
lyse

Release 0.1.dev160+g6d72bc7

labscript suite contributors

Oct 12, 2023

DOCUMENTATION

1	Introduction	3
1.1	The lyse API	3
1.2	lyse GUI	4
2	Examples	5
2.1	An analysis on a single shot	5
2.2	Single shot analysis with global file opening	6
2.3	An analysis on multiple shots	7
3	API Reference	9
3.1	lyse	9
3.2	lyse.analysis_subprocess	23
3.3	lyse.dataframe_utilities	46
3.4	lyse.figure_manager	47
3.5	lyse.tempfile2clipboard	49
3.6	lyse.__main__	49
4	<i>labscript suite</i> components	151
	Python Module Index	153
	Index	155

lyse is a component of the labsript suite. It is a combination API and GUI interface that leverages the API to run user provided analysis scripts of experiment shots. This documentation provides a brief outline of the use of lyse.

INTRODUCTION

Lyse is a data analysis system which gets *your code* running on experimental data as it is acquired. It is fundamentally based around the ideas of experimental *shots* and analysis *routines*. A shot is one trial of an experiment, and a routine is a Python script, written by you, that does something with the measurement data from one or more shots.

Analysis routines can be either *single-shot* or *multi-shot*. This determines what data and functions are available to your code when it runs. A single-shot routine has access to the data from only one shot, and functions available for saving results only to the hdf5 file for that shot. A multi-shot routine has access to the entire dataset from all the runs that are currently loaded into **lyse**, and has functions available for saving results to an hdf5 file which does not belong to any of the shots—it’s a file that exists only to save the “meta results”.

Actually things are far less magical than that. The only enforced difference between a single shot routine and a multi-shot routine is a single variable provided to your code when **lyse** runs it. Your code runs in a perfectly clean Python environment with this one exception: a variable in the global namespace called `path`, which is a path to an hdf5 file. If you have told **lyse** that your routine is a singleshot one, then this path will point to the hdf5 file for the current shot being analysed. On the other hand, if you’ve told **lyse** that your routine is a multishot one, then it will be the path to an h5 file that has been selected in **lyse** for saving results to.

The other differences listed above are conventions only (though **lyse**’s design is based around the assumption that you’ll follow these conventions most of the time), and pertain to how you use the API that **lyse** provides, which will be different depending on what sort of analysis you’re doing.

1.1 The lyse API

So great, you’ve got a single filepath. What data analysis could you possibly do with that? It might seem like you have to still do the same amount of work that you would without an analysis system! Whilst that’s not quite true, it’s intentionally been designed that way so that you can run your code outside **lyse** with very little modification. Another motivating factor is to minimise the amount of magic black box behaviour, such that an analysis routine is actually just an ordinary Python script which makes use of an API designed for our purposes. **lyse** is both a program which executes your code, and an API that your code can call on.

To use the API in an analysis routine, begin your code with:

```
from lyse import *
```

The details of the API are found in the *API reference*.

1.2 lyse GUI

The **lyse** GUI uses the API to apply single and multi-shot routines to collections of shot files, added either manually by the user or automatically by BLACS after shot completion.

Here's a screenshot of **lyse**:

Fig. 1.1: Screenshot of the Lyse GUI

1. Here's where single shot routines can be added and removed, with the plus and minus buttons. They will be executed in order on each shot (more on how that works shortly). They can be reordered, or enabled/disabled with the checkboxes on the left. The checkboxes to the right, underneath the plot icons don't currently do anything, but they are intended to provide control over how plots generated by the analysis routines are displayed and updated.
2. Here is where multi-shot routines can be added or removed. The file selection button at the top allows you to select what hdf5 file multi-shot routines will get given (to which they will save their results).
3. Allows pausing of analysis. **lyse** by default will run all single-shot routines on a shot when it arrives (either via the HTTP server or having been manually added). After all the shots have been processed, only then will the multi-shot routines be executed. So if you load ten shots in quickly, the multi-shot routines won't run until they've all been processed by the single-shot routines. However most of the time there will be sufficient delay in between shots arriving that multi-shot routines will be executed pretty much every time a new shot arrives.
4. If you want to re-run single-shot analyses on some shots, select them and click this button. They'll then be processed in order.
5. This will rerun all the multi-shot analyses.
6. Here is where shots appear, either having arrived over HTTP or having been added manually via the file browser (by clicking the plus button). Many columns will populate this part of the screen, one for each global and each of the results (as saved by single-shot routines) present in the shots. A high-priority planned feature is to be able to choose exactly which globals and results are displayed. Otherwise this display is overwhelming to the point of uselessness. The data displayed here represents the entirety of what is available to multi-shot routines via the API provided by **lyse**.
7. This is where the output of routines is displayed, errors in red. If you're putting `print` statements in your analysis code, here is where to look to see them. Likewise if there's an exception and analysis stops, look here to see why.

EXAMPLES

2.1 An analysis on a single shot

```
from lyse import *
from pylab import *

# Let's obtain our data for this shot -- globals, image attributes and
# the results of any previously run single-shot routines:
ser = data(path)

# Get a global called x:
x = ser['x']

# Get a result saved by another single-shot analysis routine which has
# already run. The result is called 'y', and the routine was called
# 'some_routine':
y = ser['some_routine','y']

# Image attributes are also stored in this series:
w_x2 = ser['side','absorption','OD','Gaussian_XW']

# If we want actual measurement data, we'll have to instantiate a Run object:
run = Run(path)

# Obtaining a trace:
t, mot_fluorecence = run.get_trace('mot fluorecence')

# Now we might do some analysis on this data. Say we've written a
# linear fit function (or we're calling some other libraries linear
# fit function):
m, c = linear_fit(t, mot_fluorecence)

# We might wish to plot the fit on the trace to show whether the fit is any good:

plot(t,mot_fluorecence,label='data')
plot(t,m*t + c,label='linear fit')
xlabel('time')
ylabel('MOT flourescence')
legend()
```

(continues on next page)

(continued from previous page)

```
# Don't call show() ! lyse will introspect what figures have been made
# and display them once this script has finished running. If you call
# show() it won't find anything. lyse keeps track of figures so that new
# figures replace old ones, rather than you getting new window popping
# up every time your script runs.

# We might wish to save this result so that we can compare it across
# shots in a multishot analysis:
run.save_result('mot loadrate', c)
```

2.2 Single shot analysis with global file opening

```
from lyse import *

# Instantiate Run object and open
# Globally opening the shot keeps the h5 file open
# This prevents excessive opening and closing of the file
# which can slow down the analysis
with Run(path).open('r+') as shot:

    # Obtaining a trace:
    t, mot_fluorecence = shot.get_trace('mot fluorecence')

    # Now we might do some analysis on this data. Say we've written a
    # linear fit function (or we're calling some other libraries linear
    # fit function):
    m, c = linear_fit(t, mot_fluorecence)
    int_tot = mot_fluorecence.sum()
    mf_min = mot_fluorecence.min()
    mf_max = mot_fluorecence.max()

    normalised_fluorecence = normalise_response(mot_fluorecence)

    # We might wish to save this result so that we can compare it across
    # shots in a multishot analysis:
    shot.save_result('mot loadrate', c)
    shot.save_result('mot integrated', int_tot)
    shot.save_results('mot fl min', mf_min, 'mot fl max', mf_max)
    shot.save_result_array('norm mot fluorecence', normalised_fluorecence)
```

2.3 An analysis on multiple shots

```
from lyse import *
from pylab import *

# Let's obtain the dataframe for all of lyse's currently loaded shots:
df = data()

# Now let's see how the MOT load rate varies with, say a global called
# 'detuning', which might be the detuning of the MOT beams:

detunings = df['detuning']

# mot load rate was saved by a routine called calculate_load_rate:

load_rates = df['calculate_load_rate', 'mot loadrate']

# Let's plot them against each other:

plot(detunings, load_rates, 'bo', label='data')

# Maybe we expect a linear relationship over the range we've got:
m, c = linear_fit(detunings, load_rates)
# (note, not a function provided by lyse, though I'm sure we'll have
# lots of stock functions like this available for import!)

plot(detunings, m*detunings + c, 'ro', label='linear fit')
legend()

#To save this result to the output hdf5 file, we have to instantiate a
#Sequence object:
seq = Sequence(path, df)
seq.save_result('detuning_loadrate_slope', c)
```


API REFERENCE

<i>lyse</i>	Lyse analysis API
<i>lyse.analysis_subprocess</i>	
<i>lyse.dataframe_utilities</i>	
<i>lyse.figure_manager</i>	
<i>lyse.tempfile2clipboard</i>	
<i>lyse.__main__</i>	Lyse GUI and supporting code

3.1 lyse

Lyse analysis API

Functions

<i>data</i> ([filepath, host, port, timeout, ...])	Get data from the lyse dataframe or a file.
<i>delay_results_return</i> ()	
<i>figure_to_clipboard</i> ([figure])	Copy a matplotlib figure to the clipboard as a png.
<i>get_plot_class</i> (identifier)	
<i>globals_diff</i> (run1, run2[, group])	Take a diff of the globals between two runs.
<i>open_file</i> (mode)	Decorator for lyse functions to allow using previously opened file with context manager.
<i>register_plot_class</i> (identifier, cls)	

3.1.1 lyse.data

`lyse.data(filepath=None, host='localhost', port=42519, timeout=5, n_sequences=None, filter_kwargs=None)`
Get data from the lyse dataframe or a file.

This function allows for either extracting information from a run's hdf5 file, or retrieving data from lyse's dataframe. If `filepath` is provided then data will be read from that file and returned as a pandas series. If `filepath` is not provided then the dataframe in lyse, or a portion of it, will be returned.

Often only part of the lyse dataframe is needed, so the `n_sequences` and `filter_kwargs` arguments provide ways to restrict what parts of the lyse dataframe are returned. The dataframe can be quite large, so only requesting a small part of it can speed up the execution of `lyse.data()` noticeably. Setting `n_sequences` makes this function return only the rows of the lyse dataframe that correspond to the `n_sequences` most recent sequences, where one sequence corresponds to one call to engage in runmanager. Additionally, the `Dataframe.filter()` method can be called on the dataframe before it is transmitted, and the arguments specified in `filter_kwargs` are passed to that method.

Parameters

- **filepath** (*str, optional*) – The path to a run's hdf5 file. If a value other than `None` is provided, then this function will return a pandas series containing data associated with the run. In particular it will contain the globals, singleshot results, multishot results, etc. that would appear in the run's row in the Lyse dataframe, but the values will be read from the file rather than extracted from the lyse dataframe. If `filepath` is `None`, then this function will instead return a section of the lyse dataframe. Note that if `filepath` is not `None`, then the other arguments will be ignored. Defaults to `None`.
- **host** (*str, optional*) – The address of the computer running lyse. Defaults to `'localhost'`.
- **port** (*int, optional*) – The port on which lyse is listening. Defaults to the entry for lyse's port in the labconfig, with a fallback value of 42519 if the labconfig has no such entry.
- **timeout** (*float, optional*) – The timeout, in seconds, for the communication with lyse. Defaults to 5.
- **n_sequences** (*int, optional*) – The maximum number of sequences to include in the returned dataframe where one sequence corresponds to one call to engage in runmanager. The dataframe rows for the most recent `n_sequences` sequences are returned. If the dataframe contains fewer than `n_sequences` sequences, then all rows will be returned. If set to `None`, then all rows are returned. Defaults to `None`.
- **filter_kwargs** (*dict, optional*) – A dictionary of keyword arguments to pass to the `Dataframe.filter()` method before the lyse dataframe is returned. For example to call `filter()` with `like='temperature'`, set `filter_kwargs` to `{'like':'temperature'}`. If set to `None` then `Dataframe.filter()` will not be called. See `pandas.DataFrame.filter()` for more information. Defaults to `None`.

Raises `ValueError` – If `n_sequences` isn't `None` or a nonnegative integer, then a `ValueError` is raised. Note that no `ValueError` is raised if `n_sequences` is greater than the number of sequences available. In that case as all available sequences are returned, i.e. the entire lyse dataframe is returned.

Returns If `filepath` is provided, then a pandas series with the data read from that file is returned. If `filepath` is omitted or set to `None` then the lyse dataframe, or a subset of it, is returned.

Return type `pandas.DataFrame` or `pandas.Series`

3.1.2 lyse.delay_results_return

`lyse.delay_results_return()`

3.1.3 lyse.figure_to_clipboard

`lyse.figure_to_clipboard(figure=None, **kwargs)`

Copy a matplotlib figure to the clipboard as a png.

If `figure` is `None`, the current figure will be copied. Copying the figure is implemented by calling `figure.savefig()` and then copying the image data from the resulting file. If `bbox_inches` keyword arg is not provided, `bbox_inches='tight'` will be used.

Parameters

- **figure** (`matplotlib.figure`, optional) – Figure to copy to the clipboard. If `None`, copies the current figure.
- ****kwargs** – Passed to `figure.savefig()` as `kwargs`.

3.1.4 lyse.get_plot_class

`lyse.get_plot_class(identifier)`

3.1.5 lyse.globals_diff

`lyse.globals_diff(run1, run2, group=None)`

Take a diff of the globals between two runs.

Uses the `labscript-utils:dict_diff` function.

Parameters

- **run1** (*Run*) – First Run to compare.
- **run2** (*Run*) – Second Run to compare.
- **group** (*str, optional*) – When `None` (default), compare all groups. Otherwise, only compare globals in `group`.

Returns Dictionary of differences between globals in the form `key:[val1,val2]` pairs. Keys unique to either dictionary are returned as `key:[val1, '-']` or `key:['-', val2]`.

Return type `dict`

3.1.6 lyse.open_file

`lyse.open_file(mode)`

Decorator for lyse functions to allow using previously opened file with context manager.

If multiple read/write operations happen on a `Run` in a single shot, opening the `h5file` once via the context manager `open` can speed up the analysis execution time by limiting the number of times the file must be opened/closed.

Note that an opened `h5py.File` can only be in modes `'r'` or `'r+'`. All other mode opening options differ in file creation logic only. In particular, all mode opening options except `'r'` are considered `'r+'` once the file is open.

Parameters `mode` (*str*) – which `h5py.File` mode to open the h5 file with. Must be ‘r’, ‘a’, ‘r+’, ‘w’, ‘w-’, or ‘x’. Lyse typically only uses ‘r’ and ‘r+’.

Returns Decorator for `Run` methods that need to read/write the shot file

Raises `PermissionError` – If the Run is set as read-only but a write mode is requested

3.1.7 lyse.register_plot_class

`lyse.register_plot_class(identifier, cls)`

Classes

<code>Run(h5_path[, no_write])</code>	A class for saving/retrieving data to/from a shot's hdf5 file.
<code>Sequence(h5_path, run_paths[, no_write])</code>	

3.1.8 lyse.Run

`class lyse.Run(h5_path, no_write=False)`

Bases: `object`

A class for saving/retrieving data to/from a shot's hdf5 file.

This class implements methods that allow the user to retrieve data from a shot's hdf5 file such as images, traces, and the values of globals. It also provides methods for saving and retrieving results from analysis.

Parameters

- `h5_path` (*str*) – The path, including file name and extension, to the hdf5 file for a shot.
- `no_write` (*bool, optional*) – Set to True to prevent editing the shot's hdf5 file. Note that doing so prohibits the ability to save results to the file. Defaults to False.

`__init__(h5_path, no_write=False)`

Methods

<code>__init__(h5_path[, no_write])</code>	
<code>get_all_image_labels()</code>	Return all existing images labels in the h5 file.
<code>get_attrs(group)</code>	Returns all attributes of the specified group as a dictionary.
<code>get_globals([group])</code>	Get globals from the shot.
<code>get_globals_expansion()</code>	Get the expansion type of each global.
<code>get_globals_raw([group])</code>	Get the raw global values from the shot.
<code>get_image(orientation, label, image)</code>	Get previously saved image from the h5 file.
<code>get_image_attributes(orientation)</code>	Return the attributes of a saved orientation image group.
<code>get_images(orientation, label, *images)</code>	Get multiple saved images from orientation and label.

continues on next page

Table 3.4 – continued from previous page

<code>get_images_dict(orientation, label, *images)</code>	Get multiple saved images from orientation and label.
<code>get_result(group, name)</code>	Retrieve result from prior calculation.
<code>get_result_array(group, name)</code>	Returns saved results data.
<code>get_result_arrays(group, *names)</code>	Retrieve multiple result arrays from the same group.
<code>get_results(group, *names)</code>	Return multiple results from the same group.
<code>get_trace(name[, raw_data])</code>	Return the saved data trace name.
<code>get_traces(*names)</code>	Retrieve multiple data traces.
<code>get_units([group])</code>	Get the units of globals.
<code>get_wait(name)</code>	Returns the wait parameters: label, timeout, duration, and time out status.
<code>get_waits()</code>	Returns the parameters of all waits in the experiment.
<code>globals_diff(other_run[, group])</code>	Take a diff between this run and another run.
<code>globals_groups()</code>	Get names of all the globals groups.
<code>open(mode)</code>	Context manager to open the Run's h5 file for successive reads/writes.
<code>save_result(name, value[, group, overwrite])</code>	Save a result to the hdf5 file.
<code>save_result_array(name, data[, group, ...])</code>	Save an array of data to the hdf5 h5 file.
<code>save_result_arrays(*args, **kwargs)</code>	Save multiple result arrays.
<code>save_results(*args, **kwargs)</code>	Save multiple results to the hdf5 file.
<code>save_results_dict(results_dict[, uncertainties])</code>	Save results dictionary.
<code>set_group(groupname)</code>	Set the default hdf5 file group for saving results.
<code>trace_names()</code>	Return a list of all saved data traces in Run.

