
<code>setFocus()</code>

---

continues on next page

Table 16 – continued from previous page

---

`setFocusPolicy(self, policy)`

---

`setFocusProxy(self, a0)`

---

`setFont(self, a0)`

---

`setForegroundRole(self, a0)`

---

`setGeometry()`

---

`setGraphicsEffect(self, effect)`

---

`setHidden(self, hidden)`

---

`setInputMethodHints(self, hints)`

---

`setLayout(self, a0)`

---

`setLayoutDirection(self, direction)`

---

`setLocale(self, locale)`

---

`setMask()`

---

`setMaximumHeight(self, maxh)`

---

`setMaximumSize()`

---

`setMaximumWidth(self, maxw)`

---

`setMinimumHeight(self, minh)`

---

`setMinimumSize()`

---

`setMinimumWidth(self, minw)`

---

`setMouseTracking(self, enable)`

---

`setObjectName(self, name)`

---

`setPalette(self, a0)`

---

`setParent()`

---

`setProperty(self, name, value)`

---

`setShortcutAutoRepeat(self, id[, en-  
abled])`

---

continues on next page

Table 16 – continued from previous page

---

<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>

---

continues on next page

Table 16 – continued from previous page

---

<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>

---

continues on next page

Table 16 – continued from previous page

---

<code>underMouse(self)</code>
<code>ungrabGesture(self, type)</code>
<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>
<code>update(-&gt; None -&gt; None)</code>
<code>updateGeometry(self)</code>
<code>updateMicroFocus(self)</code>
<code>updatesEnabled(self)</code>
<code>visibleRegion(self)</code>
<code>whatsThis(self)</code>
<code>wheelEvent(self, a0)</code>
<code>width(self)</code>
<code>widthMM(self)</code>
<code>winId(self)</code>
<code>window(self)</code>
<code>windowFilePath(self)</code>
<code>windowFlags(self)</code>
<code>windowHandle(self)</code>
<code>windowIcon(self)</code>
<code>windowIconText(self)</code>
<code>windowModality(self)</code>
<code>windowOpacity(self)</code>
<code>windowRole(self)</code>

---

continues on next page

Table 16 – continued from previous page

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL



## napari\_locan.widgets.widget\_render\_points\_3d

Render 3d point cloud.

A QWidget plugin to render SMLM data in 3d.

### Classes

---

```
RenderPoints3dQWidget(napari_viewer[,  
smlm_data])
```

---

## napari\_locan.widgets.widget\_render\_points\_3d.RenderPoints3dQWidget

```
class napari_locan.widgets.widget_render_points_3d.RenderPoints3dQWidget(napari_viewer,  
                                                                    smlm_data=<napari_lo  
                                                                    ob-  
                                                                    ject>)
```

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

```
__init__(napari_viewer[, smlm_data])
```

---

```
acceptDrops(self)
```

---

```
accessibleDescription(self)
```

---

```
accessibleName(self)
```

---

```
actionEvent(self, a0)
```

---

```
actions(self)
```

---

```
activateWindow(self)
```

---

```
addAction(self, action)
```

---

```
addActions(self, actions)
```

---

```
adjustSize(self)
```

---

```
autoFillBackground(self)
```

---

```
backgroundRole(self)
```

---

continues on next page

Table 17 – continued from previous page

---

<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>
<code>depth(self)</code>

---

continues on next page

Table 17 – continued from previous page

---

<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>

---

continues on next page

Table 17 – continued from previous page

---

`focusNextPrevChild(self, next)`

---

`focusOutEvent(self, a0)`

---

`focusPolicy(self)`

---

`focusPreviousChild(self)`

---

`focusProxy(self)`

---

`focusWidget(self)`

---

`font(self)`

---

`fontInfo(self)`

---

`fontMetrics(self)`

---

`foregroundRole(self)`

---

`frameGeometry(self)`

---

`frameSize(self)`

---

`geometry(self)`

---

`getContentsMargins(self)`

---

`grab(self[, rectangle])`

---

`grabGesture(self, type[, flags])`

---

`grabKeyboard(self)`

---

`grabMouse()`

---

`grabShortcut(self, key[, context])`

---

`graphicsEffect(self)`

---

`graphicsProxyWidget(self)`

---

`hasFocus(self)`

---

`hasHeightForWidth(self)`

---

`hasMouseTracking(self)`

---

continues on next page

Table 17 – continued from previous page

---

hasTabletTracking(self)

---

height(self)

---

heightForWidth(self, a0)

---

heightMM(self)

---

hide(self)

---

hideEvent(self, a0)

---

inherits(self, classname)

---

initPainter(self, painter)

---

inputMethodEvent(self, a0)

---

inputMethodHints(self)

---

inputMethodQuery(self, a0)

---

insertAction(self, before, action)

---

insertActions(self, before, actions)

---

installEventFilter(self, a0)

---

isActiveWindow(self)

---

isAncestorOf(self, child)

---

isEnabled(self)

---

isEnabledTo(self, a0)

---

isFullScreen(self)

---

isHidden(self)

---

isLeftToRight(self)

---

isMaximized(self)

---

isMinimized(self)

---

isModal(self)

---

continues on next page

Table 17 – continued from previous page

---

<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, signal)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, a0)</code>
<code>isWidgetType(self)</code>
<code>isWindow(self)</code>
<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>keyPressEvent(self, a0)</code>
<code>keyReleaseEvent(self, a0)</code>
<code>keyboardGrabber()</code>
<code>killTimer(self, id)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code>leaveEvent(self, a0)</code>
<code>locale(self)</code>
<code>logicalDpiX(self)</code>
<code>logicalDpiY(self)</code>
<code>lower(self)</code>
<code>mapFrom(self, a0, a1)</code>
<code>mapFromGlobal(self, a0)</code>
<code>mapFromParent(self, a0)</code>
<code>mapTo(self, a0, a1)</code>
<code>mapToGlobal(self, a0)</code>

---

continues on next page

Table 17 – continued from previous page

---

`mapToParent(self, a0)`

---

`mask(self)`

---

`maximumHeight(self)`

---

`maximumSize(self)`

---

`maximumWidth(self)`

---

`metaObject(self)`

---

`metric(self, a0)`

---

`minimumHeight(self)`

---

`minimumSize(self)`

---

`minimumSizeHint(self)`

---

`minimumWidth(self)`

---

`mouseDoubleClickEvent(self, a0)`

---

`mouseGrabber()`

---

`mouseMoveEvent(self, a0)`

---

`mousePressEvent(self, a0)`

---

`mouseReleaseEvent(self, a0)`

---

`move()`

---

`moveEvent(self, a0)`

---

`moveToThread(self, thread)`

---

`nativeEvent(self, eventType, message)`

---

`nativeParentWidget(self)`

---

`nextInFocusChain(self)`

---

`normalGeometry(self)`

---

`objectName(self)`

---

continues on next page

Table 17 – continued from previous page

<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	

continues on next page



Table 17 – continued from previous page

---

`resize()`

---

`resizeEvent(self, a0)`

---

`restoreGeometry(self, geometry)`

---

`saveGeometry(self)`

---

`screen(self)`

---

`scroll()`

---

`sender(self)`

---

`senderSignalIndex(self)`

---

`setAcceptDrops(self, on)`

---

`setAccessibleDescription(self, descrip-  
tion)`

---

`setAccessibleName(self, name)`

---

`setAttribute(self, attribute[, on])`

---

`setAutoFillBackground(self, enabled)`

---

`setBackgroundRole(self, a0)`

---

`setBaseSize()`

---

`setContentsMargins()`

---

`setContextMenuPolicy(self, policy)`

---

`setCursor(self, a0)`

---

`setDisabled(self, a0)`

---

`setEnabled(self, a0)`

---

`setFixedHeight(self, h)`

---

`setFixedSize()`

---

`setFixedWidth(self, w)`

---

`setFocus()`

---

continues on next page

Table 17 – continued from previous page

---

<code>setFocusPolicy(self, policy)</code>
<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setForegroundRole(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setHidden(self, hidden)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLocale(self, locale)</code>
<code>setMask()</code>
<code>setMaximumHeight(self, maxh)</code>
<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setMouseTracking(self, enable)</code>
<code>setObjectName(self, name)</code>
<code>setPalette(self, a0)</code>
<code>setParent()</code>
<code>setProperty(self, name, value)</code>
<code>setShortcutAutoRepeat(self, id[, en- abled])</code>

---

continues on next page

Table 17 – continued from previous page

---

<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModal- ity)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>

---

continues on next page

Table 17 – continued from previous page

---

<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>

---

continues on next page

Table 17 – continued from previous page

---

`underMouse(self)`

---

`ungrabGesture(self, type)`

---

`unsetCursor(self)`

---

`unsetLayoutDirection(self)`

---

`unsetLocale(self)`

---

`update(-> None -> None)`

---

`updateGeometry(self)`

---

`updateMicroFocus(self)`

---

`updatesEnabled(self)`

---

`visibleRegion(self)`

---

`whatsThis(self)`

---

`wheelEvent(self, a0)`

---

`width(self)`

---

`widthMM(self)`

---

`winId(self)`

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

`windowRole(self)`

---

continues on next page

Table 17 – continued from previous page

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

---

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

---

## napari\_locan.widgets.widget\_roi

Create regions of interest.

A QWidget plugin for managing regions of interest.

### Classes

---

*RoiQWidget*(napari\_viewer[, ...])

---

## napari\_locan.widgets.widget\_roi.RoiQWidget

**class** napari\_locan.widgets.widget\_roi.**RoiQWidget**(*napari\_viewer*, *region\_specifications=<napari\_locan.data\_model.region\_specifications\_specification object>*, *roi\_specifications=<napari\_locan.data\_model.roi\_specification object>*, *smlm\_data=<napari\_locan.data\_model.smlm\_data.SmlmData object>*)

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

`__init__`(napari\_viewer[, ...])

