

---

# **runmanager**

***Release 3.3.0.dev11+g7067a77***

**labscript suite contributors**

**Feb 09, 2024**



# DOCUMENTATION

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>5</b>
2.1	The graphical interface . . . . .	5
2.2	Managing global variables . . . . .	7
2.3	Parameter space scans . . . . .	13
2.4	Evaluation of globals . . . . .	14
2.5	Shot creation . . . . .	15
<b>3</b>	<b>API Reference</b>	<b>17</b>
3.1	runmanager . . . . .	17
3.2	runmanager.functions . . . . .	28
3.3	runmanager.remote . . . . .	28
3.4	runmanager.batch_compiler . . . . .	35
3.5	runmanager.globals_diff . . . . .	35
3.6	runmanager.__main__ . . . . .	35
<b>4</b>	<b>labcrypt suite components</b>	<b>209</b>
	<b>Python Module Index</b>	<b>211</b>
	<b>Index</b>	<b>213</b>



A graphical and remote interface to parameterized experiments.



## INTRODUCTION

Runmanager (see [Fig. 1.1](#)) is the primary interface for adjusting the behaviour of the experiment and generating shot files. Runmanager allows you to define parameters via a graphical interface and select the experiment logic file to use. The parameters are inserted into the experiment logic (as global variables) when shot generation is started via the 'Engage' button.

The parameters can take the form of any Python data type that can be stored in a hdf5 file. When a parameter is specified as a list or array, runmanager automatically detects that you wish to explore a parameter space. Runmanager will generate a separate shot for each point in the array. If multiple arrays are specified across two or more parameters, runmanager automatically takes the Cartesian product of those parameters in order to generate shots that span the entire parameter space. For more complex experiments, parameters can be linked together so they iterate in lock step by defining one or more zip groups. If multiple zip groups are defined, the parameter space explored will be the Cartesian product of each zip group and any other parameters containing an array or list.

The screenshot shows the runmanager interface with the following configuration:

- File menu:** Engage (225 shots), Abort, Run shot(s), Restart subprocess, Shuffle.
- Parameters:**
  - labscript file: C:\labscript\_suite\labscript\_py27\userlib\labscriptlib\krb\krb\_feshbach.py
  - Shot output folder: C:\labscript\_suite\Experiments\labscript27\krb\_feshbach\2018\04\16
  - BLACS hostname: localhost
- Groups:**
  - Output
  - Axes
  - Groups
  - Stages
  - MOT - Central
  - MOT - K
  - K Recapture
  - K Optical Pumping
  - MOT - Rb
  - Compress MOT
  - PGC
  - Rb Optical Pumping
  - Magnetic Trap
  - MT Compress
  - Microwave Evaporation
  - MT Decompress
  - Dipole Trap Load
  - Dipole Trap Evaporation
  - Imaging - Central
  - Analysis
- Parameter Table (Group name: MOT - K, Globals file: C:\labscript\_suite\Experiments\krb\globals.h5):**

Delete	Name	Value	Units	Expansion
<input checked="" type="checkbox"/>	k_central_MOT_load_time	15	s	
<input checked="" type="checkbox"/>	k_offset_frequency	0	MHz	
<input checked="" type="checkbox"/>	k_push_amplitude	212	Arb.	
<input checked="" type="checkbox"/>	k_push_frequency	96.7+k_offset_frequency+ linspace(-10,10,15)	MHz	<input checked="" type="checkbox"/> outer
<input checked="" type="checkbox"/>	k_push_repump_amplitude	537	MHz	
<input checked="" type="checkbox"/>	k_push_repump_frequency	103.3+k_offset_frequency/2	MHz	
<input checked="" type="checkbox"/>	k_repump_load_amplitude	700	Arb.	
<input checked="" type="checkbox"/>	k_repump_load_frequency	100.3+k_offset_frequency/2 + linspace(-10,10,15)/2	MHz	<input checked="" type="checkbox"/> k_MOT_freq
<input checked="" type="checkbox"/>	k_source_current_bottom	4.611	A	
<input checked="" type="checkbox"/>	k_source_current_east	array([ 95.3, 96.01428571, 96.72857143, 97.44285714, 98.15714286, 98.87142857, 99.58571429, 100.3, 101.01428571, 101.72857143, 102.44285714, 103.15714286, 103.87142857, 104.58571429, 105.3 ])	A	
<input checked="" type="checkbox"/>	k_source_current_top	5.556	A	
<input checked="" type="checkbox"/>	k_source_current_west	5.389	A	
<input checked="" type="checkbox"/>	k_trap_load_amplitude	700	Arb.	
<input checked="" type="checkbox"/>	k_trap_load_frequency	81+k_offset_frequency+ linspace(-10,10,15)	MHz	<input checked="" type="checkbox"/> k_MOT_freq
<input checked="" type="checkbox"/>	use_k	s01_use_k	Bool	
	<Click to add global>			

Fig. 1.1: The runmanager interface, used to generate experiment shot files (1) using the experiment logic defined using the labscript API in a Python file (2) and groups of parameters (3), for example those shown for the ‘MOT - K’ group (4). Further details on the runmanager interface can be found in *Usage*.

Runmanager is the primary means of defining and managing the set of experiment parameters (global variables - see §5.1.3) used in the labscript experiment logic. Runmanager also handles the creation of each hdf5 shot file, and the invocation of labscript via the execution of a user specified labscript experiment logic file.

## 2.1 The graphical interface

We believe that the manipulation of parameters, along with controls for producing shots, are best implemented in a graphical interface. Critical information on the current runmanager configuration, along with controls for generating new shots, are located in a always visible toolbar at the top of the runmanager interface. These comprise (as labelled in Fig. 2.1):

1. The engage button: This begins production of the appropriate number of shot files. The number of shot files that will be produced is displayed prominently in the button text so that any mistakes made when defining the parameter space scan can be quickly corrected prior to beginning shot generation. This button can also be 'clicked' via the F5 key on a keyboard.
2. The abort button: This stops the production of shot files prematurely.
3. The restart subprocess button: Primarily for debugging and for use during labscript development, this button restarts the subprocess that manages the execution of the labscript experiment logic file, which in turn generates and stores hardware instructions inside the hdf5 file (see *Shot creation*).
4. The shuffle checkbox: This checkbox controls the global setting for whether parameter space scans are shuffled or not. This is a tri-state checkbox (all-some-none) displaying the current shuffle state on the axes tab. Clicking the checkbox will overwrite the state of each entry on the axes tab with the new state of the global checkbox. For more details, see *Parameter space scans*.
5. The run shots checkbox: If ticked prior to clicking the engage button, shot files will be sent immediately to the BLACS queue once the hardware instructions have been generated by labscript.
6. The view shots checkbox: If ticked prior to clicking the engage button, shots will be sent to runviewer once the hardware instructions have been generated by labscript. Runviewer is assumed to be running locally, and will be launched if it is not already running once the first hdf5 file has been generated.
7. The labscript file: The Python file containing the experiment logic to be compiled into hardware instructions (see *labscript*).
8. The shot output folder: The location to store the hdf5 shot files. By default, the location is specified by the combination of a value in the laboratory PC configuration file (see *labconfig*), the name of the experiment logic Python file and the current date. The location automatically updates, at midnight, to a new folder for the day provided the folder location is left as the default.

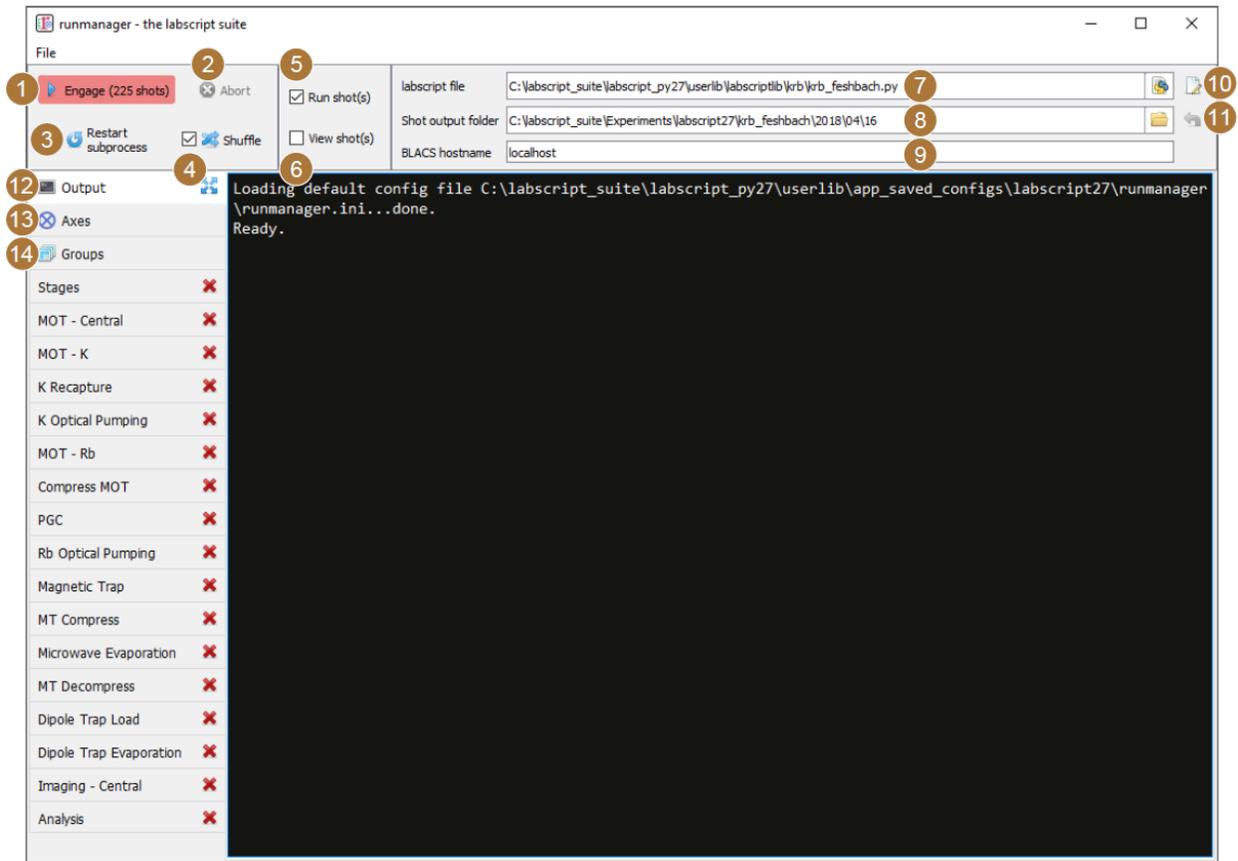


Fig. 2.1: The runmanager graphical interface with the main controls labelled as per the below text. The 'output' tab is currently shown.

9. The BLACS hostname: The network hostname of the PC the hdf5 shot files are to be sent to if the ‘run shots’ checkbox is ticked. It is expected that BLACS is running on the specified PC, and that network access (including firewalls and other network access controls) is configured appropriately.
10. The open in editor button: This button open the specified labscript experiment logic file in the text editor specified in the laboratory PC configuration file (see labconfig in the glossary).
11. The reset shot output folder button: This button resets the shot output folder to the default. This will re-enable the auto incrementation of the folder (based on the current date), which is disabled for custom locations.

These controls provide rapid access to the key functionality of runmanager (creating and distributing shot files) at all times, making for an efficient workflow. The rest of the runmanager interface exists within a set of tabs. The first 3 tabs contain further runmanager specific controls:

12. The output tab: This tab contains the terminal output of the shot creation process including the terminal output produced during the execution of the labscript experiment logic file. For example, Python `print` statements included in the experiment logic code will appear here during shot creation. This makes it easy to debug the experiment logic code using simple methods common to general purpose programming. Warnings and error messages generated by the labscript API also appear here in red text, so that any issues are immediately noticed and can be actioned. As this output is useful for debugging purposes, we allow the tab to be ‘popped out’ into a separate window so it can be visible at the same time as another tab (to avoid the need to frequently switch between the output and the tab containing the global variable(s) you are currently modifying).
13. The axes tab: This tab allows the user to control the iteration order of the parameters in the defined parameter space (see Fig. 2.2). The length of each axis of the parameter space is displayed, as is a shuffle checkbox for determining whether the points along that axis should be shuffled before the parameter space is expanded into the set of shots to be created. The global shuffle control (see item 4 in this list) is linked to the state of the shuffle checkboxes on the axes tab. This feature, along with the many benefits, is detailed further in *Parameter space scans* (see feature 3 and the paragraphs following).
14. The groups tab: This tab manages the hdf5 files that store the globals (see Fig. 2.3). Further details on managing global variables will be discussed in *Managing global variables*.

These tabs are then followed by an arbitrary number of tabs containing sets of global variables, which will be discussed further in *Managing global variables*.

In addition to this, runmanager can save and restore the entire GUI state via the relevant menu items in the ‘File’ menu. This allows rapid switching between different types of experiment logic and/or globals files.<sup>2</sup> This is particularly useful for shared experiment apparatuses, where different users want to run different experiments, and for the cases where a user wishes to rapidly switch between one of more diagnostic configurations they have previously saved.

## 2.2 Managing global variables

Runmanager provides a simple interface for grouping and modifying global variables. As mentioned previously, the ‘groups’ tab in runmanager handles creating and opening the hdf5 files that store the global variables. There are two levels of organisation for global variables:

- at the file level (globals can be stored across multiple files, the union of which is used to generate shots), and
- groups within each file.

Globals groups are created from the ‘groups’ tab in runmanager and can have arbitrary names (including spaces and special symbols). The only requirement is that a group name is unique within its file (it can however have the same name as a group in a different file). Globals within a group are then only used in the creation of shots if the ‘active’ checkbox for the group is checked on the groups tab (see Fig. 2.3). This provides a simple way of switching between

<sup>2</sup> For clarity, the values of the globals are not saved in this configuration file, but simply the location of the hdf5 file containing the globals. This means that any globals in files shared between saved runmanager configurations will share their values. For cases where global values should differ between runmanager configurations, separate globals files should be used.

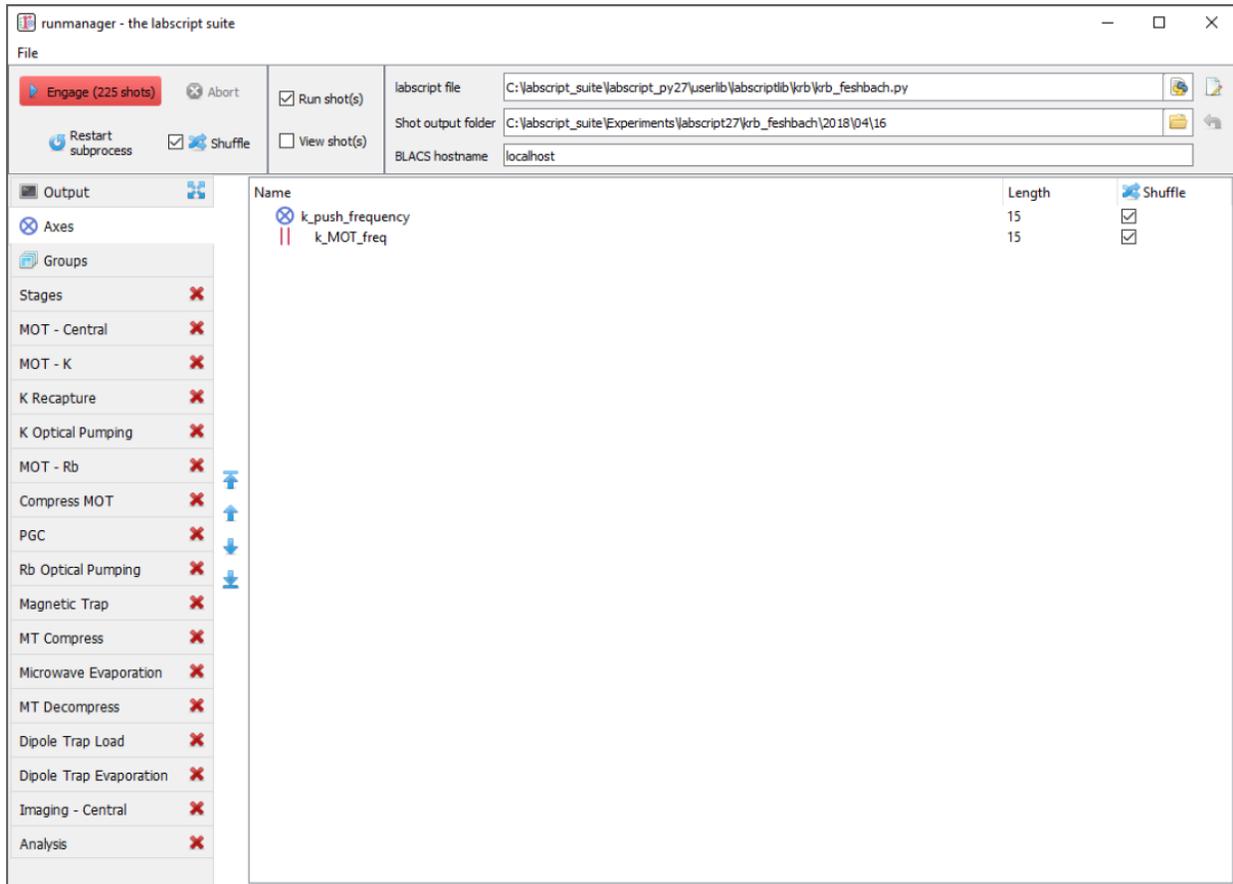


Fig. 2.2: The ‘Axes’ tab of runmanager. This tab displays a list of all global variables (indicated by the blue outer product icon) or groups of global variables (indicated by the icon with red parallel bars) that form axes of the parameter space that will be scanned over (see item 13 and *Parameter space scans* for further details). The order of the axes can be changed using the controls to the left of the list, which sets the order in which the outer product of the axes is performed (when generating the shot files).