Attributes

<code>group</code>	The group in the hdf5 file in which results are saved by default.
<code>h5_file</code>	opened h5py file handle for the shot
<code>h5_path</code>	The value provided for <code>h5_path</code> during instantiation.
<code>no_write</code>	The value provided for <code>no_write</code> during instantiation.

`get_all_image_labels()`

Return all existing images labels in the h5 file.

Returns Dictionary of the form `{orientation: [label1, label2]}`

Return type `dict`

`get_attrs(group)`

Returns all attributes of the specified group as a dictionary.

Parameters `group` (*str*) – Group for which attributes are desired.

Raises `Exception` – If the group does not exist.

Returns Dictionary of attributes.

Return type `dict`

`get_globals(group=None)`

Get globals from the shot.

Parameters `group` (*str*, *optional*) – If `None`, return all global variables. If defined, only return globals from `group`.

Returns Dictionary of globals and their values.

Return type `dict`

get_globals_expansion()

Get the expansion type of each global.

This will skip global variables that do not have an expansion.

Parameters `h5_file` (*h5py.File*, optional) – Allows passing in a handle to an already opened h5 file. If None, will open in read only mode and close after operation.

Returns Dictionary of globals with their expansion type.

Return type `dict`

get_globals_raw(group=None)

Get the raw global values from the shot.

Parameters `group` (*str*, optional) – If None, return all global variables. If defined, only return globals from `group`.

Returns Dictionary of raw globals and their values

Return type `dict`

get_image(orientation, label, image)

Get previously saved image from the h5 file.

h5 path to saved image is `/images/orientation/label/image`

Parameters

- **orientation** (*str*) – Orientation label for saved image.
- **label** (*str*) – Label of saved image.
- **image** (*str*) – Identifier of saved image.

Raises **Exception** – If the image or paths do not exist.

Returns 2-D image array.

Return type `numpy.ndarray`

get_image_attributes(orientation)

Return the attributes of a saved orientation image group.

Parameters **orientation** (*str*) – Orientation label to get attributes of.

Raises **Exception** – If images or orientation group do not exist.

Returns Dictionary of attributes and their values.

Return type `dict`

get_images(orientation, label, *images)

Get multiple saved images from orientation and label.

Iteratively calls `self.get_image(orientation, label, image)` for each image argument.

Parameters

- **orientation** (*str*) – Orientation label of saved images.
- **label** (*str*) – Label of saved images.
- ***images** (*str*) – Collection of images to return

Returns List of 2-D images.

Return type `list` of `numpy.ndarray`

get_images_dict(*orientation, label, *images*)

Get multiple saved images from orientation and label.

Iteratively calls `self.get_image(orientation, label, image)` for each image argument.

Parameters

- **orientation** (*str*) – Orientation label of saved images.
- **label** (*str*) – Label of saved images.
- ***images** (*str*) – Collection of images to return

Returns Dictionary of 2-D images.

Return type `dict` of `numpy.ndarray`

get_result(*group, name*)

Retrieve result from prior calculation.

Parameters

- **group** (*str*) – Group to look in for the result. Typically the name of the analysis script that created it.
- **name** (*str*) – Name of the result.

Raises **Exception** – If `group` or `name` do not already exist.

Returns Result with appropriate type, as determined by `labscript_utils.properties.get_attribute`.

get_result_array(*group, name*)

Returns saved results data.

Parameters

- **group** (*str*) – Group to look in for the array. Typically the name of the analysis script that created it.
- **name** (*str*) – Name of the results array to return.

Raises **Exception** – If `group` or `name` do not already exist.

Returns Numpy array of the saved data.

Return type `numpy.ndarray`

get_result_arrays(*group, *names*)

Retrieve multiple result arrays from the same group.

Iteratively calls `self.get_result_array(group, name)` with default arguments.

Parameters

- **group** (*str*) – Group to obtain the results from.
- ***names** (*str*) – Result names to retrieve.

Returns List of results.

Return type `list`

get_results(*group*, **names*)

Return multiple results from the same group.

Iteratively calls `get_result(group,name)` for each name provided.

Parameters

- **group** (*str*) – Group to look in for the results. Typically the name of the analysis script that created it.
- ***names** (*str*) – Names of results to retrieve.

Returns List of the results, in the same order as specified by names. If names does not preserve order, return order is not guaranteed.

Return type *list*

get_trace(*name*, *raw_data=False*)

Return the saved data trace name.

Parameters

- **name** (*str*) – Name of saved data trace to get.
- **raw_data** (*bool*, *optional*) – option to return the `h5_data` directly without interpreting it as a 2-D time trace.

Raises **Exception** – If name trace does not exist.

Returns Returns 2-D timetrace of times 't' and values 'values'.

Return type `numpy.ndarray`

get_traces(**names*)

Retrieve multiple data traces.

Iteratively calls `get_trace`.

Parameters ***names** (*str*) – Names of traces to retrieve

Returns List of numpy arrays.

Return type *list*

get_units(*group=None*)

Get the units of globals.

This method retrieves the values in the “Units” column of runmanager for this shot. The values are returned in a dictionary where the keys are the names of globals and the values are the corresponding units.

Parameters **group** (*str*, *optional*) – The name of the globals group for which the units will be retrieved. Globals and units from other globals groups will not be included in the returned dictionary. If set to `None` then all globals from all globals groups will be returned. If **group** is set to a value that isn't the name of a globals group, then an empty dictionary will be returned, but no error will be raised. Defaults to `None`.

Returns A dictionary in which each key is a string giving the name of a global, and each value is a string specifying the corresponding value in the “Units” column of runmanager. An empty dictionary will be returned if **group** is set to a value that isn't the name of a globals group.

Return type *dict*

get_wait(*name*)

Returns the wait paramteres: label, timeout, duration, and time out status.

Parameters **name** (*str*) – Name of the wait to get.

Raises `KeyError` if name `wait` does not exist. –

Returns Tuple of the wait parameters.

Return type `tuple`

`get_waits()`

Returns the parameters of all waits in the experiment.

Raises `Exception` – If the experiment has no waits.

Returns Returns 2D structured numpy array of the waits and their parameters.

Return type `numpy.ndarray`

`globals_diff(other_run, group=None)`

Take a diff between this run and another run.

This calls `globals_diff(self, other_run, group)`.

Parameters

- **`other_run`** (`Run`) – Run to compare to.
- **`group`** (`str`, `optional`) – When `None` (default), compare all globals. Otherwise only compare globals in `group`.

Returns Dictionary of different globals.

Return type `dict`

`globals_groups()`

Get names of all the globals groups.

Returns List of global group names.

Return type `list`

property `group`

The group in the hdf5 file in which results are saved by default.

When a `Run` instance is created from within a lyse singleshot or multishot routine, `group` will be set to the name of the running routine. If created from outside a lyse script it will be set to `None`. To change the default group for saving results, use the `set_group()` method. Note that if `self.group` is `None` and no value is provided for the optional `group` argument used by the `save...()` methods, a `ValueError` will be raised.

Attempting to directly set `self.group`'s value will automatically call `self.set_group()`.

Type `str`

property `h5_file`

opened h5py file handle for the shot

Type `h5py.File`

property `h5_path`

The value provided for `h5_path` during instantiation.

Type `str`

property `no_write`

The value provided for `no_write` during instantiation.

Type `bool`

open(mode)

Context manager to open the Run's h5 file for successive reads/writes.

Supports all h5 modes, though only 'r' and 'r+'/'a' are typically used within lyse itself.

Parameters *mode* (*str*) – which `h5py.File` mode to open the h5 file with. Must be 'r', 'a', 'r+', 'w', 'w-', or 'x'. Lyse typically only uses 'r' and 'r+'.

Yields *Run* – Handle to opened Run object.

Raises `PermissionError` – If the Run is set as read-only but a write mode is requested

Examples

Trivial example that selectively opens the file during analysis. For better performance, it would be better to combine these two openings into one.

```
>>> from lyse import *
>>> shot = Run(path)
>>> with shot.open('r'):
>>>     # shot processing that requires reads/writes
>>>     _, vals = shot.get_trace('my_trace')
>>> with shot.open('r+'):
>>>     results = vals**2
>>>     shot.save_result_array('my_result', results)
>>>     _, vals2 = shot.get_trace('my_other_trace')
>>>     shot.save_result('my_result2', vals2.mean())
>>> # Shot processing that doesn't require h5 reads/writes
>>> print(f'Max of results is {results.max():.3f}')
Max of results is 5.003
```

Open and create the shot handle for the whole analysis in a single line.

```
>>> from lyse import *
>>> with Run(path).open('r+') as shot:
>>>     # shot processing that requires reads/writes
>>>     t, vals = shot.get_trace('my_trace')
>>>     results = vals**2
>>>     shot.save_result_array('my_result', results)
>>>     # Shot processing that doesn't require h5 reads/writes
>>>     # could be outside the context
>>>     print(f'Max of results is {results.max():.3f}')
Max of results is 5.003
```

save_result(name, value, group=None, overwrite=True)

Save a result to the hdf5 file.

With the default argument values this method saves to `self.group` in the '/results' group and overwrites any existing value. Note that the result is saved as an attribute and overwriting attributes causes hdf5 file size bloat.

Parameters

- **name** (*str*) – The name of the result. This will be the name of the attribute added to the hdf5 file's group.
- **value** (*any*) – The value of the result, which will be saved as the value of the hdf5 group's attribute set by name. However note that when saving large arrays, it is better to use the

`self.save_result_array()` method which will store the results as a dataset in the hdf5 file.

- **group** (*str*, *optional*) – The group in the hdf5 file to which the result will be saved as an attribute. If set to `None`, then the result will be saved to `self.group` in `'/results'`. Note that if a value is passed for `group` here then it will NOT have `'/result'` prepended to it which allows the caller to save results anywhere in the hdf5 file. This is in contrast to using the default group set with `self.set_group()`; when the default group is set with that method it WILL have `'/results'` prepended to it when saving results. Defaults to `None`.
- **overwrite** (*bool*, *optional*) – Sets whether or not to overwrite the previous value if the attribute already exists. If set to `False` and the attribute already exists, a `PermissionError` is raised. Defaults to `True`.

Raises

- **PermissionError** – A `PermissionError` is raised if `self.no_write` is `True` because saving the result would edit the file.
- **ValueError** – A `ValueError` is raised if `self.group` is `None` and no value is provided for `group` because the method then doesn't know where to save the result.
- **PermissionError** – A `PermissionError` is raised if an attribute with name `name` already exists but `overwrite` is set to `False`.

save_result_array(*name*, *data*, *group=None*, *overwrite=True*, *keep_attrs=False*, ***kwargs*)

Save an array of data to the hdf5 h5 file.

With the default argument values this method saves to `self.group` in the `'/results'` group and overwrites any existing value without keeping the dataset's previous attributes. Additional keyword arguments are passed directly to `h5py.Group.create_dataset`.

Parameters

- **name** (*str*) – The name of the result. This will be the name of the dataset added to the hdf5 file.
- **data** (*numpy.array*) – The data to save to the hdf5 file.
- **group** (*str*, *optional*) – The group in the hdf5 file in which the result will be saved as a dataset. If set to `None`, then the result will be saved in `self.group` in `'/results'`. Note that if a value is passed for `group` here then it will NOT have `'/result'` prepended to it which allows the caller to save results anywhere in the hdf5 file. This is in contrast to using the default group set with `self.set_group()`; when the default group is set with that method it WILL have `'/results'` prepended to it when saving results. Defaults to `None`.
- **overwrite** (*bool*, *optional*) – Sets whether or not to overwrite the previous value if the dataset already exists. If set to `False` and the dataset already exists, a `PermissionError` is raised. Defaults to `True`.
- **keep_attrs** (*bool*, *optional*) – Whether or not to keep the dataset's attributes when overwriting it, i.e. if the dataset already existed. Defaults to `False`.
- ****kwargs** – Optional kwargs passed directly to `h5_file.create_dataset()`

Raises

- **PermissionError** – A `PermissionError` is raised if `self.no_write` is `True` because saving the result would edit the file.

- **ValueError** – A ValueError is raised if `self.group` is None and no value is provided for `group` because the method then doesn't know where to save the result.
- **PermissionError** – A PermissionError is raised if a dataset with name `name` already exists but `overwrite` is set to False.

save_result_arrays(*args, **kwargs)

Save multiple result arrays.

Iteratively calls `save_result_array()` on multiple data sets. Assumes arguments are ordered such that each dataset to be saved is preceded by the name to save it as. All keyword arguments are passed to each call of `save_result_array()`.

Parameters

- ***args** – Ordered arguments such that each dataset to be saved is preceded by the name to save it as.
- ****kwargs** – Passed through to `save_result_array` as `kwargs`.

save_results(*args, **kwargs)

Save multiple results to the hdf5 file.

This method iteratively calls `self.save_result()` on multiple results. It assumes arguments are ordered such that each result to be saved is preceded by the name of the attribute to save it under. Keywords arguments are passed to each call of `self.save_result()`.

Parameters

- ***args** – The names and values of results to be saved. The first entry should be a string giving the name of the first result, and the second entry should be the value for that result. After that, an arbitrary number of additional pairs of result name strings and values can be included, e.g. 'name0', value0, 'name1', value1, ...
- ****kwargs** – Keyword arguments are passed to `self.save_result()`. Note that the names and values of keyword arguments are NOT saved as results to the hdf5 file; they are only used to provide values for the optional arguments of `self.save_result()`.

Examples

```
>>> run = Run('path/to/an/hdf5/file.h5')
>>> a = 5
>>> b = 2.48
>>> run.save_results('result', a, 'other_result', b, overwrite=False)
```

save_results_dict(results_dict, uncertainties=False, **kwargs)

Save results dictionary.

Iteratively calls `self.save_result(key, value)` on the provided dictionary. If `uncertainties` is True, `value` is a two-element list where the second element is the uncertainty in the result and saved with to the same key with `u_` prepended.

Parameters

- **results_dict** (*dict*) – Dictionary of results to save. If `uncertainties` is False, form is {name:value}. If True, for is {name, [value, uncertainty]}.
- **uncertainties** (*bool*, *optional*) – Marks if uncertainties are provided.
- ****kwargs** – Extra arguments provided to `save_result`.

set_group(*groupname*)

Set the default hdf5 file group for saving results.

The `save...()` methods will save their results to `self.group` if an explicit value for their optional `group` argument is not given. This method updates `self.group`, making sure to create the group in the hdf5 file if it does not already exist.

Parameters `groupname` (*str*) – The name of the hdf5 file group in which to save results by default. The group will be created in the `'/results'` group of the hdf5 file.

trace_names()

Return a list of all saved data traces in Run.

Raises `KeyError` – If the group `'/data/traces/'` does not yet exist.

Returns List of keys in the h5 file's `'/data/traces/'` group.

Return type `list`

3.1.9 lyse.Sequence

class `lyse.Sequence`(*h5_path, run_paths, no_write=False*)

Bases: `lyse.Run`

__init__(*h5_path, run_paths, no_write=False*)

Generic results storage that is not associated with a specific Run.

This is typically used to save results from a multi-shot analysis to an independent h5 file.

Parameters

- **h5_path** (*str*) – Path to h5 file to save to. If file does not exist, will try to create it assuming `no_write=False`. If file exists, opens a handle to it.
- **run_paths** (`list` or `pandas.DataFrame`) – List of runs to associate with the sequence. If a dataframe is supplied, will introspect the runs from the `'filepath'` data.
- **no_write** (*bool, optional*) – If True, opens file in read-only mode.

Raises `PermissionError` – If trying to create a file in read-only mode.

