

---

# lyse

*Release 3.0.0b3.dev8+gc8d46e9*

labscript suite contributors

Jun 14, 2020



# CONTENTS

1	Release Notes	3
2	lyse	5
3	Indices and tables	7



Functions, classes, and methods for use in lyse analysis routines.



## RELEASE NOTES

### 1.1 lyse 2.6.0

*Release date: 2020-02-17*

This release includes six bugfixes, two changes to work with newer versions of a library, and two enhancements.

#### 1.1.1 Bug Fixes

- Fix bug where lyse would fail to terminate subprocesses upon quit. Normal subprocesses would eventually terminate themselves, but this caused lyse to hang whilst closing waiting for them. (PR #63)
- Increase the timeout for starting subprocesses to 30 seconds. Slow computers starting lyse, or when starting many subprocesses at once, would often cause timeouts which necessitated restarting lyse. (PR #64)
- Fix bug where lyse's close button was unresponsive if there were no analysis routines. (PR #67)
- Fix incorrect logic for updating the Qt model and dataframe in lyse - it is hoped that this will resolve an intermittent issue (issue #45) in which the dataframe and Qt model are sometimes not updated after analysis completes. If anyone sees the buggy behaviour again, please reopen issue #45 or a new bug report. (PR #68)
- Fix a bug where lyse could crash if loading shots with different numbers of levels in the hierarchy of their corresponding dataframes. (PR #71)
- In analysis routine subprocesses, force the multiprocessing module to spawn new processes instead of forking. This allows analysis routines to use the multiprocessing module, whereas otherwise this causes crashes due to zmq not being fork-safe. This only affects Unix, where forking is the default behaviour of the multiprocessing module - on Windows this was not an issue since processes were always spawned fresh. (PR #65)

#### 1.1.2 Enhancements

- Expand the allowed datatypes that may be saved/loaded as HDF5 attributes via `Run.save_result()` etc to anything supported by `labscript_utils.properties.get/set_attribute(s)`. This means that datatypes not supported natively by HDF5 are JSON encoded and saved as a string with a prefix indicating this. (PRs #66 and #69)
- Allow instantiating a `Run()` object from within a function, not just at the global scope. The default group that results will be saved to will be the name of the file the `Run()` object is instantiated in. (PR #72)

### 1.1.3 API Changes

- Update to no longer use the deprecated `pandas.convert_objects()` method, in favour of the newer `infer_objects` method. ([PR #70](#))
- Do not throw a version error about `pandas v1.0`



## LYSE

**lyse** is a data analysis framework for experiments controlled by the labscript suite. Analysis routines are Python scripts that can be run on a labscript shot file. We break analysis into two distinct groups:

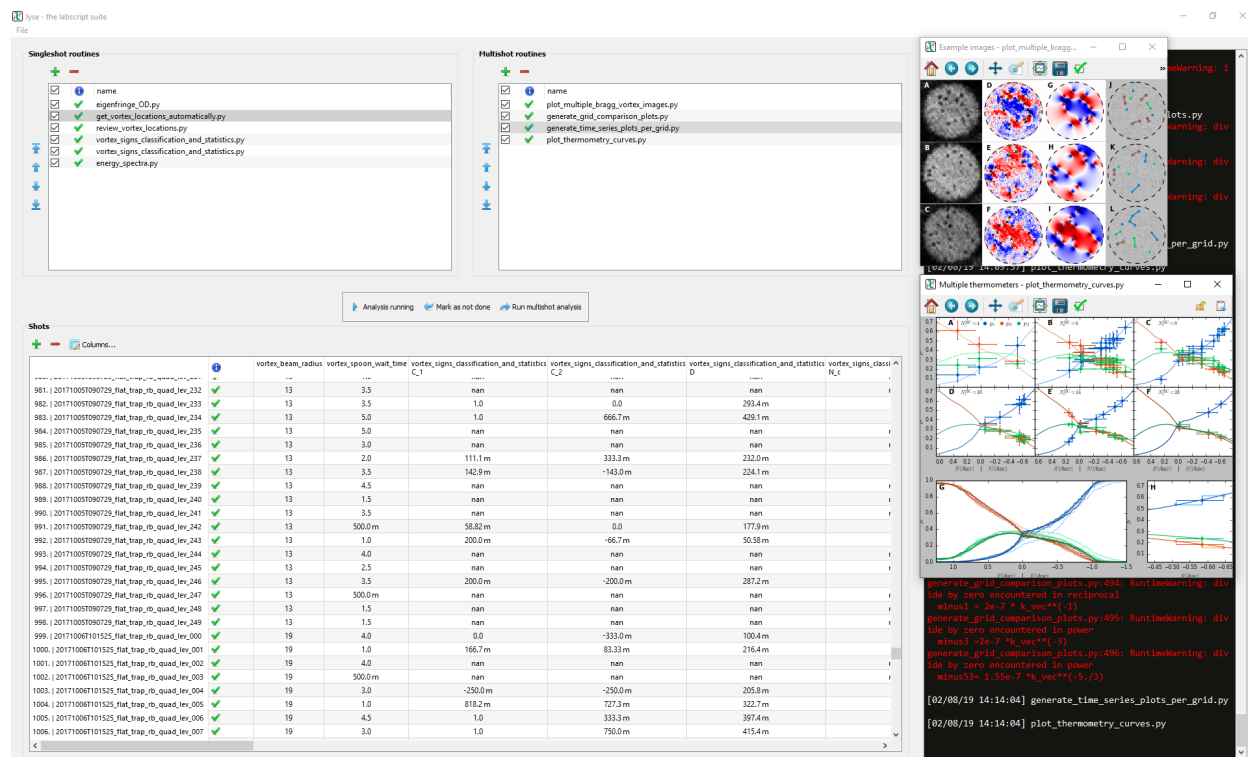
1. **single-shot analysis** routines analyse a single hdf5 file corresponding to one realisation of an experimental sequence, or *shot*; and
2. **multi-shot analysis** routines perform higher-order analysis on data from many shots.

A set of Python scripts can be loaded for each analysis category, which will run (when appropriate) on new shot files as they are sent to lyse from **blacs**. Acquired data, global variables from **runmanager** and analysis results generated in lyse, for all loaded shots, are stored in a table (a [pandas DataFrame](#)). This is a powerful analysis schema that takes advantage of the self-documenting nature of labscript experiment shot files.

The DataFrame is displayed in the lyse GUI and can also be accessed remotely from another PC as follows:

```
import lyse
df = lyse.data(host='127.0.0.1', port=42519)
```

Moreover, the lyse DataFrame can be serialised so that an analysis session can be restored in lyse or in a headless environment.



The lyse interface, comprising:

- Single- and multi-shot analysis routines (Python scripts);
- A graphical representation of the Pandas DataFrame;
- Figures generated by the analysis scripts.<sup>†</sup>
- The output log from lyse and the analysis routines.

<sup>†</sup> In this example, these are publication quality figures generated for the manuscript *Science* **364**, pp. 1267 (2019) doi:10.1126/science.aat5793.

## 2.1 Installation

lyse can be installed on Python 3.6 and later by running:

```
$ pip install lyse
```

Or using conda, from the *labscript suite* channel on *Anaconda Cloud*:

```
$ conda install -c labscript-suite lyse
```

For further details, see the *labscript suite* installation documentation.

## INDICES AND TABLES

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