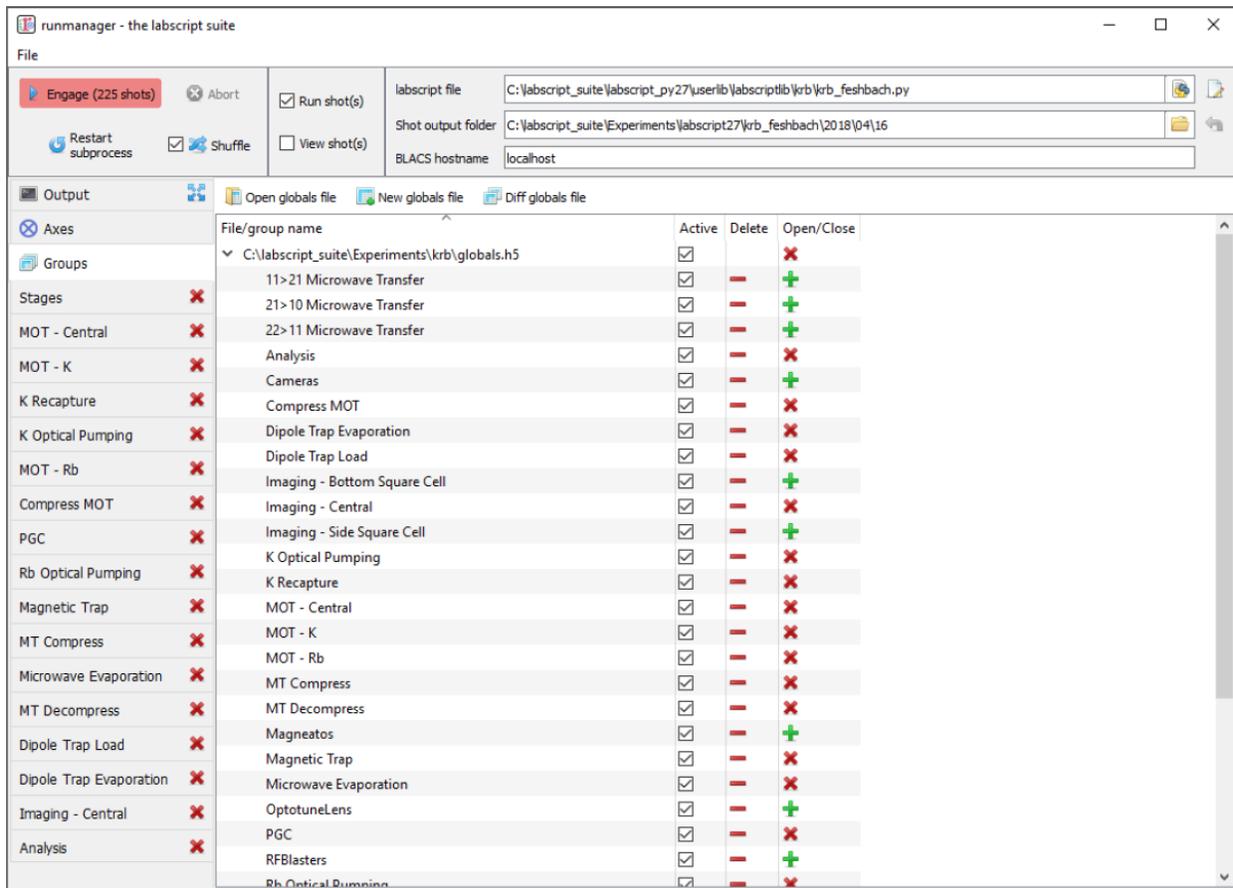


Fig. 2.3: The ‘Groups’ tab of runmanager. This tab displays the groups of global variables (stored in hdf5 files) that have been loaded into runmanager. From this tab, users can enabled/disable the use of these globals when compiling shots (using the ‘active’ checkboxes) and open/close an editing tab for each group. The editing tabs, when open, are displayed as additional tabs on the left most edge of the runmanager interface. See [Managing global variables](#) for further details on managing globals.

different groups of globals, allowing labs to maintain a common set of parameters for their experiments as well as individual parameter sets for specific users and/or experiments. For example, rather than modifying a set of globals in a group, a user could instead deactivate the group containing those globals, and instead ask runmanager to pull those globals from a separate file.

Each group of globals can be opened for editing in a new tab. We provide columns for the global name, value and units. The global name must be a [valid Python variable name](#), and must not conflict with any member of the pylab library, python keywords, or existing items in the Python `__builtin__` module. This ensures that it can be injected into the labscrip experiment logic (see [labscrip](#)) without conflicting with existing Python functionality. The global name must also be unique across all active groups, as global groups are joined into a single set before passing the globals into the labscrip experiment logic.

The value of a global can be any Python expression (including the use of functions from the numpy module), that evaluates to a datatype supported by hdf5, such as, but not limited to:

- a number: 1234 or 780e-9,
- a string: 'N\_atoms',
- a list or numpy array (which will be treated as an axis of a parameter space to scan, where the global variable will contain only one of the elements of the list or array in each shot): [1, 2, 3] or `array([1, 2, 3])`,
- a tuple (which despite being list like, will not be treated as an axis of a parameter space to scan and will instead be passed into labscrip as the tuple specified): (1, 2, 3),
- a Boolean: True or False
- an equation: 1+2
- a Python inbuilt, or numpy, function call that returns a valid value: `linspace(0, 10, 10)`,
- an expression that references another defined global variable by name (the value of this global variable is used in its place): `2*other_global` or `linspace(0, 10, other_global)`,
- a Boolean expression: `(other_global1 and other_global2)` or `(other_global3 == 7)` or `(other_global4)`, or
- any of the above plus a Python comment: `780e-9 #This was previously 781e-9.`

As these expressions can become quite complex (see [Fig. 2.4](#)), the tooltip for the value cells displays the evaluated result of the Python expression. The value cell is also colour coded to the successful evaluation of the expression, so that mistakes can be easily identified (see [Fig. 2.5](#)).

Listing 2.1: Full expression for drop\_time global

```
[ drop_time_rb if x else drop_time_k for x in central_image_order ] if
s11_imaging_both_species else ( drop_time_k if central_image_k == True
else ( drop_time_rb if central_image_rb == True else drop_time_general))
```

The units of the global are not currently passed into the labscrip experiment logic code, but are a way to provide context to the user within runmanager. For example, if the labscrip experiment logic multiplied a global variable for a frequency by 1e6 everywhere it was used (or the keyword argument `units="MHZ"` was used everywhere), then you could type 'MHZ' into the units column of runmanager so that a later user would know that the global was expected to be of that magnitude and would not accidentally enter it in kHz or Hz. In addition to this, globals whose values are explicitly specified as either True or False have their units automatically set to 'Bool', a checkbox is placed in the units column for easy toggling, and the units cell is colour coded to this checkbox for easy observation of the state. We frequently use this functionality to enable/disable various stages of our experiment logic file (see [Fig. 2.6](#)).

While we recommend storing globals in a dedicated set of files, the storage format for the globals is identical to that in any shot, which allows a user to easily load in globals from existing shots (even ones that have been executed and analysed). However, once pointed at an existing shot file, any modification to globals will modify that shot file, thus

The screenshot shows the 'runmanager - the labscript suite' window. The 'File' menu is open, showing options like 'Engage (225 shots)', 'Abort', 'Run shot(s)', 'Restart subprocess', and 'Shuffle'. The 'labscript file' is set to 'C:\labscript\_suite\labscript\_py27\userlib\labscriptlib\krb\krb\_feshbach.py'. The 'Shot output folder' is 'C:\labscript\_suite\Experiments\labscript27\krb\_feshbach\2018\04\16'. The 'BLACS hostname' is 'localhost'. The main area shows the 'Imaging - Central' group with a 'Globals file' at 'C:\labscript\_suite\Experiments\krb\globals.h5'. A table of global variables is displayed:

Delete	Name	Value	Units	Expansion
<input type="checkbox"/>	SG_current	60	A	
<input type="checkbox"/>	SG_imaging	False	Bool	
<input type="checkbox"/>	central_image_k	[not x for x in central_image_order] if s11_imaging_both_species else s11_central_image_k	Bool	
<input type="checkbox"/>	central_image_k_fluoro	s11_central_image_k_fluoro		
<input type="checkbox"/>	central_image_k_recapture	s11_central_image_k_recapture	Bool	
<input type="checkbox"/>	central_image_order	[True, False] if s11_imaging_both_species else 0	Bool	
<input type="checkbox"/>	central_image_rb	central_image_order if s11_imaging_both_species else s11_central_image_rb	Bool	
<input type="checkbox"/>	central_image_rb_fluoro	s11_central_image_order if s11_imaging_both_species else s11_central_image_rb_fluoro		
<input type="checkbox"/>	central_imaging_bias_x	0#0.32	A	
<input type="checkbox"/>	central_imaging_bias_y	1.3	A	
<input type="checkbox"/>	central_imaging_bias_z	0.061	A	
<input type="checkbox"/>	dipole_image_delay	0#drop_time-0.5e-3	s	
<input type="checkbox"/>	drop_time	[drop_time_rb if x else drop_time_k for x in central_image_order] if s11_imaging_both_species else 0	s	
<input type="checkbox"/>	drop_time_general	1e-3#15e-3#0.55e-3	s	
<input type="checkbox"/>	drop_time_k	1e-3	s	
<input type="checkbox"/>	drop_time_rb	15e-3	s	
<input type="checkbox"/>	imaging_pulse_time	.1e-3	s	
<input type="checkbox"/>	interframe_time	200e-3	s	
<input type="checkbox"/>	k_fluoro_imaging_time	5e-3	s	
<input type="checkbox"/>	k_imaging_MOT_repump	True	Bool	
<input type="checkbox"/>	k_imaging_amplitude	850	Arb.	
<input type="checkbox"/>	k_imaging_frequency	92#94.2	MHz	
<input type="checkbox"/>	k_imaging_repump	True	Bool	
<input type="checkbox"/>	k_imaging_repump_amplitude	600	Arb.	

Fig. 2.4: An example of complex global variables that utilise Python expressions to define their value. Note, for example the `drop_time` global variable, whose full expression is shown below. The `drop_time` used is always drawn from one of three global variables, but the global variable selected is determined by a separate global variable (a Boolean) and may contain a list of drop times if the user wishes to image multiple species. In the case of the expression generating a list, this global becomes an axis of a parameter space, running two shots for every other data point in the parameter space (one shot to image each of the two species our experiment supports). Such an expression could not be defined within experiment logic as parameter spaces must be defined within runmanager, not labscript. In order to simplify the view of globals with complex expressions, the tooltip (shown for the `central_image_rb` global) shows the value(s) the global will take in the next compiled shot(s).

The screenshot shows the 'runmanager - the labscript suite' window. The main area displays a configuration table for the 'Imaging - Central' group. The table has columns for 'Delete', 'Name', 'Value', 'Units', and 'Expansion'. The 'drop\_time\_k' row is highlighted in red, and a tooltip shows a 'SyntaxError: 1e-3'.

Delete	Name	Value	Units	Expansion
<input type="checkbox"/>	SG_current	60	A	
<input type="checkbox"/>	SG_imaging	False	Bool	
<input type="checkbox"/>	central_image_k	[not x for x in central_image_order] if s11_imaging_both_species else s11_central_image_k	Bool	
<input type="checkbox"/>	central_image_k_fluoro	s11_central_image_k_fluoro		
<input type="checkbox"/>	central_image_k_recapture	s11_central_image_k_recapture	Bool	
<input type="checkbox"/>	central_image_order	[True, False] if s11_imaging_both_species else 0	Bool	
<input type="checkbox"/>	central_image_rb	central_image_order if s11_imaging_both_species else s11_central_image_rb	Bool	
<input type="checkbox"/>	central_image_rb_fluoro	s11_central_image_rb_fluoro		
<input type="checkbox"/>	central_imaging_bias_x	0#0.32	A	
<input type="checkbox"/>	central_imaging_bias_y	1.3	A	
<input type="checkbox"/>	central_imaging_bias_z	0.061	A	
<input type="checkbox"/>	dipole_image_delay	0#drop_time-0.5e-3	s	
<input type="checkbox"/>	drop_time	[drop_time_rb if x else drop_time_k for x in central_image_order] if s11_imaging_both_s...	s	
<input type="checkbox"/>	drop_time_general	1e-3#15e-3#0.55e-3	s	
<input type="checkbox"/>	drop_time_k	1e-3	s	
<input type="checkbox"/>	drop_time_rb	15e-3	s	
<input type="checkbox"/>	imaging_pulse_time	.1e-3	s	
<input type="checkbox"/>	interframe_time	200e-3	s	
<input type="checkbox"/>	k_fluoro_imaging_time	5e-3	s	
<input type="checkbox"/>	k_imaging_MOT_repump	True	Bool	
<input type="checkbox"/>	k_imaging_amplitude	850	Arb.	
<input type="checkbox"/>	k_imaging_frequency	92#94.2	MHz	
<input type="checkbox"/>	k_imaging_repump	True	Bool	
<input type="checkbox"/>	k_imaging_repump_amplitude	600	Arb	

Fig. 2.5: An example of an evaluation error in a global variable. The user is notified of the error in two places: an icon appears next to the tab name and the global in question is highlighted in red. The tooltip displays the cause of the error, in this case a Python syntax error.

partially destroying the complete record of the experiment.<sup>3</sup> Thus, we encourage this feature to only be used for the cases where you wish to look at the globals from an old shot or where you wish to use the globals, without modification, to compile new shots.

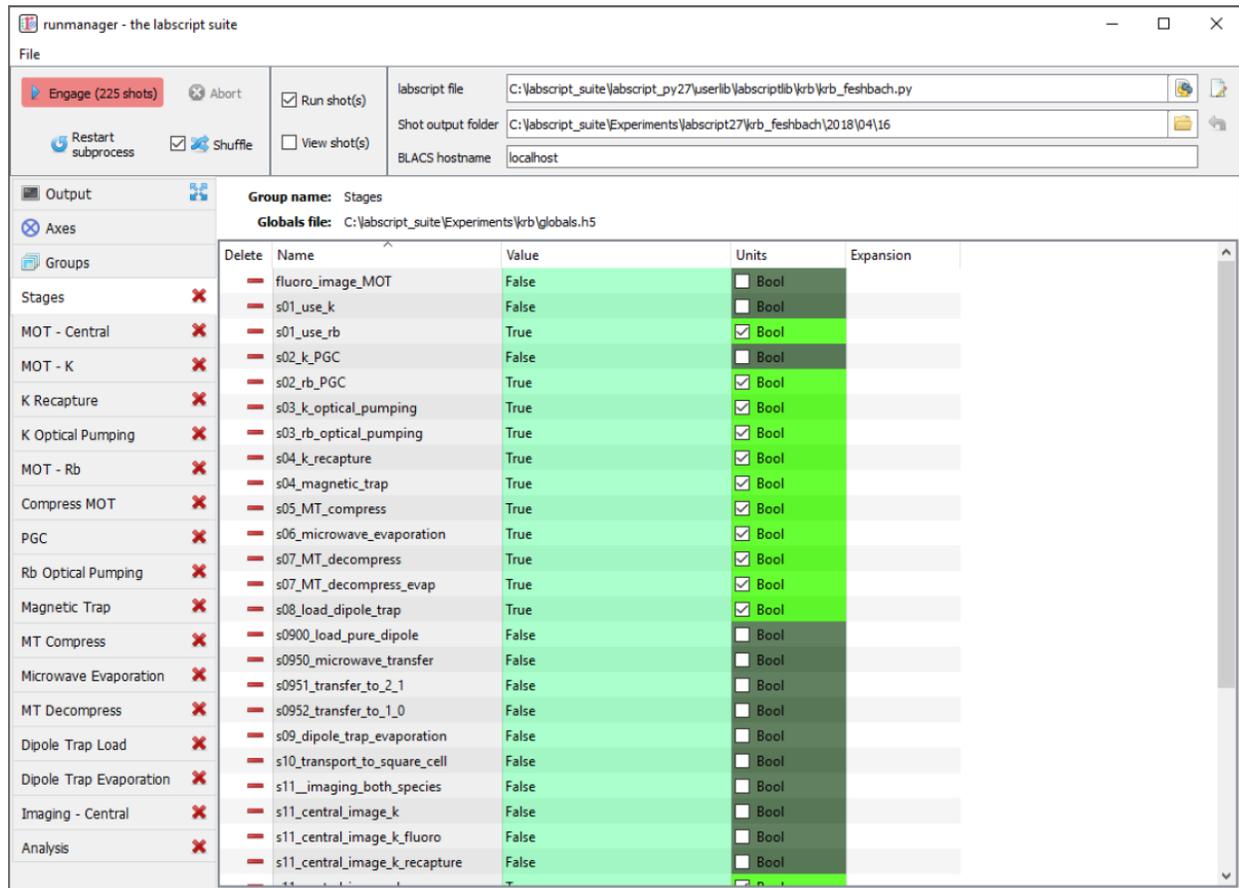


Fig. 2.6: An example of how a labscrip experiment can be parameterised by a series of Boolean global variables. Here we split up the production of a BEC into several stages. We name each global with a prefix that increments in order to keep the globals in an appropriate sort order. Runmanager detects the Boolean type of the global, and provides a simple checkbox toggle in the units column. By using these global variables in our labscrip experiment logic file, as the Boolean expression for an if statement, we can quickly turn on/off various stages of the BEC production process (which is very useful when debugging or optimising the BEC production process).