---

`acceptDrops`(self)

---

`accessibleDescription`(self)

---

`accessibleName`(self)

---

`actionEvent`(self, a0)

---

`actions`(self)

---

`activateWindow`(self)

---

`addAction`(self, action)

---

`addActions`(self, actions)

---

`adjustSize`(self)

---

`autoFillBackground`(self)

---

continues on next page



Table 18 – continued from previous page

---

<code>backgroundRole(self)</code>
<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>

---

continues on next page

Table 18 – continued from previous page

---

<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>

---

continues on next page

Table 18 – continued from previous page

---

<code>focusNextChild(self)</code>
<code>focusNextPrevChild(self, next)</code>
<code>focusOutEvent(self, a0)</code>
<code>focusPolicy(self)</code>
<code>focusPreviousChild(self)</code>
<code>focusProxy(self)</code>
<code>focusWidget(self)</code>
<code>font(self)</code>
<code>fontInfo(self)</code>
<code>fontMetrics(self)</code>
<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>
<code>frameSize(self)</code>
<code>geometry(self)</code>
<code>getContentsMargins(self)</code>
<code>grab(self[, rectangle])</code>
<code>grabGesture(self, type[, flags])</code>
<code>grabKeyboard(self)</code>
<code>grabMouse()</code>
<code>grabShortcut(self, key[, context])</code>
<code>graphicsEffect(self)</code>
<code>graphicsProxyWidget(self)</code>
<code>hasFocus(self)</code>
<code>hasHeightForWidth(self)</code>

---

continues on next page

Table 18 – continued from previous page

---

`hasMouseTracking(self)`

---

`hasTabletTracking(self)`

---

`height(self)`

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

`isAncestorOf(self, child)`

---

`isEnabled(self)`

---

`isEnabledTo(self, a0)`

---

`isFullScreen(self)`

---

`isHidden(self)`

---

`isLeftToRight(self)`

---

`isMaximized(self)`

---

`isMinimized(self)`

---

continues on next page

Table 18 – continued from previous page

---

`isModal(self)`

---

`isRightToLeft(self)`

---

`isSignalConnected(self, signal)`

---

`isVisible(self)`

---

`isVisibleTo(self, a0)`

---

`isWidgetType(self)`

---

`isWindow(self)`

---

`isWindowModified(self)`

---

`isWindowType(self)`

---

`keyPressEvent(self, a0)`

---

`keyReleaseEvent(self, a0)`

---

`keyboardGrabber()`

---

`killTimer(self, id)`

---

`layout(self)`

---

`layoutDirection(self)`

---

`leaveEvent(self, a0)`

---

`locale(self)`

---

`logicalDpiX(self)`

---

`logicalDpiY(self)`

---

`lower(self)`

---

`mapFrom(self, a0, a1)`

---

`mapFromGlobal(self, a0)`

---

`mapFromParent(self, a0)`

---

`mapTo(self, a0, a1)`

---

continues on next page

Table 18 – continued from previous page

---

`mapToGlobal(self, a0)`

---

`mapToParent(self, a0)`

---

`mask(self)`

---

`maximumHeight(self)`

---

`maximumSize(self)`

---

`maximumWidth(self)`

---

`metaObject(self)`

---

`metric(self, a0)`

---

`minimumHeight(self)`

---

`minimumSize(self)`

---

`minimumSizeHint(self)`

---

`minimumWidth(self)`

---

`mouseDoubleClickEvent(self, a0)`

---

`mouseGrabber()`

---

`mouseMoveEvent(self, a0)`

---

`mousePressEvent(self, a0)`

---

`mouseReleaseEvent(self, a0)`

---

`move()`

---

`moveEvent(self, a0)`

---

`moveToThread(self, thread)`

---

`nativeEvent(self, eventType, message)`

---

`nativeParentWidget(self)`

---

`nextInFocusChain(self)`

---

`normalGeometry(self)`

---

continues on next page

Table 18 – continued from previous page

<code>objectName(self)</code>	
<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	

continues on next page

Table 18 – continued from previous page

---

<code>repaint(-&gt; None -&gt; None)</code>
<code>resize()</code>
<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>sender(self)</code>
<code>senderSignalIndex(self)</code>
<code>setAcceptDrops(self, on)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setEnabled(self, a0)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>

---

continues on next page



Table 18 – continued from previous page

---

<code>setFocus()</code>
<code>setFocusPolicy(self, policy)</code>
<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setForegroundRole(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setHidden(self, hidden)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLocale(self, locale)</code>
<code>setMask()</code>
<code>setMaximumHeight(self, maxh)</code>
<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setMouseTracking(self, enable)</code>
<code>setObjectName(self, name)</code>
<code>setPalette(self, a0)</code>
<code>setParent()</code>
<code>setProperty(self, name, value)</code>

---

continues on next page

Table 18 – continued from previous page

---

<code>setShortcutAutoRepeat(self, id[, enabled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>

---

continues on next page

Table 18 – continued from previous page

---

<code>setWindowTitle(self, a0)</code>
<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>

---

continues on next page

Table 18 – continued from previous page

---

`tr(self, sourceText[, disambiguation, n])`

---

`underMouse(self)`

---

`ungrabGesture(self, type)`

---

`unsetCursor(self)`

---

`unsetLayoutDirection(self)`

---

`unsetLocale(self)`

---

`update(-> None -> None)`

---

`updateGeometry(self)`

---

`updateMicroFocus(self)`

---

`updatesEnabled(self)`

---

`visibleRegion(self)`

---

`whatIsThis(self)`

---

`wheelEvent(self, a0)`

---

`width(self)`

---

`widthMM(self)`

---

`winId(self)`

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

continues on next page

Table 18 – continued from previous page

---

`windowRole(self)`

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

## napari\_locan.widgets.widget\_run\_script

Run python script.

A QWidget plugin with a simple interface to handle python scripts for localization analysis.

### Classes

---

*RunScriptQWidget*(napari\_viewer)

---

## napari\_locan.widgets.widget\_run\_script.RunScriptQWidget

**class** napari\_locan.widgets.widget\_run\_script.RunScriptQWidget(*napari\_viewer*)

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

`__init__(napari_viewer)`

---

`acceptDrops(self)`

---

`accessibleDescription(self)`

---

`accessibleName(self)`

---

`actionEvent(self, a0)`

---

`actions(self)`

---

`activateWindow(self)`

---

`addAction(self, action)`

---

`addActions(self, actions)`

---

`adjustSize(self)`

---

`autoFillBackground(self)`

---

`backgroundRole(self)`

---

`baseSize(self)`

---

`blockSignals(self, b)`

---

continues on next page

Table 19 – continued from previous page

---

<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>
<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>

---

continues on next page



Table 19 – continued from previous page

---

<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>
<code>focusNextPrevChild(self, next)</code>
<code>focusOutEvent(self, a0)</code>

---

continues on next page

Table 19 – continued from previous page

---

`focusPolicy(self)`

---

`focusPreviousChild(self)`

---

`focusProxy(self)`

---

`focusWidget(self)`

---

`font(self)`

---

`fontInfo(self)`

---

`fontMetrics(self)`

---

`foregroundRole(self)`

---

`frameGeometry(self)`

---

`frameSize(self)`

---

`geometry(self)`

---

`getContentsMargins(self)`

---

`grab(self[, rectangle])`

---

`grabGesture(self, type[, flags])`

---

`grabKeyboard(self)`

---

`grabMouse()`

---

`grabShortcut(self, key[, context])`

---

`graphicsEffect(self)`

---

`graphicsProxyWidget(self)`

---

`hasFocus(self)`

---

`hasHeightForWidth(self)`

---

`hasMouseTracking(self)`

---

`hasTabletTracking(self)`

---

`height(self)`

---

continues on next page

Table 19 – continued from previous page

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

`isAncestorOf(self, child)`

---

`isEnabled(self)`

---

`isEnabledTo(self, a0)`

---

`isFullScreen(self)`

---

`isHidden(self)`

---

`isLeftToRight(self)`

---

`isMaximized(self)`

---

`isMinimized(self)`

---

`isModal(self)`

---

`isRightToLeft(self)`

---

`isSignalConnected(self, signal)`

---

continues on next page

Table 19 – continued from previous page

---

<code>isVisible(self)</code>
<code>isVisibleTo(self, a0)</code>
<code>isWidgetType(self)</code>
<code>isWindow(self)</code>
<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>keyPressEvent(self, a0)</code>
<code>keyReleaseEvent(self, a0)</code>
<code>keyboardGrabber()</code>
<code>killTimer(self, id)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code>leaveEvent(self, a0)</code>
<code>locale(self)</code>
<code>logicalDpiX(self)</code>
<code>logicalDpiY(self)</code>
<code>lower(self)</code>
<code>mapFrom(self, a0, a1)</code>
<code>mapFromGlobal(self, a0)</code>
<code>mapFromParent(self, a0)</code>
<code>mapTo(self, a0, a1)</code>
<code>mapToGlobal(self, a0)</code>
<code>mapToParent(self, a0)</code>
<code>mask(self)</code>

---

continues on next page

Table 19 – continued from previous page

---

`maximumHeight(self)`

---

`maximumSize(self)`

---

`maximumWidth(self)`

---

`metaObject(self)`

---

`metric(self, a0)`

---

`minimumHeight(self)`

---

`minimumSize(self)`

---

`minimumSizeHint(self)`

---

`minimumWidth(self)`

---

`mouseDoubleClickEvent(self, a0)`

---

`mouseGrabber()`

---

`mouseMoveEvent(self, a0)`

---

`mousePressEvent(self, a0)`

---

`mouseReleaseEvent(self, a0)`

---

`move()`

---

`moveEvent(self, a0)`

---

`moveToThread(self, thread)`

---

`nativeEvent(self, eventType, message)`

---

`nativeParentWidget(self)`

---

`nextInFocusChain(self)`

---

`normalGeometry(self)`

---

`objectName(self)`

---

`overrideWindowFlags(self, type)`

---

`overrideWindowState(self, state)`

---

continues on next page

Table 19 – continued from previous page

<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	
<code>resize()</code>	
<code>resizeEvent(self, a0)</code>	

continues on next page

Table 19 – continued from previous page

---

`restoreGeometry(self, geometry)`

---

`saveGeometry(self)`

---

`screen(self)`

---

`scroll()`

---

`sender(self)`

---

`senderSignalIndex(self)`

---

`setAcceptDrops(self, on)`

---

`setAccessibleDescription(self, description)`

---

`setAccessibleName(self, name)`

---

`setAttribute(self, attribute[, on])`

---

`setAutoFillBackground(self, enabled)`

---

`setBackgroundRole(self, a0)`

---

`setBaseSize()`

---

`setContentsMargins()`

---

`setContextMenuPolicy(self, policy)`

---

`setCursor(self, a0)`

---

`setDisabled(self, a0)`

---

`setEnabled(self, a0)`

---

`setFixedHeight(self, h)`

---

`setFixedSize()`

---

`setFixedWidth(self, w)`

---

`setFocus()`

---

`setFocusPolicy(self, policy)`

---

`setFocusProxy(self, a0)`

---

continues on next page

Table 19 – continued from previous page

---

<code>setFont(self, a0)</code>
<code>setForegroundRole(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setHidden(self, hidden)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLocale(self, locale)</code>
<code>setMask()</code>
<code>setMaximumHeight(self, maxh)</code>
<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setMouseTracking(self, enable)</code>
<code>setObjectName(self, name)</code>
<code>setPalette(self, a0)</code>
<code>setParent()</code>
<code>setProperty(self, name, value)</code>
<code>setShortcutAutoRepeat(self, id[, en- abled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>