Methods

<code>__init__(h5_path, run_paths[, no_write])</code>	Generic results storage that is not associated with a specific Run.
<code>get_all_image_labels()</code>	Return all existing images labels in the h5 file.
<code>get_attrs(group)</code>	Returns all attributes of the specified group as a dictionary.
<code>get_globals([group])</code>	Get globals from the shot.
<code>get_globals_expansion()</code>	Get the expansion type of each global.
<code>get_globals_raw([group])</code>	Get the raw global values from the shot.
<code>get_image(*args)</code>	Not implemented!
<code>get_image_attributes(orientation)</code>	Return the attributes of a saved orientation image group.
<code>get_images(orientation, label, *images)</code>	Get multiple saved images from orientation and label.
<code>get_images_dict(orientation, label, *images)</code>	Get multiple saved images from orientation and label.
<code>get_result(group, name)</code>	Retrieve result from prior calculation.

continues on next page

Table 3.6 – continued from previous page

<code>get_result_array(*args)</code>	Get the specified result array from each run in the sequence.
<code>get_result_arrays(*args)</code>	Not implemented!
<code>get_results(group, *names)</code>	Return multiple results from the same group.
<code>get_trace(*args)</code>	Get the named trace from each run in the sequence.
<code>get_traces(*args)</code>	Not implemented!
<code>get_units([group])</code>	Get the units of globals.
<code>get_wait(name)</code>	Returns the wait parameters: label, timeout, duration, and time out status.
<code>get_waits()</code>	Returns the parameters of all waits in the experiment.
<code>globals_diff(other_run[, group])</code>	Take a diff between this run and another run.
<code>globals_groups()</code>	Get names of all the globals groups.
<code>open(mode)</code>	Context manager to open the Run's h5 file for successive reads/writes.
<code>save_result(name, value[, group, overwrite])</code>	Save a result to the hdf5 file.
<code>save_result_array(name, data[, group, ...])</code>	Save an array of data to the hdf5 h5 file.
<code>save_result_arrays(*args, **kwargs)</code>	Save multiple result arrays.
<code>save_results(*args, **kwargs)</code>	Save multiple results to the hdf5 file.
<code>save_results_dict(results_dict[, uncertainties])</code>	Save results dictionary.
<code>set_group(groupname)</code>	Set the default hdf5 file group for saving results.
<code>trace_names()</code>	Return a list of all saved data traces in Run.

Attributes

<code>group</code>	The group in the hdf5 file in which results are saved by default.
<code>h5_file</code>	opened h5py file handle for the shot
<code>h5_path</code>	The value provided for <code>h5_path</code> during instantiation.
<code>no_write</code>	The value provided for <code>no_write</code> during instantiation.

`get_image(*args)`
Not implemented!

Attention: Not implemented, but could be.

`get_result_array(*args)`
Get the specified result array from each run in the sequence.

Parameters `*args` (*str*) – Passed directly to `get_result_array`. Should be group and name to result to obtain.

Returns Dictionary of path:result pairs for each run.

Return type `dict`

`get_result_arrays(*args)`
Not implemented!

Attention: Not implemented, but could be.

get_trace(*args)

Get the named trace from each run in the sequence.

Parameters *args (*str*) – Name of trace. Passed directly to *get_trace*.

Returns Dictionary of path:trace pairs for each run.

Return type dict

get_traces(*args)

Not implemented!

Attention: Not implemented, but could be.

3.2 lyse.analysis_subprocess

Classes

AnalysisWorker(filepath, to_parent, from_parent)

Plot(figure, identifier, filepath)

PlotWindow(plot, *args, **kwargs)

PlotWindowCloseEvent(force, *args, **kwargs)

3.2.1 lyse.analysis_subprocess.AnalysisWorker

class lyse.analysis_subprocess.**AnalysisWorker**(filepath, to_parent, from_parent)

Bases: object

__init__(filepath, to_parent, from_parent)

Methods

__init__(filepath, to_parent, from_parent)

do_analysis(path)

mainloop()

new_figure(fig, identifier)

continues on next page

Table 3.9 – continued from previous page

`post_analysis_plot_actions()`

`pre_analysis_plot_actions()`

`reset_figs()`

`do_analysis(path)``mainloop()``new_figure(fig, identifier)``post_analysis_plot_actions()``pre_analysis_plot_actions()``reset_figs()`

3.2.2 lyse.analysis_subprocess.Plot

class `lyse.analysis_subprocess.Plot`(*figure, identifier, filepath*)Bases: `object``__init__`(*figure, identifier, filepath*)

Methods

`__init__`(*figure, identifier, filepath*)

`analysis_complete`(*figure_in_use*) To be overridden by subclasses.

`clear`()

`draw`()

`get_window_state`() Called when the Plot window is about to be closed due to a change in registered Plot window class

`on_close`() Called when the window is closed.

`on_copy_to_clipboard_triggered`()

`on_lock_axes_triggered`()

`restore_axis_limits`()

`restore_window_state`(*state*) Called when the Plot window is recreated due to a change in registered Plot window class.

`save_axis_limits`()

`set_window_title`(*identifier, filepath*)

`show`()

continues on next page

Table 3.10 – continued from previous page

`update_window_size()`

Attributes

`is_shown`

analysis_complete(*figure_in_use*)

To be overridden by subclasses. Called as part of the post analysis plot actions

clear()**draw**()**get_window_state**()

Called when the Plot window is about to be closed due to a change in registered Plot window class

Can be overridden by subclasses if custom information should be saved (although bear in mind that you will be passing the information from the previous Plot subclass which might not be what you want unless the old and new classes have a common ancestor, or the change in Plot class is triggered by a reload of the module containing your Plot subclass).

Returns a dictionary of information on the window state.

If you have overridden this method, please call the base method first and then update the returned dictionary with your additional information before returning it from your method.

property is_shown**on_close**()

Called when the window is closed.

Note that this only happens if the Plot window class has changed. Clicking the “X” button in the window title bar has been overridden to hide the window instead of closing it.

on_copy_to_clipboard_triggered()**on_lock_axes_triggered**()**restore_axis_limits**()**restore_window_state**(*state*)

Called when the Plot window is recreated due to a change in registered Plot window class.

Can be overridden by subclasses if custom information should be restored (although bear in mind that you will get the information from the previous Plot subclass which might not be what you want unless the old and new classes have a common ancestor, or the change in Plot class is triggered by a reload of the module containing your Plot subclass).

If overriding, please call the parent method in addition to your new code

Parameters *state* – A dictionary of information to restore

save_axis_limits()**set_window_title**(*identifier, filepath*)**show**()**update_window_size**()

3.2.3 lyse.analysis_subprocess.PlotWindow

class lyse.analysis_subprocess.PlotWindow(plot, *args, **kwargs)

Bases: PyQt5.QtWidgets.QWidget

__init__(plot, *args, **kwargs)

Methods

__init__(plot, *args, **kwargs)

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)

changeEvent(self, a0)

childAt(-> QWidget)

childEvent(self, a0)

children(self)

childrenRect(self)

childrenRegion(self)

clearFocus(self)

continues on next page

Table 3.12 – continued from previous page

<code>clearMask(self)</code>
<code>close(self)</code>
<code><i>closeEvent</i>(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, destroySubWindows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-> bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>
<code>dragMoveEvent(self, a0)</code>

continues on next page

Table 3.12 – continued from previous page

<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(-> QObject)</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>

continues on next page

Table 3.12 – continued from previous page

<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>
<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>

continues on next page

Table 3.12 – continued from previous page

<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>
<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>

continues on next page

Table 3.12 – continued from previous page

<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>
<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>

continues on next page

Table 3.12 – continued from previous page

<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>	
<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 3.12 – 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(-> None -> 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>

continues on next page

Table 3.12 – continued from previous page

<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>
<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>

continues on next page

Table 3.12 – continued from previous page

<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[, 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>

continues on next page

Table 3.12 – continued from previous page

<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>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>

continues on next page

Table 3.12 – 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(-> None -> 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>

continues on next page

Table 3.12 – continued from previous page

<code>windowIconText(self)</code>	
<code>windowModality(self)</code>	
<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>	
<hr/>	
Attributes	
<hr/>	
<code>DrawChildren</code>	
<code>DrawWindowBackground</code>	
<code>IgnoreMask</code>	
<code>PdmDepth</code>	
<code>PdmDevicePixelRatio</code>	
<code>PdmDevicePixelRatioScaled</code>	
<code>PdmDpiX</code>	
<code>PdmDpiY</code>	
<code>PdmHeight</code>	
<code>PdmHeightMM</code>	
<code>PdmNumColors</code>	
<code>PdmPhysicalDpiX</code>	
<code>PdmPhysicalDpiY</code>	
<code>PdmWidth</code>	

continues on next page

Table 3.13 – continued from previous page

PdmWidthMM	
<i>close_signal</i>	
customContextMenuRequested	QPoint) [signal]
destroyed	typing.Optional[QObject] = None) [signal]
objectNameChanged	str) [signal]
staticMetaObject	
windowIconChanged	QIcon) [signal]
windowIconTextChanged	str) [signal]
windowTitleChanged	str) [signal]

closeEvent(*self*, *a0*: *QCloseEvent*)

close_signal

3.2.4 lyse.analysis_subprocess.PlotWindowCloseEvent

class lyse.analysis_subprocess.PlotWindowCloseEvent(*force*, **args*, ***kwargs*)

Bases: PyQt5.QtGui.QCloseEvent

__init__(*force*, **args*, ***kwargs*)

Methods

__init__(*force*, **args*, ***kwargs*)

accept(*self*)

ignore(*self*)

isAccepted(*self*)

registerEventType([*hint*])

setAccepted(*self*, *accepted*)

spontaneous(*self*)

type(*self*)

Attributes

ActionAdded

ActionChanged

ActionRemoved

ActivationChange

ApplicationActivate

ApplicationActivated

ApplicationDeactivate

ApplicationDeactivated

ApplicationFontChange

ApplicationLayoutDirectionChange

ApplicationPaletteChange

ApplicationStateChange

ApplicationWindowIconChange

ChildAdded

ChildPolished

ChildRemoved

Clipboard

Close

CloseSoftwareInputPanel

ContentsRectChange

ContextMenu

CursorChange

DeferredDelete

DragEnter

continues on next page

Table 3.15 – continued from previous page

DragLeave
DragMove
Drop
DynamicPropertyChange
EnabledChange
Enter
EnterWhatsThisMode
Expose
FileOpen
FocusAboutToChange
FocusIn
FocusOut
FontChange
Gesture
GestureOverride
GrabKeyboard
GrabMouse
GraphicsSceneContextMenu
GraphicsSceneDragEnter
GraphicsSceneDragLeave
GraphicsSceneDragMove
GraphicsSceneDrop
GraphicsSceneHelp
GraphicsSceneHoverEnter
GraphicsSceneHoverLeave

continues on next page

Table 3.15 – continued from previous page

GraphicsSceneHoverMove
GraphicsSceneMouseDoubleClick
GraphicsSceneMouseMove
GraphicsSceneMousePress
GraphicsSceneMouseRelease
GraphicsSceneMove
GraphicsSceneResize
GraphicsSceneWheel
Hide
HideToParent
HoverEnter
HoverLeave
HoverMove
IconDrag
IconTextChange
InputMethod
InputMethodQuery
KeyPress
KeyRelease
KeyboardLayoutChange
LanguageChange
LayoutDirectionChange
LayoutRequest
Leave
LeaveWhatsThisMode

continues on next page

Table 3.15 – continued from previous page

LocaleChange
MacSizeChange
MaxUser
MetaCall
ModifiedChange
MouseButtonDbClick
MouseButtonPress
MouseButtonRelease
MouseMove
MouseTrackingChange
Move
NonClientAreaMouseButtonDbClick
NonClientAreaMouseButtonPress
NonClientAreaMouseButtonRelease
NonClientAreaMouseMove
None_
OkRequest
OrientationChange
Paint
PaletteChange
ParentAboutToChange
ParentChange
PlatformPanel
PlatformSurface
Polish

continues on next page

Table 3.15 – continued from previous page

PolishRequest
QueryWhatsThis
ReadOnlyChange
RequestSoftwareInputPanel
Resize
Scroll
ScrollPrepare
Shortcut
ShortcutOverride
Show
ShowToParent
SockAct
StateMachineSignal
StateMachineWrapped
StatusTip
StyleChange
TabletEnterProximity
TabletLeaveProximity
TabletMove
TabletPress
TabletRelease
TabletTrackingChange
ThreadChange
Timer
ToolBarChange

continues on next page

Table 3.15 – continued from previous page

ToolTip
ToolTipChange
TouchBegin
TouchCancel
TouchEnd
TouchUpdate
UngrabKeyboard
UngrabMouse
UpdateLater
UpdateRequest
User
WhatsThis
WhatsThisClicked
Wheel
WinEventAct
WinIdChange
WindowActivate
WindowBlocked
WindowDeactivate
WindowIconChange
WindowStateChange
WindowTitleChange
WindowUnblocked
ZOrderChange

3.3 lyse.dataframe_utilities

Functions

<code>asdatetime(timestr)</code>	
<code>concat_with_padding(*dataframes)</code>	Concatenates dataframes with MultiIndex column labels, padding shallower hierarchies such that the MultiIndexes have the same nlevels.
<code>flat_dict_to_flat_series(dictionary)</code>	
<code>flat_dict_to_hierarchical_dataframe(dictionary)</code>	Make all the keys tuples of the same length
<code>flatten_dict(dictionary[, keys])</code>	Takes a nested dictionary whose keys are strings, and returns a flat dictionary whose keys are tuples of strings, each element of which is the key for one level of the hierarchy.
<code>get_dataframe_from_shot(filepath)</code>	
<code>get_dataframe_from_shots(filepaths)</code>	
<code>get_nested_dict_from_shot(filepath)</code>	
<code>get_series_from_shot(filepath)</code>	
<code>pad_columns(df, n)</code>	Add depth to hiererchical column labels with empty strings
<code>replace_with_padding(df, row, index)</code>	

3.3.1 lyse.dataframe_utilities.asdatetime

`lyse.dataframe_utilities.asdatetime(timestr)`

3.3.2 lyse.dataframe_utilities.concat_with_padding

`lyse.dataframe_utilities.concat_with_padding(*dataframes)`

Concatenates dataframes with MultiIndex column labels, padding shallower hierarchies such that the MultiIndexes have the same nlevels.

3.3.3 lyse.dataframe_utilities.flat_dict_to_flat_series

`lyse.dataframe_utilities.flat_dict_to_flat_series(dictionary)`

3.3.4 lyse.dataframe_utilities.flat_dict_to_hierarchical_dataframe

lyse.dataframe_utilities.flat_dict_to_hierarchical_dataframe(*dictionary*)
Make all the keys tuples of the same length

3.3.5 lyse.dataframe_utilities.flatten_dict

lyse.dataframe_utilities.flatten_dict(*dictionary*, *keys=()*)
Takes a nested dictionary whose keys are strings, and returns a flat dictionary whose keys are tuples of strings, each element of which is the key for one level of the hierarchy.

3.3.6 lyse.dataframe_utilities.get_dataframe_from_shot

lyse.dataframe_utilities.get_dataframe_from_shot(*filepath*)

3.3.7 lyse.dataframe_utilities.get_dataframe_from_shots

lyse.dataframe_utilities.get_dataframe_from_shots(*filepaths*)

3.3.8 lyse.dataframe_utilities.get_nested_dict_from_shot

lyse.dataframe_utilities.get_nested_dict_from_shot(*filepath*)

3.3.9 lyse.dataframe_utilities.get_series_from_shot

lyse.dataframe_utilities.get_series_from_shot(*filepath*)

3.3.10 lyse.dataframe_utilities.pad_columns

lyse.dataframe_utilities.pad_columns(*df*, *n*)
Add depth to hiererchical column labels with empty strings

3.3.11 lyse.dataframe_utilities.replace_with_padding

lyse.dataframe_utilities.replace_with_padding(*df*, *row*, *index*)

3.4 lyse.figure_manager

Functions

install()

3.4.1 lyse.figure_manager.install

lyse.figure_manager.install()

Classes

FigureManager()

3.4.2 lyse.figure_manager.FigureManager

class lyse.figure_manager.FigureManager

Bases: `object`

`__init__()`

Methods

`__init__()`

`close([identifier])`

`get_first_empty_figure(identifier, *args, ...)`

`reset()`

`set_first_figure_current()`

`show()`

`close(identifier=None)`

`get_first_empty_figure(identifier, *args, **kwargs)`

`reset()`

`set_first_figure_current()`

`show()`

3.5 lyse.tempfile2clipboard

Functions

main()

3.5.1 lyse.tempfile2clipboard.main

lyse.tempfile2clipboard.**main()**

3.6 lyse.__main__

Lyse GUI and supporting code

Functions

error_dialog(message)

get_screen_geometry()

Return the a list of the geometries of each screen: each a tuple of left, top, width and height

question_dialog(message)

scientific_notation(x[, sigfigs, mode])

Returns a unicode string of the float f in scientific notation

3.6.1 lyse.__main__.error_dialog

lyse.__main__.**error_dialog**(*message*)

3.6.2 lyse.__main__.get_screen_geometry

lyse.__main__.**get_screen_geometry**()

Return the a list of the geometries of each screen: each a tuple of left, top, width and height

3.6.3 lyse.__main__.question_dialog

`lyse.__main__.question_dialog(message)`

3.6.4 lyse.__main__.scientific_notation

`lyse.__main__.scientific_notation(x, sigfigs=4, mode='eng')`
Returns a unicode string of the float f in scientific notation

Classes

AnalysisRoutine(filepath, model, output_box_port)

DataFrameModel(view, exp_config)

EditColumns(filebox, column_names, ...)