## 2.3 Parameter space scans

One of the key features of runmanager (and critical goals of our scientific control system) is the ability to easily automate the traversal of a large parameter space, an increasingly important requirement for performing modern ultracold atom experiments. Runmanager provides four features for managing parameter space scans:

1. The automatic detection of global variables that are defined as a list.<sup>4</sup> Such globals are labelled ‘outer’ in the expansions column as all such globals will be combined, via an outer product, into the parameter space to be

<sup>3</sup> Note that in an executed shot file, globals exist in two formats: the evaluated format (one point in the parameter space) used by labscrip, and the raw strings as displayed in runmanager. Only the latter would be overwritten if globals were edited in the manner described in the main body text.

<sup>4</sup> Runmanager considers both Python lists and numpy arrays to be what we refer to as ‘lists’ in this section.

scanned. The number of shots to be generated, which is simply the product of the lengths of all ‘outer’ product globals, is displayed next to the engage button.

2. The ability to define what we term ‘zip groups’ after the Python function `zip`. Two (or more) globals (specified as lists) can be grouped together so that they iterate through values in lockstep. In this instance, the zip group is used as a single axis of the outer product rather than one axis for each global.
3. The third feature is the ability to define the order in which the axes of the parameter space are iterated over when producing individual shots (see the ‘Axes’ tab discussed previously in *The graphical interface*).
4. The ability to randomly shuffle the order of values within each global (or zip group) defined as a list. This can be done on a per global basis or on the entire set of shots that spans the defined parameter space.

These features provide a powerful basis for performing complex experiments.

Consider the following example. Many of the early stages of BEC production (for instance the MOT or magnetic trap stages) should be optimised for best phase-space density. Phase space density is calculated from several parameters; the most important being atom number and atom cloud temperature. While atom number can be easily measured from an absorption image from a single shot, temperature is most commonly determined from analysing the result of multiple shots. In this case, the drop time (the time between releasing the atoms from the trap and taking the absorption image) is varied for each shot and the temperature determined by fitting to the linearised relationship between atom cloud size and drop time. Already, it can be seen that measuring the phase space density for a single set of parameters requires several shots, which can be easily automated via the feature 1 described above.

Now consider the optimisation of MOT or magnetic trap parameters. Many of these are coupled and can not be independently optimised. As such, it is preferable to optimise two or three variables at once, measuring the phase-space density at each point to determine the optimal set of parameters. Such a parameter space typically takes several hours to complete due to the large number of shots that must be run. A BEC apparatus is likely to undergo systematic drifts during this time, which may invalidate the results. However, with careful thought, features 3 and 4 can be used to counteract this. For example, systematic drift will effect the linearity of the data when determining temperature, especially if the acquisition of each data point is separated by a significant period of time. However, by defining the drop time to be the inner most item of the outer product, you ensure that all shots needed to determine the phase-space density for a single set of MOT parameters are executed as close together in time as possible. Shuffling the order of the drop time then eliminates short term systematic drift, as does separately shuffling the order of the values in each remaining axes of the outer product (the MOT parameters). If long term systematic drifts need to be quantified, then an additional axes to the outer product can be added at the outer most layer in order to repeat each of the shots a prescribed number of times (by defining an additional ‘dummy’ global variable as `range(N)` where `N` is the number of times to repeat each shot).

While the above example may seem complicated, runmanager makes it trivial to implement. A user simply defines the list of values to scan over for each parameter, sets the order in which the outer product should use each axis, and specifies whether the values for each axis should be shuffled. Once done, clicking the engage button generates the sequence of shots and sends them to BLACS to be executed on the experiment.

## 2.4 Evaluation of globals

All global variable expressions are automatically evaluated after a change to any global variable. This serves to both update the tooltip with the result of the expression, detect axes of a parameter space to scan (and group them into zip groups if appropriate) and warn the user of any errors during the evaluation of the globals. As discussed previously, runmanager allows these global variable expressions to reference other global variables. This allows a user to maintain a record of a set of parameters, and all relevant quantities derived from one or more of those parameters, without ever storing a parameter more than once. This ensures that important quantities need not be derived (from globals) in the labscript experiment logic script, and that they are accessible directly during the analysis stage (see *lyse*).

To implement this, we take advantage of the Python built-in function `exec` which not only evaluates a string containing a Python expression, but can do so from within a controlled namespace. This has a two-fold benefit. The first is that

it allows us to provide access to a specific set of functions that can be used from within the Python expressions (such as numpy functions like `linspace`). The second is that it allows us to keep track of the relationship between global variables, which is critical for both descriptive error messages and automatically detecting which globals should be combined into a zip groups.

The Python `exec` function is given access to a namespace to work in via an optional argument in the form of a dictionary. Keys and values in this dictionary correspond to variable names in the namespace and their associated values respectively. Rather than using a native Python dictionary for the namespace, we subclass the Python dictionary and override the built-in dictionary method for looking up entries in the dictionary. When combined with `exec`, this translates to our `dict` subclasses tracking each time the `exec` function requests the value of a variable in the namespace. This then provides us with a mapping of each global variable, and the names of global variables that it depends on. In order to resolve both the order in which global variable expressions are evaluated in, and detection of any recursive relationships, we begin by evaluating all global expressions and then recursively re-evaluate the set of globals that did not evaluate in the previous iteration. The first iteration will evaluate any (correctly defined) independent globals, and subsequent iterations will then be able to evaluate globals that depend on other globals (once those other globals have been evaluated by a previous iteration).

The hierarchy of global interdependencies is then used to determine automatic zip group names, which are based on the name of the global in the hierarchy that does not depend on any other. If a global depends on multiple other globals, then the zip group name is chosen semi-randomly based on the order of the items in the Python dictionary (which depends on a hash of the dictionary key names and the size of the dictionary). However, it is of course always possible to overwrite the automatic zip group name with something else should our algorithm choose incorrectly.

We believe that this complex evaluation of global variables is only possible due to the use of an interpreted language that has tools for parsing its own syntax. As such, the choice of Python as our programming language has allowed us to implement extremely useful, advanced features that might otherwise be too difficult to produce in more low level languages such as C++.

## 2.5 Shot creation

The internal process for generating shot files is quite complex. This is primarily motivated by the desire for modularity (for example, to separate shot file generation from hardware instruction generation) and the desire for robustness. As runmanager ultimately initiates the execution of user code (the labscript experiment logic file), there is a risk that problems in the user code could crash runmanager. We mitigate this by using a multi-process architecture.

We originally spawned a new Python process for each shot (in order to guarantee the internal state of labscript was fresh). However the time required to start a Python process (especially on Windows) was a considerable fraction of the entire shot generation time. As such we now use a single, long-lived, Python process and clean-up the internal state of labscript and Python explicitly after each shot.

To generate shot files, runmanager:

1. Re-evaluates all globals (see *Evaluation of globals*). This both determines the number of shots to produce, and generates the evaluated set of global variables for each shot.
2. The globals are then written to hdf5 files, one file for each shot. We also write the unevaluated globals into every hdf5 file, in order to provide a complete record of the experiment (the unevaluated globals contain information about the parameter space that is not available when looking at the single point of parameter space in the evaluated globals of a single shot file).
3. In a thread (in order to keep the GUI responsive), we iterate over the set of files and send their file paths to a long-running subprocess (launched by runmanager at startup) that is used to execute labscript code in an isolated environment. We call this process the ‘compilation subprocess’.
4. The subprocess, which has the labscript API imported, calls an initialisation method to inform the labscript API of the hdf5 file to write hardware instructions to.

5. The subprocess loads the global variables from runmanager into the `__builtin__` dictionary.
6. The subprocess then executes the labscript experiment logic file (using the Python function `exec`) in an isolated namespace, which invokes the labscript API via the users experiment logic and generates the required hardware instructions and saves them in the hdf5 file. Terminal output (for example, `print` statements) are sent back to the parent runmanager process and placed in the output tab.
7. The subprocess restores the `__builtin__` dictionary to its original state to prevent globals from polluting subsequent shots. A clean-up method from the labscript API is also called so that the internal state of the labscript Python module is also reset.

Once shot files are created, the file paths are sent to runviewer or BLACS, as determined by the checkboxes in the runmanager GUI, for viewing and/or executing the shots respectively. This architecture also has several unrealised benefits:

1. If the need arose, we could easily parallelise the generation of hardware instructions by instantiating multiple instances of the compilation subprocess.
2. We could use runmanager as a generic parameter (space) management software by replacing the compilation subprocess with something else. For example, runmanager could be used to manage parameters for simulations, producing one shot file per simulation to be run in the same way we do for real experiments. These files could then be sent to a scheduling program (like BLACS) that feeds them to the simulation software.

## API REFERENCE

<code>runmanager</code>	
<code>runmanager.functions</code>	
<code>runmanager.remote</code>	
<code>runmanager.batch_compiler</code>	
<code>runmanager.globals_diff</code>	Script that runs <code>runmanager.globals_diff_shots()</code> between two shot files.
<code>runmanager.__main__</code>	Runmanager GUI and supporting code

### 3.1 runmanager

#### Functions

<code>add_expansion_groups(filename)</code>	backward compatability, for globals files which don't have expansion groups.
<code>compile_labscript(labscript_file, run_file)</code>	Compiles labscript_file with the run file, returning the processes return code, stdout and stderr.
<code>compile_labscript_async(labscript_file, ...)</code>	Compiles labscript_file with run_file.
<code>compile_labscript_with_globals_files(...)</code>	Creates a run file output_path, using all the globals from globals_files.
<code>compile_labscript_with_globals_files_async(..)</code>	Same as compile_labscript_with_globals_files, except it launches a thread to do
<code>compile_multishot_async(labscript_file, ...)</code>	Compiles labscript_file with run_files.
<code>copy_group(source_globals_file, ..., ...)</code>	This function copies the group source_groupname from source_globals_file to dest_globals_file and renames the new group so that there is no name collision.
<code>delete_global(filename, groupname, globalname)</code>	
<code>delete_group(filename, groupname)</code>	
<code>dict_diff(dict1, dict2)</code>	Return the difference between two dictionaries as a dictionary of key: [val1, val2] pairs.
<code>evaluate_globals(sequence_globals[, ...])</code>	Takes a dictionary of globals as returned by get_globals.

continues on next page

Table 3.1 – continued from previous page

<code>expand_globals(sequence_globals, evaled_globals)</code>	Expands iterable globals according to their expansion settings.
<code>find_comments(src)</code>	Return a list of start and end indices for where comments are in given Python source.
<code>flatten_globals(sequence_globals[, evaluated])</code>	Flattens the data structure of the globals.
<code>get_all_groups(h5_files)</code>	returns a dictionary of group_name: h5_path pairs from a list of h5_files.
<code>get_expansion(filename, groupname, globalname)</code>	
<code>get_globals(groups)</code>	Takes a dictionary of group_name: h5_file pairs and pulls the globals out of the groups in their files.
<code>get_globalslist(filename, groupname)</code>	
<code>get_grouplist(filename)</code>	
<code>get_shot_globals(filepath)</code>	Returns the evaluated globals for a shot, for use by lab-script or lyse.
<code>get_units(filename, groupname, globalname)</code>	
<code>get_value(filename, groupname, globalname)</code>	
<code>globals_diff_groups(active_groups, other_groups)</code>	Given two sets of globals groups, perform a diff of the raw and evaluated globals.
<code>globals_diff_shots(file1, file2[, max_cols])</code>	
<code>guess_expansion_type(value)</code>	
<code>is_valid_hdf5_group_name(name)</code>	Ensure that a string is a valid name for an hdf5 group.
<code>is_valid_python_identifier(name)</code>	
<code>iterator_to_tuple(iterator[, max_length])</code>	
<code>make_run_file_from_globals_files(...[, config])</code>	Creates a run file output_path, using all the globals from globals_files.
<code>make_run_files(output_folder, ...[, shuffle])</code>	Does what it says.
<code>make_single_run_file(filename, ...)</code>	Does what it says.
<code>new_global(filename, groupname, globalname)</code>	
<code>new_globals_file(filename)</code>	
<code>new_group(filename, groupname)</code>	
<code>new_sequence_details(script_path[, config, ...])</code>	Generate the details for a new sequence: the toplevel attrs sequence_date, sequence_index, sequence_id; and the the output directory and filename prefix for the shot files, according to labconfig settings.
<code>next_sequence_index(shot_basedir, dt[, ...])</code>	Return the next sequence index for sequences in the given base directory (i.e. <experiment_shot_storage>/<script_basename>) and the date of the given datetime object, and increment the sequence index atomically on disk if increment=True.

continues on next page

Table 3.1 – continued from previous page

<code>remove_comments_and_tokenify(src)</code>	Removes comments from source code, leaving it otherwise intact, and returns it.
<code>rename_global(filename, groupname, ...)</code>	
<code>rename_group(filename, oldgroupname, ...)</code>	
<code>set_expansion(filename, groupname, ...)</code>	
<code>set_units(filename, groupname, globalname, units)</code>	
<code>set_value(filename, groupname, globalname, value)</code>	

### 3.1.1 runmanager.add\_expansion\_groups

`runmanager.add_expansion_groups(filename)`

backward compatability, for globals files which don't have expansion groups. Create them if they don't exist. Guess expansion settings based on datatypes, if possible.

### 3.1.2 runmanager.compile\_labscript

`runmanager.compile_labscript(labscript_file, run_file)`

Compiles labscript\_file with the run file, returning the processes return code, stdout and stderr.

### 3.1.3 runmanager.compile\_labscript\_async

`runmanager.compile_labscript_async(labscript_file, run_file, stream_port, done_callback)`

Compiles labscript\_file with run\_file. This function is designed to be called in a thread. The stdout and stderr from the compilation will be shovelled into stream\_port via zmq push as it spews forth, and when compilation is complete, done\_callback will be called with a boolean argument indicating success. Note that the zmq communication will be encrypted, or not, according to security settings in labconfig. If you want to receive the data on a zmq socket, do so using a PULL socket created from a labscript\_utils.ls\_zprocess.Context, or using a labscript\_utils.ls\_zprocess.ZMQServer. These subclasses will also be configured with the appropriate security settings and will be able to receive the messages.

### 3.1.4 runmanager.compile\_labscript\_with\_globals\_files

`runmanager.compile_labscript_with_globals_files(labscript_file, globals_files, output_path)`

Creates a run file output\_path, using all the globals from globals\_files. Compiles labscript\_file with the run file, returning the processes return code, stdout and stderr.

### 3.1.5 runmanager.compile\_labscript\_with\_globals\_files\_async

`runmanager.compile_labscript_with_globals_files_async(labscript_file, globals_files, output_path, stream_port, done_callback)`

**Same as `compile_labscript_with_globals_files`, except it launches a thread to do**

the work and does not return anything. Instead, `stderr` and `stdout` will be put to `stream_port` via `zmq` push in the multipart message format `['stdout', 'hello, world`

`']`

etc. When compilation is finished, the function `done_callback` will be called a boolean argument indicating success or failure. If you want to receive the data on a `zmq` socket, do so using a `PULL` socket created from a `labscript_utils.ls_zprocess.Context`, or using a `labscript_utils.ls_zprocess.ZMQServer`. These subclasses will also be configured with the appropriate security settings and will be able to receive the messages.

### 3.1.6 runmanager.compile\_multishot\_async

`runmanager.compile_multishot_async(labscript_file, run_files, stream_port, done_callback)`

Compiles `labscript_file` with `run_files`. This function is designed to be called in a thread. The `stdout` and `stderr` from the compilation will be shovelled into `stream_port` via `zmq` push as it spews forth, and when each compilation is complete, `done_callback` will be called with a boolean argument indicating success. Compilation will stop after the first failure. If you want to receive the data on a `zmq` socket, do so using a `PULL` socket created from a `labscript_utils.ls_zprocess.Context`, or using a `labscript_utils.ls_zprocess.ZMQServer`. These subclasses will also be configured with the appropriate security settings and will be able to receive the messages.

### 3.1.7 runmanager.copy\_group

`runmanager.copy_group(source_globals_file, source_groupname, dest_globals_file, delete_source_group=False)`

This function copies the group `source_groupname` from `source_globals_file` to `dest_globals_file` and renames the new group so that there is no name collision. If `delete_source_group` is `False` the copied files have a suffix `'_copy'`.

### 3.1.8 runmanager.delete\_global

`runmanager.delete_global(filename, groupname, globalname)`

### 3.1.9 runmanager.delete\_group

`runmanager.delete_group(filename, groupname)`

### 3.1.10 runmanager.dict\_diff

`runmanager.dict_diff(dict1, dict2)`

Return the difference between two dictionaries as a dictionary of key: [val1, val2] pairs. Keys unique to either dictionary are included as key: [val1, '-'] or key: ['- ', val2].

### 3.1.11 runmanager.evaluate\_globals

`runmanager.evaluate_globals(sequence_globals, raise_exceptions=True)`

Takes a dictionary of globals as returned by `get_globals`. These globals are unevaluated strings. Evaluates them all in the same namespace so that the expressions can refer to each other. Iterates to allow for `NameErrors` to be resolved by subsequently defined globals. Throws an exception if this does not result in all errors going away. The exception contains the messages of all exceptions which failed to be resolved. If `raise_exceptions` is `False`, any evaluations resulting in an exception will instead return the exception object in the results dictionary

### 3.1.12 runmanager.expand\_globals

`runmanager.expand_globals(sequence_globals, evald_globals, expansion_config=None, return_dimensions=False)`

Expands iterable globals according to their expansion settings. Creates a number of 'axes' which are to be outer product'ed together. Some of these axes have only one element, these are globals that do not vary. Some have a set of globals being zipped together, iterating in lock-step. Others contain a single global varying across its values (the globals set to 'outer' expansion). Returns a list of shots, each element of which is a dictionary for that shot's globals.