---

continues on next page



Table 19 – continued from previous page

---

<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>
<code>sharedPainter(self)</code>
<code>show(self)</code>

---

continues on next page

Table 19 – continued from previous page

---

<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>
<code>underMouse(self)</code>
<code>ungrabGesture(self, type)</code>

---

continues on next page

Table 19 – continued from previous page

---

`unsetCursor(self)`

---

`unsetLayoutDirection(self)`

---

`unsetLocale(self)`

---

`update(-> None -> None)`

---

`updateGeometry(self)`

---

`updateMicroFocus(self)`

---

`updatesEnabled(self)`

---

`visibleRegion(self)`

---

`whatsThis(self)`

---

`wheelEvent(self, a0)`

---

`width(self)`

---

`widthMM(self)`

---

`winId(self)`

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

`windowRole(self)`

---

`windowState(self)`

---

`windowTitle(self)`

---

continues on next page

Table 19 – continued from previous page

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

## napari\_locan.widgets.widget\_select

Select localizations from SMLM dataset.

A QWidget plugin to select localizations in current SMLM dataset based on a filter specification. A new SMLM dataset will be created.

### Classes

---

```
SelectQWidget(napari_viewer[, smlm_data, ...])
```

---

## napari\_locan.widgets.widget\_select.SelectQWidget

```
class napari_locan.widgets.widget_select.SelectQWidget(napari_viewer,
                                                    smlm_data=<napari_locan.data_model.smlm_
                                                    object>, fil-
                                                    ter_specifications=<napari_locan.data_model.
                                                    object>)
```

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

```
__init__(napari_viewer[, smlm_data, ...])
```

---

```
acceptDrops(self)
```

---

```
accessibleDescription(self)
```

---

```
accessibleName(self)
```

---

```
actionEvent(self, a0)
```

---

```
actions(self)
```

---

```
activateWindow(self)
```

---

```
addAction(self, action)
```

---

```
addActions(self, actions)
```

---

```
adjustSize(self)
```

---

```
autoFillBackground(self)
```

---

continues on next page

Table 20 – continued from previous page

---

`backgroundRole(self)`

---

`baseSize(self)`

---

`blockSignals(self, b)`

---

`changeEvent(self, a0)`

---

`childAt(-> Optional[QWidget])`

---

`childEvent(self, a0)`

---

`children(self)`

---

`childrenRect(self)`

---

`childrenRegion(self)`

---

`clearFocus(self)`

---

`clearMask(self)`

---

`close(self)`

---

`closeEvent(self, a0)`

---

`colorCount(self)`

---

`connectNotify(self, signal)`

---

`contentsMargins(self)`

---

`contentsRect(self)`

---

`contextMenuEvent(self, a0)`

---

`contextMenuPolicy(self)`

---

`create(self[, window, initializeWindow, ...])`

---

`createWindowContainer(window[, parent,  
flags])`

---

`cursor(self)`

---

`customEvent(self, a0)`

---

`deleteLater(self)`

---

continues on next page

Table 20 – continued from previous page

---

<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>

---

continues on next page



Table 20 – continued from previous page

---

<code>focusNextChild(self)</code>
<code>focusNextPrevChild(self, next)</code>
<code>focusOutEvent(self, a0)</code>
<code>focusPolicy(self)</code>
<code>focusPreviousChild(self)</code>
<code>focusProxy(self)</code>
<code>focusWidget(self)</code>
<code>font(self)</code>
<code>fontInfo(self)</code>
<code>fontMetrics(self)</code>
<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>
<code>frameSize(self)</code>
<code>geometry(self)</code>
<code>getContentsMargins(self)</code>
<code>grab(self[, rectangle])</code>
<code>grabGesture(self, type[, flags])</code>
<code>grabKeyboard(self)</code>
<code>grabMouse()</code>
<code>grabShortcut(self, key[, context])</code>
<code>graphicsEffect(self)</code>
<code>graphicsProxyWidget(self)</code>
<code>hasFocus(self)</code>
<code>hasHeightForWidth(self)</code>

---

continues on next page

Table 20 – continued from previous page

---

`hasMouseTracking(self)`

---

`hasTabletTracking(self)`

---

`height(self)`

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

`isAncestorOf(self, child)`

---

`isEnabled(self)`

---

`isEnabledTo(self, a0)`

---

`isFullScreen(self)`

---

`isHidden(self)`

---

`isLeftToRight(self)`

---

`isMaximized(self)`

---

`isMinimized(self)`

---

continues on next page

Table 20 – continued from previous page

---

`isModal(self)`

---

`isRightToLeft(self)`

---

`isSignalConnected(self, signal)`

---

`isVisible(self)`

---

`isVisibleTo(self, a0)`

---

`isWidgetType(self)`

---

`isWindow(self)`

---

`isWindowModified(self)`

---

`isWindowType(self)`

---

`keyPressEvent(self, a0)`

---

`keyReleaseEvent(self, a0)`

---

`keyboardGrabber()`

---

`killTimer(self, id)`

---

`layout(self)`

---

`layoutDirection(self)`

---

`leaveEvent(self, a0)`

---

`locale(self)`

---

`logicalDpiX(self)`

---

`logicalDpiY(self)`

---

`lower(self)`

---

`mapFrom(self, a0, a1)`

---

`mapFromGlobal(self, a0)`

---

`mapFromParent(self, a0)`

---

`mapTo(self, a0, a1)`

---

continues on next page

Table 20 – continued from previous page

<code>mapToGlobal(self, a0)</code>
<code>mapToParent(self, a0)</code>
<code>mask(self)</code>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumWidth(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>mouseDoubleClickEvent(self, a0)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, a0)</code>
<code>mousePressEvent(self, a0)</code>
<code>mouseReleaseEvent(self, a0)</code>
<code>move()</code>
<code>moveEvent(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>
<code>nextInFocusChain(self)</code>
<code>normalGeometry(self)</code>

continues on next page

Table 20 – continued from previous page

<code>objectName(self)</code>	
<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	

continues on next page

Table 20 – continued from previous page

---

repaint(-> None -> None)

---

resize()

---

resizeEvent(self, a0)

---

restoreGeometry(self, geometry)

---

saveGeometry(self)

---

screen(self)

---

scroll()

---

sender(self)

---

senderSignalIndex(self)

---

setAcceptDrops(self, on)

---

setAccessibleDescription(self, description)

---

setAccessibleName(self, name)

---

setAttribute(self, attribute[, on])

---

setAutoFillBackground(self, enabled)

---

setBackgroundRole(self, a0)

---

setBaseSize()

---

setContentsMargins()

---

setContextMenuPolicy(self, policy)

---

setCursor(self, a0)

---

setDisabled(self, a0)

---

setEnabled(self, a0)

---

setFixedHeight(self, h)

---

setFixedSize()

---

setFixedWidth(self, w)

---

continues on next page

Table 20 – continued from previous page

---

`setFocus()`

---

`setFocusPolicy(self, policy)`

---

`setFocusProxy(self, a0)`

---

`setFont(self, a0)`

---

`setForegroundRole(self, a0)`

---

`setGeometry()`

---

`setGraphicsEffect(self, effect)`

---

`setHidden(self, hidden)`

---

`setInputMethodHints(self, hints)`

---

`setLayout(self, a0)`

---

`setLayoutDirection(self, direction)`

---

`setLocale(self, locale)`

---

`setMask()`

---

`setMaximumHeight(self, maxh)`

---

`setMaximumSize()`

---

`setMaximumWidth(self, maxw)`

---

`setMinimumHeight(self, minh)`

---

`setMinimumSize()`

---

`setMinimumWidth(self, minw)`

---

`setMouseTracking(self, enable)`

---

`setObjectName(self, name)`

---

`setPalette(self, a0)`

---

`setParent()`

---

`setProperty(self, name, value)`

---

continues on next page

Table 20 – continued from previous page

---

<code>setShortcutAutoRepeat(self, id[, enabled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>

---

continues on next page



Table 20 – continued from previous page

---

`setWindowTitle(self, a0)`

---

`sharedPainter(self)`

---

`show(self)`

---

`showEvent(self, a0)`

---

`showFullScreen(self)`

---

`showMaximized(self)`

---

`showMinimized(self)`

---

`showNormal(self)`

---

`signalsBlocked(self)`

---

`size(self)`

---

`sizeHint(self)`

---

`sizeIncrement(self)`

---

`sizePolicy(self)`

---

`stackUnder(self, a0)`

---

`startTimer(self, interval[, timerType])`

---

`statusTip(self)`

---

`style(self)`

---

`styleSheet(self)`

---

`tabletEvent(self, a0)`

---

`testAttribute(self, attribute)`

---

`thread(self)`

---

`timerEvent(self, a0)`

---

`toolTip(self)`

---

`toolTipDuration(self)`

---

continues on next page

Table 20 – continued from previous page

---

`tr(self, sourceText[, disambiguation, n])`

---

`underMouse(self)`

---

`ungrabGesture(self, type)`

---

`unsetCursor(self)`

---

`unsetLayoutDirection(self)`

---

`unsetLocale(self)`

---

`update(-> None -> None)`

---

`updateGeometry(self)`

---

`updateMicroFocus(self)`

---

`updatesEnabled(self)`

---

`visibleRegion(self)`

---

`whatIsThis(self)`

---

`wheelEvent(self, a0)`

---

`width(self)`

---

`widthMM(self)`

---

`winId(self)`

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

continues on next page

Table 20 – continued from previous page

---

`windowRole(self)`

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

## napari\_locan.widgets.widget\_show\_data

Show data statistics for a SMLM dataset.