EditColumnsDialog()

FileBox(container, exp_config, ...)

ItemDelegate(view, model, col_status, ...)

An item delegate with a fixed height and a progress bar
in one column

Lyse()

LyseMainWindow(*args, **kwargs)

RoutineBox(container, exp_config, filebox, ...)

TableView(*args)

TreeView(*args)

UneditableModel

WebServer(*args, **kwargs)

3.6.5 lyse.__main__.AnalysisRoutine

class `lyse.__main__.AnalysisRoutine`(filepath, model, output_box_port, checked=2)

Bases: `object`

`__init__`(filepath, model, output_box_port, checked=2)

Methods

<code>__init__(filepath, model, output_box_port[, ...])</code>	
<code>check_child_exited(worker, timeout_time[, ...])</code>	
<code>do_analysis(filepath)</code>	
<code>enabled()</code>	
<code>end_child([restart])</code>	
<code>get_row_index()</code>	Returns the row index for this routine's row in the model
<code>remove()</code>	End the child process and remove from the treeview
<code>restart()</code>	
<code>set_status(status)</code>	
<code>start_worker()</code>	

check_child_exited(*worker, timeout_time, kill=False, restart=False*)

do_analysis(*filepath*)

enabled()

end_child(*restart=False*)

get_row_index()

Returns the row index for this routine's row in the model

remove()

End the child process and remove from the treeview

restart()

set_status(*status*)

start_worker()

3.6.6 lyse.__main__.DataFrameModel

class lyse.__main__.**DataFrameModel**(*view, exp_config*)

Bases: `PyQt5.QtCore.QObject`

__init__(*view, exp_config*)

Methods

<code>__init__(view, exp_config)</code>	
<code>add_files(filepaths, new_row_data[, done])</code>	Add files to the dataframe model.
<code>blockSignals(self, b)</code>	
<code>childEvent(self, a0)</code>	
<code>children(self)</code>	
<code>connectNotify(self, signal)</code>	
<code>connect_signals()</code>	
<code>customEvent(self, a0)</code>	
<code>deleteLater(self)</code>	
<code>disconnect(-> 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(-> QObject)</code>	
<code>findChildren(...)</code>	
<code>get_first_incomplete()</code>	Returns the filepath of the first shot in the model that has not been analysed
<code>infer_objects()</code>	Convert columns in the dataframe with dtype 'object' into compatible, more specific types, if possible.
<code>inherits(self, classname)</code>	
<code>installEventFilter(self, a0)</code>	
<code>isSignalConnected(self, signal)</code>	
<code>isWidgetType(self)</code>	
<code>isWindowType(self)</code>	

continues on next page

Table 3.24 – continued from previous page

<code>killTimer(self, id)</code>	
<code>mark_as_deleted_off_disk(filepath)</code>	
<code>mark_selection_not_done()</code>	
<code>metaObject(self)</code>	
<code>moveToThread(self, thread)</code>	
<code>new_row(filepath[, done])</code>	
<code>objectName(self)</code>	
<code>on_double_click(index)</code>	
<code>on_remove_selection()</code>	
<code>on_view_context_menu_requested(point)</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>remove_selection([confirm])</code>	
<code>renumber_rows([add_from])</code>	Add/update row indices - the rows are numbered in simple sequential order for easy comparison with the dataframe.
<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_columns_visible(columns_visible)</code>	
<code>set_status_percent(filepath, status_percent)</code>	
<code>signalsBlocked(self)</code>	

continues on next page

Table 3.24 – continued from previous page

<code>startTimer(self, interval[, timerType])</code>	
<code>thread(self)</code>	
<code>timerEvent(self, a0)</code>	
<code>tr(self, sourceText[, disambiguation, n])</code>	
<code>update_column_levels()</code>	Pads the keys and values of our lists of column names so that they still match those in the dataframe after the number of levels in its multiindex has increased (the number of levels never decreases, given the current implementation of <code>concat_with_padding()</code>)
<code>update_row(filepath[, ...])</code>	"Updates a row in the dataframe and Qt model to the data in the HDF5 file for that shot.

Attributes

<code>COL_FILEPATH</code>	
<code>COL_STATUS</code>	
<code>ROLE_DELETED_OFF_DISK</code>	
<code>ROLE_STATUS_PERCENT</code>	
<code>columns_changed</code>	
<code>destroyed</code>	<code>typing.Optional[QObject] = None</code> [signal]
<code>objectNameChanged</code>	<code>str</code> [signal]
<code>staticMetaObject</code>	

`COL_FILEPATH = 1`

`COL_STATUS = 0`

`ROLE_DELETED_OFF_DISK = 258`

`ROLE_STATUS_PERCENT = 257`

`add_files(filepaths, new_row_data, done=False)`

Add files to the dataframe model. `New_row_data` should be a dataframe containing the new rows.

`columns_changed`

`connect_signals()`

`get_first_incomplete()`

Returns the filepath of the first shot in the model that has not been analysed

`infer_objects()`

Convert columns in the dataframe with dtype 'object' into compatible, more specific types, if possible. This improves pickling performance and ensures multishot analysis code does not encounter columns with

dtype 'object' for non-mixed numerical data, which it might choke on.

mark_as_deleted_off_disk(*filepath*)

mark_selection_not_done()

new_row(*filepath*, *done=False*)

on_double_click(*index*)

on_remove_selection()

on_view_context_menu_requested(*point*)

remove_selection(*confirm=True*)

renumber_rows(*add_from=0*)

Add/update row indices - the rows are numbered in simple sequential order for easy comparison with the dataframe. *add_from* allows you to only add numbers for new rows from the given index as a performance optimisation, though if the number of digits changes, all rows will still be renumbered. *add_from* should not be used if rows have been deleted.

set_columns_visible(*columns_visible*)

set_status_percent(*filepath*, *status_percent*)

update_column_levels()

Pads the keys and values of our lists of column names so that they still match those in the dataframe after the number of levels in its multiindex has increased (the number of levels never decreases, given the current implementation of *concat_with_padding*())

update_row(*filepath*, *dataframe_already_updated=False*, *new_row_data=None*, *updated_row_data=None*)

“Updates a row in the dataframe and Qt model to the data in the HDF5 file for that shot.

3.6.7 lyse.__main__.EditColumns

class lyse.__main__.EditColumns(*filebox*, *column_names*, *columns_visible*)

Bases: `object`

__init__(*filebox*, *column_names*, *columns_visible*)

Methods

`__init__`(*filebox*, *column_names*, *columns_visible*)

`cancel`()

`close`()

`connect_signals`()

`do_sort`()

`make_it_so`()

`on_filter_text_edited`(*text*)

continues on next page

Table 3.26 – continued from previous page

<code>on_model_item_changed(item)</code>
<code>on_select_all_state_changed(state)</code>
<code>on_set_selected_triggered(visible)</code>
<code>on_treeView_context_menu_requested(point)</code>
<code>populate_model(column_names, columns_visible)</code>
<code>show()</code>
<code>update_columns(column_names, columns_visible)</code>
<code>update_select_all_checkstate()</code>
<code>update_visible_state(item, state)</code>

Attributes

`COL_NAME`

`COL_VISIBLE`

`ROLE_SORT_DATA`

```
COL_NAME = 1
COL_VISIBLE = 0
ROLE_SORT_DATA = 257
cancel()
close()
connect_signals()
do_sort()
make_it_so()
on_filter_text_edited(text)
on_model_item_changed(item)
on_select_all_state_changed(state)
on_set_selected_triggered(visible)
on_treeView_context_menu_requested(point)
populate_model(column_names, columns_visible)
show()
```

```

update_columns(column_names, columns_visible)
update_select_all_checkstate()
update_visible_state(item, state)

```

3.6.8 lyse.__main__.EditColumnsDialog

```

class lyse.__main__.EditColumnsDialog
    Bases: PyQt5.QtWidgets.QDialog
    __init__()

```

Methods

```
__init__()
```

```
accept(self)
```

```
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)
```

```
changeEvent(self, a0)
```

```
childAt(-> QWidget)
```

```
childEvent(self, a0)
```

continues on next page

Table 3.28 – continued from previous page

<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><i>closeEvent</i>(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, destroySubWindows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-> bool)</code>

continues on next page

Table 3.28 – continued from previous page

<code>disconnectNotify(self, signal)</code>
<code>done(self, a0)</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>exec(self)</code>
<code>exec_(self)</code>
<code>find(a0)</code>
<code>findChild(-> QObject)</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>

continues on next page

Table 3.28 – continued from previous page

<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>
<code>height(self)</code>
<code>heightForWidth(self, a0)</code>
<code>heightMM(self)</code>
<code>hide(self)</code>

continues on next page

Table 3.28 – continued from previous page

<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>
<code>isSignalConnected(self, signal)</code>
<code>isSizeGripEnabled(self)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, a0)</code>
<code>isWidgetType(self)</code>

continues on next page

Table 3.28 – continued from previous page

<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>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumWidth(self)</code>
<code>metaObject(self)</code>

continues on next page

Table 3.28 – continued from previous page

<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>open(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>

continues on next page

Table 3.28 – continued from previous page

<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>reject(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(-> None -> None)</code>	
<code>resize()</code>	
<code>resizeEvent(self, a0)</code>	
<code>restoreGeometry(self, geometry)</code>	
<code>result(self)</code>	
<code>saveGeometry(self)</code>	
<code>screen(self)</code>	

continues on next page

Table 3.28 – continued from previous page

<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>
<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setForegroundRole(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>

continues on next page

Table 3.28 – continued from previous page

<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>setModal(self, modal)</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>setResult(self, r)</code>
<code>setShortcutAutoRepeat(self, id[, enabled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeGripEnabled(self, a0)</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>

continues on next page

Table 3.28 – continued from previous page

<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>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>

continues on next page

Table 3.28 – continued from previous page

<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>
<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>
<code>update(-> None -> None)</code>

continues on next page

Table 3.28 – continued from previous page

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

Attributes

Accepted	
DrawChildren	
DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
Rejected	
accepted	accepted(self) [signal]
<i>close_signal</i>	
customContextMenuRequested	QPoint) [signal]
destroyed	typing.Optional[QObject] = None) [signal]
finished	int) [signal]
objectNameChanged	str) [signal]
rejected	rejected(self) [signal]
staticMetaObject	
windowIconChanged	QIcon) [signal]
windowIconTextChanged	str) [signal]
windowTitleChanged	str) [signal]

closeEvent(self, a0: *QCloseEvent*)

close_signal

3.6.9 lyse.__main__.FileBox

class lyse.__main__.FileBox(*container, exp_config, to_singleshot, from_singleshot, to_multishot, from_multishot*)

Bases: object

__init__(*container, exp_config, to_singleshot, from_singleshot, to_multishot, from_multishot*)

Methods

__init__(*container, exp_config, ...*)

analysis_loop()

connect_signals()

do_multishot_analysis()

do_singleshot_analysis(*filepath*)

incoming_buffer_loop()

We use a queue as a buffer for incoming shots.

on_add_shot_files_clicked()

on_analysis_running_toggled(*pressed*)

on_columns_changed()

on_edit_columns_clicked()

on_mark_selection_not_done_clicked()

on_run_multishot_analysis_clicked()

pause_analysis()

set_add_shots_progress(*completed, total, message*)

set_columns_visible(*columns_visible*)

analysis_loop()

connect_signals()

do_multishot_analysis()

do_singleshot_analysis(*filepath*)

incoming_buffer_loop()

We use a queue as a buffer for incoming shots. We don't want to hang and not respond to a client submitting shots, so we just let shots pile up here until we can get to them. The downside to this is that we can't return errors to the client if the shot cannot be added, but the suggested workflow is to handle errors here anyway. A client running shots shouldn't stop the experiment on account of errors from the analysis stage, so what's the point of passing errors to it? We'll just raise errors here and the user can decide what to do with them.

```
on_add_shot_files_clicked()
on_analysis_running_toggled(pressed)
on_columns_changed()
on_edit_columns_clicked()
on_mark_selection_not_done_clicked()
on_run_multishot_analysis_clicked()
pause_analysis()
set_add_shots_progress(completed, total, message)
set_columns_visible(columns_visible)
```

3.6.10 lyse.__main__.ItemDelegate

```
class lyse.__main__.ItemDelegate(view, model, col_status, role_status_percent)
```

```
    Bases: PyQt5.QtWidgets.QStyledItemDelegate
```

An item delegate with a fixed height and a progress bar in one column

```
    __init__(view, model, col_status, role_status_percent)
```

Methods

```
__init__(view, model, col_status, ...)
```

```
blockSignals(self, b)
```

```
childEvent(self, a0)
```

```
children(self)
```

```
connectNotify(self, signal)
```

```
createEditor(self, parent, option, index)
```

```
customEvent(self, a0)
```

```
deleteLater(self)
```

```
destroyEditor(self, editor, index)
```

```
disconnect(-> bool)
```

```
disconnectNotify(self, signal)
```

```
displayText(self, value, locale)
```

```
dumpObjectInfo(self)
```

continues on next page

Table 3.31 – continued from previous page

<code>dumpObjectTree(self)</code>	
<code>dynamicPropertyNames(self)</code>	
<code>editorEvent(self, event, model, option, index)</code>	
<code>event(self, a0)</code>	
<code>eventFilter(self, object, event)</code>	
<code>findChild(-> QObject)</code>	
<code>findChildren(...)</code>	
<code>helpEvent(self, event, view, option, index)</code>	
<code>inherits(self, classname)</code>	
<code>initStyleOption(self, option, index)</code>	
<code>installEventFilter(self, a0)</code>	
<code>isSignalConnected(self, signal)</code>	
<code>isWidgetType(self)</code>	
<code>isWindowType(self)</code>	
<code>itemEditorFactory(self)</code>	
<code>killTimer(self, id)</code>	
<code>metaObject(self)</code>	
<code>moveToThread(self, thread)</code>	
<code>objectName(self)</code>	
<code><i>paint</i>(self, painter, option, index)</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 3.32 – continued from previous page

objectNameChanged	str) [signal]
sizeHintChanged	QModelIndex) [signal]
staticMetaObject	

EXTRA_ROW_HEIGHT = 2

paint(self, painter: *QPainter*, option: *QStyleOptionViewItem*, index: *QModelIndex*)

sizeHint(self, option: *QStyleOptionViewItem*, index: *QModelIndex*) → *QSize*

3.6.11 lyse.__main__.Lyse

class lyse.__main__.Lyse

Bases: *object*

__init__()

Methods

__init__()

are_you_sure()

connect_signals()

delete_items(confirm) Delete items from whichever box has focus, with optional confirmation dialog

get_save_data()

load_configuration(filename[, ...])

load_window_geometry_configuration(filename) Load only the window geometry from the config file.
on_close_event()

on_load_configuration_triggered()

on_load_dataframe_triggered()

on_revert_configuration_triggered()

on_save_configuration_as_triggered()

on_save_configuration_triggered()

on_save_dataframe_triggered([choose_folder])

only_window_geometry_is_different(...)

continues on next page

Table 3.33 – continued from previous page

`save_configuration(save_file)`

`setup_config()`

`terminate_all_workers()`

`workers_terminated()`

`are_you_sure()``connect_signals()``delete_items(confirm)`

Delete items from whichever box has focus, with optional confirmation dialog

`get_save_data()``load_configuration(filename, restore_window_geometry=True)``load_window_geometry_configuration(filename)`

Load only the window geometry from the config file. It's useful to have this separate from the rest of `load_configuration` so that it can be called before the window is shown.