### 3.1.13 runmanager.find\_comments

`runmanager.find_comments(src)`

Return a list of start and end indices for where comments are in given Python source. Comments on separate lines with only whitespace in between them are coalesced. Whitespace preceding a comment is counted as part of the comment.

### 3.1.14 runmanager.flatten\_globals

`runmanager.flatten_globals(sequence_globals, evaluated=False)`

Flattens the data structure of the globals. If `evaluated=False`, saves only the value expression string of the global, not the units or expansion.

### 3.1.15 runmanager.get\_all\_groups

`runmanager.get_all_groups(h5_files)`

returns a dictionary of group\_name: h5\_path pairs from a list of h5\_files.

### 3.1.16 runmanager.get\_expansion

`runmanager.get_expansion(filename, groupname, globalname)`

### 3.1.17 runmanager.get\_globals

`runmanager.get_globals(groups)`

Takes a dictionary of group\_name: h5\_file pairs and pulls the globals out of the groups in their files. The globals are strings storing python expressions at this point. All these globals are packed into a new dictionary, keyed by group\_name, where the values are dictionaries which look like {global\_name: (expression, units, expansion), ... }

### 3.1.18 runmanager.get\_globalslist

`runmanager.get_globalslist(filename, groupname)`

### 3.1.19 runmanager.get\_grouplist

`runmanager.get_grouplist(filename)`

### 3.1.20 runmanager.get\_shot\_globals

`runmanager.get_shot_globals(filepath)`

Returns the evaluated globals for a shot, for use by labscrip or lyse. Simple dictionary access as in `dict(h5py.File(filepath).attrs)` would be fine except we want to apply some hacks, so it's best to do that in one place.

Deprecated: use identical function `labscrip_utils.shot_utils.get_shot_globals`

### 3.1.21 runmanager.get\_units

`runmanager.get_units(filename, groupname, globalname)`

### 3.1.22 runmanager.get\_value

`runmanager.get_value(filename, groupname, globalname)`

### 3.1.23 runmanager.globals\_diff\_groups

`runmanager.globals_diff_groups(active_groups, other_groups, max_cols=1000, return_string=True)`

Given two sets of globals groups, perform a diff of the raw and evaluated globals.

### 3.1.24 runmanager.globals\_diff\_shots

`runmanager.globals_diff_shots(file1, file2, max_cols=100)`

### 3.1.25 runmanager.guess\_expansion\_type

`runmanager.guess_expansion_type(value)`

### 3.1.26 runmanager.is\_valid\_hdf5\_group\_name

`runmanager.is_valid_hdf5_group_name(name)`

Ensure that a string is a valid name for an hdf5 group.

The names of hdf5 groups may only contain ASCII characters. Furthermore, the characters “/” and “.” are not allowed.

**Parameters**

**name** (*str*) – The potential name for an hdf5 group.

**Returns**

**Whether or not name is a valid name for an hdf5 group. This will be True if it is a valid name or False otherwise.**

**Return type**

bool

### 3.1.27 runmanager.is\_valid\_python\_identifier

`runmanager.is_valid_python_identifier(name)`

### 3.1.28 runmanager.iterator\_to\_tuple

`runmanager.iterator_to_tuple(iterator, max_length=1000000)`

### 3.1.29 runmanager.make\_run\_file\_from\_globals\_files

`runmanager.make_run_file_from_globals_files(labscript_file, globals_files, output_path, config=None)`

Creates a run file `output_path`, using all the globals from `globals_files`. Uses `labscript_file` to determine the `sequence_attrs` only

### 3.1.30 runmanager.make\_run\_files

`runmanager.make_run_files(output_folder, sequence_globals, shots, sequence_attrs, filename_prefix, shuffle=False)`

Does what it says. `sequence_globals` and `shots` are of the datatypes returned by `get_globals` and `get_shots`, one is a nested dictionary with string values, and the other a flat dictionary. `sequence_attrs` is a dict of the attributes pertaining to this sequence to be initially set at the top-level group of the h5 file, as returned by `new_sequence_details`. `output_folder` and `filename_prefix` determine the directory shot files will be output to, as well as their filenames (this function will generate filenames with the shot number and `.h5` extension appended to `filename_prefix`). Sensible defaults for these are also returned by `new_sequence_details()`, so preferably these should be used.

`Shuffle` will randomise the order that the run files are generated in with respect to which element of `shots` they come from. This function returns a *generator*. The run files are not actually created until you loop over this generator (which gives you the filepaths). This is useful for not having to clean up as many unused files in the event of failed compilation of labscripts. If you want all the run files to be created at some point, simply convert the returned generator to a list. The filenames the run files are given is simply the `sequence_id` with increasing integers appended.

### 3.1.31 runmanager.make\_single\_run\_file

`runmanager.make_single_run_file(filename, sequenceglobals, runglobals, sequence_attrs, run_no, n_runs)`

Does what it says. `runglobals` is a dict of this run's globals, the format being the same as that of one element of the list returned by `expand_globals`. `sequence_globals` is a nested dictionary of the type returned by `get_globals`. `sequence_attrs` is a dict of attributes pertaining to this sequence, as returned by `new_sequence_details`. `run_no` and `n_runs` must be provided, if this run file is part of a sequence, then they should reflect how many run files are being generated in this sequence, all of which must have identical `sequence_attrs`.

### 3.1.32 runmanager.new\_global

`runmanager.new_global(filename, groupname, globalname)`

### 3.1.33 runmanager.new\_globals\_file

`runmanager.new_globals_file(filename)`

### 3.1.34 runmanager.new\_group

`runmanager.new_group(filename, groupname)`

### 3.1.35 runmanager.new\_sequence\_details

`runmanager.new_sequence_details(script_path, config=None, increment_sequence_index=True)`

Generate the details for a new sequence: the toplevel attrs `sequence_date`, `sequence_index`, `sequence_id`; and the the output directory and filename prefix for the shot files, according to labconfig settings. If `increment_sequence_index=True`, then we are claiming the resulting sequence index for use such that it cannot be used by anyone else. This should be done if the sequence details are immediately about to be used to compile a sequence. Otherwise, set `increment_sequence_index` to `False`, but in that case the results are indicative only and one should call this function again with `increment_sequence_index=True` before compiling the sequence, as otherwise the `sequence_index` may be used by other code in the meantime.

### 3.1.36 runmanager.next\_sequence\_index

`runmanager.next_sequence_index(shot_basedir, dt, increment=True)`

Return the next sequence index for sequences in the given base directory (i.e. `<experiment_shot_storage>/<script_basename>`) and the date of the given datetime object, and increment the sequence index atomically on disk if `increment=True`. If not setting `increment=True`, then the result is indicative only and may be used by other code at any time. One must increment the sequence index prior to use.

### 3.1.37 runmanager.remove\_comments\_and\_tokenify

`runmanager.remove_comments_and_tokenify(src)`

Removes comments from source code, leaving it otherwise intact, and returns it. Also returns the raw tokens for the code, allowing comparisons between source to be made without being sensitive to whitespace.

### 3.1.38 runmanager.rename\_global

`runmanager.rename_global(filename, groupname, oldglobalname, newglobalname)`

### 3.1.39 runmanager.rename\_group

`runmanager.rename_group(filename, oldgroupname, newgroupname)`

### 3.1.40 runmanager.set\_expansion

`runmanager.set_expansion(filename, groupname, globalname, expansion)`

### 3.1.41 runmanager.set\_units

`runmanager.set_units(filename, groupname, globalname, units)`

### 3.1.42 runmanager.set\_value

`runmanager.set_value(filename, groupname, globalname, value)`

## Classes

---

```
TraceDictionary(*args, **kwargs)
```

---

### 3.1.43 runmanager.TraceDictionary

```
class runmanager.TraceDictionary(*args, **kwargs)
```

```
    Bases: dict
```

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

## Methods

<code>__init__(*args, **kwargs)</code>	
<code>clear()</code>	
<code>copy()</code>	
<code>fromkeys([value])</code>	Create a new dictionary with keys from iterable and values set to value.
<code>get(key[, default])</code>	Return the value for key if key is in the dictionary, else default.
<code>items()</code>	
<code>keys()</code>	
<code>pop(k[,d])</code>	If the key is not found, return the default if given; otherwise, raise a <code>KeyError</code> .
<code>popitem()</code>	Remove and return a (key, value) pair as a 2-tuple.
<code>setdefault(key[, default])</code>	Insert key with a value of default if key is not in the dictionary.
<code>start_trace()</code>	
<code>stop_trace()</code>	
<code>update([E, ]**F)</code>	If E is present and has a <code>.keys()</code> method, then does: for k in E: D[k] = E[k] If E is present and lacks a <code>.keys()</code> method, then does: for k, v in E: D[k] = v In either case, this is followed by: for k in F: D[k] = F[k]
<code>values()</code>	

`start_trace()`

`stop_trace()`

## Exceptions

<code>ExpansionError</code>	An exception class so that error handling code can tell when a parsing exception was caused by a mismatch with the expansion mode
-----------------------------	---

### 3.1.44 runmanager.ExpansionError

**exception** runmanager.ExpansionError

An exception class so that error handling code can tell when a parsing exception was caused by a mismatch with the expansion mode

## 3.2 runmanager.functions

### Functions

---

*drop\_times*(t\_min, t\_max, n\_points[, ...])

*first*()

Infinite iterator.

*quadspace*(t\_min, t\_max, n\_points[, ...])

---

#### 3.2.1 runmanager.functions.drop\_times

runmanager.functions.**drop\_times**(t\_min, t\_max, n\_points, randomise=False, repeats=1)

#### 3.2.2 runmanager.functions.first

runmanager.functions.**first**()

Infinite iterator. Its first return value is true, subsequent return values are False

#### 3.2.3 runmanager.functions.quadspace

runmanager.functions.**quadspace**(t\_min, t\_max, n\_points, randomise=False, repeats=1)

## 3.3 runmanager.remote

## Functions

<code>abort()</code>	Trigger abort compilation/submission
<code>engage()</code>	Trigger shot compilation/submission
<code>error_in_globals()</code>	True if any tab of an active group contains error(s)
<code>get_globals([raw])</code>	Return all active globals as a dict of the form: { '<global_name>': value }.
<code>get_labscript_file()</code>	Get the path of the current experiment script
<code>get_run_shots()</code>	Get boolean state of 'Run shot(s)' checkbox
<code>get_shot_output_folder()</code>	Get the current shot output folder
<code>get_shuffle()</code>	Get boolean state of 'Shuffle' checkbox
<code>get_version()</code>	Return the version of runmanager the server is running in
<code>get_view_shots()</code>	Get boolean state of 'View shot(s)' checkbox
<code>is_output_folder_default()</code>	True if shot output folder is not the default path
<code>n_shots()</code>	Get the number of prospective shots from pressing 'Engage'
<code>reset_shot_output_folder()</code>	Reset the shot output folder to the default path
<code>say_hello()</code>	Ping the runmanager server for a response
<code>set_globals(globals[, raw])</code>	For a dict of the form { '<global_name>': value }, set the given globals to the given values.
<code>set_labscript_file(value)</code>	Set the current experiment script
<code>set_run_shots(value)</code>	Set boolean state of 'Run shot(s)' checkbox
<code>set_shot_output_folder(value)</code>	Set the shot output folder
<code>set_shuffle(value)</code>	Set boolean state of 'Shuffle' checkbox
<code>set_view_shots(value)</code>	Set boolean state of 'View shot(s)' checkbox

### 3.3.1 runmanager.remote.abort

`runmanager.remote.abort()`

Trigger abort compilation/submission

### 3.3.2 runmanager.remote.engage

`runmanager.remote.engage()`

Trigger shot compilation/submission

### 3.3.3 runmanager.remote.error\_in\_globals

`runmanager.remote.error_in_globals()`

True if any tab of an active group contains error(s)

### 3.3.4 runmanager.remote.get\_globals

`runmanager.remote.get_globals(raw=False)`

Return all active globals as a dict of the form: { '<global\_name>': value }. If `raw=True`, then the global values are returned as their string representations, as stored in the runmanager GUI and globals HDF5 file, otherwise they are evaluated as python objects and then returned.

### 3.3.5 runmanager.remote.get\_labscript\_file

`runmanager.remote.get_labscript_file()`

Get the path of the current experiment script

### 3.3.6 runmanager.remote.get\_run\_shots

`runmanager.remote.get_run_shots()`

Get boolean state of 'Run shot(s)' checkbox

### 3.3.7 runmanager.remote.get\_shot\_output\_folder

`runmanager.remote.get_shot_output_folder()`

Get the current shot output folder

### 3.3.8 runmanager.remote.get\_shuffle

`runmanager.remote.get_shuffle()`

Get boolean state of 'Shuffle' checkbox

### 3.3.9 runmanager.remote.get\_version

`runmanager.remote.get_version()`

Return the version of runmanager the server is running in

### 3.3.10 runmanager.remote.get\_view\_shots

`runmanager.remote.get_view_shots()`

Get boolean state of 'View shot(s)' checkbox

### 3.3.11 runmanager.remote.is\_output\_folder\_default

`runmanager.remote.is_output_folder_default()`

True if shot output folder is not the default path

### 3.3.12 runmanager.remote.n\_shots

`runmanager.remote.n_shots()`

Get the number of prospective shots from pressing 'Engage'

### 3.3.13 runmanager.remote.reset\_shot\_output\_folder

`runmanager.remote.reset_shot_output_folder()`

Reset the shot output folder to the default path

### 3.3.14 runmanager.remote.say\_hello

`runmanager.remote.say_hello()`

Ping the runmanager server for a response

### 3.3.15 runmanager.remote.set\_globals

`runmanager.remote.set_globals(globals, raw=False)`

For a dict of the form {'<global\_name>': value}, set the given globals to the given values. If `raw=True`, then global values will be treated as the string representations of Python objects rather than the objects themselves, and written directly to the HDF5 file and runmanager GUI without calling `repr()` on them first.

### 3.3.16 runmanager.remote.set\_labscript\_file

`runmanager.remote.set_labscript_file(value)`

Set the current experiment script

### 3.3.17 runmanager.remote.set\_run\_shots

`runmanager.remote.set_run_shots(value)`

Set boolean state of 'Run shot(s)' checkbox

### 3.3.18 runmanager.remote.set\_shot\_output\_folder

`runmanager.remote.set_shot_output_folder(value)`

Set the shot output folder

### 3.3.19 runmanager.remote.set\_shuffle

`runmanager.remote.set_shuffle(value)`

Set boolean state of 'Shuffle' checkbox

### 3.3.20 runmanager.remote.set\_view\_shots

`runmanager.remote.set_view_shots(value)`

Set boolean state of 'View shot(s)' checkbox

## Classes

---

*Client*([host, port, timeout])

A ZMQClient for communication with runmanager

---

### 3.3.21 runmanager.remote.Client

**class** `runmanager.remote.Client` (*host=None, port=None, timeout=None*)

Bases: `ZMQClient`

A ZMQClient for communication with runmanager

`__init__` (*host=None, port=None, timeout=None*)

## Methods

<code>__init__([host, port, timeout])</code>	
<code>abort()</code>	Trigger abort compilation/submission
<code>clear_interrupt()</code>	Clear our internal Interruptor object so that future <code>get*()/push*()</code> calls can proceed as normal.
<code>engage()</code>	Trigger shot compilation/submission
<code>error_in_globals()</code>	True if any tab of an active group contains error(s)
<code>get_globals([raw])</code>	Return all active globals as a dict of the form: <code>{'&lt;global_name&gt;': value}</code> .
<code>get_labscript_file()</code>	Get the path of the current experiment script
<code>get_run_shots()</code>	Get boolean state of 'Run shot(s)' checkbox
<code>get_shot_output_folder()</code>	Get the current shot output folder
<code>get_shuffle()</code>	Get boolean state of 'Shuffle' checkbox
<code>get_version()</code>	Return the version of runmanager the server is running in
<code>get_view_shots()</code>	Get boolean state of 'View shot(s)' checkbox
<code>instance()</code>	
<code>interrupt([reason])</code>	Interrupt any current and future <code>get*()/push*()</code> calls, causing them to raise <code>Interrupted(reason)</code> until <code>clear_interrupt()</code> is called.
<code>is_output_folder_default()</code>	True if shot output folder is not the default path
<code>n_shots()</code>	Get the number of prospective shots from pressing 'Engage'
<code>request(command, *args, **kwargs)</code>	
<code>reset_shot_output_folder()</code>	Reset the shot output folder to the default path
<code>say_hello()</code>	Ping the runmanager server for a response
<code>set_globals(globals[, raw])</code>	For a dict of the form <code>{'&lt;global_name&gt;': value}</code> , set the given globals to the given values.
<code>set_labscript_file(value)</code>	Set the current experiment script
<code>set_run_shots(value)</code>	Set boolean state of 'Run shot(s)' checkbox
<code>set_shot_output_folder(value)</code>	Set the shot output folder
<code>set_shuffle(value)</code>	Set boolean state of 'Shuffle' checkbox
<code>set_view_shots(value)</code>	Set boolean state of 'View shot(s)' checkbox

### **abort()**

Trigger abort compilation/submission

### **engage()**

Trigger shot compilation/submission

### **error\_in\_globals()**

True if any tab of an active group contains error(s)

### **get\_globals(*raw=False*)**

Return all active globals as a dict of the form: `{'<global_name>': value}`. If `raw=True`, then the global values are returned as their string representations, as stored in the runmanager GUI and globals HDF5 file, otherwise they are evaluated as python objects and then returned.