A QWidget plugin for showing locdata data statistics (locdata.data.describe()).

### Classes

---

*ShowDataQWidget*(napari\_viewer[,  
smlm\_data])

---

*TableModel*(data)

---

## napari\_locan.widgets.widget\_show\_data.ShowDataQWidget

**class** napari\_locan.widgets.widget\_show\_data.**ShowDataQWidget**(*napari\_viewer*,  
*smlm\_data*=<napari\_locan.data\_model.  
*object*>)

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

`__init__(napari_viewer[, smlm_data])`

---

`acceptDrops(self)`

---

`accessibleDescription(self)`

---

`accessibleName(self)`

---

`actionEvent(self, a0)`

---

`actions(self)`

---

`activateWindow(self)`

---

`addAction(self, action)`

---

`addActions(self, actions)`

---

`adjustSize(self)`

---

`autoFillBackground(self)`

---

`backgroundRole(self)`

---

continues on next page

Table 21 – continued from previous page

---

`baseSize(self)`

---

`blockSignals(self, b)`

---

`changeEvent(self, a0)`

---

`childAt(-> Optional[QWidget])`

---

`childEvent(self, a0)`

---

`children(self)`

---

`childrenRect(self)`

---

`childrenRegion(self)`

---

`clearFocus(self)`

---

`clearMask(self)`

---

`close(self)`

---

`closeEvent(self, a0)`

---

`colorCount(self)`

---

`connectNotify(self, signal)`

---

`contentsMargins(self)`

---

`contentsRect(self)`

---

`contextMenuEvent(self, a0)`

---

`contextMenuPolicy(self)`

---

`create(self[, window, initializeWindow, ...])`

---

`createWindowContainer(window[, parent,  
flags])`

---

`cursor(self)`

---

`customEvent(self, a0)`

---

`deleteLater(self)`

---

`depth(self)`

---

continues on next page

Table 21 – continued from previous page

---

<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>

---

continues on next page

Table 21 – continued from previous page

---

`focusNextPrevChild(self, next)`

---

`focusOutEvent(self, a0)`

---

`focusPolicy(self)`

---

`focusPreviousChild(self)`

---

`focusProxy(self)`

---

`focusWidget(self)`

---

`font(self)`

---

`fontInfo(self)`

---

`fontMetrics(self)`

---

`foregroundRole(self)`

---

`frameGeometry(self)`

---

`frameSize(self)`

---

`geometry(self)`

---

`getContentsMargins(self)`

---

`grab(self[, rectangle])`

---

`grabGesture(self, type[, flags])`

---

`grabKeyboard(self)`

---

`grabMouse()`

---

`grabShortcut(self, key[, context])`

---

`graphicsEffect(self)`

---

`graphicsProxyWidget(self)`

---

`hasFocus(self)`

---

`hasHeightForWidth(self)`

---

`hasMouseTracking(self)`

---

continues on next page



Table 21 – continued from previous page

---

hasTabletTracking(self)

---

height(self)

---

heightForWidth(self, a0)

---

heightMM(self)

---

hide(self)

---

hideEvent(self, a0)

---

inherits(self, classname)

---

initPainter(self, painter)

---

inputMethodEvent(self, a0)

---

inputMethodHints(self)

---

inputMethodQuery(self, a0)

---

insertAction(self, before, action)

---

insertActions(self, before, actions)

---

installEventFilter(self, a0)

---

isActiveWindow(self)

---

isAncestorOf(self, child)

---

isEnabled(self)

---

isEnabledTo(self, a0)

---

isFullScreen(self)

---

isHidden(self)

---

isLeftToRight(self)

---

isMaximized(self)

---

isMinimized(self)

---

isModal(self)

---

continues on next page

Table 21 – continued from previous page

---

<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, signal)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, a0)</code>
<code>isWidgetType(self)</code>
<code>isWindow(self)</code>
<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>keyPressEvent(self, a0)</code>
<code>keyReleaseEvent(self, a0)</code>
<code>keyboardGrabber()</code>
<code>killTimer(self, id)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code>leaveEvent(self, a0)</code>
<code>locale(self)</code>
<code>logicalDpiX(self)</code>
<code>logicalDpiY(self)</code>
<code>lower(self)</code>
<code>mapFrom(self, a0, a1)</code>
<code>mapFromGlobal(self, a0)</code>
<code>mapFromParent(self, a0)</code>
<code>mapTo(self, a0, a1)</code>
<code>mapToGlobal(self, a0)</code>

---

continues on next page

Table 21 – continued from previous page

<code>mapToParent(self, a0)</code>
<code>mask(self)</code>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumWidth(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>mouseDoubleClickEvent(self, a0)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, a0)</code>
<code>mousePressEvent(self, a0)</code>
<code>mouseReleaseEvent(self, a0)</code>
<code>move()</code>
<code>moveEvent(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>
<code>nextInFocusChain(self)</code>
<code>normalGeometry(self)</code>
<code>objectName(self)</code>

continues on next page

Table 21 – continued from previous page

<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	

continues on next page

Table 21 – continued from previous page

---

<code>resize()</code>
<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>sender(self)</code>
<code>senderSignalIndex(self)</code>
<code>setAcceptDrops(self, on)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setEnabled(self, a0)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>
<code>setFocus()</code>

---

continues on next page

Table 21 – continued from previous page

---

<code>setFocusPolicy(self, policy)</code>
<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setForegroundRole(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setHidden(self, hidden)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLocale(self, locale)</code>
<code>setMask()</code>
<code>setMaximumHeight(self, maxh)</code>
<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setMouseTracking(self, enable)</code>
<code>setObjectName(self, name)</code>
<code>setPalette(self, a0)</code>
<code>setParent()</code>
<code>setProperty(self, name, value)</code>
<code>setShortcutAutoRepeat(self, id[, en- abled])</code>

---

continues on next page

Table 21 – continued from previous page

---

<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>

---

continues on next page

Table 21 – continued from previous page

---

<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>

---

continues on next page



Table 21 – continued from previous page

---

`underMouse(self)`

---

`ungrabGesture(self, type)`

---

`unsetCursor(self)`

---

`unsetLayoutDirection(self)`

---

`unsetLocale(self)`

---

`update(-> None -> None)`

---

`updateGeometry(self)`

---

`updateMicroFocus(self)`

---

`updatesEnabled(self)`

---

`visibleRegion(self)`

---

`whatsThis(self)`

---

`wheelEvent(self, a0)`

---

`width(self)`

---

`widthMM(self)`

---

`winId(self)`

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

`windowRole(self)`

---

continues on next page

Table 21 – continued from previous page

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

## napari\_locan.widgets.widget\_show\_data.TableModel

**class** napari\_locan.widgets.widget\_show\_data.TableModel(*data*)

Bases: PyQt5.QtCore.QAbstractTableModel

### Methods

---

`__init__(data)`

---

`beginInsertColumns(self, parent, first, last)`

---

`beginInsertRows(self, parent, first, last)`

---

`beginMoveColumns(self, sourceParent, ...)`

---

`beginMoveRows(self, sourceParent, ...)`

---

`beginRemoveColumns(self, parent, first, last)`

---

`beginRemoveRows(self, parent, first, last)`

---

`beginResetModel(self)`

---

`blockSignals(self, b)`

---

`buddy(self, index)`

---

`canDropMimeData(self, data, action, row, ...)`

---

`canFetchMore(self, parent)`

---

`changePersistentIndex(self, from_, to)`

---

`changePersistentIndexList(self, from_,  
to)`

---

`checkIndex(self, index[, options])`

---

`childEvent(self, a0)`

---

`children(self)`

---

`columnCount([parent])`

**rtype** int

---

`connectNotify(self, signal)`

---

`createIndex(self, row, column[, object])`

---

continues on next page

Table 22 – continued from previous page

<code>customEvent(self, a0)</code>	
<code>data(index, role)</code>	<b>rtype str</b>
<code>decodeData(self, row, column, parent, stream)</code>	
<code>deleteLater(self)</code>	
<code>disconnect(-&gt; bool)</code>	
<code>disconnectNotify(self, signal)</code>	
<code>dropMimeData(self, data, action, row, ...)</code>	
<code>dumpObjectInfo(self)</code>	
<code>dumpObjectTree(self)</code>	
<code>dynamicPropertyNames(self)</code>	
<code>encodeData(self, indexes, stream)</code>	
<code>endInsertColumns(self)</code>	
<code>endInsertRows(self)</code>	
<code>endMoveColumns(self)</code>	
<code>endMoveRows(self)</code>	
<code>endRemoveColumns(self)</code>	
<code>endRemoveRows(self)</code>	
<code>endResetModel(self)</code>	
<code>event(self, a0)</code>	
<code>eventFilter(self, a0, a1)</code>	
<code>fetchMore(self, parent)</code>	
<code>findChild(-&gt; QObjectT)</code>	
<code>findChildren(...)</code>	
<code>flags(self, index)</code>	

continues on next page

Table 22 – continued from previous page

---

<code>hasChildren(self[, parent])</code>
<code>hasIndex(self, row, column[, parent])</code>
<code><i>headerData</i>(self, section, orientation[, role])</code>
<code>index(self, row, column[, parent])</code>
<code>inherits(self, classname)</code>
<code>insertColumn(self, column[, parent])</code>
<code>insertColumns(self, column, count[, parent])</code>
<code>insertRow(self, row[, parent])</code>
<code>insertRows(self, row, count[, parent])</code>
<code>installEventFilter(self, a0)</code>
<code>isSignalConnected(self, signal)</code>
<code>isWidgetType(self)</code>
<code>isWindowType(self)</code>
<code>itemData(self, index)</code>
<code>killTimer(self, id)</code>
<code>match(self, start, role, value[, hits, flags])</code>
<code>metaObject(self)</code>
<code>mimeData(self, indexes)</code>
<code>mimeTypes(self)</code>
<code>moveColumn(self, sourceParent, sourceColumn, ...)</code>
<code>moveColumns(self, sourceParent, ...)</code>
<code>moveRow(self, sourceParent, sourceRow, ...)</code>
<code>moveRows(self, sourceParent, sourceRow, ...)</code>
<code>moveToThread(self, thread)</code>