`on_close_event()``on_load_configuration_triggered()``on_load_dataframe_triggered()``on_revert_configuration_triggered()``on_save_configuration_as_triggered()``on_save_configuration_triggered()``on_save_dataframe_triggered(choose_folder=True)``only_window_geometry_is_different(current_data, old_data)``save_configuration(save_file)``setup_config()``terminate_all_workers()``workers_terminated()`

3.6.12 lyse.__main__.LyseMainWindow

```
class lyse.__main__.LyseMainWindow(*args, **kwargs)
```

```
    Bases: PyQt5.QtWidgets.QMainWindow
```

```
    __init__(*args, **kwargs)
```


Methods

`__init__(*args, **kwargs)`

`acceptDrops(self)`

`accessibleDescription(self)`

`accessibleName(self)`

`actionEvent(self, a0)`

`actions(self)`

`activateWindow(self)`

`addAction(self, action)`

`addActions(self, actions)`

`addDockWidget()`

`addToolBar(-> None)`

`addToolBarBreak(self[, area])`

`adjustSize(self)`

`autoFillBackground(self)`

`backgroundRole(self)`

`baseSize(self)`

`blockSignals(self, b)`

`centralWidget(self)`

`changeEvent(self, a0)`

`childAt(-> QWidget)`

`childEvent(self, a0)`

`children(self)`

`childrenRect(self)`

`childrenRegion(self)`

continues on next page

Table 3.34 – continued from previous page

<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code><i>closeEvent</i>(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, event)</code>
<code>contextMenuPolicy(self)</code>
<code>corner(self, corner)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createPopupMenu(self)</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code><i>delayedClose</i>(timeout_time)</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(-> bool)</code>

continues on next page

Table 3.34 – continued from previous page

<code>disconnectNotify(self, signal)</code>
<code>dockOptions(self)</code>
<code>dockWidgetArea(self, dockwidget)</code>
<code>documentMode(self)</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, event)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-> QObject)</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>

continues on next page

Table 3.34 – continued from previous page

<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>
<code>height(self)</code>
<code>heightForWidth(self, a0)</code>
<code>heightMM(self)</code>
<code>hide(self)</code>

continues on next page

Table 3.34 – continued from previous page

<code>hideEvent(self, a0)</code>
<code>iconSize(self)</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>insertToolBar(self, before, toolbar)</code>
<code>insertToolBarBreak(self, before)</code>
<code>installEventFilter(self, a0)</code>
<code>isActiveWindow(self)</code>
<code>isAncestorOf(self, child)</code>
<code>isAnimated(self)</code>
<code>isDockNestingEnabled(self)</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 3.34 – continued from previous page

<code>isSeparator(self, pos)</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>
<code>mapToParent(self, a0)</code>

continues on next page

Table 3.34 – continued from previous page

<code>mask(self)</code>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumWidth(self)</code>
<code>menuBar(self)</code>
<code>menuItem(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 3.34 – continued from previous page

<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code><i>paintEvent</i>(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>removeDockWidget(self, dockwidget)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>removeToolBar(self, toolbar)</code>	
<code>removeToolBarBreak(self, before)</code>	

continues on next page

Table 3.34 – continued from previous page

<code>render(, sourceRegion, flags, ...)</code>
<code>repaint(-> None -> None)</code>
<code>resize()</code>
<code>resizeDocks(self, docks, sizes, orientation)</code>
<code>resizeEvent(self, a0)</code>
<code>restoreDockWidget(self, dockwidget)</code>
<code>restoreGeometry(self, geometry)</code>
<code>restoreState(self, state[, version])</code>
<code>saveGeometry(self)</code>
<code>saveState(self[, version])</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>setAnimated(self, enabled)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setCentralWidget(self, widget)</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>

continues on next page

Table 3.34 – continued from previous page

<code>setCorner(self, corner, area)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setDockNestingEnabled(self, enabled)</code>
<code>setDockOptions(self, options)</code>
<code>setDocumentMode(self, enabled)</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>
<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>setIconSize(self, iconSize)</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>

continues on next page

Table 3.34 – continued from previous page

<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMenuBar(self, menubar)</code>
<code>setMenuWidget(self, menubar)</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[, enabled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusbar(self, statusbar)</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabPosition(self, areas, tabPosition)</code>
<code>setTabShape(self, tabShape)</code>
<code>setTabletTracking(self, enable)</code>
<code>setToolButtonStyle(self, toolButtonStyle)</code>

continues on next page

Table 3.34 – continued from previous page

<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUnifiedTitleAndToolBarOnMac(self, set)</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>
<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>

continues on next page

Table 3.34 – continued from previous page

<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>splitDockWidget(self, after, dockwidget, ...)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusBar(self)</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabPosition(self, area)</code>
<code>tabShape(self)</code>
<code>tabifiedDockWidgets(self, dockwidget)</code>
<code>tabifyDockWidget(self, first, second)</code>
<code>tabletEvent(self, a0)</code>
<code>takeCentralWidget(self)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolBarArea(self, toolbar)</code>
<code>toolBarBreak(self, toolbar)</code>
<code>toolButtonStyle(self)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>

continues on next page

Table 3.34 – continued from previous page

<code>tr(self, sourceText[, disambiguation, n])</code>
<code>underMouse(self)</code>
<code>ungrabGesture(self, type)</code>
<code>unifiedTitleAndToolBarOnMac(self)</code>
<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>
<code>update(-> None -> 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>

continues on next page

Table 3.34 – continued from previous page

windowRole(self)	
WindowState(self)	
windowTitle(self)	
windowType(self)	
x(self)	
y(self)	
Attributes	
AllowNestedDocks	
AllowTabbedDocks	
AnimatedDocks	
DrawChildren	
DrawWindowBackground	
ForceTabbedDocks	
GroupedDragging	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	

continues on next page

Table 3.35 – continued from previous page

PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
VerticalTabs	
customContextMenuRequested	QPoint) [signal]
destroyed	typing.Optional[QObject] = None) [signal]
<i>firstPaint</i>	
iconSizeChanged	QSize) [signal]
objectNameChanged	str) [signal]
staticMetaObject	
tabifiedDockWidgetActivated	QDockWidget) [signal]
toolButtonStyleChanged	Qt.ToolButtonStyle) [signal]
windowIconChanged	QIcon) [signal]
windowIconTextChanged	str) [signal]
windowTitleChanged	str) [signal]

closeEvent(*self*, *a0*: *QCloseEvent*)

delayedClose(*timeout_time*)

firstPaint

paintEvent(*self*, *a0*: *QPaintEvent*)

3.6.13 lyse.__main__.RoutineBox

class lyse.__main__.**RoutineBox**(*container*, *exp_config*, *filebox*, *from_filebox*, *to_filebox*, *output_box_port*, *multishot=False*)

Bases: *object*

__init__(*container*, *exp_config*, *filebox*, *from_filebox*, *to_filebox*, *output_box_port*, *multishot=False*)

Methods

__init__(*container*, *exp_config*, *filebox*, ...)

add_routines(*routine_files*[, *clear_existing*]) Add routines to the routine box, where *routine_files* is a list of tuples containing the filepath and whether the routine is enabled or not when it is added.

analysis_loop()

connect_signals()

do_analysis(*filepath*) Run all analysis routines once on the given filepath, which is a shot file if we are a singleshot routine box

continues on next page

Table 3.36 – continued from previous page

<code>on_add_routines_clicked()</code>	
<code>on_model_item_changed(item)</code>	
<code>on_move_down_clicked()</code>	
<code>on_move_to_bottom_clicked()</code>	
<code>on_move_to_top_clicked()</code>	
<code>on_move_up_clicked()</code>	
<code>on_remove_selection()</code>	
<code>on_restart_selected_triggered()</code>	
<code>on_select_all_state_changed(state)</code>	
<code>on_set_selected_triggered(active)</code>	
<code>on_treeView_context_menu_requested(point)</code>	
<code>on_treeview_double_left_clicked(index)</code>	
<code>remove_selection([confirm])</code>	
<code>reorder(order)</code>	
<code>todo()</code>	How many analysis routines are not done?
<code>update_select_all_checkstate()</code>	

Attributes`COL_ACTIVE``COL_NAME``COL_STATUS``ROLE_FULLPATH``ROLE_SORTINDEX``COL_ACTIVE = 0``COL_NAME = 2``COL_STATUS = 1``ROLE_FULLPATH = 257`

ROLE_SORTINDEX = 258

add_routines(*routine_files*, *clear_existing=False*)

Add routines to the routine box, where *routine_files* is a list of tuples containing the filepath and whether the routine is enabled or not when it is added. if *clear_existing == True*, then any existing analysis routines will be cleared before the new ones are added.

analysis_loop()

connect_signals()

do_analysis(*filepath*)

Run all analysis routines once on the given filepath, which is a shot file if we are a singleshot routine box

on_add_routines_clicked()

on_model_item_changed(*item*)

on_move_down_clicked()

on_move_to_bottom_clicked()

on_move_to_top_clicked()

on_move_up_clicked()

on_remove_selection()

on_restart_selected_triggered()

on_select_all_state_changed(*state*)

on_set_selected_triggered(*active*)

on_treeView_context_menu_requested(*point*)

on_treeview_double_left_clicked(*index*)

remove_selection(*confirm=True*)

reorder(*order*)

todo()

How many analysis routines are not done?

update_select_all_checkstate()

3.6.14 lyse.__main__.TableView

```
class lyse.__main__.TableView(*args)
```

```
    Bases: PyQt5.QtWidgets.QTableView
```

```
    __init__(*args)
```

Methods

`__init__(*args)`

`acceptDrops(self)`

`accessibleDescription(self)`

`accessibleName(self)`

`actionEvent(self, a0)`

`actions(self)`

`activateWindow(self)`

`addAction(self, action)`

`addActions(self, actions)`

`addScrollBarWidget(self, widget, alignment)`

`adjustSize(self)`

`alternatingRowColors(self)`

`autoFillBackground(self)`

`autoScrollMargin(self)`

`backgroundRole(self)`

`baseSize(self)`

`blockSignals(self, b)`

`changeEvent(self, a0)`

`childAt(-> QWidget)`

`childEvent(self, a0)`

`children(self)`

`childrenRect(self)`

`childrenRegion(self)`

`clearFocus(self)`

continues on next page

Table 3.38 – continued from previous page

<code>clearMask(self)</code>
<code>clearSelection(self)</code>
<code>clearSpans(self)</code>
<code>close(self)</code>
<code>closeEditor(self, editor, hint)</code>
<code>closeEvent(self, a0)</code>
<code>closePersistentEditor(self, index)</code>
<code>colorCount(self)</code>
<code>columnAt(self, x)</code>
<code>columnCountChanged(self, oldCount, newCount)</code>
<code>columnMoved(self, column, oldIndex, newIndex)</code>
<code>columnResized(self, column, oldWidth, newWidth)</code>
<code>columnSpan(self, row, column)</code>
<code>columnViewportPosition(self, column)</code>
<code>columnWidth(self, column)</code>
<code>commitData(self, editor)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>cornerWidget(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>currentChanged(self, current, previous)</code>

continues on next page

Table 3.38 – continued from previous page

<code>currentIndex(self)</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>dataChanged(self, topLeft, bottomRight[, roles])</code>
<code>defaultDropAction(self)</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>dirtyRegionOffset(self)</code>
<code>disconnect(-> bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragDropMode(self)</code>
<code>dragDropOverwriteMode(self)</code>
<code>dragEnabled(self)</code>
<code>dragEnterEvent(self, e)</code>
<code>dragLeaveEvent(self, e)</code>
<code>dragMoveEvent(self, e)</code>
<code>drawFrame(self, a0)</code>
<code>dropEvent(self, e)</code>
<code>dropIndicatorPosition(self)</code>
<code>dumpObjectInfo(self)</code>

continues on next page

Table 3.38 – continued from previous page

<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>edit()</code>
<code>editTriggers(self)</code>
<code>editorDestroyed(self, editor)</code>
<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, event)</code>
<code>eventFilter(self, object, event)</code>
<code>executeDelayedItemsLayout(self)</code>
<code>find(a0)</code>
<code>findChild(-> QObject)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, e)</code>
<code>focusNextChild(self)</code>
<code>focusNextPrevChild(self, next)</code>
<code>focusOutEvent(self, e)</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>

continues on next page

Table 3.38 – continued from previous page

<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>
<code>frameRect(self)</code>
<code>frameShadow(self)</code>
<code>frameShape(self)</code>
<code>frameSize(self)</code>
<code>frameStyle(self)</code>
<code>frameWidth(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>gridStyle(self)</code>
<code>hasAutoScroll(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>

continues on next page

Table 3.38 – continued from previous page

<code>heightMM(self)</code>
<code>hide(self)</code>
<code>hideColumn(self, column)</code>
<code>hideEvent(self, a0)</code>
<code>hideRow(self, row)</code>
<code>horizontalHeader(self)</code>
<code>horizontalOffset(self)</code>
<code>horizontalScrollBar(self)</code>
<code>horizontalScrollBarPolicy(self)</code>
<code>horizontalScrollMode(self)</code>
<code>horizontalScrollbarAction(self, action)</code>
<code>horizontalScrollbarValueChanged(self, value)</code>
<code>iconSize(self)</code>
<code>indexOf(self, p)</code>
<code>indexOfWidget(self, index)</code>
<code>inherits(self, classname)</code>
<code>initPainter(self, painter)</code>
<code>initStyleOption(self, option)</code>
<code>inputMethodEvent(self, event)</code>
<code>inputMethodHints(self)</code>
<code>inputMethodQuery(self, query)</code>
<code>insertAction(self, before, action)</code>
<code>insertActions(self, before, actions)</code>
<code>installEventFilter(self, a0)</code>
<code>isActiveWindow(self)</code>

continues on next page

Table 3.38 – continued from previous page

<code>isAncestorOf(self, child)</code>
<code>isColumnHidden(self, column)</code>
<code>isCornerButtonEnabled(self)</code>
<code>isEnabled(self)</code>
<code>isEnabledTo(self, a0)</code>
<code>isFullScreen(self)</code>
<code>isHidden(self)</code>
<code>isIndexHidden(self, index)</code>
<code>isLeftToRight(self)</code>
<code>isMaximized(self)</code>
<code>isMinimized(self)</code>
<code>isModal(self)</code>
<code>isPersistentEditorOpen(self, index)</code>
<code>isRightToLeft(self)</code>
<code>isRowHidden(self, row)</code>
<code>isSignalConnected(self, signal)</code>
<code>isSortingEnabled(self)</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>itemDelegate(-> QAbstractItemDelegate)</code>
<code>itemDelegateForColumn(self, column)</code>

continues on next page

Table 3.38 – continued from previous page

<code>itemDelegateForRow(self, row)</code>
<code>keyPressEvent(self, e)</code>
<code>keyReleaseEvent(self, a0)</code>
<code>keyboardGrabber()</code>
<code>keyboardSearch(self, search)</code>
<code>killTimer(self, id)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code><i>leaveEvent</i>(self, a0)</code>
<code>lineWidth(self)</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>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumViewportSize(self)</code>
<code>maximumWidth(self)</code>

continues on next page

Table 3.38 – continued from previous page

<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>midLineWidth(self)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>model(self)</code>
<code><i>mouseDoubleClickEvent</i>(self, e)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, e)</code>
<code><i>mousePressEvent</i>(self, e)</code>
<code><i>mouseReleaseEvent</i>(self, e)</code>
<code>move()</code>
<code>moveCursor(self, cursorAction, modifiers)</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>openPersistentEditor(self, index)</code>
<code>overrideWindowFlags(self, type)</code>
<code>overrideWindowState(self, state)</code>

continues on next page

Table 3.38 – continued from previous page

<code>paintEngine(self)</code>	
<code>paintEvent(self, e)</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(-> None -> None)</code>	
<code>reset(self)</code>	
<code>resetHorizontalScrollMode(self)</code>	
<code>resetVerticalScrollMode(self)</code>	

continues on next page

Table 3.38 – continued from previous page

<code>resize()</code>
<code>resizeColumnToContents(self, column)</code>
<code>resizeColumnsToContents(self)</code>
<code>resizeEvent(self, e)</code>
<code>resizeRowToContents(self, row)</code>
<code>resizeRowsToContents(self)</code>
<code>restoreGeometry(self, geometry)</code>
<code>rootIndex(self)</code>
<code>rowAt(self, y)</code>
<code>rowCountChanged(self, oldCount, newCount)</code>
<code>rowHeight(self, row)</code>
<code>rowMoved(self, row, oldIndex, newIndex)</code>
<code>rowResized(self, row, oldHeight, newHeight)</code>
<code>rowSpan(self, row, column)</code>
<code>rowViewportPosition(self, row)</code>
<code>rowsAboutToBeRemoved(self, parent, start, end)</code>
<code>rowsInserted(self, parent, start, end)</code>
<code>saveGeometry(self)</code>
<code>scheduleDelayedItemsLayout(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>scrollBarWidgets(self, alignment)</code>
<code>scrollContentsBy(self, dx, dy)</code>
<code>scrollDirtyRegion(self, dx, dy)</code>
<code>scrollTo(self, index[, hint])</code>

continues on next page

Table 3.38 – continued from previous page

<code>scrollToBottom(self)</code>
<code>scrollToTop(self)</code>
<code>selectAll(self)</code>
<code>selectColumn(self, column)</code>
<code>selectRow(self, row)</code>
<code>selectedIndexes(self)</code>
<code>selectionBehavior(self)</code>
<code>selectionChanged(self, selected, deselected)</code>
<code>selectionCommand(self, index[, event])</code>
<code>selectionMode(self)</code>
<code>selectionModel(self)</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>setAlternatingRowColors(self, enable)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setAutoScroll(self, enable)</code>
<code>setAutoScrollMargin(self, margin)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setColumnHidden(self, column, hide)</code>
<code>setColumnWidth(self, column, width)</code>

continues on next page

Table 3.38 – continued from previous page

<code>setContentMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCornerButtonEnabled(self, enable)</code>
<code>setCornerWidget(self, widget)</code>
<code>setCurrentIndex(self, index)</code>
<code>setCursor(self, a0)</code>
<code>setDefaultDropAction(self, dropAction)</code>
<code>setDirtyRegion(self, region)</code>
<code>setDisabled(self, a0)</code>
<code>setDragDropMode(self, behavior)</code>
<code>setDragDropOverwriteMode(self, overwrite)</code>
<code>setDragEnabled(self, enable)</code>
<code>setDropIndicatorShown(self, enable)</code>
<code>setEditTriggers(self, triggers)</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>
<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setForegroundRole(self, a0)</code>
<code>setFrameRect(self, a0)</code>
<code>setFrameShadow(self, a0)</code>