### **get\_labscript\_file()**

Get the path of the current experiment script

**get\_run\_shots()**

Get boolean state of 'Run shot(s)' checkbox

**get\_shot\_output\_folder()**

Get the current shot output folder

**get\_shuffle()**

Get boolean state of 'Shuffle' checkbox

**get\_version()**

Return the version of runmanager the server is running in

**get\_view\_shots()**

Get boolean state of 'View shot(s)' checkbox

**is\_output\_folder\_default()**

True if shot output folder is not the default path

**n\_shots()**

Get the number of prospective shots from pressing 'Engage'

**request(*command*, *\*args*, *\*\*kwargs*)**

**reset\_shot\_output\_folder()**

Reset the shot output folder to the default path

**say\_hello()**

Ping the runmanager server for a response

**set\_globals(*globals*, *raw=False*)**

For a dict of the form {'<global\_name>': value}, set the given globals to the given values. If *raw=True*, then global values will be treated as the string representations of Python objects rather than the objects themselves, and written directly to the HDF5 file and runmanager GUI without calling repr() on them first.

**set\_labscript\_file(*value*)**

Set the current experiment script

**set\_run\_shots(*value*)**

Set boolean state of 'Run shot(s)' checkbox

**set\_shot\_output\_folder(*value*)**

Set the shot output folder

**set\_shuffle(*value*)**

Set boolean state of 'Shuffle' checkbox

**set\_view\_shots(*value*)**

Set boolean state of 'View shot(s)' checkbox

## 3.4 runmanager.batch\_compiler

### Classes

```
BatchProcessor(to_parent, from_parent, kill_lock)
```

### 3.4.1 runmanager.batch\_compiler.BatchProcessor

```
class runmanager.batch_compiler.BatchProcessor(to_parent, from_parent, kill_lock)
```

```
Bases: object
```

```
__init__(to_parent, from_parent, kill_lock)
```

#### Methods

```
__init__(to_parent, from_parent, kill_lock)
```

```
compile(labscript_file, run_file)
```

```
mainloop()
```

```
compile(labscript_file, run_file)
```

```
mainloop()
```

## 3.5 runmanager.globals\_diff

Script that runs `runmanager.globals_diff_shots()` between two shot files.

It is run from the command prompt:

```
$ python runmanager.global_diffs(shot1,shot2)
```

## 3.6 runmanager.\_\_main\_\_

Runmanager GUI and supporting code

## Functions

<code>composite_colors(r0, g0, b0, a0, r1, g1, b1, a1)</code>	composite a second colour over a first with given alpha values and return the result
<code>error_dialog(message)</code>	
<code>log_if_global(g, g_list, message)</code>	logs a message if the global name "g" is in "g_list"
<code>nested(*contextmanagers)</code>	
<code>question_dialog(message)</code>	
<code>scroll_view_to_row_if_current(view, item)</code>	Checks to see if the item is in the row of the current item.

### 3.6.1 runmanager.\_\_main\_\_.composite\_colors

`runmanager.__main__.composite_colors(r0, g0, b0, a0, r1, g1, b1, a1)`

composite a second colour over a first with given alpha values and return the result

### 3.6.2 runmanager.\_\_main\_\_.error\_dialog

`runmanager.__main__.error_dialog(message)`

### 3.6.3 runmanager.\_\_main\_\_.log\_if\_global

`runmanager.__main__.log_if_global(g, g_list, message)`

logs a message if the global name "g" is in "g\_list"

useful if you want to print out a message inside a loop over globals, but only for a particular global (or set of globals).

If `g_list` is empty, then it will use the hardcoded list below (useful if you want to change the behaviour globally)

### 3.6.4 runmanager.\_\_main\_\_.nested

`runmanager.__main__.nested(*contextmanagers)`

### 3.6.5 runmanager.\_\_main\_\_.question\_dialog

`runmanager.__main__.question_dialog(message)`

### 3.6.6 runmanager.\_\_main\_\_.scroll\_view\_to\_row\_if\_current

`runmanager.__main__.scroll_view_to_row_if_current(view, item)`

Checks to see if the item is in the row of the current item. If it is, scrolls the treeview/tableview vertically to ensure that row is visible. This is done by recording the horizontal scroll position, then using `view.scrollTo()`, and then restoring the horizontal position

#### Classes

<code>AlternatingColorModel(view)</code>	
<code>Editor(parent)</code>	Popup editor with word wrapping and automatic resizing.
<code>FingerTabBarWidget([parent, minwidth, minheight])</code>	A TabBar with the tabs on the left and the text horizontal.
<code>FingerTabWidget(parent, *args)</code>	A QTabWidget equivalent which uses our FingerTabBarWidget
<code>GroupTab(tabWidget, globals_file, group_name)</code>	
<code>ItemDelegate(*args, **kwargs)</code>	An item delegate with a larger row height and column width, faint grey vertical lines between columns, and a custom editor for handling multi-line data
<code>ItemView(*args)</code>	Mixin for QTableView and QTreeView 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>PoppedOutOutputBoxWindow</code>	
<code>RemoteServer()</code>	
<code>RunManager()</code>	
<code>RunmanagerMainWindow(*args, **kwargs)</code>	
<code>TabToolButton(*args, **kwargs)</code>	
<code>TableView([parent])</code>	TableView version of our customised ItemView
<code>TreeView([parent])</code>	Treewiew version of our customised ItemView

### 3.6.7 runmanager.\_\_main\_\_.AlternatingColorModel

`class runmanager.__main__.AlternatingColorModel(view)`

Bases: `QStandardItemModel`

`__init__(view)`

## Methods

`__init__(view)`  
`addColumn(self, items)`  
`appendRow()`  
`beginInsertColumns(self, parent, first, last)`  
`beginInsertRows(self, parent, first, last)`  
`beginMoveColumns(self, sourceParent, ...)`  
`beginMoveRows(self, sourceParent, ...)`  
`beginRemoveColumns(self, parent, first, last)`  
`beginRemoveRows(self, parent, first, last)`  
`beginResetModel(self)`  
`blockSignals(self, b)`  
`buddy(self, index)`  
`canDropMimeData(self, data, action, row, ...)`  
`canFetchMore(self, parent)`  
`changePersistentIndex(self, from_, to)`  
`changePersistentIndexList(self, from_, to)`  
`checkIndex(self, index[, options])`  
`childEvent(self, a0)`  
`children(self)`  
`clear(self)`  
`clearItemData(self, index)`  
`columnCount(self[, parent])`  
`connectNotify(self, signal)`  
`createIndex(self, row, column[, object])`  
`customEvent(self, a0)`

---

continues on next page

Table 3.2 – continued from previous page

<i>data</i> (index, role)	When background color data is being requested, returns modified colours for every second row, according to the palette of the view.
decodeData(self, row, column, parent, stream)	
deleteLater(self)	
disconnect(-> bool)	
disconnectNotify(self, signal)	
dropMimeData(self, data, action, row, ...)	
dumpObjectInfo(self)	
dumpObjectTree(self)	
dynamicPropertyNames(self)	
encodeData(self, indexes, stream)	
endInsertColumns(self)	
endInsertRows(self)	
endMoveColumns(self)	
endMoveRows(self)	
endRemoveColumns(self)	
endRemoveRows(self)	
endResetModel(self)	
event(self, a0)	
eventFilter(self, a0, a1)	
fetchMore(self, parent)	
findChild(-> QObjectT)	
findChildren(...)	
findItems(self, text[, flags, column])	
flags(self, index)	
<i>get_bgbrush</i> (normal_brush, alternate, selected)	Get cell colour as a function of its ordinary colour, whether it is on an odd row, and whether it is selected.

continues on next page

Table 3.2 – continued from previous page

<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>
<code>inherits(self, classname)</code>
<code>insertColumn()</code>
<code>insertColumns(self, column, count[, parent])</code>
<code>insertRow(-&gt; 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>

continues on next page

Table 3.2 – continued from previous page

<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>	
<code>parent(-&gt; 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>	

continues on next page

Table 3.2 – continued from previous page

<code>setItem()</code>
<code>setItemData(self, index, roles)</code>
<code>setItemPrototype(self, item)</code>
<code>setItemRoleNames(self, roleNames)</code>
<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>

continues on next page

Table 3.2 – continued from previous page

<code>timerEvent(self, a0)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>
<code>verticalHeaderItem(self, row)</code>

## Attributes

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

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

When background color data is being requested, returns modified colours for every second row, according to the palette of the view. This has the effect of making the alternate colours visible even when custom colors have been set - the same shading will be applied to the custom colours. Only really looks sensible when the normal and alternate colors are similar. Also applies selection highlight colour (using `ItemView.COLOR_HIGHLIGHT`), similarly with alternate-row shading, for the case of a `QTableView`.

**get\_bgbrush**(*normal\_brush, alternate, selected*)

Get cell colour as a function of its ordinary colour, whether it is on an odd row, and whether it is selected.

### 3.6.8 runmanager.\_\_main\_\_.Editor

**class** runmanager.\_\_main\_\_.Editor(*parent*)

Bases: `QTextEdit`

Popup editor with word wrapping and automatic resizing.

**\_\_init\_\_**(*parent*)

#### Methods

`__init__(parent)`

`acceptDrops(self)`

`acceptRichText(self)`

`accessibleDescription(self)`

`accessibleName(self)`

`actionEvent(self, a0)`

`actions(self)`

`activateWindow(self)`

`addAction(self, action)`

`addActions(self, actions)`

`addScrollBarWidget(self, widget, alignment)`

`adjustSize(self)`

`alignment(self)`

`anchorAt(self, pos)`

`append(self, text)`

continues on next page

Table 3.3 – continued from previous page

---

<code>autoFillBackground(self)</code>
<code>autoFormatting(self)</code>
<code>backgroundRole(self)</code>
<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>canInsertFromMimeData(self, source)</code>
<code>canPaste(self)</code>
<code>changeEvent(self, e)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clear(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, e)</code>
<code>contextMenuPolicy(self)</code>
<code>copy(self)</code>
<code>cornerWidget(self)</code>

---

continues on next page

Table 3.3 – continued from previous page

<code>create(self[, window, initializeWindow, ...])</code>
<code>createMimeDataFromSelection(self)</code>
<code>createStandardContextMenu(-&gt; Optional[QMenu])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>currentCharFormat(self)</code>
<code>currentFont(self)</code>
<code>cursor(self)</code>
<code>cursorForPosition(self, pos)</code>
<code>cursorRect(-&gt; QRect)</code>
<code>cursorWidth(self)</code>
<code>customEvent(self, a0)</code>
<code>cut(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>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>document(self)</code>
<code>documentTitle(self)</code>
<code>dragEnterEvent(self, e)</code>
<code>dragLeaveEvent(self, e)</code>
<code>dragMoveEvent(self, e)</code>

continues on next page

Table 3.3 – continued from previous page

<code>drawFrame(self, a0)</code>
<code>dropEvent(self, e)</code>
<code>dumpObjectInfo(self)</code>
<code>dumpObjectTree(self)</code>
<code>dynamicPropertyNames(self)</code>
<code>effectiveWinId(self)</code>
<code>ensureCursorVisible(self)</code>
<code>ensurePolished(self)</code>
<code>enterEvent(self, a0)</code>
<code>event(self, e)</code>
<code>eventFilter(self, a0, a1)</code>
<code>extraSelections(self)</code>
<code>find() -&gt; bool ) -&gt; bool ))</code>
<code>findChild(-&gt; QObjectT)</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>fontFamily(self)</code>
<code>fontInfo(self)</code>

continues on next page

Table 3.3 – continued from previous page

<code>fontItalic(self)</code>
<code>fontMetrics(self)</code>
<code>fontPointSize(self)</code>
<code>fontUnderline(self)</code>
<code>fontWeight(self)</code>
<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>hasFocus(self)</code>
<code>hasHeightForWidth(self)</code>
<code>hasMouseTracking(self)</code>
<code>hasTabletTracking(self)</code>

continues on next page

Table 3.3 – continued from previous page

<code>height(self)</code>
<code>heightForWidth(self, a0)</code>
<code>heightMM(self)</code>
<code>hide(self)</code>
<code>hideEvent(self, a0)</code>
<code>horizontalScrollBar(self)</code>
<code>horizontalScrollBarPolicy(self)</code>
<code>inherits(self, classname)</code>
<code>initPainter(self, painter)</code>
<code>initStyleOption(self, option)</code>
<code>inputMethodEvent(self, a0)</code>
<code>inputMethodHints(self)</code>
<code>inputMethodQuery(-&gt; Any)</code>
<code>insertAction(self, before, action)</code>
<code>insertActions(self, before, actions)</code>
<code>insertFromMimeData(self, source)</code>
<code>insertHtml(self, text)</code>
<code>insertPlainText(self, text)</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>

continues on next page

Table 3.3 – continued from previous page

<code>isMaximized(self)</code>
<code>isMinimized(self)</code>
<code>isModal(self)</code>
<code>isReadOnly(self)</code>
<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, signal)</code>
<code>isUndoRedoEnabled(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>keyPressEvent(self, e)</code>
<code>keyReleaseEvent(self, e)</code>
<code>keyboardGrabber()</code>
<code>killTimer(self, id)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code>leaveEvent(self, a0)</code>
<code>lineWidth(self)</code>
<code>lineWrapColumnOrWidth(self)</code>
<code>lineWrapMode(self)</code>
<code>loadResource(self, type, name)</code>
<code>locale(self)</code>
<code>logicalDpiX(self)</code>

continues on next page

Table 3.3 – continued from previous page

<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>
<code>mergeCurrentCharFormat(self, modifier)</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>mouseDoubleClickEvent(self, e)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, e)</code>
<code>mousePressEvent(self, e)</code>
<code>mouseReleaseEvent(self, e)</code>

continues on next page

Table 3.3 – continued from previous page

<code>move()</code>
<code>moveCursor(self, operation[, mode])</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>overwriteMode(self)</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>paste(self)</code>
<code>physicalDpiX(self)</code>
<code>physicalDpiY(self)</code>
<code>placeholderText(self)</code>
<code>pos(self)</code>
<code>previousInFocusChain(self)</code>
<code>print(self, printer)</code>
<code>print_(self, printer)</code>

continues on next page

Table 3.3 – continued from previous page

<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>redo(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, id)</code>	
<code>removeAction(self, action)</code>	
<code>removeEventFilter(self, a0)</code>	
<code>render(, sourceRegion, flags, ...)</code>	
<code>repaint(-&gt; None -&gt; None)</code>	
<code>resize()</code>	
<code><i>resizeEvent</i>(self, a0)</code>	
<code>restoreGeometry(self, geometry)</code>	
<code>saveGeometry(self)</code>	
<code>screen(self)</code>	
<code>scroll()</code>	
<code>scrollBarWidgets(self, alignment)</code>	
<code>scrollContentsBy(self, dx, dy)</code>	
<code>scrollToAnchor(self, name)</code>	
<code>selectAll(self)</code>	
<code>sender(self)</code>	
<code>senderSignalIndex(self)</code>	
<code>setAcceptDrops(self, on)</code>	

continues on next page

Table 3.3 – continued from previous page

<code>setAcceptRichText(self, accept)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAlignment(self, a)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setAutoFormatting(self, features)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCornerWidget(self, widget)</code>
<code>setCurrentCharFormat(self, format)</code>
<code>setCurrentFont(self, f)</code>
<code>setCursor(self, a0)</code>
<code>setCursorWidth(self, width)</code>
<code>setDisabled(self, a0)</code>
<code>setDocument(self, document)</code>
<code>setDocumentTitle(self, title)</code>
<code>setEnabled(self, a0)</code>
<code>setExtraSelections(self, selections)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>
<code>setFocus()</code>
<code>setFocusPolicy(self, policy)</code>

continues on next page

Table 3.3 – continued from previous page

---

<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setFontFamily(self, fontFamily)</code>
<code>setFontItalic(self, b)</code>
<code>setFontPointSize(self, s)</code>
<code>setFontUnderline(self, b)</code>
<code>setFontWeight(self, w)</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>setHidden(self, hidden)</code>
<code>setHorizontalScrollBar(self, scrollbar)</code>
<code>setHorizontalScrollBarPolicy(self, a0)</code>
<code>setHtml(self, text)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLineWidth(self, a0)</code>
<code>setLineWrapColumnOrWidth(self, w)</code>
<code>setLineWrapMode(self, mode)</code>
<code>setLocale(self, locale)</code>
<code>setMarkdown(self, markdown)</code>