---

continues on next page

Table 22 – continued from previous page

<code>objectName(self)</code>	
<code>parent(self)</code>	
<code>persistentIndexList(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>receivers(self, signal)</code>	
<code>removeColumn(self, column[, parent])</code>	
<code>removeColumns(self, column, count[, parent])</code>	
<code>removeEventFilter(self, a0)</code>	
<code>removeRow(self, row[, parent])</code>	
<code>removeRows(self, row, count[, parent])</code>	
<code>resetInternalData(self)</code>	
<code>revert(self)</code>	
<code>roleNames(self)</code>	
<code>rowCount([parent])</code>	<b>rtype int</b>
<code>sender(self)</code>	
<code>senderSignalIndex(self)</code>	
<code>setData(self, index, value[, role])</code>	
<code>setHeaderData(self, section, orientation, value)</code>	
<code>setItemData(self, index, roles)</code>	
<code>setObjectName(self, name)</code>	
<code>setParent(self, a0)</code>	
<code>setProperty(self, name, value)</code>	
<code>sibling(self, row, column, idx)</code>	

continues on next page

Table 22 – continued from previous page

---

`signalsBlocked(self)`

---

`sort(self, column[, order])`

---

`span(self, index)`

---

`startTimer(self, interval[, timerType])`

---

`submit(self)`

---

`supportedDragActions(self)`

---

`supportedDropActions(self)`

---

`thread(self)`

---

`timerEvent(self, a0)`

---

`tr(self, sourceText[, disambiguation, n])`

---



## Attributes

HorizontalSortHint	
NoLayoutChangeHint	
VerticalSortHint	
columnsAboutToBeInserted	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
columnsAboutToBeMoved	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
columnsAboutToBeRemoved	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
columnsInserted	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
columnsMoved	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
columnsRemoved	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
dataChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
headerDataChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
layoutAboutToBeChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
layoutChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
modelAboutToBeReset	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
modelReset	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
rowsAboutToBeInserted	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
rowsAboutToBeMoved	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
rowsAboutToBeRemoved	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
rowsInserted	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
rowsMoved	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
rowsRemoved	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	

**columnCount** (*parent=None*)

**Return type** int

**data** (*index, role*)

**Return type** str

**headerData** (*section, orientation, role*)

**rowCount** (*parent=None*)

**Return type** int

## napari\_locan.widgets.widget\_show\_metadata

Show metadata for a SMLM dataset.

QWidget plugin for showing metadata for a single SMLM dataset (locdata.meta).

## Classes

---

*ShowMetadataQWidget*(*napari\_viewer*[,  
*smlm\_data*])

---

## napari\_locan.widgets.widget\_show\_metadata.ShowMetadataQWidget

**class** `napari_locan.widgets.widget_show_metadata.ShowMetadataQWidget` (*napari\_viewer*,  
*smlm\_data=<napari\_locan.da*  
*object>*)

Bases: `PyQt5.QtWidgets.QWidget`

## Methods

---

`__init__`(*napari\_viewer*[, *smlm\_data*])

---

`acceptDrops`(*self*)

---

`accessibleDescription`(*self*)

---

`accessibleName`(*self*)

---

`actionEvent`(*self*, *a0*)

---

`actions`(*self*)

---

`activateWindow`(*self*)

---

continues on next page

Table 23 – continued from previous page

---

<code>addAction(self, action)</code>
<code>addActions(self, actions)</code>
<code>adjustSize(self)</code>
<code>autoFillBackground(self)</code>
<code>backgroundRole(self)</code>
<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>

---

continues on next page

Table 23 – continued from previous page

---

<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>
<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySubWindows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>

---

continues on next page

Table 23 – continued from previous page

---

<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>
<code>focusNextPrevChild(self, next)</code>
<code>focusOutEvent(self, a0)</code>
<code>focusPolicy(self)</code>
<code>focusPreviousChild(self)</code>
<code>focusProxy(self)</code>
<code>focusWidget(self)</code>
<code>font(self)</code>
<code>fontInfo(self)</code>
<code>fontMetrics(self)</code>
<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>
<code>frameSize(self)</code>
<code>geometry(self)</code>
<code>getContentsMargins(self)</code>
<code>grab(self[, rectangle])</code>
<code>grabGesture(self, type[, flags])</code>
<code>grabKeyboard(self)</code>
<code>grabMouse()</code>
<code>grabShortcut(self, key[, context])</code>

---

continues on next page

Table 23 – continued from previous page

---

<code>graphicsEffect(self)</code>
<code>graphicsProxyWidget(self)</code>
<code>hasFocus(self)</code>
<code>hasHeightForWidth(self)</code>
<code>hasMouseTracking(self)</code>
<code>hasTabletTracking(self)</code>
<code>height(self)</code>
<code>heightForWidth(self, a0)</code>
<code>heightMM(self)</code>
<code>hide(self)</code>
<code>hideEvent(self, a0)</code>
<code>inherits(self, classname)</code>
<code>initPainter(self, painter)</code>
<code>inputMethodEvent(self, a0)</code>
<code>inputMethodHints(self)</code>
<code>inputMethodQuery(self, a0)</code>
<code>insertAction(self, before, action)</code>
<code>insertActions(self, before, actions)</code>
<code>installEventFilter(self, a0)</code>
<code>isActiveWindow(self)</code>
<code>isAncestorOf(self, child)</code>
<code>isEnabled(self)</code>
<code>isEnabledTo(self, a0)</code>
<code>isFullScreen(self)</code>

---

continues on next page

Table 23 – continued from previous page

<code>isHidden(self)</code>
<code>isLeftToRight(self)</code>
<code>isMaximized(self)</code>
<code>isMinimized(self)</code>
<code>isModal(self)</code>
<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, signal)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, a0)</code>
<code>isWidgetType(self)</code>
<code>isWindow(self)</code>
<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>keyPressEvent(self, a0)</code>
<code>keyReleaseEvent(self, a0)</code>
<code>keyboardGrabber()</code>
<code>killTimer(self, id)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code>leaveEvent(self, a0)</code>
<code>locale(self)</code>
<code>logicalDpiX(self)</code>
<code>logicalDpiY(self)</code>
<code>lower(self)</code>

continues on next page

Table 23 – continued from previous page

---

`mapFrom(self, a0, a1)`

---

`mapFromGlobal(self, a0)`

---

`mapFromParent(self, a0)`

---

`mapTo(self, a0, a1)`

---

`mapToGlobal(self, a0)`

---

`mapToParent(self, a0)`

---

`mask(self)`

---

`maximumHeight(self)`

---

`maximumSize(self)`

---

`maximumWidth(self)`

---

`metaObject(self)`

---

`metric(self, a0)`

---

`minimumHeight(self)`

---

`minimumSize(self)`

---

`minimumSizeHint(self)`

---

`minimumWidth(self)`

---

`mouseDoubleClickEvent(self, a0)`

---

`mouseGrabber()`

---

`mouseMoveEvent(self, a0)`

---

`mousePressEvent(self, a0)`

---

`mouseReleaseEvent(self, a0)`

---

`move()`

---

`moveEvent(self, a0)`

---

`moveToThread(self, thread)`

---

continues on next page



Table 23 – continued from previous page

<code>nativeEvent(self, eventType, message)</code>	
<code>nativeParentWidget(self)</code>	
<code>nextInFocusChain(self)</code>	
<code>normalGeometry(self)</code>	
<code>objectName(self)</code>	
<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	

continues on next page

Table 23 – continued from previous page

<code>releaseShortcut(self, id)</code>
<code>removeAction(self, action)</code>
<code>removeEventFilter(self, a0)</code>
<code>render(, sourceRegion, flags, ...)</code>
<code>repaint(-&gt; None -&gt; None)</code>
<code>resize()</code>
<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>sender(self)</code>
<code>senderSignalIndex(self)</code>
<code>setAcceptDrops(self, on)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>

continues on next page

Table 23 – continued from previous page

---

<code>setEnabled(self, a0)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>
<code>setFocus()</code>
<code>setFocusPolicy(self, policy)</code>
<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setForegroundRole(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setHidden(self, hidden)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLocale(self, locale)</code>
<code>setMask()</code>
<code>setMaximumHeight(self, maxh)</code>
<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setMouseTracking(self, enable)</code>

---

continues on next page

Table 23 – continued from previous page

---

<code>setObjectName(self, name)</code>
<code>setPalette(self, a0)</code>
<code>setParent()</code>
<code>setProperty(self, name, value)</code>
<code>setShortcutAutoRepeat(self, id[, enabled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>

---

continues on next page

Table 23 – continued from previous page

---

`setWindowModified(self, a0)`

---

`setWindowOpacity(self, level)`

---

`setWindowRole(self, a0)`

---

`setWindowState(self, state)`

---

`setWindowTitle(self, a0)`

---

`sharedPainter(self)`

---

`show(self)`

---

`showEvent(self, a0)`

---

`showFullScreen(self)`

---

`showMaximized(self)`

---

`showMinimized(self)`

---

`showNormal(self)`

---

`signalsBlocked(self)`

---

`size(self)`

---

`sizeHint(self)`

---

`sizeIncrement(self)`

---

`sizePolicy(self)`

---

`stackUnder(self, a0)`

---

`startTimer(self, interval[, timerType])`

---

`statusTip(self)`

---

`style(self)`

---

`styleSheet(self)`

---

`tabletEvent(self, a0)`

---

`testAttribute(self, attribute)`

---

continues on next page

Table 23 – continued from previous page

---

<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>
<code>underMouse(self)</code>
<code>ungrabGesture(self, type)</code>
<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>
<code>update(-&gt; None -&gt; None)</code>
<code>updateGeometry(self)</code>
<code>updateMicroFocus(self)</code>
<code>updatesEnabled(self)</code>
<code>visibleRegion(self)</code>
<code>whatsThis(self)</code>
<code>wheelEvent(self, a0)</code>
<code>width(self)</code>
<code>widthMM(self)</code>
<code>winId(self)</code>
<code>window(self)</code>
<code>windowFilePath(self)</code>
<code>windowFlags(self)</code>
<code>windowHandle(self)</code>

---

continues on next page

Table 23 – continued from previous page

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

`windowRole(self)`

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

---

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

---



## napari\_locan.widgets.widget\_show\_properties

Show locdata properties for a SMLM dataset.

QWidget plugin for showing the aggregated properties for a single SMLM dataset (locdata.properties).