continues on next page

Table 3.38 – continued from previous page

<code>setFrameShape(self, a0)</code>
<code>setFrameStyle(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setGridStyle(self, style)</code>
<code>setHidden(self, hidden)</code>
<code>setHorizontalHeader(self, header)</code>
<code>setHorizontalScrollBar(self, scrollbar)</code>
<code>setHorizontalScrollBarPolicy(self, a0)</code>
<code>setHorizontalScrollMode(self, mode)</code>
<code>setIconSize(self, size)</code>
<code>setIndexWidget(self, index, widget)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setItemDelegate(self, delegate)</code>
<code>setItemDelegateForColumn(self, column, delegate)</code>
<code>setItemDelegateForRow(self, row, delegate)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLineWidth(self, a0)</code>
<code>setLocale(self, locale)</code>
<code>setMask()</code>
<code>setMaximumHeight(self, maxh)</code>
<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMidLineWidth(self, a0)</code>

continues on next page

Table 3.38 – continued from previous page

<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setModel(self, model)</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>setRootIndex(self, index)</code>
<code>setRowHeight(self, row, height)</code>
<code>setRowHidden(self, row, hide)</code>
<code>setSelection(self, rect, command)</code>
<code>setSelectionBehavior(self, behavior)</code>
<code>setSelectionMode(self, mode)</code>
<code>setSelectionModel(self, selectionModel)</code>
<code>setShortcutAutoRepeat(self, id[, enabled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setShowGrid(self, show)</code>
<code>setSizeAdjustPolicy(self, policy)</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setSortingEnabled(self, enable)</code>
<code>setSpan(self, row, column, rowSpan, columnSpan)</code>
<code>setState(self, state)</code>

continues on next page

Table 3.38 – continued from previous page

<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabKeyNavigation(self, enable)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setTextElideMode(self, mode)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVerticalHeader(self, header)</code>
<code>setVerticalScrollBar(self, scrollbar)</code>
<code>setVerticalScrollBarPolicy(self, a0)</code>
<code>setVerticalScrollMode(self, mode)</code>
<code>setViewport(self, widget)</code>
<code>setViewportMargins()</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>

continues on next page

Table 3.38 – continued from previous page

<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>
<code>setWindowState(self, state)</code>
<code>setWindowTitle(self, a0)</code>
<code>setWordWrap(self, on)</code>
<code>setupViewport(self, viewport)</code>
<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showColumn(self, column)</code>
<code>showDropIndicator(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showGrid(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>showRow(self, row)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeAdjustPolicy(self)</code>
<code>sizeHint(self)</code>
<code>sizeHintForColumn(self, column)</code>
<code>sizeHintForIndex(self, index)</code>
<code>sizeHintForRow(self, row)</code>
<code>sizeIncrement(self)</code>

continues on next page

Table 3.38 – continued from previous page

<code>sizePolicy(self)</code>
<code>sortByColumn(self, column, order)</code>
<code>stackUnder(self, a0)</code>
<code>startDrag(self, supportedActions)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>state(self)</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabKeyNavigation(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>textElideMode(self)</code>
<code>thread(self)</code>
<code>timerEvent(self, event)</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()</code>
<code>updateEditorData(self)</code>

continues on next page

Table 3.38 – continued from previous page

<code>updateEditorGeometries(self)</code>
<code>updateGeometries(self)</code>
<code>updateGeometry(self)</code>
<code>updateMicroFocus(self)</code>
<code>updatesEnabled(self)</code>
<code>verticalHeader(self)</code>
<code>verticalOffset(self)</code>
<code>verticalScrollBar(self)</code>
<code>verticalScrollBarPolicy(self)</code>
<code>verticalScrollMode(self)</code>
<code>verticalScrollbarAction(self, action)</code>
<code>verticalScrollbarValueChanged(self, value)</code>
<code>viewOptions(self)</code>
<code>viewport(self)</code>
<code>viewportEvent(self, e)</code>
<code>viewportMargins(self)</code>
<code>viewportSizeHint(self)</code>
<code>visibleRegion(self)</code>
<code>visualRect(self, index)</code>
<code>visualRegionForSelection(self, selection)</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 3.38 – continued from previous page

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

Attributes

<code>AboveItem</code>
<code>AdjustIgnored</code>
<code>AdjustToContents</code>
<code>AdjustToContentsOnFirstShow</code>
<code>AllEditTriggers</code>
<code>AnimatingState</code>
<code>AnyKeyPressed</code>
<code>BelowItem</code>

continues on next page

Table 3.39 – continued from previous page

Box
CollapsingState
ContiguousSelection
CurrentChanged
DoubleClicked
DragDrop
DragOnly
DragSelectingState
DraggingState
DrawChildren
DrawWindowBackground
DropOnly
EditKeyPressed
EditingState
EnsureVisible
ExpandingState
ExtendedSelection
HLine
IgnoreMask
InternalMove
MoveDown
MoveEnd
MoveHome
MoveLeft
MoveNext

continues on next page

Table 3.39 – continued from previous page

MovePageDown
MovePageUp
MovePrevious
MoveRight
MoveUp
MultiSelection
NoDragDrop
NoEditTriggers
NoFrame
NoSelection
NoState
OnItem
OnViewport
Panel
PdmDepth
PdmDevicePixelRatio
PdmDevicePixelRatioScaled
PdmDpiX
PdmDpiY
PdmHeight
PdmHeightMM
PdmNumColors
PdmPhysicalDpiX
PdmPhysicalDpiY
PdmWidth

continues on next page

Table 3.39 – continued from previous page

PdmWidthMM	
Plain	
PositionAtBottom	
PositionAtCenter	
PositionAtTop	
Raised	
ScrollPerItem	
ScrollPerPixel	
SelectColumns	
SelectItems	
SelectRows	
SelectedClicked	
Shadow_Mask	
Shape_Mask	
SingleSelection	
StyledPanel	
Sunken	
VLine	
WinPanel	
activated	QModelIndex) [signal]
clicked	QModelIndex) [signal]
customContextMenuRequested	QPoint) [signal]
destroyed	typing.Optional[QObject] = None) [signal]
doubleClicked	QModelIndex) [signal]
<i>doubleLeftClicked</i>	A QTableView that emits a custom signal leftClicked(index) after a left click on a valid index, and doubleLeftClicked(index) (in addition) on double click.
entered	QModelIndex) [signal]
iconSizeChanged	QSize) [signal]
<i>leftClicked</i>	

continues on next page

Table 3.39 – continued from previous page

objectNameChanged	str) [signal]
pressed	QModelIndex) [signal]
staticMetaObject	
viewportEntered	viewportEntered(self) [signal]
windowIconChanged	QIcon) [signal]
windowIconTextChanged	str) [signal]
windowTitleChanged	str) [signal]

doubleLeftClicked

A `QTableView` that emits a custom signal `leftClicked(index)` after a left click on a valid index, and `doubleLeftClicked(index)` (in addition) on double click. Multiple inheritance of `QObject`s is not possible, so we are forced to duplicate code instead of sharing code with the extremely similar `TreeView` class in this module

leaveEvent (*self*, *a0*: *QEvent*)

leftClicked

mouseDoubleClickEvent (*self*, *e*: *QMouseEvent*)

mousePressEvent (*self*, *e*: *QMouseEvent*)

mouseReleaseEvent (*self*, *e*: *QMouseEvent*)

3.6.15 lyse.__main__.TreeView

```
class lyse.__main__.TreeView(*args)
    Bases: PyQt5.QtWidgets.QTreeView
    __init__(*args)
```

Methods

```
__init__(*args)
```

```
acceptDrops(self)
```

```
accessibleDescription(self)
```

```
accessibleName(self)
```

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

```
actions(self)
```

```
activateWindow(self)
```

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

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

continues on next page

Table 3.40 – continued from previous page

<code>addScrollBarWidget(self, widget, alignment)</code>
<code>adjustSize(self)</code>
<code>allColumnsShowFocus(self)</code>
<code>alternatingRowColors(self)</code>
<code>autoExpandDelay(self)</code>
<code>autoFillBackground(self)</code>
<code>autoScrollMargin(self)</code>
<code>backgroundRole(self)</code>
<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-> 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>clearSelection(self)</code>
<code>close(self)</code>
<code>closeEditor(self, editor, hint)</code>
<code>closeEvent(self, a0)</code>
<code>closePersistentEditor(self, index)</code>
<code>collapse(self, index)</code>
<code>collapseAll(self)</code>

continues on next page

Table 3.40 – continued from previous page

<code>colorCount(self)</code>
<code>columnAt(self, x)</code>
<code>columnCountChanged(self, oldCount, newCount)</code>
<code>columnMoved(self)</code>
<code>columnResized(self, column, oldSize, newSize)</code>
<code>columnViewportPosition(self, column)</code>
<code>columnWidth(self, column)</code>
<code>commitData(self, editor)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>cornerWidget(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>currentChanged(self, current, previous)</code>
<code>currentIndex(self)</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>dataChanged(self, topLeft, bottomRight[, roles])</code>
<code>defaultDropAction(self)</code>
<code>deleteLater(self)</code>
<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySubWindows])</code>

continues on next page

Table 3.40 – continued from previous page

<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>dirtyRegionOffset(self)</code>
<code>disconnect(-> bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragDropMode(self)</code>
<code>dragDropOverwriteMode(self)</code>
<code>dragEnabled(self)</code>
<code>dragEnterEvent(self, e)</code>
<code>dragLeaveEvent(self, e)</code>
<code>dragMoveEvent(self, event)</code>
<code>drawBranches(self, painter, rect, index)</code>
<code>drawFrame(self, a0)</code>
<code>drawRow(self, painter, options, index)</code>
<code>drawTree(self, painter, region)</code>
<code>dropEvent(self, e)</code>
<code>dropIndicatorPosition(self)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>edit()</code>
<code>editTriggers(self)</code>
<code>editorDestroyed(self, editor)</code>

continues on next page

Table 3.40 – continued from previous page

<code>effectiveWinId(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, event)</code>
<code>eventFilter(self, object, event)</code>
<code>executeDelayedItemsLayout(self)</code>
<code>expand(self, index)</code>
<code>expandAll(self)</code>
<code>expandRecursively(self, index[, depth])</code>
<code>expandToDepth(self, depth)</code>
<code>expandsOnDoubleClick(self)</code>
<code>find(a0)</code>
<code>findChild(-> QObject)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, e)</code>
<code>focusNextChild(self)</code>
<code>focusNextPrevChild(self, next)</code>
<code>focusOutEvent(self, e)</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>

continues on next page

Table 3.40 – continued from previous page

<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>
<code>frameRect(self)</code>
<code>frameShadow(self)</code>
<code>frameShape(self)</code>
<code>frameSize(self)</code>
<code>frameStyle(self)</code>
<code>frameWidth(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>hasAutoScroll(self)</code>
<code>hasFocus(self)</code>
<code>hasHeightForWidth(self)</code>
<code>hasMouseTracking(self)</code>
<code>hasTabletTracking(self)</code>
<code>header(self)</code>
<code>height(self)</code>
<code>heightForWidth(self, a0)</code>

continues on next page

Table 3.40 – continued from previous page

<code>heightMM(self)</code>
<code>hide(self)</code>
<code>hideColumn(self, column)</code>
<code>hideEvent(self, a0)</code>
<code>horizontalOffset(self)</code>
<code>horizontalScrollBar(self)</code>
<code>horizontalScrollBarPolicy(self)</code>
<code>horizontalScrollMode(self)</code>
<code>horizontalScrollbarAction(self, action)</code>
<code>horizontalScrollbarValueChanged(self, value)</code>
<code>iconSize(self)</code>
<code>indentation(self)</code>
<code>indexAbove(self, index)</code>
<code>indexAt(self, p)</code>
<code>indexBelow(self, index)</code>
<code>indexRowSizeHint(self, index)</code>
<code>indexWidget(self, index)</code>
<code>inherits(self, classname)</code>
<code>initPainter(self, painter)</code>
<code>initStyleOption(self, option)</code>
<code>inputMethodEvent(self, event)</code>
<code>inputMethodHints(self)</code>
<code>inputMethodQuery(self, query)</code>
<code>insertAction(self, before, action)</code>
<code>insertActions(self, before, actions)</code>

continues on next page

Table 3.40 – continued from previous page

<code>installEventFilter(self, a0)</code>
<code>isActiveWindow(self)</code>
<code>isAncestorOf(self, child)</code>
<code>isAnimated(self)</code>
<code>isColumnHidden(self, column)</code>
<code>isEnabled(self)</code>
<code>isEnabledTo(self, a0)</code>
<code>isExpanded(self, index)</code>
<code>isFirstColumnSpanned(self, row, parent)</code>
<code>isFullScreen(self)</code>
<code>isHeaderHidden(self)</code>
<code>isHidden(self)</code>
<code>isIndexHidden(self, index)</code>
<code>isLeftToRight(self)</code>
<code>isMaximized(self)</code>
<code>isMinimized(self)</code>
<code>isModal(self)</code>
<code>isPersistentEditorOpen(self, index)</code>
<code>isRightToLeft(self)</code>
<code>isRowHidden(self, row, parent)</code>
<code>isSignalConnected(self, signal)</code>
<code>isSortingEnabled(self)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, a0)</code>
<code>isWidgetType(self)</code>

continues on next page

Table 3.40 – continued from previous page

<code>isWindow(self)</code>
<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>itemDelegate(-> QAbstractItemDelegate)</code>
<code>itemDelegateForColumn(self, column)</code>
<code>itemDelegateForRow(self, row)</code>
<code>itemsExpandable(self)</code>
<code>keyPressEvent(self, event)</code>
<code>keyReleaseEvent(self, a0)</code>
<code>keyboardGrabber()</code>
<code>keyboardSearch(self, search)</code>
<code>killTimer(self, id)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code><i>leaveEvent</i>(self, a0)</code>
<code>lineWidth(self)</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 3.40 – continued from previous page

<code>mapToParent(self, a0)</code>
<code>mask(self)</code>
<code>maximumHeight(self)</code>
<code>maximumSize(self)</code>
<code>maximumViewportSize(self)</code>
<code>maximumWidth(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>midLineWidth(self)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>model(self)</code>
<code><i>mouseDoubleClickEvent</i>(self, e)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, event)</code>
<code><i>mousePressEvent</i>(self, e)</code>
<code><i>mouseReleaseEvent</i>(self, event)</code>
<code>move()</code>
<code>moveCursor(self, cursorAction, modifiers)</code>
<code>moveEvent(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>

continues on next page

Table 3.40 – continued from previous page

<code>nextInFocusChain(self)</code>	
<code>normalGeometry(self)</code>	
<code>objectName(self)</code>	
<code>openPersistentEditor(self, index)</code>	
<code>overrideWindowFlags(self, type)</code>	
<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, e)</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>reexpand(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	

continues on next page

Table 3.40 – continued from previous page

<code>removeAction(self, action)</code>
<code>removeEventFilter(self, a0)</code>
<code>render(, sourceRegion, flags, ...)</code>
<code>repaint(-> None -> None)</code>
<code>reset(self)</code>
<code>resetHorizontalScrollMode(self)</code>
<code>resetIndentation(self)</code>
<code>resetVerticalScrollMode(self)</code>
<code>resize()</code>
<code>resizeColumnToContents(self, column)</code>
<code>resizeEvent(self, e)</code>
<code>restoreGeometry(self, geometry)</code>
<code>rootIndex(self)</code>
<code>rootIsDecorated(self)</code>
<code>rowHeight(self, index)</code>
<code>rowsAboutToBeRemoved(self, parent, start, end)</code>
<code>rowsInserted(self, parent, start, end)</code>
<code>rowsRemoved(self, parent, first, last)</code>
<code>saveGeometry(self)</code>
<code>scheduleDelayedItemsLayout(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>scrollBarWidgets(self, alignment)</code>
<code>scrollContentsBy(self, dx, dy)</code>
<code>scrollDirtyRegion(self, dx, dy)</code>

continues on next page

Table 3.40 – continued from previous page

<code>scrollTo(self, index[, hint])</code>
<code>scrollToBottom(self)</code>
<code>scrollToTop(self)</code>
<code>selectAll(self)</code>
<code>selectedIndexes(self)</code>
<code>selectionBehavior(self)</code>
<code>selectionChanged(self, selected, deselected)</code>
<code>selectionCommand(self, index[, event])</code>
<code>selectionMode(self)</code>
<code>selectionModel(self)</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>setAllColumnsShowFocus(self, enable)</code>
<code>setAlternatingRowColors(self, enable)</code>
<code>setAnimated(self, enable)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoExpandDelay(self, delay)</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setAutoScroll(self, enable)</code>
<code>setAutoScrollMargin(self, margin)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>