---

continues on next page

Table 3.3 – continued from previous page

<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>setMouseTracking(self, enable)</code>
<code>setObjectName(self, name)</code>
<code>setOverwriteMode(self, overwrite)</code>
<code>setPalette(self, a0)</code>
<code>setParent()</code>
<code>setPlaceholderText(self, placeholderText)</code>
<code>setPlainText(self, text)</code>
<code>setProperty(self, name, value)</code>
<code>setReadOnly(self, ro)</code>
<code>setShortcutAutoRepeat(self, id[, enabled])</code>
<code>setShortcutEnabled(self, id[, enabled])</code>
<code>setSizeAdjustPolicy(self, policy)</code>
<code>setSizeIncrement()</code>
<code>setSizePolicy()</code>
<code>setStatusTip(self, a0)</code>
<code>setStyle(self, a0)</code>
<code>setStyleSheet(self, styleSheet)</code>
<code>setTabChangesFocus(self, b)</code>

continues on next page

Table 3.3 – continued from previous page

<code>setTabOrder(a0, a1)</code>
<code>setTabStopDistance(self, distance)</code>
<code>setTabStopWidth(self, width)</code>
<code>setTabletTracking(self, enable)</code>
<code>setText(self, text)</code>
<code>setTextBackgroundColor(self, c)</code>
<code>setTextColor(self, c)</code>
<code>setTextCursor(self, cursor)</code>
<code>setTextInteractionFlags(self, flags)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUndoRedoEnabled(self, enable)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVerticalScrollBar(self, scrollbar)</code>
<code>setVerticalScrollBarPolicy(self, a0)</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 3.3 – 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>setWordWrapMode(self, policy)</code>
<code>setupViewport(self, viewport)</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>sizeAdjustPolicy(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>tabChangesFocus(self)</code>
<code>tabStopDistance(self)</code>

continues on next page

Table 3.3 – continued from previous page

---

<code>tabStopWidth(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>textBackgroundColor(self)</code>
<code>textColor(self)</code>
<code>textCursor(self)</code>
<code>textInteractionFlags(self)</code>
<code>thread(self)</code>
<code>timerEvent(self, e)</code>
<code>toHtml(self)</code>
<code>toMarkdown(self[, features])</code>
<code>toPlainText(self)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>
<code>underMouse(self)</code>
<code>undo(self)</code>
<code>ungrabGesture(self, type)</code>
<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>
<code>update(-&gt; None -&gt; None)</code>
<code>updateGeometry(self)</code>
<code>updateMicroFocus(self)</code>
<code><i>update_size()</i></code>
<code>updatesEnabled(self)</code>

---

continues on next page

Table 3.3 – continued from previous page

<code>verticalScrollBar(self)</code>
<code>verticalScrollBarPolicy(self)</code>
<code>viewport(self)</code>
<code>viewportEvent(self, a0)</code>
<code>viewportMargins(self)</code>
<code>viewportSizeHint(self)</code>
<code>visibleRegion(self)</code>
<code>whatsThis(self)</code>
<code>wheelEvent(self, e)</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>wordWrapMode(self)</code>
<code>x(self)</code>

continues on next page

Table 3.3 – continued from previous page

y(self)
zoomIn(self[, range])
zoomOut(self[, range])

## Attributes

AdjustIgnored
AdjustToContents
AdjustToContentsOnFirstShow
AutoAll
AutoBulletList
AutoNone
Box
DrawChildren
DrawWindowBackground
FixedColumnWidth
FixedPixelWidth
HLine
IgnoreMask
NoFrame
NoWrap
Panel
PdmDepth
PdmDevicePixelRatio
PdmDevicePixelRatioScaled
PdmDpiX

continues on next page

Table 3.4 – continued from previous page

PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
Plain	
Raised	
Shadow_Mask	
Shape_Mask	
StyledPanel	
Sunken	
VLine	
WidgetWidth	
WinPanel	
copyAvailable	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
currentCharFormatChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
cursorPositionChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
redoAvailable	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
selectionChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	

continues on next page

Table 3.4 – continued from previous page

textChanged	int = ..., arguments: Sequence = ...)	->
undoAvailable	int = ..., arguments: Sequence = ...)	->
windowIconChanged	int = ..., arguments: Sequence = ...)	->
windowIconTextChanged	int = ..., arguments: Sequence = ...)	->
windowTitleChanged	int = ..., arguments: Sequence = ...)	->

**resizeEvent**(self, a0: *QResizeEvent* | *None*)

**update\_size**()

### 3.6.9 runmanager.\_\_main\_\_.FingerTabBarWidget

**class** runmanager.\_\_main\_\_.FingerTabBarWidget(*parent=None, minwidth=180, minheight=30, \*\*kwargs*)

Bases: *QTabBar*

A TabBar with the tabs on the left and the text horizontal. Credit to @LegoStormtroopr, <https://gist.github.com/LegoStormtroopr/5075267>. We will promote the TabBar from the ui file to one of these.

**\_\_init\_\_**(*parent=None, minwidth=180, minheight=30, \*\*kwargs*)

#### Methods

<b>__init__</b> ([parent, minwidth, minheight])
<b>acceptDrops</b> (self)
<b>accessibleDescription</b> (self)
<b>accessibleName</b> (self)
<b>accessibleTabName</b> (self, index)
<b>actionEvent</b> (self, a0)
<b>actions</b> (self)
<b>activateWindow</b> (self)
<b>addAction</b> (self, action)
<b>addActions</b> (self, actions)
<b>addTab</b> (-> int)

continues on next page

Table 3.5 – continued from previous page

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

continues on next page

Table 3.5 – continued from previous page

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

continues on next page

Table 3.5 – continued from previous page

---

<code>eventFilter(self, a0, a1)</code>
<code>expanding(self)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>
<code>focusNextPrevChild(self, next)</code>
<code>focusOutEvent(self, a0)</code>
<code>focusPolicy(self)</code>
<code>focusPreviousChild(self)</code>
<code>focusProxy(self)</code>
<code>focusWidget(self)</code>
<code>font(self)</code>
<code>fontInfo(self)</code>
<code>fontMetrics(self)</code>
<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>
<code>frameSize(self)</code>
<code>geometry(self)</code>
<code>getContentsMargins(self)</code>
<code>grab(self[, rectangle])</code>
<code>grabGesture(self, type[, flags])</code>
<code>grabKeyboard(self)</code>
<code>grabMouse()</code>
<code>grabShortcut(self, key[, context])</code>

---

continues on next page

Table 3.5 – continued from previous page

<code>graphicsEffect(self)</code>
<code>graphicsProxyWidget(self)</code>
<code>hasFocus(self)</code>
<code>hasHeightForWidth(self)</code>
<code>hasMouseTracking(self)</code>
<code>hasTabletTracking(self)</code>
<code>height(self)</code>
<code>heightForWidth(self, a0)</code>
<code>heightMM(self)</code>
<code>hide(self)</code>
<code>hideEvent(self, a0)</code>
<code>iconSize(self)</code>
<code><i>indexAtPos</i>(point)</code>
<code>inherits(self, classname)</code>
<code>initPainter(self, painter)</code>
<code>initStyleOption(self, option, tabIndex)</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>insertTab(-&gt; int)</code>
<code>installEventFilter(self, a0)</code>
<code>isActiveWindow(self)</code>
<code>isAncestorOf(self, child)</code>
<code>isEnabled(self)</code>

continues on next page

Table 3.5 – continued from previous page

<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><i>isMovable</i>(self)</code>
<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, signal)</code>
<code>isTabEnabled(self, index)</code>
<code>isTabVisible(self, index)</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>

continues on next page

Table 3.5 – continued from previous page

<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>minimumTabSizeHint(self, index)</code>
<code>minimumWidth(self)</code>
<code>mouseDoubleClickEvent(self, a0)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, a0)</code>
<code><i>mousePressEvent</i>(self, a0)</code>
<code><i>mouseReleaseEvent</i>(self, a0)</code>
<code>move()</code>

continues on next page

Table 3.5 – continued from previous page

<code>moveEvent(self, a0)</code>	
<code>moveTab(self, from_, to)</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><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>	

continues on next page

Table 3.5 – continued from previous page

---

<code>releaseMouse(self)</code>
<code>releaseShortcut(self, id)</code>
<code>removeAction(self, action)</code>
<code>removeEventFilter(self, a0)</code>
<code>removeTab(self, index)</code>
<code>render(, sourceRegion, flags, ...)</code>
<code>repaint(-&gt; None -&gt; None)</code>
<code>resize()</code>
<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>selectionBehaviorOnRemove(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>setAccessibleTabName(self, index, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setAutoHide(self, hide)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setChangeCurrentOnDrag(self, change)</code>

---

continues on next page

Table 3.5 – continued from previous page

<code>setContentMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCurrentIndex(self, index)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setDocumentMode(self, set)</code>
<code>setDrawBase(self, drawTheBase)</code>
<code>setElideMode(self, a0)</code>
<code>setEnabled(self, a0)</code>
<code>setExpanding(self, enabled)</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, size)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLocale(self, locale)</code>

continues on next page

Table 3.5 – continued from previous page

<code>setMask()</code>	
<code>setMaximumHeight(self, maxh)</code>	
<code>setMaximumSize()</code>	
<code>setMaximumWidth(self, maxw)</code>	
<code>setMinimumHeight(self, minh)</code>	
<code>setMinimumSize()</code>	
<code>setMinimumWidth(self, minw)</code>	
<code>setMouseTracking(self, enable)</code>	
<code>setMovable(movable[, index])</code>	Set tabs movable on an individual basis, or set for all tabs if no index specified
<code>setObjectName(self, name)</code>	
<code>setPalette(self, a0)</code>	
<code>setParent()</code>	
<code>setProperty(self, name, value)</code>	
<code>setSelectionBehaviorOnRemove(self, behavior)</code>	
<code>setShape(self, shape)</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>setTabButton(self, index, position, widget)</code>	
<code>setTabData(self, index, data)</code>	
<code>setTabEnabled(self, index, a1)</code>	
<code>setTabIcon(self, index, icon)</code>	

continues on next page

Table 3.5 – continued from previous page

<code>setTabOrder(a0, a1)</code>
<code>setTabText(self, index, text)</code>
<code>setTabTextColor(self, index, color)</code>
<code>setTabToolTip(self, index, tip)</code>
<code>setTabVisible(self, index, visible)</code>
<code>setTabWhatsThis(self, index, text)</code>
<code>setTabletTracking(self, enable)</code>
<code>setTabsClosable(self, closable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setUsesScrollButtons(self, useButtons)</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>shape(self)</code>

continues on next page

Table 3.5 – continued from previous page

---

<code>sharedPainter(self)</code>
<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>stylesheet(self)</code>
<code>tabAt(self, pos)</code>
<code>tabButton(self, index, position)</code>
<code>tabData(self, index)</code>
<code>tabIcon(self, index)</code>
<code>tabInserted(self, index)</code>
<code><i>tabLayoutChange</i>(self)</code>
<code>tabRect(self, index)</code>
<code>tabRemoved(self, index)</code>
<code><i>tabSizeHint</i>(self, index)</code>

---

continues on next page

Table 3.5 – continued from previous page

<code>tabText(self, index)</code>
<code>tabTextColor(self, index)</code>
<code>tabToolTip(self, index)</code>
<code>tabWhatsThis(self, index)</code>
<code>tabletEvent(self, a0)</code>
<code>tabsClosable(self)</code>
<code>testAttribute(self, attribute)</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(-&gt; None -&gt; None)</code>
<code>updateGeometry(self)</code>
<code>updateMicroFocus(self)</code>
<code>updatesEnabled(self)</code>
<code>usesScrollButtons(self)</code>
<code>visibleRegion(self)</code>
<code>whatsThis(self)</code>
<code>wheelEvent(self, event)</code>
<code>width(self)</code>

continues on next page

Table 3.5 – continued from previous page

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)
x(self)
y(self)

**Attributes**

DrawChildren
DrawWindowBackground
IgnoreMask
LeftSide
PdmDepth
PdmDevicePixelRatio
PdmDevicePixelRatioScaled

continues on next page

Table 3.6 – continued from previous page

PdmDpiX			
PdmDpiY			
PdmHeight			
PdmHeightMM			
PdmNumColors			
PdmPhysicalDpiX			
PdmPhysicalDpiY			
PdmWidth			
PdmWidthMM			
RightSide			
RoundedEast			
RoundedNorth			
RoundedSouth			
RoundedWest			
SelectLeftTab			
SelectPreviousTab			
SelectRightTab			
TriangularEast			
TriangularNorth			
TriangularSouth			
TriangularWest			
currentChanged	int = ..., arguments:	Sequence = ...)	->
	PYQT_SIGNAL		
customContextMenuRequested	int = ..., arguments:	Sequence = ...)	->
	PYQT_SIGNAL		
destroyed	int = ..., arguments:	Sequence = ...)	->
	PYQT_SIGNAL		
objectNameChanged	int = ..., arguments:	Sequence = ...)	->
	PYQT_SIGNAL		
staticMetaObject			

continues on next page

Table 3.6 – continued from previous page

tabBarClicked	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
tabBarDoubleClicked	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
tabCloseRequested	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
tabMoved	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
windowIconChanged	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
windowIconTextChanged	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
windowTitleChanged	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	

**indexAtPos**(*point*)

**isMovable**(*self*) → bool

**mousePressEvent**(*self*, *a0*: *QMouseEvent* | *None*)

**mouseReleaseEvent**(*self*, *a0*: *QMouseEvent* | *None*)

**paintEvent**(*self*, *a0*: *QPaintEvent* | *None*)

**setMovable**(*movable*, *index=None*)

Set tabs movable on an individual basis, or set for all tabs if no index specified

**setTabButton**(*self*, *index: int*, *position: QTabBar.ButtonPosition*, *widget: QWidget* | *None*)

**tabLayoutChange**(*self*)

**tabSizeHint**(*self*, *index: int*) → QSize

### 3.6.10 runmanager.\_\_main\_\_.FingerTabWidget

**class** runmanager.\_\_main\_\_.FingerTabWidget(*parent*, \**args*)

Bases: QTabWidget

A QTabWidget equivalent which uses our FingerTabBarWidget

**\_\_init\_\_**(*parent*, \**args*)

#### Methods

**\_\_init\_\_**(*parent*, \**args*)

**acceptDrops**(*self*)

**accessibleDescription**(*self*)

continues on next page

Table 3.7 – continued from previous page

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

continues on next page

Table 3.7 – continued from previous page

---

<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>cornerWidget(self[, corner])</code>
<code>count(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>currentIndex(self)</code>
<code>currentWidget(self)</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>
<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySubWin- dows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>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>

---

continues on next page

Table 3.7 – continued from previous page

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

continues on next page

Table 3.7 – continued from previous page

---

<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, width)</code>
<code>heightMM(self)</code>
<code>hide(self)</code>
<code>hideEvent(self, a0)</code>
<code>iconSize(self)</code>
<code>indexOf(self, widget)</code>
<code>inherits(self, classname)</code>
<code>initPainter(self, painter)</code>
<code>initStyleOption(self, option)</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 3.7 – continued from previous page

<code>insertActions(self, before, actions)</code>
<code>insertTab(-&gt; int)</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>isMovable(self)</code>
<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, signal)</code>
<code>isTabEnabled(self, index)</code>
<code>isTabVisible(self, index)</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>

continues on next page

Table 3.7 – continued from previous page

<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>
<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>

continues on next page

Table 3.7 – continued from previous page

<code>mouseGrabber()</code>	
<code>mouseMoveEvent(self, a0)</code>	
<code>mousePressEvent(self, a0)</code>	
<code>mouseReleaseEvent(self, a0)</code>	
<code>move()</code>	
<code>moveEvent(self, a0)</code>	
<code>moveToThread(self, thread)</code>	
<code>nativeEvent(self, eventType, message)</code>	
<code>nativeParentWidget(self)</code>	
<code>nextInFocusChain(self)</code>	
<code>normalGeometry(self)</code>	
<code>objectName(self)</code>	
<code>overrideWindowFlags(self, type)</code>	
<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.

continues on next page

Table 3.7 – continued from previous page

<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>removeTab(self, index)</code>
<code>render(, sourceRegion, flags, ...)</code>
<code>repaint(-&gt; None -&gt; None)</code>
<code>resize()</code>
<code>resizeEvent(self, a0)</code>
<code>restoreGeometry(self, geometry)</code>
<code>saveGeometry(self)</code>
<code>screen(self)</code>
<code>scroll()</code>
<code>sender(self)</code>
<code>senderSignalIndex(self)</code>
<code>setAcceptDrops(self, on)</code>
<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>

continues on next page

Table 3.7 – continued from previous page

<code>setContentMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCornerWidget(self, widget[, corner])</code>
<code>setCurrentIndex(self, index)</code>
<code>setCurrentWidget(self, widget)</code>
<code>setCursor(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setDocumentMode(self, set)</code>
<code>setElideMode(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>setIconSize(self, size)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLocale(self, locale)</code>

continues on next page

Table 3.7 – continued from previous page

---

<code>setMask()</code>
<code>setMaximumHeight(self, maxh)</code>
<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setMouseTracking(self, enable)</code>
<code>setMovable(self, movable)</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>setTabBar(self, a0)</code>
<code>setTabBarAutoHide(self, enabled)</code>
<code><i>setTabClosable</i>(index, closable)</code>
<code>setTabEnabled(self, index, a1)</code>
<code>setTabIcon(self, index, icon)</code>
<code>setTabOrder(a0, a1)</code>

---

continues on next page

Table 3.7 – continued from previous page

<code>setTabPosition(self, a0)</code>
<code>setTabShape(self, s)</code>
<code>setTabText(self, index, a1)</code>
<code>setTabToolTip(self, index, tip)</code>
<code>setTabVisible(self, index, visible)</code>
<code>setTabWhatsThis(self, index, text)</code>
<code>setTabletTracking(self, enable)</code>
<code>setTabsClosable(self, closeable)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setUsesScrollButtons(self, useButtons)</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>

continues on next page

Table 3.7 – continued from previous page

---

<code>show(self)</code>
<code>showEvent(self, a0)</code>
<code>showFullScreen(self)</code>
<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabBar(self)</code>
<code>tabBarAutoHide(self)</code>
<code>tabIcon(self, index)</code>
<code>tabInserted(self, index)</code>
<code>tabPosition(self)</code>
<code>tabRemoved(self, index)</code>
<code>tabShape(self)</code>
<code>tabText(self, index)</code>
<code>tabToolTip(self, index)</code>
<code>tabWhatsThis(self, index)</code>