### Classes

---

```
ShowPropertiesQWidget(napari_viewer[,  
smlm_data])
```

---

## napari\_locan.widgets.widget\_show\_properties.ShowPropertiesQWidget

```
class napari_locan.widgets.widget_show_properties.ShowPropertiesQWidget(napari_viewer,  
                                                                    smlm_data=<napari_locan.widgets.widget_show_properties.ShowPropertiesQWidget object>)
```

Bases: PyQt5.QtWidgets.QWidget

### Methods

---

```
__init__(napari_viewer[, smlm_data])
```

---

```
acceptDrops(self)
```

---

```
accessibleDescription(self)
```

---

```
accessibleName(self)
```

---

```
actionEvent(self, a0)
```

---

```
actions(self)
```

---

```
activateWindow(self)
```

---

```
addAction(self, action)
```

---

```
addActions(self, actions)
```

---

```
adjustSize(self)
```

---

```
autoFillBackground(self)
```

---

```
backgroundRole(self)
```

---

continues on next page

Table 24 – continued from previous page

---

`baseSize(self)`

---

`blockSignals(self, b)`

---

`changeEvent(self, a0)`

---

`childAt(-> Optional[QWidget])`

---

`childEvent(self, a0)`

---

`children(self)`

---

`childrenRect(self)`

---

`childrenRegion(self)`

---

`clearFocus(self)`

---

`clearMask(self)`

---

`close(self)`

---

`closeEvent(self, a0)`

---

`colorCount(self)`

---

`connectNotify(self, signal)`

---

`contentsMargins(self)`

---

`contentsRect(self)`

---

`contextMenuEvent(self, a0)`

---

`contextMenuPolicy(self)`

---

`create(self[, window, initializeWindow, ...])`

---

`createWindowContainer(window[, parent,  
flags])`

---

`cursor(self)`

---

`customEvent(self, a0)`

---

`deleteLater(self)`

---

`depth(self)`

---

continues on next page

Table 24 – continued from previous page

---

<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>

---

continues on next page

Table 24 – continued from previous page

---

<code>focusNextPrevChild(self, next)</code>
<code>focusOutEvent(self, a0)</code>
<code>focusPolicy(self)</code>
<code>focusPreviousChild(self)</code>
<code>focusProxy(self)</code>
<code>focusWidget(self)</code>
<code>font(self)</code>
<code>fontInfo(self)</code>
<code>fontMetrics(self)</code>
<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>
<code>frameSize(self)</code>
<code>geometry(self)</code>
<code>getContentsMargins(self)</code>
<code>grab(self[, rectangle])</code>
<code>grabGesture(self, type[, flags])</code>
<code>grabKeyboard(self)</code>
<code>grabMouse()</code>
<code>grabShortcut(self, key[, context])</code>
<code>graphicsEffect(self)</code>
<code>graphicsProxyWidget(self)</code>
<code>hasFocus(self)</code>
<code>hasHeightForWidth(self)</code>
<code>hasMouseTracking(self)</code>

---

continues on next page

Table 24 – continued from previous page

---

`hasTabletTracking(self)`

---

`height(self)`

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

`isAncestorOf(self, child)`

---

`isEnabled(self)`

---

`isEnabledTo(self, a0)`

---

`isFullScreen(self)`

---

`isHidden(self)`

---

`isLeftToRight(self)`

---

`isMaximized(self)`

---

`isMinimized(self)`

---

`isModal(self)`

---

continues on next page

Table 24 – continued from previous page

---

`isRightToLeft(self)`

---

`isSignalConnected(self, signal)`

---

`isVisible(self)`

---

`isVisibleTo(self, a0)`

---

`isWidgetType(self)`

---

`isWindow(self)`

---

`isWindowModified(self)`

---

`isWindowType(self)`

---

`keyPressEvent(self, a0)`

---

`keyReleaseEvent(self, a0)`

---

`keyboardGrabber()`

---

`killTimer(self, id)`

---

`layout(self)`

---

`layoutDirection(self)`

---

`leaveEvent(self, a0)`

---

`locale(self)`

---

`logicalDpiX(self)`

---

`logicalDpiY(self)`

---

`lower(self)`

---

`mapFrom(self, a0, a1)`

---

`mapFromGlobal(self, a0)`

---

`mapFromParent(self, a0)`

---

`mapTo(self, a0, a1)`

---

`mapToGlobal(self, a0)`

---

continues on next page

Table 24 – continued from previous page

---

`mapToParent(self, a0)`

---

`mask(self)`

---

`maximumHeight(self)`

---

`maximumSize(self)`

---

`maximumWidth(self)`

---

`metaObject(self)`

---

`metric(self, a0)`

---

`minimumHeight(self)`

---

`minimumSize(self)`

---

`minimumSizeHint(self)`

---

`minimumWidth(self)`

---

`mouseDoubleClickEvent(self, a0)`

---

`mouseGrabber()`

---

`mouseMoveEvent(self, a0)`

---

`mousePressEvent(self, a0)`

---

`mouseReleaseEvent(self, a0)`

---

`move()`

---

`moveEvent(self, a0)`

---

`moveToThread(self, thread)`

---

`nativeEvent(self, eventType, message)`

---

`nativeParentWidget(self)`

---

`nextInFocusChain(self)`

---

`normalGeometry(self)`

---

`objectName(self)`

---

continues on next page

Table 24 – continued from previous page

<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	

continues on next page



Table 24 – continued from previous page

---

<code>resize()</code>
<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>sender(self)</code>
<code>senderSignalIndex(self)</code>
<code>setAcceptDrops(self, on)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setEnabled(self, a0)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>
<code>setFocus()</code>

---

continues on next page

Table 24 – continued from previous page

---

`setFocusPolicy(self, policy)`

---

`setFocusProxy(self, a0)`

---

`setFont(self, a0)`

---

`setForegroundRole(self, a0)`

---

`setGeometry()`

---

`setGraphicsEffect(self, effect)`

---

`setHidden(self, hidden)`

---

`setInputMethodHints(self, hints)`

---

`setLayout(self, a0)`

---

`setLayoutDirection(self, direction)`

---

`setLocale(self, locale)`

---

`setMask()`

---

`setMaximumHeight(self, maxh)`

---

`setMaximumSize()`

---

`setMaximumWidth(self, maxw)`

---

`setMinimumHeight(self, minh)`

---

`setMinimumSize()`

---

`setMinimumWidth(self, minw)`

---

`setMouseTracking(self, enable)`

---

`setObjectName(self, name)`

---

`setPalette(self, a0)`

---

`setParent()`

---

`setProperty(self, name, value)`

---

`setShortcutAutoRepeat(self, id[, en-  
abled])`

---

continues on next page

Table 24 – continued from previous page

---

<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>

---

continues on next page

Table 24 – continued from previous page

---

<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>

---

continues on next page

Table 24 – continued from previous page

---

`underMouse(self)`

---

`ungrabGesture(self, type)`

---

`unsetCursor(self)`

---

`unsetLayoutDirection(self)`

---

`unsetLocale(self)`

---

`update(-> None -> None)`

---

`updateGeometry(self)`

---

`updateMicroFocus(self)`

---

`updatesEnabled(self)`

---

`visibleRegion(self)`

---

`whatsThis(self)`

---

`wheelEvent(self, a0)`

---

`width(self)`

---

`widthMM(self)`

---

`winId(self)`

---

`window(self)`

---

`windowFilePath(self)`

---

`windowFlags(self)`

---

`windowHandle(self)`

---

`windowIcon(self)`

---

`windowIconText(self)`

---

`windowModality(self)`

---

`windowOpacity(self)`

---

`windowRole(self)`

---

continues on next page

Table 24 – continued from previous page

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

## napari\_locan.widgets.widget\_slm\_data

The data model for localization-based SMLM data.

QWidget plugin to access SMLM datasets from which images are rendered and localization-based analysis procedures are computed. Each dataset is kept as locdata, i.e. a `locan.LocData` object with metadata, aggregated properties, and localization properties for all localizations.

### Classes

---

```
SmlmDataQWidget(napari_viewer[,  
smlm_data])
```

---

## napari\_locan.widgets.widget\_slm\_data.SmlmDataQWidget

```
class napari_locan.widgets.widget_slm_data.SmlmDataQWidget(napari_viewer,  
                                                    smlm_data=<napari_locan.data_model.  
                                                    object>)
```

Bases: `PyQt5.QtWidgets.QWidget`

### Methods

---

```
__init__(napari_viewer[, smlm_data])
```

---

```
acceptDrops(self)
```

---

```
accessibleDescription(self)
```

---

```
accessibleName(self)
```

---

```
actionEvent(self, a0)
```

---

```
actions(self)
```

---

```
activateWindow(self)
```

---

```
addAction(self, action)
```

---

```
addActions(self, actions)
```

---

```
adjustSize(self)
```

---

```
autoFillBackground(self)
```

---

```
backgroundRole(self)
```

---

continues on next page



Table 25 – continued from previous page

---

<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>
<code>depth(self)</code>

---

continues on next page

Table 25 – continued from previous page

---

<code>destroy(self[, destroyWindow, destroySub- Windows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>
<code>dropEvent(self, a0)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, a0)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>