continues on next page

Table 3.40 – continued from previous page

<code>setColumnHidden(self, column, hide)</code>
<code>setColumnWidth(self, column, width)</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCornerWidget(self, widget)</code>
<code>setCurrentIndex(self, index)</code>
<code>setCursor(self, a0)</code>
<code>setDefaultDropAction(self, dropAction)</code>
<code>setDirtyRegion(self, region)</code>
<code>setDisabled(self, a0)</code>
<code>setDragDropMode(self, behavior)</code>
<code>setDragDropOverwriteMode(self, overwrite)</code>
<code>setDragEnabled(self, enable)</code>
<code>setDropIndicatorShown(self, enable)</code>
<code>setEditTriggers(self, triggers)</code>
<code>setEnabled(self, a0)</code>
<code>setExpanded(self, index, expand)</code>
<code>setExpandsOnDoubleClick(self, enable)</code>
<code>setFirstColumnSpanned(self, row, parent, span)</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>

continues on next page

Table 3.40 – continued from previous page

<code>setFont(self, a0)</code>
<code>setForegroundColor(self, a0)</code>
<code>setFrameRect(self, a0)</code>
<code>setFrameShadow(self, a0)</code>
<code>setFrameShape(self, a0)</code>
<code>setFrameStyle(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setHeader(self, header)</code>
<code>setHeaderHidden(self, hide)</code>
<code>setHidden(self, hidden)</code>
<code>setHorizontalScrollBar(self, scrollbar)</code>
<code>setHorizontalScrollBarPolicy(self, a0)</code>
<code>setHorizontalScrollMode(self, mode)</code>
<code>setIconSize(self, size)</code>
<code>setIndentation(self, i)</code>
<code>setIndexWidget(self, index, widget)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setItemDelegate(self, delegate)</code>
<code>setItemDelegateForColumn(self, column, delegate)</code>
<code>setItemDelegateForRow(self, row, delegate)</code>
<code>setItemsExpandable(self, enable)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLineWidth(self, a0)</code>

continues on next page

Table 3.40 – continued from previous page

<code>setLocale(self, locale)</code>
<code>setMask()</code>
<code>setMaximumHeight(self, maxh)</code>
<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMidLineWidth(self, a0)</code>
<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setModel(self, model)</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>setRootIndex(self, index)</code>
<code>setRootIsDecorated(self, show)</code>
<code>setRowHidden(self, row, parent, hide)</code>
<code>setSelection(self, rect, command)</code>
<code>setSelectionBehavior(self, behavior)</code>
<code>setSelectionMode(self, mode)</code>
<code>setSelectionModel(self, selectionModel)</code>
<code>setShortcutAutoRepeat(self, id[, enabled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeAdjustPolicy(self, policy)</code>

continues on next page

Table 3.40 – continued from previous page

<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setSortingEnabled(self, enable)</code>
<code>setState(self, state)</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabKeyNavigation(self, enable)</code>
<code>setTabOrder(a0, a1)</code>
<code>setTabletTracking(self, enable)</code>
<code>setTextElideMode(self, mode)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setTreePosition(self, logicalIndex)</code>
<code>setUniformRowHeights(self, uniform)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVerticalScrollBar(self, scrollbar)</code>
<code>setVerticalScrollBarPolicy(self, a0)</code>
<code>setVerticalScrollMode(self, mode)</code>
<code>setViewport(self, widget)</code>
<code>setViewportMargins()</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 3.40 – 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>setWordWrap(self, on)</code>
<code>setupViewport(self, viewport)</code>
<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showColumn(self, column)</code>
<code>showDropIndicator(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>sizeAdjustPolicy(self)</code>
<code>sizeHint(self)</code>
<code>sizeHintForColumn(self, column)</code>

continues on next page

Table 3.40 – continued from previous page

<code>sizeHintForIndex(self, index)</code>
<code>sizeHintForRow(self, row)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>sortByColumn(self, column, order)</code>
<code>stackUnder(self, a0)</code>
<code>startDrag(self, supportedActions)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>state(self)</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabKeyNavigation(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>textElideMode(self)</code>
<code>thread(self)</code>
<code>timerEvent(self, event)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>
<code>treePosition(self)</code>
<code>underMouse(self)</code>
<code>ungrabGesture(self, type)</code>
<code>uniformRowHeights(self)</code>

continues on next page

Table 3.40 – continued from previous page

<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>
<code>update()</code>
<code>updateEditorData(self)</code>
<code>updateEditorGeometries(self)</code>
<code>updateGeometries(self)</code>
<code>updateGeometry(self)</code>
<code>updateMicroFocus(self)</code>
<code>updatesEnabled(self)</code>
<code>verticalOffset(self)</code>
<code>verticalScrollBar(self)</code>
<code>verticalScrollBarPolicy(self)</code>
<code>verticalScrollMode(self)</code>
<code>verticalScrollbarAction(self, action)</code>
<code>verticalScrollbarValueChanged(self, value)</code>
<code>viewOptions(self)</code>
<code>viewport(self)</code>
<code>viewportEvent(self, event)</code>
<code>viewportMargins(self)</code>
<code>viewportSizeHint(self)</code>
<code>visibleRegion(self)</code>
<code>visualRect(self, index)</code>
<code>visualRegionForSelection(self, selection)</code>
<code>whatsThis(self)</code>

continues on next page

Table 3.40 – continued from previous page

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)
windowType(self)
wordWrap(self)
x(self)
y(self)

Attributes

AboveItem
AdjustIgnored
AdjustToContents
AdjustToContentsOnFirstShow

continues on next page

Table 3.41 – continued from previous page

AllEditTriggers
AnimatingState
AnyKeyPressed
BelowItem
Box
CollapsingState
ContiguousSelection
CurrentChanged
DoubleClicked
DragDrop
DragOnly
DragSelectingState
DraggingState
DrawChildren
DrawWindowBackground
DropOnly
EditKeyPressed
EditingState
EnsureVisible
ExpandingState
ExtendedSelection
HLine
IgnoreMask
InternalMove
MoveDown

continues on next page

Table 3.41 – continued from previous page

MoveEnd
MoveHome
MoveLeft
MoveNext
MovePageDown
MovePageUp
MovePrevious
MoveRight
MoveUp
MultiSelection
NoDragDrop
NoEditTriggers
NoFrame
NoSelection
NoState
OnItem
OnViewport
Panel
PdmDepth
PdmDevicePixelRatio
PdmDevicePixelRatioScaled
PdmDpiX
PdmDpiY
PdmHeight
PdmHeightMM

continues on next page

Table 3.41 – continued from previous page

PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
Plain	
PositionAtBottom	
PositionAtCenter	
PositionAtTop	
Raised	
ScrollPerItem	
ScrollPerPixel	
SelectColumns	
SelectItems	
SelectRows	
SelectedClicked	
Shadow_Mask	
Shape_Mask	
SingleSelection	
StyledPanel	
Sunken	
VLine	
WinPanel	
activated	QModelIndex) [signal]
clicked	QModelIndex) [signal]
collapsed	QModelIndex) [signal]
customContextMenuRequested	QPoint) [signal]
destroyed	typing.Optional[QObject] = None) [signal]

continues on next page

Table 3.41 – continued from previous page

<code>doubleClicked</code>	<code>QModelIndex</code> [signal]
<code>doubleLeftClicked</code>	A <code>QTreeView</code> that emits a custom signal <code>leftClicked(index)</code> after a left click on a valid index, and <code>doubleLeftClicked(index)</code> (in addition) on double click.
<code>entered</code>	<code>QModelIndex</code> [signal]
<code>expanded</code>	<code>QModelIndex</code> [signal]
<code>iconSizeChanged</code>	<code>QSize</code> [signal]
<code>leftClicked</code>	
<code>objectNameChanged</code>	<code>str</code> [signal]
<code>pressed</code>	<code>QModelIndex</code> [signal]
<code>staticMetaObject</code>	
<code>viewportEntered</code>	<code>viewportEntered(self)</code> [signal]
<code>windowIconChanged</code>	<code>QIcon</code> [signal]
<code>windowIconTextChanged</code>	<code>str</code> [signal]
<code>windowTitleChanged</code>	<code>str</code> [signal]

doubleLeftClicked

A `QTreeView` that emits a custom signal `leftClicked(index)` after a left click on a valid index, and `doubleLeftClicked(index)` (in addition) on double click.

`leaveEvent(self, a0: QEvent)`

`leftClicked`

`mouseDoubleClickEvent(self, e: QMouseEvent)`

`mousePressEvent(self, e: QMouseEvent)`

`mouseReleaseEvent(self, event: QMouseEvent)`

3.6.16 lyse.__main__.UneditableModel

class `lyse.__main__.UneditableModel`

Bases: `PyQt5.QtGui.QStandardItemModel`

`__init__(*args, **kwargs)`

Methods

`__init__(*args, **kwargs)`

`addColumn(self, items)`

`appendRow()`

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

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

continues on next page

Table 3.42 – continued from previous page

<code>beginMoveColumns(self, sourceParent, ...)</code>
<code>beginMoveRows(self, sourceParent, ...)</code>
<code>beginRemoveColumns(self, parent, first, last)</code>
<code>beginRemoveRows(self, parent, first, last)</code>
<code>beginResetModel(self)</code>
<code>blockSignals(self, b)</code>
<code>buddy(self, index)</code>
<code>canDropMimeData(self, data, action, row, ...)</code>
<code>canFetchMore(self, parent)</code>
<code>changePersistentIndex(self, from_, to)</code>
<code>changePersistentIndexList(self, from_, to)</code>
<code>checkIndex(self, index[, options])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>clear(self)</code>
<code>clearItemData(self, index)</code>
<code>columnCount(self[, parent])</code>
<code>connectNotify(self, signal)</code>
<code>createIndex(self, row, column[, object])</code>
<code>customEvent(self, a0)</code>
<code>data(self, index[, role])</code>
<code>decodeData(self, row, column, parent, stream)</code>
<code>deleteLater(self)</code>
<code>disconnect(-> bool)</code>
<code>disconnectNotify(self, signal)</code>

continues on next page

Table 3.42 – continued from previous page

<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(-> QObject)</code>	
<code>findChildren(...)</code>	
<code>findItems(self, text[, flags, column])</code>	
<code><i>flags</i>(index)</code>	Return flags as normal except that the <code>ItemIsEditable</code> flag is always <code>False</code>
<code>hasChildren(self[, parent])</code>	
<code>hasIndex(self, row, column[, parent])</code>	
<code>headerData(self, section, orientation[, role])</code>	
<code>horizontalHeaderItem(self, column)</code>	
<code>index(self, row, column[, parent])</code>	
<code>indexFromItem(self, item)</code>	

continues on next page

Table 3.42 – continued from previous page

<code>inherits(self, classname)</code>
<code>insertColumn()</code>
<code>insertColumns(self, column, count[, parent])</code>
<code>insertRow(-> None)</code>
<code>insertRows(self, row, count[, parent])</code>
<code>installEventFilter(self, a0)</code>
<code>invisibleRootItem(self)</code>
<code>isSignalConnected(self, signal)</code>
<code>isWidgetType(self)</code>
<code>isWindowType(self)</code>
<code>item(self, row[, column])</code>
<code>itemData(self, index)</code>
<code>itemFromIndex(self, index)</code>
<code>itemPrototype(self)</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>
<code>objectName(self)</code>

continues on next page

Table 3.42 – continued from previous page

<code>parent(-> QModelIndex)</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(self[, parent])</code>	
<code>sender(self)</code>	
<code>senderSignalIndex(self)</code>	
<code>setColumnCount(self, columns)</code>	
<code>setData(self, index, value[, role])</code>	
<code>setHeaderData(self, section, orientation, value)</code>	
<code>setHorizontalHeaderItem(self, column, item)</code>	
<code>setHorizontalHeaderLabels(self, labels)</code>	
<code>setItem()</code>	
<code>setItemData(self, index, roles)</code>	
<code>setItemPrototype(self, item)</code>	
<code>setItemRoleNames(self, roleNames)</code>	

continues on next page

Table 3.42 – continued from previous page

<code>setObjectName(self, name)</code>
<code>setParent(self, a0)</code>
<code>setProperty(self, name, value)</code>
<code>setRowCount(self, rows)</code>
<code>setSortRole(self, role)</code>
<code>setVerticalHeaderItem(self, row, item)</code>
<code>setVerticalHeaderLabels(self, labels)</code>
<code>sibling(self, row, column, idx)</code>
<code>signalsBlocked(self)</code>
<code>sort(self, column[, order])</code>
<code>sortRole(self)</code>
<code>span(self, index)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>submit(self)</code>
<code>supportedDragActions(self)</code>
<code>supportedDropActions(self)</code>
<code>takeColumn(self, column)</code>
<code>takeHorizontalHeaderItem(self, column)</code>
<code>takeItem(self, row[, column])</code>
<code>takeRow(self, row)</code>
<code>takeVerticalHeaderItem(self, row)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>
<code>verticalHeaderItem(self, row)</code>

Attributes

HorizontalSortHint	
NoLayoutChangeHint	
VerticalSortHint	
columnsAboutToBeInserted	int, last: int) [signal]
columnsAboutToBeMoved	int, sourceEnd: int, destinationParent: QModelIndex, destinationColumn: int) [signal]
columnsAboutToBeRemoved	int, last: int) [signal]
columnsInserted	int, last: int) [signal]
columnsMoved	int, end: int, destination: QModelIndex, column: int) [signal]
columnsRemoved	int, last: int) [signal]
dataChanged	QModelIndex, roles: Iterable[int] = []) [signal]
destroyed	typing.Optional[QObject] = None) [signal]
headerDataChanged	int, last: int) [signal]
itemChanged	QStandardItem) [signal]
layoutAboutToBeChanged	QAbstractItemModel.LayoutChangeHint = QAbstractItemModel.NoLayoutChangeHint) [signal]
layoutChanged	QAbstractItemModel.LayoutChangeHint = QAbstractItemModel.NoLayoutChangeHint) [signal]
modelAboutToBeReset	modelAboutToBeReset(self) [signal]
modelReset	modelReset(self) [signal]
objectNameChanged	str) [signal]
rowsAboutToBeInserted	int, last: int) [signal]
rowsAboutToBeMoved	int, sourceEnd: int, destinationParent: QModelIndex, destinationRow: int) [signal]
rowsAboutToBeRemoved	int, last: int) [signal]
rowsInserted	int, last: int) [signal]
rowsMoved	int, end: int, destination: QModelIndex, row: int) [signal]
rowsRemoved	int, last: int) [signal]
staticMetaObject	

flags(*index*)

Return flags as normal except that the `ItemsIsEditable` flag is always `False`

3.6.17 lyse.__main__.WebServer

```
class lyse.__main__.WebServer(*args, **kwargs)
    Bases: labscript_utils.ls_zprocess.ZMQServer
```

Parameters

- **args** (*Any*) –
- **kwargs** (*Any*) –

Return type *Any*

`__init__(*args, **kwargs)`

Parameters

- **args** (*Any*) –
- **kwargs** (*Any*) –

Return type None

Methods

`__init__(*args, **kwargs)`

`handler(request_data)`

handler(*request_data*)

LABSCRIPT SUITE COMPONENTS

The *labscript suite* is modular by design, and is comprised of:

Table 4.1: Python libraries

	labscript — Expressive composition of hardware-timed experiments
	labscript-devices — Plugin architecture for controlling experiment hardware
	labscript-utils — Shared modules used by the <i>labscript suite</i>

Table 4.2: Graphical applications

	runmanager — Graphical and remote interface to parameterized experiments
	blacs — Graphical interface to scientific instruments and experiment supervision
	lyse — Online analysis of live experiment data
	runviewer — Visualize hardware-timed experiment instructions

PYTHON MODULE INDEX

|

lyse, 9

lyse.__main__, 49

lyse.analysis_subprocess, 23

lyse.dataframe_utilities, 46

lyse.figure_manager, 47

lyse.tempfile2clipboard, 49

Symbols

__init__() (lyse.Run method), 12
 __init__() (lyse.Sequence method), 21
 __init__() (lyse.__main__.AnalysisRoutine method), 50
 __init__() (lyse.__main__.DataFrameModel method), 51
 __init__() (lyse.__main__.EditColumns method), 55
 __init__() (lyse.__main__.EditColumnsDialog method), 57
 __init__() (lyse.__main__.FileBox method), 71
 __init__() (lyse.__main__.ItemDelegate method), 72
 __init__() (lyse.__main__.Lyse method), 75
 __init__() (lyse.__main__.LyseMainWindow method), 76
 __init__() (lyse.__main__.RoutineBox method), 92
 __init__() (lyse.__main__.TableView method), 94
 __init__() (lyse.__main__.TreeView method), 118
 __init__() (lyse.__main__.UneditableModel method), 142
 __init__() (lyse.__main__.WebServer method), 148
 __init__() (lyse.analysis_subprocess.AnalysisWorker method), 23
 __init__() (lyse.analysis_subprocess.Plot method), 24
 __init__() (lyse.analysis_subprocess.PlotWindow method), 26
 __init__() (lyse.analysis_subprocess.PlotWindowCloseEvent method), 39
 __init__() (lyse.figure_manager.FigureManager method), 48