---

continues on next page

Table 3.7 – continued from previous page

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

continues on next page

Table 3.7 – continued from previous page

windowFilePath(self)
windowFlags(self)
windowHandle(self)
windowIcon(self)
windowIconText(self)
windowModality(self)
windowOpacity(self)
windowRole(self)
windowState(self)
windowTitle(self)
windowType(self)
x(self)
y(self)

### Attributes

DrawChildren
DrawWindowBackground
East
IgnoreMask
North
PdmDepth
PdmDevicePixelRatio
PdmDevicePixelRatioScaled
PdmDpiX
PdmDpiY

continues on next page

Table 3.8 – continued from previous page

PdmHeight			
PdmHeightMM			
PdmNumColors			
PdmPhysicalDpiX			
PdmPhysicalDpiY			
PdmWidth			
PdmWidthMM			
Rounded			
South			
Triangular			
West			
currentChanged	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
customContextMenuRequested	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
destroyed	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
objectNameChanged	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
staticMetaObject			
tabBarClicked	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
tabBarDoubleClicked	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
tabCloseRequested	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
windowIconChanged	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
windowIconTextChanged	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
windowTitleChanged	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->

**addTab**(*self*, *widget*: *QWidget* | *None*, *a1*: *str* | *None*) → int

**addTab**(*self*, *widget*: *QWidget* | *None*, *icon*: *QIcon*, *label*: *str* | *None*) → int

**setTabClosable**(*index*, *closable*)

### 3.6.11 runmanager.\_\_main\_\_.GroupTab

**class** runmanager.\_\_main\_\_.GroupTab(*tabWidget, globals\_file, group\_name*)

Bases: `object`

**\_\_init\_\_**(*tabWidget, globals\_file, group\_name*)

## Methods

<code>__init__(tabWidget, globals_file, group_name)</code>	
<code>change_global_expansion(global_name, ...)</code>	
<code>change_global_units(global_name, ...)</code>	
<code>change_global_value(global_name, ...[, ...])</code>	
<code>check_for_boolean_values(item)</code>	Checks if the value is 'True' or 'False'.
<code>close()</code>	
<code>complete_change_global_value(global_name, ...)</code>	
<code>connect_signals()</code>	
<code>delete_global(global_name[, confirm])</code>	
<code>do_model_sort()</code>	
<code>get_global_item_by_name(global_name, column)</code>	Returns an item from the row representing a global in the globals model.
<code>globals_changed()</code>	Called whenever something about a global has changed.
<code>make_global_row(name[, value, units, expansion])</code>	
<code>new_global(global_name)</code>	
<code>on_globals_delete_selected_triggered()</code>	
<code>on_globals_model_expansion_changed(item)</code>	
<code>on_globals_model_item_changed(item)</code>	
<code>on_globals_model_name_changed(item)</code>	Handles global renaming and creation of new globals due to the user editing the <click to add global> item
<code>on_globals_model_units_changed(item)</code>	
<code>on_globals_model_value_changed(item)</code>	
<code>on_globals_set_selected_bools_triggered(st</code>	
<code>on_tableView_globals_context_menu_request</code>	
<code>on_tableView_globals_leftClicked(index)</code>	
<code>populate_model()</code>	
<code>rename_global(previous_global_name, ...)</code>	
<code>set_file_and_group_name(globals_file, group_name)</code>	Provided as a separate method so the main app can call it if the group gets renamed
<code>set_tab_icon(icon_string)</code>	

## Attributes

<i>COLOR_BOOL_OFF</i>
<i>COLOR_BOOL_ON</i>
<i>COLOR_ERROR</i>
<i>COLOR_OK</i>
<i>GLOBALS_COL_DELETE</i>
<i>GLOBALS_COL_EXPANSION</i>
<i>GLOBALS_COL_NAME</i>
<i>GLOBALS_COL_UNITS</i>
<i>GLOBALS_COL_VALUE</i>
<i>GLOBALS_DUMMY_ROW_TEXT</i>
<i>GLOBALS_ROLE_IS_BOOL</i>
<i>GLOBALS_ROLE_IS_DUMMY_ROW</i>
<i>GLOBALS_ROLE_PREVIOUS_TEXT</i>
<i>GLOBALS_ROLE_SORT_DATA</i>

**COLOR\_BOOL\_OFF = '#608060'**

**COLOR\_BOOL\_ON = '#63F731'**

**COLOR\_ERROR = '#F79494'**

**COLOR\_OK = '#A5F7C6'**

**GLOBALS\_COL\_DELETE = 0**

**GLOBALS\_COL\_EXPANSION = 4**

**GLOBALS\_COL\_NAME = 1**

**GLOBALS\_COL\_UNITS = 3**

**GLOBALS\_COL\_VALUE = 2**

**GLOBALS\_DUMMY\_ROW\_TEXT = '<Click to add global>'**

**GLOBALS\_ROLE\_IS\_BOOL = 260**

**GLOBALS\_ROLE\_IS\_DUMMY\_ROW = 257**

**GLOBALS\_ROLE\_PREVIOUS\_TEXT** = 259

**GLOBALS\_ROLE\_SORT\_DATA** = 258

**change\_global\_expansion**(*global\_name, previous\_expansion, new\_expansion*)

**change\_global\_units**(*global\_name, previous\_units, new\_units*)

**change\_global\_value**(*global\_name, previous\_value, new\_value, interactive=True*)

**check\_for\_boolean\_values**(*item*)

Checks if the value is 'True' or 'False'. If either, makes the units cell checkable, uneditable, and coloured to indicate the state. The units cell can then be clicked to toggle the value.

**close**()

**complete\_change\_global\_value**(*global\_name, previous\_value, new\_value, item, previous\_background, previous\_icon, interactive=True*)

**connect\_signals**()

**delete\_global**(*global\_name, confirm=True*)

**do\_model\_sort**()

**get\_global\_item\_by\_name**(*global\_name, column, previous\_name=None*)

Returns an item from the row representing a global in the globals model. Which item is returned is set by the column argument.

**globals\_changed**()

Called whenever something about a global has changed. call `app.globals_changed` to inform the main application that it needs to parse globals again. `self.update_parse_indication` will be called by the main app when parsing is done, and will set the colours and tooltips appropriately

**make\_global\_row**(*name, value="", units="", expansion=""*)

**new\_global**(*global\_name*)

**on\_globals\_delete\_selected\_triggered**()

**on\_globals\_model\_expansion\_changed**(*item*)

**on\_globals\_model\_item\_changed**(*item*)

**on\_globals\_model\_name\_changed**(*item*)

Handles global renaming and creation of new globals due to the user editing the <click to add global> item

**on\_globals\_model\_units\_changed**(*item*)

**on\_globals\_model\_value\_changed**(*item*)

**on\_globals\_set\_selected\_bools\_triggered**(*state*)

**on\_tableView\_globals\_context\_menu\_requested**(*point*)

**on\_tableView\_globals\_leftClicked**(*index*)

**populate\_model**()

**rename\_global**(*previous\_global\_name, new\_global\_name*)

**set\_file\_and\_group\_name**(*globals\_file, group\_name*)

Provided as a separate method so the main app can call it if the group gets renamed

**set\_tab\_icon**(*icon\_string*)

**update\_parse\_indication**(*active\_groups, sequence\_globals, evaled\_globals*)

### 3.6.12 runmanager.\_\_main\_\_.ItemDelegate

**class** runmanager.\_\_main\_\_.ItemDelegate(\*args, \*\*kwargs)

Bases: [QStyledItemDelegate](#)

An item delegate with a larger row height and column width, faint grey vertical lines between columns, and a custom editor for handling multi-line data

**\_\_init\_\_**(\*args, \*\*kwargs)

#### Methods

[\\_\\_init\\_\\_](#)(\*args, \*\*kwargs)

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)

dumpObjectTree(self)

dynamicPropertyNames(self)

editorEvent(self, event, model, option, index)

---

continues on next page

Table 3.9 – continued from previous page

<code>event(self, a0)</code>	
<code>eventFilter(obj, event)</code>	Filter events before they get to the editor, so that editing is ended when the user presses tab, shift-tab or enter (which otherwise would not end editing in a <code>QTextEdit</code> ).
<code>findChild(-&gt; QObjectT)</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>paint(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>	
<code>sender(self)</code>	
<code>senderSignalIndex(self)</code>	
<code>setEditorData(self, editor, index)</code>	

continues on next page

Table 3.9 – continued from previous page

---

<code>setItemEditorFactory(self, factory)</code>
<code><i>setModelData</i>(self, editor, model, index)</code>
<code>setObjectName(self, name)</code>
<code>setParent(self, a0)</code>
<code>setProperty(self, name, value)</code>
<code>signalsBlocked(self)</code>
<code><i>sizeHint</i>(self, option, index)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>
<code>updateEditorGeometry(self, editor, option, index)</code>

---

## Attributes

<i>EXTRA_COL_WIDTH</i>			
<i>EXTRA_ROW_HEIGHT</i>			
EditNextItem			
EditPreviousItem			
<i>MIN_ROW_HEIGHT</i>			
NoHint			
RevertModelCache			
SubmitModelCache			
closeEditor	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
commitData	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
destroyed	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
objectNameChanged	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
sizeHintChanged	int = ..., arguments: PYQT_SIGNAL	Sequence = ...)	->
staticMetaObject			

**EXTRA\_COL\_WIDTH = 20**

**EXTRA\_ROW\_HEIGHT = 6**

**MIN\_ROW\_HEIGHT = 22**

**createEditor**(*self*, *parent*: *QWidget* | *None*, *option*: *QStyleOptionViewItem*, *index*: *QModelIndex*) → *QWidget* | *None*

**eventFilter**(*obj*, *event*)

Filter events before they get to the editor, so that editing is ended when the user presses tab, shift-tab or enter (which otherwise would not end editing in a *QTextEdit*).

**paint**(*self*, *painter*: *QPainter* | *None*, *option*: *QStyleOptionViewItem*, *index*: *QModelIndex*)

**setEditorData**(*self*, *editor*: *QWidget* | *None*, *index*: *QModelIndex*)

**setModelData**(*self*, *editor*: *QWidget* | *None*, *model*: *QAbstractItemModel* | *None*, *index*: *QModelIndex*)

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

### 3.6.13 runmanager.\_\_main\_\_.ItemView

**class** runmanager.\_\_main\_\_.ItemView(\*args)

Bases: `object`

Mixin for QTableView and QTreeView that emits a custom signal `leftClicked(index)` after a left click on a valid index, and `doubleLeftClicked(index)` (in addition) on double click. Also has modified tab and arrow key behaviour and custom selection highlighting.

`__init__`(\*args)

#### Methods

<code>__init__</code> (*args)
<code>keyPressEvent</code> (event)
<code>leaveEvent</code> (event)
<code>mouseDoubleClickEvent</code> (event)
<code>mousePressEvent</code> (event)
<code>mouseReleaseEvent</code> (event)
<code>moveCursor</code> (cursor_action, keyboard_modifiers)

#### Attributes

<code>COLOR_HIGHLIGHT</code>	
<code>doubleLeftClicked</code>	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
<code>leftClicked</code>	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

`COLOR_HIGHLIGHT = '#40308CC6'`

#### `doubleLeftClicked`

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

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

#### Type

`pyqtSignal`(\*types, name)

**Type**

str = ..., revision

**keyPressEvent**(*event*)**leaveEvent**(*event*)**leftClicked**

int = ..., arguments: Sequence = ...) -&gt; PYQT\_SIGNAL

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

**Type**

pyqtSignal(\*types, name

**Type**

str = ..., revision

**mouseDoubleClickEvent**(*event*)**mousePressEvent**(*event*)**mouseReleaseEvent**(*event*)**moveCursor**(*cursor\_action*, *keyboard\_modifiers*)

### 3.6.14 runmanager.\_\_main\_\_.PoppedOutOutputBoxWindow

**class** runmanager.\_\_main\_\_.PoppedOutOutputBoxWindow

Bases: QDialog

**\_\_init\_\_**(\*args, \*\*kwargs)**Methods**`__init__(*args, **kwargs)``accept(self)``acceptDrops(self)``accessibleDescription(self)``accessibleName(self)``actionEvent(self, a0)``actions(self)`

continues on next page

Table 3.10 – continued from previous page

---

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

---

continues on next page

Table 3.10 – continued from previous page

<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>deleteLater(self)</code>
<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySubWindows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>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>

continues on next page

Table 3.10 – continued from previous page

---

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

---

continues on next page

Table 3.10 – continued from previous page

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

continues on next page

Table 3.10 – continued from previous page

---

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

---

continues on next page

Table 3.10 – continued from previous page

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

continues on next page

Table 3.10 – continued from previous page

<code>overrideWindowState(self, state)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, a0)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, name)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, signal)</code>	
<code>rect(self)</code>	
<code>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(-&gt; None -&gt; None)</code>	
<code>resize()</code>	
<code>resizeEvent(self, a0)</code>	

continues on next page

Table 3.10 – continued from previous page

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

continues on next page

Table 3.10 – continued from previous page

---

<code>setForegroundColor(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setHidden(self, hidden)</code>
<code>setInputMethodHints(self, hints)</code>
<code>setLayout(self, a0)</code>
<code>setLayoutDirection(self, direction)</code>
<code>setLocale(self, locale)</code>
<code>setMask()</code>
<code>setMaximumHeight(self, maxh)</code>
<code>setMaximumSize()</code>
<code>setMaximumWidth(self, maxw)</code>
<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>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>

---

continues on next page

Table 3.10 – continued from previous page

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

continues on next page

Table 3.10 – continued from previous page

<code>showMaximized(self)</code>
<code>showMinimized(self)</code>
<code>showNormal(self)</code>
<code>signalsBlocked(self)</code>
<code>size(self)</code>
<code>sizeHint(self)</code>
<code>sizeIncrement(self)</code>
<code>sizePolicy(self)</code>
<code>stackUnder(self, a0)</code>
<code>startTimer(self, interval[, timerType])</code>
<code>statusTip(self)</code>
<code>style(self)</code>
<code>styleSheet(self)</code>
<code>tabletEvent(self, a0)</code>
<code>testAttribute(self, attribute)</code>
<code>thread(self)</code>
<code>timerEvent(self, a0)</code>
<code>toolTip(self)</code>
<code>toolTipDuration(self)</code>
<code>tr(self, sourceText[, disambiguation, n])</code>
<code>underMouse(self)</code>
<code>ungrabGesture(self, type)</code>
<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>
<code>update(-&gt; None -&gt; None)</code>

continues on next page

Table 3.10 – 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	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
finished	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
objectNameChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
rejected	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
staticMetaObject	
windowIconChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
3.6. runmanager.__main__	

`closeEvent(self, a0: QCloseEvent | None)`

### 3.6.15 runmanager.\_\_main\_\_.RemoteServer

`class runmanager.__main__.RemoteServer`

Bases: `ZMQServer`

`__init__()`

## Methods

<code>__init__()</code>	
<code>handle_abort()</code>	
<code>handle_engage()</code>	
<code>handle_error_in_globals()</code>	
<code>handle_get_globals([raw])</code>	
<code>handle_get_labscript_file()</code>	
<code>handle_get_run_shots()</code>	
<code>handle_get_shot_output_folder()</code>	
<code>handle_get_shuffle()</code>	
<code>handle_get_view_shots()</code>	
<code>handle_is_output_folder_default()</code>	
<code>handle_n_shots()</code>	
<code>handle_reset_shot_output_folder()</code>	
<code>handle_set_globals(globals[, raw])</code>	
<code>handle_set_labscript_file(value)</code>	
<code>handle_set_run_shots(value)</code>	
<code>handle_set_shot_output_folder(value)</code>	
<code>handle_set_shuffle(value)</code>	
<code>handle_set_view_shots(value)</code>	
<code>handler(request_data)</code>	To be overridden by subclasses.
<code>mainloop()</code>	
<code>setup_auth(context)</code>	Deprecated.
<code>shutdown()</code>	
<code>shutdown_on_interrupt()</code>	
<code>timeout()</code>	A function to call every self.timeout_interval seconds in the same thread as the handler.