---

continues on next page

Table 25 – continued from previous page

---

`focusNextPrevChild(self, next)`

---

`focusOutEvent(self, a0)`

---

`focusPolicy(self)`

---

`focusPreviousChild(self)`

---

`focusProxy(self)`

---

`focusWidget(self)`

---

`font(self)`

---

`fontInfo(self)`

---

`fontMetrics(self)`

---

`foregroundRole(self)`

---

`frameGeometry(self)`

---

`frameSize(self)`

---

`geometry(self)`

---

`getContentsMargins(self)`

---

`grab(self[, rectangle])`

---

`grabGesture(self, type[, flags])`

---

`grabKeyboard(self)`

---

`grabMouse()`

---

`grabShortcut(self, key[, context])`

---

`graphicsEffect(self)`

---

`graphicsProxyWidget(self)`

---

`hasFocus(self)`

---

`hasHeightForWidth(self)`

---

`hasMouseTracking(self)`

---

continues on next page

Table 25 – continued from previous page

---

`hasTabletTracking(self)`

---

`height(self)`

---

`heightForWidth(self, a0)`

---

`heightMM(self)`

---

`hide(self)`

---

`hideEvent(self, a0)`

---

`inherits(self, classname)`

---

`initPainter(self, painter)`

---

`inputMethodEvent(self, a0)`

---

`inputMethodHints(self)`

---

`inputMethodQuery(self, a0)`

---

`insertAction(self, before, action)`

---

`insertActions(self, before, actions)`

---

`installEventFilter(self, a0)`

---

`isActiveWindow(self)`

---

`isAncestorOf(self, child)`

---

`isEnabled(self)`

---

`isEnabledTo(self, a0)`

---

`isFullScreen(self)`

---

`isHidden(self)`

---

`isLeftToRight(self)`

---

`isMaximized(self)`

---

`isMinimized(self)`

---

`isModal(self)`

---

continues on next page

Table 25 – continued from previous page

---

<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, signal)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, a0)</code>
<code>isWidgetType(self)</code>
<code>isWindow(self)</code>
<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>keyPressEvent(self, a0)</code>
<code>keyReleaseEvent(self, a0)</code>
<code>keyboardGrabber()</code>
<code>killTimer(self, id)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code>leaveEvent(self, a0)</code>
<code>locale(self)</code>
<code>logicalDpiX(self)</code>
<code>logicalDpiY(self)</code>
<code>lower(self)</code>
<code>mapFrom(self, a0, a1)</code>
<code>mapFromGlobal(self, a0)</code>
<code>mapFromParent(self, a0)</code>
<code>mapTo(self, a0, a1)</code>
<code>mapToGlobal(self, a0)</code>

---

continues on next page

Table 25 – continued from previous page

---

`mapToParent(self, a0)`

---

`mask(self)`

---

`maximumHeight(self)`

---

`maximumSize(self)`

---

`maximumWidth(self)`

---

`metaObject(self)`

---

`metric(self, a0)`

---

`minimumHeight(self)`

---

`minimumSize(self)`

---

`minimumSizeHint(self)`

---

`minimumWidth(self)`

---

`mouseDoubleClickEvent(self, a0)`

---

`mouseGrabber()`

---

`mouseMoveEvent(self, a0)`

---

`mousePressEvent(self, a0)`

---

`mouseReleaseEvent(self, a0)`

---

`move()`

---

`moveEvent(self, a0)`

---

`moveToThread(self, thread)`

---

`nativeEvent(self, eventType, message)`

---

`nativeParentWidget(self)`

---

`nextInFocusChain(self)`

---

`normalGeometry(self)`

---

`objectName(self)`

---

continues on next page

Table 25 – continued from previous page

<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	

continues on next page

Table 25 – continued from previous page

---

<code>resize()</code>
<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>sender(self)</code>
<code>senderSignalIndex(self)</code>
<code>setAcceptDrops(self, on)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setEnabled(self, a0)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>
<code>setFocus()</code>

---

continues on next page



Table 25 – continued from previous page

---

`setFocusPolicy(self, policy)`

---

`setFocusProxy(self, a0)`

---

`setFont(self, a0)`

---

`setForegroundRole(self, a0)`

---

`setGeometry()`

---

`setGraphicsEffect(self, effect)`

---

`setHidden(self, hidden)`

---

`setInputMethodHints(self, hints)`

---

`setLayout(self, a0)`

---

`setLayoutDirection(self, direction)`

---

`setLocale(self, locale)`

---

`setMask()`

---

`setMaximumHeight(self, maxh)`

---

`setMaximumSize()`

---

`setMaximumWidth(self, maxw)`

---

`setMinimumHeight(self, minh)`

---

`setMinimumSize()`

---

`setMinimumWidth(self, minw)`

---

`setMouseTracking(self, enable)`

---

`setObjectName(self, name)`

---

`setPalette(self, a0)`

---

`setParent()`

---

`setProperty(self, name, value)`

---

`setShortcutAutoRepeat(self, id[, en-  
abled])`

---

continues on next page

Table 25 – continued from previous page

<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>

continues on next page

Table 25 – continued from previous page

---

<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>

---

continues on next page

Table 25 – continued from previous page

---

<code>underMouse(self)</code>
<code>ungrabGesture(self, type)</code>
<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>
<code>update(-&gt; None -&gt; None)</code>
<code>updateGeometry(self)</code>
<code>updateMicroFocus(self)</code>
<code>updatesEnabled(self)</code>
<code>visibleRegion(self)</code>
<code>whatsThis(self)</code>
<code>wheelEvent(self, a0)</code>
<code>width(self)</code>
<code>widthMM(self)</code>
<code>winId(self)</code>
<code>window(self)</code>
<code>windowFilePath(self)</code>
<code>windowFlags(self)</code>
<code>windowHandle(self)</code>
<code>windowIcon(self)</code>
<code>windowIconText(self)</code>
<code>windowModality(self)</code>
<code>windowOpacity(self)</code>
<code>windowRole(self)</code>

---

continues on next page

Table 25 – continued from previous page

---

`windowState(self)`

---

`windowTitle(self)`

---

`windowType(self)`

---

`x(self)`

---

`y(self)`

---

## Attributes

---

DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

---

## CHANGELOG

### 7.1 0.6 - dev

#### 7.1.1 New Features

- 

#### 7.1.2 API Changes

- 

#### 7.1.3 Bug Fixes

- 

#### 7.1.4 Other Changes and Additions

- 

### 7.2 0.5 - 2023-12-07

#### 7.2.1 Bug Fixes

- fix version readout with readthedocs
- fix use of new locan.colormaps module

## 7.2.2 Other Changes and Additions

- add GitHub action for deploying to PyPI and TestPyPI
- configure setuptools\_scm for branching model
- bump version requirements for dependencies

## 7.3 0.4.0 - 2023-11-08

### 7.3.1 New Features

- add widget to save and load project
- add button to get regions from locdata hulls

### 7.3.2 Other Changes and Additions

- add abstract base class for DataModels
- add data model for region specifications
- add data model for roi specifications
- add data model for filter specifications
- minor modifications like button rearrangement

## 7.4 0.3.0 - 2023-11-01

### 7.4.1 Bug Fixes

- add dependency for matplotlib<3.8.0

### 7.4.2 Other Changes and Additions

- additions to the documentation

## 7.5 0.2.0 - 2023-10-31

### 7.5.1 Bug Fixes

- correct requirements
- always open dialog with load button
- remove unknown\_file\_format from load options
- fix dockerfile
- fix readthedocs



## 7.5.2 Other Changes and Additions

- add to documentation

## 7.6 0.1.0 - 2023-10-29

### 7.6.1 New Features

- sample data for 2d
- data models from SmlmData and FilterSpecifications
- **widgets for**
  - SMLM data
  - Show metadata
  - Show properties
  - Show localization data
  - Show localization property distributions
  - Load
  - Filter specifications
  - Select
  - Region of interest
  - Render points 2D / 3D
  - Render image 2D / 3D
  - Render features of a SMLM dataset
  - Cluster
  - Render collection as 2D / 3D point cloud
  - Render collection features
  - Run script
- documentation via readthedocs

**LICENSE**

BSD 3-Clause License

Copyright (c) 2022-23, Biotechnologie und Biophysik - Universität Würzburg All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of napari-locan nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## DEVELOPMENT

We welcome any contributions for improving or further developing this package. However, please excuse that we are limited in time for development and support. Some things to keep in mind when adding code...

### 9.1 Install

A few extra libraries are needed for development:

```
pip install .[test,dev,docs]
```

### 9.2 Import Conventions

The following import conventions are used throughout Locan source code and documentation:

```
import locan as lc
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import scipy as sp
import pandas as pd
```

This is enforced through ruff following specifications in pyproject.toml.

### 9.3 Unit tests

For testing we use `py.test`.

A test suite is provided in `/src/napari_locan/_tests`.

Tests can also be run with `tox`.

## 9.4 Coverage

For measuring code coverage in testing we use `coverage.py`.

Configurations are kept in `pyproject.toml`.

## 9.5 Code checks

We use `black` for formatting and `ruff` for code linting.

Configurations are kept in `pyproject.toml`.

## 9.6 Versioning

We use `SemVer` for versioning. For all versions available, see the [releases in this repository](#).

## 9.7 Documentation

Documentation is provided as restructured text, myst markdown, and as docstrings within the code. HTML pages and other documentation formats are build using `Sphinx`.

We follow standard recommendations for [python documentation](#) and the [numpy conventions](#).

To update the documentation from sources delete `/docs/sources/generated` and run:

```
sphinx-build -b html -E YOUR_PATH\nnapari-locan\docs YOUR_PATH\nnapari-locan\  
↪docs\_build
```

## 9.8 Type hints

We try to make use of type checking using `mypy` as much as possible.

Configurations are kept in `pyproject.toml`.

## 9.9 To remember

- The plugin is strongly linked to `locan` and its development procedures.

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

## PYTHON MODULE INDEX

n

- napari\_locan, 8
- napari\_locan.data\_model, 8
- napari\_locan.data\_model.data\_model\_base, 8
- napari\_locan.data\_model.filter\_specifications, 14
- napari\_locan.data\_model.region\_specifications, 18
- napari\_locan.data\_model.roi\_specifications, 21
- napari\_locan.data\_model.smlm\_data, 25
- napari\_locan.sample\_data, 30
- napari\_locan.sample\_data.sample\_data, 31
- napari\_locan.scripts, 32
- napari\_locan.widgets, 32
- napari\_locan.widgets.widget\_clustering, 33
- napari\_locan.widgets.widget\_filter\_specifications, 48
- napari\_locan.widgets.widget\_load, 63
- napari\_locan.widgets.widget\_napari\_locan\_project, 78
- napari\_locan.widgets.widget\_property\_distribution, 93
- napari\_locan.widgets.widget\_render\_collection\_2d, 108
- napari\_locan.widgets.widget\_render\_collection\_features, 123
- napari\_locan.widgets.widget\_render\_features, 138
- napari\_locan.widgets.widget\_render\_image\_2d, 153
- napari\_locan.widgets.widget\_render\_image\_3d, 168
- napari\_locan.widgets.widget\_render\_points\_2d, 183
- napari\_locan.widgets.widget\_render\_points\_3d, 198
- napari\_locan.widgets.widget\_roi, 213
- napari\_locan.widgets.widget\_run\_script, 228
- napari\_locan.widgets.widget\_select, 243
- napari\_locan.widgets.widget\_show\_data, 258
- napari\_locan.widgets.widget\_show\_metadata, 279
- napari\_locan.widgets.widget\_show\_properties, 294
- napari\_locan.widgets.widget\_smlm\_data, 309