A

add_files() (lyse.__main__.DataFrameModel method), 54
 add_routines() (lyse.__main__.RoutineBox method), 94
 analysis_complete() (lyse.analysis_subprocess.Plot method), 25
 analysis_loop() (lyse.__main__.FileBox method), 71
 analysis_loop() (lyse.__main__.RoutineBox method), 94
 AnalysisRoutine (class in lyse.__main__), 50

AnalysisWorker (class in lyse.analysis_subprocess), 23
 are_you_sure() (lyse.__main__.Lyse method), 76
 asdatetime() (in module lyse.dataframe_utilities), 46

C

cancel() (lyse.__main__.EditColumns method), 56
 check_child_exited() (lyse.__main__.AnalysisRoutine method), 51
 clear() (lyse.analysis_subprocess.Plot method), 25
 close() (lyse.__main__.EditColumns method), 56
 close() (lyse.figure_manager.FigureManager method), 48
 close_signal (lyse.__main__.EditColumnsDialog attribute), 70
 close_signal (lyse.analysis_subprocess.PlotWindow attribute), 39
 closeEvent() (lyse.__main__.EditColumnsDialog method), 70
 closeEvent() (lyse.__main__.LyseMainWindow method), 92
 closeEvent() (lyse.analysis_subprocess.PlotWindow method), 39
 COL_ACTIVE (lyse.__main__.RoutineBox attribute), 93
 COL_FILEPATH (lyse.__main__.DataFrameModel attribute), 54
 COL_NAME (lyse.__main__.EditColumns attribute), 56
 COL_NAME (lyse.__main__.RoutineBox attribute), 93
 COL_STATUS (lyse.__main__.DataFrameModel attribute), 54
 COL_STATUS (lyse.__main__.RoutineBox attribute), 93
 COL_VISIBLE (lyse.__main__.EditColumns attribute), 56
 columns_changed (lyse.__main__.DataFrameModel attribute), 54
 concat_with_padding() (in module lyse.dataframe_utilities), 46
 connect_signals() (lyse.__main__.DataFrameModel method), 54
 connect_signals() (lyse.__main__.EditColumns method), 56
 connect_signals() (lyse.__main__.FileBox method), 71

- connect_signals() (*lyse.__main__.Lyse method*), 76
connect_signals() (*lyse.__main__.RoutineBox method*), 94
- ## D
- data() (*in module lyse*), 10
DataFrameModel (*class in lyse.__main__*), 51
delay_results_return() (*in module lyse*), 11
delayedClose() (*lyse.__main__.LyseMainWindow method*), 92
delete_items() (*lyse.__main__.Lyse method*), 76
do_analysis() (*lyse.__main__.AnalysisRoutine method*), 51
do_analysis() (*lyse.__main__.RoutineBox method*), 94
do_analysis() (*lyse.analysis_subprocess.AnalysisWorker method*), 24
do_multishot_analysis() (*lyse.__main__.FileBox method*), 71
do_singleshots_analysis() (*lyse.__main__.FileBox method*), 71
do_sort() (*lyse.__main__.EditColumns method*), 56
doubleLeftClicked (*lyse.__main__.TableView attribute*), 118
doubleLeftClicked (*lyse.__main__.TreeView attribute*), 142
draw() (*lyse.analysis_subprocess.Plot method*), 25
- ## E
- EditColumns (*class in lyse.__main__*), 55
EditColumnsDialog (*class in lyse.__main__*), 57
enabled() (*lyse.__main__.AnalysisRoutine method*), 51
end_child() (*lyse.__main__.AnalysisRoutine method*), 51
error_dialog() (*in module lyse.__main__*), 49
EXTRA_ROW_HEIGHT (*lyse.__main__.ItemDelegate attribute*), 75
- ## F
- figure_to_clipboard() (*in module lyse*), 11
FigureManager (*class in lyse.figure_manager*), 48
FileBox (*class in lyse.__main__*), 71
firstPaint (*lyse.__main__.LyseMainWindow attribute*), 92
flags() (*lyse.__main__.UneditableModel method*), 148
flat_dict_to_flat_series() (*in module lyse.dataframe_utilities*), 46
flat_dict_to_hierarchical_dataframe() (*in module lyse.dataframe_utilities*), 47
flatten_dict() (*in module lyse.dataframe_utilities*), 47
- ## G
- get_all_image_labels() (*lyse.Run method*), 13
get_attrs() (*lyse.Run method*), 13
get_dataframe_from_shot() (*in module lyse.dataframe_utilities*), 47
get_dataframe_from_shots() (*in module lyse.dataframe_utilities*), 47
get_first_empty_figure() (*lyse.figure_manager.FigureManager method*), 48
get_first_incomplete() (*lyse.__main__.DataFrameModel method*), 54
get_globals() (*lyse.Run method*), 13
get_globals_expansion() (*lyse.Run method*), 14
get_globals_raw() (*lyse.Run method*), 14
get_image() (*lyse.Run method*), 14
get_image() (*lyse.Sequence method*), 22
get_image_attributes() (*lyse.Run method*), 14
get_images() (*lyse.Run method*), 14
get_images_dict() (*lyse.Run method*), 15
get_nested_dict_from_shot() (*in module lyse.dataframe_utilities*), 47
get_plot_class() (*in module lyse*), 11
get_result() (*lyse.Run method*), 15
get_result_array() (*lyse.Run method*), 15
get_result_array() (*lyse.Sequence method*), 22
get_result_arrays() (*lyse.Run method*), 15
get_result_arrays() (*lyse.Sequence method*), 22
get_results() (*lyse.Run method*), 15
get_row_index() (*lyse.__main__.AnalysisRoutine method*), 51
get_save_data() (*lyse.__main__.Lyse method*), 76
get_screen_geometry() (*in module lyse.__main__*), 49
get_series_from_shot() (*in module lyse.dataframe_utilities*), 47
get_trace() (*lyse.Run method*), 16
get_trace() (*lyse.Sequence method*), 23
get_traces() (*lyse.Run method*), 16
get_traces() (*lyse.Sequence method*), 23
get_units() (*lyse.Run method*), 16
get_wait() (*lyse.Run method*), 16
get_waits() (*lyse.Run method*), 17
get_window_state() (*lyse.analysis_subprocess.Plot method*), 25
globals_diff() (*in module lyse*), 11
globals_diff() (*lyse.Run method*), 17
globals_groups() (*lyse.Run method*), 17
group (*lyse.Run property*), 17
- ## H
- h5_file (*lyse.Run property*), 17
h5_path (*lyse.Run property*), 17
handler() (*lyse.__main__.WebServer method*), 149

- I**
- `incoming_buffer_loop()` (*lyse.__main__.FileBox method*), 71
 - `infer_objects()` (*lyse.__main__.DataFrameModel method*), 54
 - `install()` (*in module lyse.figure_manager*), 48
 - `is_shown` (*lyse.analysis_subprocess.Plot property*), 25
 - `ItemDelegate` (*class in lyse.__main__*), 72
- L**
- `leaveEvent()` (*lyse.__main__.TableView method*), 118
 - `leaveEvent()` (*lyse.__main__.TreeView method*), 142
 - `leftClicked` (*lyse.__main__.TableView attribute*), 118
 - `leftClicked` (*lyse.__main__.TreeView attribute*), 142
 - `load_configuration()` (*lyse.__main__.Lyse method*), 76
 - `load_window_geometry_configuration()` (*lyse.__main__.Lyse method*), 76
 - `lyse`
 - module, 9
 - `Lyse` (*class in lyse.__main__*), 75
 - `lyse.__main__`
 - module, 49
 - `lyse.analysis_subprocess`
 - module, 23
 - `lyse.dataframe_utilities`
 - module, 46
 - `lyse.figure_manager`
 - module, 47
 - `lyse.tempfile2clipboard`
 - module, 49
 - `LyseMainWindow` (*class in lyse.__main__*), 76
- M**
- `main()` (*in module lyse.tempfile2clipboard*), 49
 - `mainloop()` (*lyse.analysis_subprocess.AnalysisWorker method*), 24
 - `make_it_so()` (*lyse.__main__.EditColumns method*), 56
 - `mark_as_deleted_off_disk()` (*lyse.__main__.DataFrameModel method*), 55
 - `mark_selection_not_done()` (*lyse.__main__.DataFrameModel method*), 55
 - `module`
 - `lyse`, 9
 - `lyse.__main__`, 49
 - `lyse.analysis_subprocess`, 23
 - `lyse.dataframe_utilities`, 46
 - `lyse.figure_manager`, 47
 - `lyse.tempfile2clipboard`, 49
 - `mouseDoubleClickEvent()` (*lyse.__main__.TableView method*), 118
 - `mouseDoubleClickEvent()` (*lyse.__main__.TreeView method*), 142
 - `mousePressEvent()` (*lyse.__main__.TableView method*), 118
 - `mousePressEvent()` (*lyse.__main__.TreeView method*), 142
 - `mouseReleaseEvent()` (*lyse.__main__.TableView method*), 118
 - `mouseReleaseEvent()` (*lyse.__main__.TreeView method*), 142
- N**
- `new_figure()` (*lyse.analysis_subprocess.AnalysisWorker method*), 24
 - `new_row()` (*lyse.__main__.DataFrameModel method*), 55
 - `no_write` (*lyse.Run property*), 17
- O**
- `on_add_routines_clicked()` (*lyse.__main__.RoutineBox method*), 94
 - `on_add_shot_files_clicked()` (*lyse.__main__.FileBox method*), 71
 - `on_analysis_running_toggled()` (*lyse.__main__.FileBox method*), 72
 - `on_close()` (*lyse.analysis_subprocess.Plot method*), 25
 - `on_close_event()` (*lyse.__main__.Lyse method*), 76
 - `on_columns_changed()` (*lyse.__main__.FileBox method*), 72
 - `on_copy_to_clipboard_triggered()` (*lyse.analysis_subprocess.Plot method*), 25
 - `on_double_click()` (*lyse.__main__.DataFrameModel method*), 55
 - `on_edit_columns_clicked()` (*lyse.__main__.FileBox method*), 72
 - `on_filter_text_edited()` (*lyse.__main__.EditColumns method*), 56
 - `on_load_configuration_triggered()` (*lyse.__main__.Lyse method*), 76
 - `on_load_dataframe_triggered()` (*lyse.__main__.Lyse method*), 76
 - `on_lock_axes_triggered()` (*lyse.analysis_subprocess.Plot method*), 25
 - `on_mark_selection_not_done_clicked()` (*lyse.__main__.FileBox method*), 72
 - `on_model_item_changed()` (*lyse.__main__.EditColumns method*), 56
 - `on_model_item_changed()` (*lyse.__main__.RoutineBox method*), 94
 - `on_move_down_clicked()` (*lyse.__main__.RoutineBox method*), 94

- on_move_to_bottom_clicked() (lyse.__main__.RoutineBox method), 94
 on_move_to_top_clicked() (lyse.__main__.RoutineBox method), 94
 on_move_up_clicked() (lyse.__main__.RoutineBox method), 94
 on_remove_selection() (lyse.__main__.DataFrameModel method), 55
 on_remove_selection() (lyse.__main__.RoutineBox method), 94
 on_restart_selected_triggered() (lyse.__main__.RoutineBox method), 94
 on_revert_configuration_triggered() (lyse.__main__.Lyse method), 76
 on_run_multishot_analysis_clicked() (lyse.__main__.FileBox method), 72
 on_save_configuration_as_triggered() (lyse.__main__.Lyse method), 76
 on_save_configuration_triggered() (lyse.__main__.Lyse method), 76
 on_save_dataframe_triggered() (lyse.__main__.Lyse method), 76
 on_select_all_state_changed() (lyse.__main__.EditColumns method), 56
 on_select_all_state_changed() (lyse.__main__.RoutineBox method), 94
 on_set_selected_triggered() (lyse.__main__.EditColumns method), 56
 on_set_selected_triggered() (lyse.__main__.RoutineBox method), 94
 on_treeView_context_menu_requested() (lyse.__main__.EditColumns method), 56
 on_treeView_context_menu_requested() (lyse.__main__.RoutineBox method), 94
 on_treeview_double_left_clicked() (lyse.__main__.RoutineBox method), 94
 on_view_context_menu_requested() (lyse.__main__.DataFrameModel method), 55
 only_window_geometry_is_different() (lyse.__main__.Lyse method), 76
 open() (lyse.Run method), 17
 open_file() (in module lyse), 11
- ## P
- pad_columns() (in module lyse.dataframe_utilities), 47
 paint() (lyse.__main__.ItemDelegate method), 75
 paintEvent() (lyse.__main__.LyseMainWindow method), 92
 pause_analysis() (lyse.__main__.FileBox method), 72
 Plot (class in lyse.analysis_subprocess), 24
 PlotWindow (class in lyse.analysis_subprocess), 26
 PlotWindowCloseEvent (class in lyse.analysis_subprocess), 39
 populate_model() (lyse.__main__.EditColumns method), 56
 post_analysis_plot_actions() (lyse.analysis_subprocess.AnalysisWorker method), 24
 pre_analysis_plot_actions() (lyse.analysis_subprocess.AnalysisWorker method), 24
- ## Q
- question_dialog() (in module lyse.__main__), 50
- ## R
- register_plot_class() (in module lyse), 12
 remove() (lyse.__main__.AnalysisRoutine method), 51
 remove_selection() (lyse.__main__.DataFrameModel method), 55
 remove_selection() (lyse.__main__.RoutineBox method), 94
 renumber_rows() (lyse.__main__.DataFrameModel method), 55
 reorder() (lyse.__main__.RoutineBox method), 94
 replace_with_padding() (in module lyse.dataframe_utilities), 47
 reset() (lyse.figure_manager.FigureManager method), 48
 reset_figs() (lyse.analysis_subprocess.AnalysisWorker method), 24
 restart() (lyse.__main__.AnalysisRoutine method), 51
 restore_axis_limits() (lyse.analysis_subprocess.Plot method), 25
 restore_window_state() (lyse.analysis_subprocess.Plot method), 25
 ROLE_DELETED_OFF_DISK (lyse.__main__.DataFrameModel attribute), 54
 ROLE_FULLPATH (lyse.__main__.RoutineBox attribute), 93
 ROLE_SORT_DATA (lyse.__main__.EditColumns attribute), 56
 ROLE_SORTINDEX (lyse.__main__.RoutineBox attribute), 94
 ROLE_STATUS_PERCENT (lyse.__main__.DataFrameModel attribute), 54
 RoutineBox (class in lyse.__main__), 92
 Run (class in lyse), 12
- ## S
- save_axis_limits() (lyse.analysis_subprocess.Plot method), 25

save_configuration() (*lyse.__main__.Lyse method*), 76
 save_result() (*lyse.Run method*), 18
 save_result_array() (*lyse.Run method*), 19
 save_result_arrays() (*lyse.Run method*), 20
 save_results() (*lyse.Run method*), 20
 save_results_dict() (*lyse.Run method*), 20
 scientific_notation() (*in module lyse.__main__*), 50
 Sequence (*class in lyse*), 21
 set_add_shots_progress() (*lyse.__main__.FileBox method*), 72
 set_columns_visible() (*lyse.__main__.DataFrameModel method*), 55
 set_columns_visible() (*lyse.__main__.FileBox method*), 72
 set_first_figure_current() (*lyse.figure_manager.FigureManager method*), 48
 set_group() (*lyse.Run method*), 20
 set_status() (*lyse.__main__.AnalysisRoutine method*), 51
 set_status_percent() (*lyse.__main__.DataFrameModel method*), 55
 set_window_title() (*lyse.analysis_subprocess.Plot method*), 25
 setup_config() (*lyse.__main__.Lyse method*), 76
 show() (*lyse.__main__.EditColumns method*), 56
 show() (*lyse.analysis_subprocess.Plot method*), 25
 show() (*lyse.figure_manager.FigureManager method*), 48
 sizeHint() (*lyse.__main__.ItemDelegate method*), 75
 start_worker() (*lyse.__main__.AnalysisRoutine method*), 51

T

TableView (*class in lyse.__main__*), 94
 terminate_all_workers() (*lyse.__main__.Lyse method*), 76
 todo() (*lyse.__main__.RoutineBox method*), 94
 trace_names() (*lyse.Run method*), 21
 TreeView (*class in lyse.__main__*), 118

U

UneditableModel (*class in lyse.__main__*), 142
 update_column_levels() (*lyse.__main__.DataFrameModel method*), 55
 update_columns() (*lyse.__main__.EditColumns method*), 56
 update_row() (*lyse.__main__.DataFrameModel method*), 55

update_select_all_checkstate() (*lyse.__main__.EditColumns method*), 57
 update_select_all_checkstate() (*lyse.__main__.RoutineBox method*), 94
 update_visible_state() (*lyse.__main__.EditColumns method*), 57
 update_window_size() (*lyse.analysis_subprocess.Plot method*), 25

W

WebServer (*class in lyse.__main__*), 148
 workers_terminated() (*lyse.__main__.Lyse method*), 76