## INDEX

**A**

`append_item()` (*napari\_locan.data\_model.smlm\_data.SmlmData* method), 29

`append_item()` (*napari\_locan.data\_model.data\_model\_base.DataModel* method), 12

**C**

`ClusteringQWidget` (class in *napari\_locan.widgets.widget\_clustering*), 33

`columnCount()` (*napari\_locan.widgets.widget\_show\_data.TableModel* method), 279

`count` (*napari\_locan.data\_model.data\_model\_base.DataModel* attribute), 12

**D**

`data()` (*napari\_locan.widgets.widget\_show\_data.TableModel* method), 279

`DataModel` (class in *napari\_locan.data\_model.data\_model\_base*), 9

`dataset` (*napari\_locan.data\_model.data\_model\_base.DataModel* property), 12

`datasets` (*napari\_locan.data\_model.data\_model\_base.DataModel* property), 12

`datasets_changed_signal` (*napari\_locan.data\_model.data\_model\_base.DataModel* attribute), 12

`delete_all()` (*napari\_locan.data\_model.data\_model\_base.DataModel* method), 13

`delete_all()` (*napari\_locan.data\_model.smlm\_data.SmlmData* method), 29

`delete_item()` (*napari\_locan.data\_model.data\_model\_base.DataModel* method), 13

`delete_item()` (*napari\_locan.data\_model.smlm\_data.SmlmData* method), 29

**F**

`filter_condition` (*napari\_locan.data\_model.filter\_specifications.FilterSpecifications* property), 17

`FilterSpecifications` (class in *napari\_locan.data\_model.filter\_specifications*), 14

`FilterSpecificationsQWidget` (class in *napari\_locan.widgets.widget\_filter\_specifications*), 48

**H**

`headerData()` (*napari\_locan.widgets.widget\_show\_data.TableModel* method), 279

**I**

`index` (*napari\_locan.data\_model.data\_model\_base.DataModel* property), 13

`index` (*napari\_locan.data\_model.smlm\_data.SmlmData* property), 29

`index_changed_signal` (*napari\_locan.data\_model.data\_model\_base.DataModel* attribute), 13

`index_changed_signal` (*napari\_locan.data\_model.smlm\_data.SmlmData* attribute), 29

**L**

`LoadQWidget` (class in *napari\_locan.widgets.widget\_load*), 63

`locdata` (*napari\_locan.data\_model.smlm\_data.SmlmData* property), 30

`locdata_name` (*napari\_locan.data\_model.smlm\_data.SmlmData* property), 30

locdata_names	( <i>napari_locan.data_model.slm_data.SlmData</i> property), 30	<i>napari_locan.widgets.widget_render_collection_features</i> , 123
locdata_names_changed_signal	( <i>napari_locan.data_model.slm_data.SlmData</i> attribute), 30	<i>napari_locan.widgets.widget_render_features</i> , 138
locdatas	( <i>napari_locan.data_model.slm_data.SlmData</i> property), 30	<i>napari_locan.widgets.widget_render_image_2d</i> , 153
		<i>napari_locan.widgets.widget_render_image_3d</i> , 168
		<i>napari_locan.widgets.widget_render_points_2d</i> , 183
		<i>napari_locan.widgets.widget_render_points_3d</i> , 198
make_image_npc()	(in module <i>napari_locan.sample_data.sample_data</i> ), 31	<i>napari_locan.widgets.widget_roi</i> , 213
make_image_tubulin()	(in module <i>napari_locan.sample_data.sample_data</i> ), 31	<i>napari_locan.widgets.widget_run_script</i> , 228
make_points_npc()	(in module <i>napari_locan.sample_data.sample_data</i> ), 31	<i>napari_locan.widgets.widget_select</i> , 243
make_points_tubulin()	(in module <i>napari_locan.sample_data.sample_data</i> ), 32	<i>napari_locan.widgets.widget_show_data</i> , 258
module		<i>napari_locan.widgets.widget_show_metadata</i> , 279
<i>napari_locan</i>	, 8	<i>napari_locan.widgets.widget_show_properties</i> , 294
<i>napari_locan.data_model</i>	, 8	<i>napari_locan.widgets.widget_slm_data</i> , 309
<i>napari_locan.data_model.data_model_base</i>	, 8	
<i>napari_locan.data_model.filter_specifications</i>	(in module <i>napari_locan.data_model.data_model_base.DataModel</i> property), 13	
<i>napari_locan.data_model.region_specifications</i>	(in module <i>napari_locan.data_model.data_model_base.DataModel</i> property), 13	
<i>napari_locan.data_model.roi_specifications</i>	<i>times_changed_signal</i> ( <i>napari_locan.data_model.data_model_base.DataModel</i> attribute), 13	
<i>napari_locan.data_model.slm_data</i>	, 25	<i>napari_locan</i> module, 8
<i>napari_locan.sample_data</i>	, 30	<i>napari_locan.data_model</i> module, 8
<i>napari_locan.sample_data.sample_data</i>	, 31	<i>napari_locan.data_model.data_model_base</i> module, 8
<i>napari_locan.scripts</i>	, 32	<i>napari_locan.data_model.filter_specifications</i> module, 14
<i>napari_locan.widgets</i>	, 32	<i>napari_locan.data_model.region_specifications</i> module, 18
<i>napari_locan.widgets.widget_clustering</i>	, 33	<i>napari_locan.data_model.roi_specifications</i> module, 21
<i>napari_locan.widgets.widget_filter_specifications</i>	, 48	<i>napari_locan.data_model.slm_data</i> module, 25
<i>napari_locan.widgets.widget_load</i>	, 63	<i>napari_locan.sample_data</i> module, 30
<i>napari_locan.widgets.widget_napari_locan</i>	, 78	<i>napari_locan.sample_data.sample_data</i> module, 31
<i>napari_locan.widgets.widget_property_distribution</i>	, 93	
<i>napari_locan.widgets.widget_render_collection_2d</i>	, 108	



module, 31  
 napari\_locan.scripts  
   module, 32  
 napari\_locan.widgets  
   module, 32  
 napari\_locan.widgets.widget\_clustering  
   module, 33  
 napari\_locan.widgets.widget\_filter\_specifications  
   module, 48  
 napari\_locan.widgets.widget\_load  
   module, 63  
 napari\_locan.widgets.widget\_napari\_locan\_project  
   module, 78  
 napari\_locan.widgets.widget\_property\_distribution  
   module, 93  
 napari\_locan.widgets.widget\_render\_collection\_2d  
   module, 108  
 napari\_locan.widgets.widget\_render\_collection\_features  
   module, 123  
 napari\_locan.widgets.widget\_render\_features  
   module, 138  
 napari\_locan.widgets.widget\_render\_image\_2d  
   module, 153  
 napari\_locan.widgets.widget\_render\_image\_3d  
   module, 168  
 napari\_locan.widgets.widget\_render\_points\_2d  
   module, 183  
 napari\_locan.widgets.widget\_render\_points\_3d  
   module, 198  
 napari\_locan.widgets.widget\_roi  
   module, 213  
 napari\_locan.widgets.widget\_run\_script  
   module, 228  
 napari\_locan.widgets.widget\_select  
   module, 243  
 napari\_locan.widgets.widget\_show\_data  
   module, 258  
 napari\_locan.widgets.widget\_show\_metadata  
   module, 279  
 napari\_locan.widgets.widget\_show\_properties  
   module, 294  
 napari\_locan.widgets.widget\_smlm\_data  
   module, 309  
 NapariLocanProjectQWidget (class in *napari\_locan.widgets.widget\_napari\_locan\_project*),  
   78

**P**  
 PropertyDistributionQWidget (class in *napari\_locan.widgets.widget\_property\_distribution*),  
   93

**Q**  
 QABCMeta (class in *napari\_locan.data\_model.data\_model\_base*),  
   14

**R**  
 RegionSpecifications (class in *napari\_locan.data\_model.region\_specifications*),  
   18  
 RenderCollection2dQWidget (class in *napari\_locan.widgets.widget\_render\_collection\_2d*),  
   108  
 RenderCollectionFeaturesQWidget (class in *napari\_locan.widgets.widget\_render\_collection\_features*),  
   138  
 RenderFeaturesQWidget (class in *napari\_locan.widgets.widget\_render\_features*),  
   138  
 RenderImage2dQWidget (class in *napari\_locan.widgets.widget\_render\_image\_2d*),  
   153  
 RenderImage3dQWidget (class in *napari\_locan.widgets.widget\_render\_image\_3d*),  
   168  
 RenderPoints2dQWidget (class in *napari\_locan.widgets.widget\_render\_points\_2d*),  
   183  
 RenderPoints3dQWidget (class in *napari\_locan.widgets.widget\_render\_points\_3d*),  
   198  
 RoiQWidget (class in *napari\_locan.widgets.widget\_roi*), 213  
 RoiSpecifications (class in *napari\_locan.data\_model.roi\_specifications*),  
   22  
 rowCount() (napari\_locan.widgets.widget\_show\_data.TableModel method), 279  
 RunScriptQWidget (class in *napari\_locan.widgets.widget\_run\_script*),  
   228

**S**  
 SelectQWidget (class in *napari\_locan.widgets.widget\_select*),  
   243  
 set\_datasets\_and\_names() (napari\_locan.data\_model.data\_model\_base.DataModel method), 13  
 set\_index\_slot() (napari\_locan.data\_model.data\_model\_base.DataModel

*method*), 14

`set_index_slot()` (*na-  
 pari\_locan.data\_model.smlm\_data.SmlmData  
 method*), 30

`ShowDataQWidget` (*class in na-  
 pari\_locan.widgets.widget\_show\_data*),  
 258

`ShowMetadataQWidget` (*class in na-  
 pari\_locan.widgets.widget\_show\_metadata*),  
 279

`ShowPropertiesQWidget` (*class in na-  
 pari\_locan.widgets.widget\_show\_properties*),  
 294

`SmlmData` (*class in na-  
 pari\_locan.data\_model.smlm\_data*),  
 26

`SmlmDataQWidget` (*class in na-  
 pari\_locan.widgets.widget\_smlm\_data*),  
 309

## T

`TableModel` (*class in na-  
 pari\_locan.widgets.widget\_show\_data*),  
 273