### `handle_abort()`

```
handle_engage()
handle_error_in_globals()
handle_get_globals(raw=False)
handle_get_labscript_file()
handle_get_run_shots()
handle_get_shot_output_folder()
handle_get_shuffle()
handle_get_view_shots()
handle_is_output_folder_default()
handle_n_shots()
handle_reset_shot_output_folder()
handle_set_globals(globals, raw=False)
handle_set_labscript_file(value)
handle_set_run_shots(value)
handle_set_shot_output_folder(value)
handle_set_shuffle(value)
handle_set_view_shots(value)
handler(request_data)
```

To be overridden by subclasses. This is an example implementation

### 3.6.16 runmanager.\_\_main\_\_.RunManager

```
class runmanager.__main__.RunManager
```

Bases: `object`

```
__init__()
```

#### Methods

```
__init__()
```

```
add_item_to_axes_model(expansion_name,
shuffle)
```

```
check_child_exited(timeout_time[, kill])
```

```
check_output_folder_update()
```

Do a single check of whether the output folder needs updating.

continues on next page

Table 3.11 – continued from previous page

<code>close_current_tab()</code>	
<code>close_globals_file(globals_file[, confirm])</code>	
<code>close_group(globals_file, group_name)</code>	
<code>compile_loop()</code>	
<code>connect_signals()</code>	
<code>copy_group(source_globals_file, ...[, ...])</code>	This function copies a group of globals with the name <code>source_group_name</code> from the file <code>source_globals_file</code> to a new file <code>dest_globals_file</code> .
<code>delete_group(globals_file, group_name[, confirm])</code>	
<code>do_model_sort()</code>	
<code>get_active_groups([interactive])</code>	Returns active groups in the format <code>{group_name: globals_file}</code> .
<code>get_default_output_folder()</code>	Returns what the default output folder would be right now, based on the current date and selected labscript file.
<code>get_group_item_by_name(globals_file, ...[, ...])</code>	Returns an item from the row representing a globals group in the groups model.
<code>get_save_data()</code>	
<code>globals_changed()</code>	Called from either self, a GroupTab, or the Remote-Server to inform runmanager that something about globals has changed, and that they need parsing again.
<code>guess_expansion_modes(active_groups, ...)</code>	This function is designed to be called iteratively.
<code>load_configuration(filename)</code>	
<code>make_group_row(group_name)</code>	Returns a new row representing one group in the groups tab, ready to be inserted into the model.
<code>make_h5_files(labscript_file, output_folder, ...)</code>	
<code>new_group(globals_file, group_name)</code>	
<code>on_abort_clicked()</code>	
<code>on_axes_check_selected_triggered(*args)</code>	
<code>on_axes_item_changed(item)</code>	
<code>on_axes_uncheck_selected_triggered(*args)</code>	
<code>on_axis_down_clicked(checked)</code>	
<code>on_axis_to_bottom_clicked(checked)</code>	
<code>on_axis_to_top_clicked(checked)</code>	

continues on next page

Table 3.11 – continued from previous page

<code>on_axis_up_clicked</code> (checked)	
<code>on_close_event</code> ()	
<code>on_diff_globals_file_clicked</code> ()	
<code>on_edit_labscript_file_clicked</code> (checked)	
<code>on_engage_clicked</code> ()	
<code>on_groups_close_selected_files_triggered</code> ()	
<code>on_groups_close_selected_groups_triggered</code>	
<code>on_groups_copy_selected_groups_triggered</code> (	
<code>on_groups_delete_selected_triggered</code> ()	
<code>on_groups_model_active_changed</code> (item)	Sets the sort data for the item in response to its check state changing.
<code>on_groups_model_item_changed</code> (item)	This function is for responding to data changes in the model.
<code>on_groups_model_name_changed</code> (item)	Handles group renaming and creation of new groups due to the user editing the <click to add group> item
<code>on_groups_model_openclose_changed</code> (item)	Sets item sort data and icon in response to the open/close state of a group changing.
<code>on_groups_open_selected_triggered</code> ()	
<code>on_groups_set_selection_active_triggered</code> (.	
<code>on_labscript_file_text_changed</code> (text)	
<code>on_load_configuration_triggered</code> ()	
<code>on_master_shuffle_clicked</code> (state)	
<code>on_new_globals_file_clicked</code> ()	
<code>on_open_globals_file_clicked</code> ()	
<code>on_output_popout_button_clicked</code> ()	
<code>on_reset_shot_output_folder_clicked</code> (checke	
<code>on_restart_subprocess_clicked</code> ()	
<code>on_revert_configuration_triggered</code> ()	
<code>on_save_configuration_as_triggered</code> ()	
<code>on_save_configuration_triggered</code> ()	

continues on next page

Table 3.11 – continued from previous page

<code>on_select_labscript_file_clicked</code> (checked)	
<code>on_select_shot_output_folder_clicked</code> (check	
<code>on_shot_output_folder_text_changed</code> (text)	
<code>on_tabCloseRequested</code> (index)	
<code>on_treeView_axes_context_menu_requested</code> (pc	
<code>on_treeView_groups_context_menu_requested</code>	
<code>on_treeView_groups_doubleLeftClicked</code> (index)	
<code>on_treeView_groups_leftClicked</code> (index)	Here we respond to user clicks on the treeview.
<code>open_globals_file</code> (globals_file)	
<code>open_group</code> (globals_file, group_name[, ...])	
<code>parse_globals</code> (active_groups[, ...])	
<code>preparse_globals</code> ()	
<code>preparse_globals_loop</code> ()	Runs in a thread, waiting on a <code>threading.Event</code> that tells us when some globals have changed, and calls <code>parse_globals</code> to evaluate them all before feeding the results back to the relevant tabs to be displayed.
<code>rename_group</code> (globals_file, ...)	
<code>rollover_shot_output_folder</code> ()	Runs in a thread, checking every 30 seconds if the default output folder has changed, likely because the date has changed, but also possible because another instance of runmanager has incremented the sequence index.
<code>save_configuration</code> (save_file)	
<code>send_to_BLACS</code> (run_file, BLACS_hostname)	
<code>send_to_runviewer</code> (run_file)	
<code>setup_axes_tab</code> ()	
<code>setup_config</code> ()	
<code>setup_groups_tab</code> ()	
<code>switch_tabs</code> (change)	
<code>update_axes_indentation</code> ()	
<code>update_axes_tab</code> (expansions, dimensions)	

continues on next page

Table 3.11 – continued from previous page

<code>update_global_shuffle_state(*args, **kwargs)</code>	
<code>update_tabs_parsing_indication(...)</code>	
<code>wait_until_preparse_complete()</code>	Block until the preparse loop has finished pending work

### Attributes

<code>AXES_COL_LENGTH</code>
<code>AXES_COL_NAME</code>
<code>AXES_COL_SHUFFLE</code>
<code>AXES_ROLE_NAME</code>
<code>GROUPS_COL_ACTIVE</code>
<code>GROUPS_COL_DELETE</code>
<code>GROUPS_COL_NAME</code>
<code>GROUPS_COL_OPENCLOSE</code>
<code>GROUPS_DUMMY_ROW_TEXT</code>
<code>GROUPS_ROLE_GROUP_IS_OPEN</code>
<code>GROUPS_ROLE_IS_DUMMY_ROW</code>
<code>GROUPS_ROLE_PREVIOUS_NAME</code>
<code>GROUPS_ROLE_SORT_DATA</code>

`AXES_COL_LENGTH = 1`

`AXES_COL_NAME = 0`

`AXES_COL_SHUFFLE = 2`

`AXES_ROLE_NAME = 257`

`GROUPS_COL_ACTIVE = 1`

`GROUPS_COL_DELETE = 2`

`GROUPS_COL_NAME = 0`

`GROUPS_COL_OPENCLOSE = 3`

**GROUPS\_DUMMY\_ROW\_TEXT** = '<Click to add group>'

**GROUPS\_ROLE\_GROUP\_IS\_OPEN** = 260

**GROUPS\_ROLE\_IS\_DUMMY\_ROW** = 257

**GROUPS\_ROLE\_PREVIOUS\_NAME** = 258

**GROUPS\_ROLE\_SORT\_DATA** = 259

**add\_item\_to\_axes\_model**(*expansion\_name, shuffle, dimensions=None*)

**check\_child\_exited**(*timeout\_time, kill=False*)

**check\_output\_folder\_update**()

Do a single check of whether the output folder needs updating. This is implemented as a separate function to the above loop so that the whole check happens at once in the Qt main thread and hence is atomic and can't be interfered with by other Qt calls in the program.

**close\_current\_tab**()

**close\_globals\_file**(*globals\_file, confirm=True*)

**close\_group**(*globals\_file, group\_name*)

**compile\_loop**()

**connect\_signals**()

**copy\_group**(*source\_globals\_file, source\_group\_name, dest\_globals\_file=None, delete\_source\_group=False*)

This function copies a group of globals with the name *source\_group\_name* from the file *source\_globals\_file* to a new file *dest\_globals\_file*. If *delete\_source\_group* is *True* the source group is deleted after copying

**delete\_group**(*globals\_file, group\_name, confirm=True*)

**do\_model\_sort**()

**get\_active\_groups**(*interactive=True*)

Returns active groups in the format {*group\_name*: *globals\_file*}. Displays an error dialog and returns *None* if multiple groups of the same name are selected, this is invalid - selected groups must be uniquely named. If *interactive=False*, raises the exception instead.

**get\_default\_output\_folder**()

Returns what the default output folder would be right now, based on the current date and selected labscript file. Returns empty string if no labscript file is selected. Does not create the default output folder, does not check if it exists.

**get\_group\_item\_by\_name**(*globals\_file, group\_name, column, previous\_name=None*)

Returns an item from the row representing a globals group in the groups model. Which item is returned is set by the *column* argument.

**get\_save\_data**()

**globals\_changed**()

Called from either self, a *GroupTab*, or the *RemoteServer* to inform runmanager that something about globals has changed, and that they need parsing again.

**guess\_expansion\_modes**(*active\_groups, evaled\_globals, global\_hierarchy, expansions*)

This function is designed to be called iteratively. It changes the expansion type of globals that reference other globals - such that globals referencing an iterable global will be zipped with it, rather than outer producted. Each time this method is called, `self.parse_globals` should also be called, so that the globals are evaluated with their new expansion modes, if they changed. This should be performed repeatedly until there are no more changes. Note that this method does not return what expansion types it thinks globals should have - it *actually writes them to the globals HDF5 file*. So it is up to later code to ensure it re-reads the expansion mode from the HDF5 file before proceeding. At present this method is only called from `self.prepare_globals()`, so see there to see how it fits in with everything else. This method uses four instance attributes to store state: `self.previous_evaluated_globals`, `self.previous_global_hierarchy`, `self.previous_expansion_types` and `self.previous_expansions`. This is necessary so that it can detect changes.

**load\_configuration**(*filename*)

**make\_group\_row**(*group\_name*)

Returns a new row representing one group in the groups tab, ready to be inserted into the model.

**make\_h5\_files**(*labscript\_file, output\_folder, sequence\_globals, shots, shuffle*)

**new\_group**(*globals\_file, group\_name*)

**on\_abort\_clicked**()

**on\_axes\_check\_selected\_triggered**(\*args)

**on\_axes\_item\_changed**(*item*)

**on\_axes\_uncheck\_selected\_triggered**(\*args)

**on\_axis\_down\_clicked**(*checked*)

**on\_axis\_to\_bottom\_clicked**(*checked*)

**on\_axis\_to\_top\_clicked**(*checked*)

**on\_axis\_up\_clicked**(*checked*)

**on\_close\_event**()

**on\_diff\_globals\_file\_clicked**()

**on\_edit\_labscript\_file\_clicked**(*checked*)

**on\_engage\_clicked**()

**on\_groups\_close\_selected\_files\_triggered**()

**on\_groups\_close\_selected\_groups\_triggered**()

**on\_groups\_copy\_selected\_groups\_triggered**(*dest\_globals\_file=None, delete\_source\_group=False*)

**on\_groups\_delete\_selected\_triggered**()

**on\_groups\_model\_active\_changed**(*item*)

Sets the sort data for the item in response to its check state changing. Also, if this is the first time this function has been called on the stack, that is, the change was initiated externally instead of via recursion from this function itself, then set the check state of other items for consistency. This entails checking/unchecking all group rows in response to the file row's check state changing, or changing the file row's check state to

reflect the check state of the child group rows. That's why we need to keep track of the recursion depth - so that those changes we make don't in turn cause further changes. But we don't disconnect the `on_changed` signal altogether, because we still want to do the update of the sort data, and anything else that might be added in future.

#### `on_groups_model_item_changed(item)`

This function is for responding to data changes in the model. The methods for responding to changes different columns do different things. Mostly they make other data changes for model consistency, but also group creation and renaming is handled in response to changes to the 'name' column. When we change things elsewhere, we prefer to only change one thing, and the rest of the changes are triggered here. So here we do the following:

Be careful not to recurse unsafely into this method - changing something that itself triggers further changes is fine so long as they peter out and don't get stuck in a loop. If recursion needs to be stopped, one can disconnect the signal temporarily with the context manager `self.groups_model_item_changed_disconnected`. But use this sparingly, otherwise there's the risk that some required data updates will be forgotten about and won't happen.

#### `on_groups_model_name_changed(item)`

Handles group renaming and creation of new groups due to the user editing the <click to add group> item

#### `on_groups_model_openclose_changed(item)`

Sets item sort data and icon in response to the open/close state of a group changing.

#### `on_groups_open_selected_triggered()`

#### `on_groups_set_selection_active_triggered(checked_state)`

#### `on_labscript_file_text_changed(text)`

#### `on_load_configuration_triggered()`

#### `on_master_shuffle_clicked(state)`

#### `on_new_globals_file_clicked()`

#### `on_open_globals_file_clicked()`

#### `on_output_popout_button_clicked()`

#### `on_reset_shot_output_folder_clicked(checked)`

#### `on_restart_subprocess_clicked()`

#### `on_revert_configuration_triggered()`

#### `on_save_configuration_as_triggered()`

#### `on_save_configuration_triggered()`

#### `on_select_labscript_file_clicked(checked)`

#### `on_select_shot_output_folder_clicked(checked)`

#### `on_shot_output_folder_text_changed(text)`

#### `on_tabCloseRequested(index)`

#### `on_treeView_axes_context_menu_requested(point)`

**on\_treeView\_groups\_context\_menu\_requested**(*point*)

**on\_treeView\_groups\_doubleLeftClicked**(*index*)

**on\_treeView\_groups\_leftClicked**(*index*)

Here we respond to user clicks on the treeview. We do the following:

- If the user clicks on the <click to add group> dummy row, we go into edit mode on it so they can enter the name of the new group they want.
- If the user clicks on the icon to open or close a globals file or a group, we call the appropriate open and close methods and update the open/close data role on the model.
- If the user clicks delete on a globals group, we call a delete method, which deletes it after confirmation, and closes it if it was open.

**open\_globals\_file**(*globals\_file*)

**open\_group**(*globals\_file, group\_name, trigger\_preparse=True*)

**parse\_globals**(*active\_groups, raise\_exceptions=True, expand\_globals=True, expansion\_order=None, return\_dimensions=False*)

**preparse\_globals**()

**preparse\_globals\_loop**()

Runs in a thread, waiting on a `threading.Event` that tells us when some globals have changed, and calls `parse_globals` to evaluate them all before feeding the results back to the relevant tabs to be displayed.

**rename\_group**(*globals\_file, previous\_group\_name, new\_group\_name*)

**rollover\_shot\_output\_folder**()

Runs in a thread, checking every 30 seconds if the default output folder has changed, likely because the date has changed, but also possible because another instance of runmanager has incremented the sequence index. If the default output folder has changed, and if runmanager is configured to use the default output folder, sets the folder in which compiled shots will be put. Does not create the folder if it does not already exist, this will be done at compile-time.

**save\_configuration**(*save\_file*)

**send\_to\_BLACS**(*run\_file, BLACS\_hostname*)

**send\_to\_runviewer**(*run\_file*)

**setup\_axes\_tab**()

**setup\_config**()

**setup\_groups\_tab**()

**switch\_tabs**(*change*)

**update\_axes\_indentation**()

**update\_axes\_tab**(*expansions, dimensions*)

**update\_global\_shuffle\_state**(\*args, \*\*kwargs)

**update\_tabs\_parsing\_indication**(*active\_groups, sequence\_globals, evaled\_globals, n\_shots*)

**wait\_until\_preparse\_complete**()

Block until the preparse loop has finished pending work

### 3.6.17 runmanager.\_\_main\_\_.RunmanagerMainWindow

**class** runmanager.\_\_main\_\_.RunmanagerMainWindow(\*args, \*\*kwargs)

Bases: QMainWindow

**\_\_init\_\_**(\*args, \*\*kwargs)

#### Methods

<code>__init__(*args, **kwargs)</code>
<code>acceptDrops(self)</code>
<code>accessibleDescription(self)</code>
<code>accessibleName(self)</code>
<code>actionEvent(self, a0)</code>
<code>actions(self)</code>
<code>activateWindow(self)</code>
<code>addAction(self, action)</code>
<code>addActions(self, actions)</code>
<code>addDockWidget()</code>
<code>addToolBar(-&gt; None)</code>
<code>addToolBarBreak(self[, area])</code>
<code>adjustSize(self)</code>
<code>autoFillBackground(self)</code>
<code>backgroundRole(self)</code>
<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>centralWidget(self)</code>
<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>

continues on next page

Table 3.12 – 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, 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>deleteLater(self)</code>
<code>depth(self)</code>
<code>destroy(self[, destroyWindow, destroySubWin- dows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>

continues on next page

Table 3.12 – continued from previous page

<code>disconnect(-&gt; bool)</code>
<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(-&gt; QObjectT)</code>
<code>findChildren(...)</code>
<code>focusInEvent(self, a0)</code>
<code>focusNextChild(self)</code>
<code>focusNextPrevChild(self, next)</code>
<code>focusOutEvent(self, a0)</code>
<code>focusPolicy(self)</code>
<code>focusPreviousChild(self)</code>

continues on next page

Table 3.12 – 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>
<code>hideEvent(self, a0)</code>

continues on next page

Table 3.12 – continued from previous page

<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>
<code>isSeparator(self, pos)</code>
<code>isSignalConnected(self, signal)</code>

continues on next page

Table 3.12 – continued from previous page

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

continues on next page

Table 3.12 – continued from previous page

<code>maximumWidth(self)</code>
<code>menuBar(self)</code>
<code>menuWidget(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>
<code>minimumHeight(self)</code>
<code>minimumSize(self)</code>
<code>minimumSizeHint(self)</code>
<code>minimumWidth(self)</code>
<code>mouseDoubleClickEvent(self, a0)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, a0)</code>
<code>mousePressEvent(self, a0)</code>
<code>mouseReleaseEvent(self, a0)</code>
<code>move()</code>
<code>moveEvent(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>
<code>nextInFocusChain(self)</code>
<code>normalGeometry(self)</code>
<code>objectName(self)</code>
<code>overrideWindowFlags(self, type)</code>
<code>overrideWindowState(self, state)</code>
<code>paintEngine(self)</code>
<code><i>paintEvent</i>(self, a0)</code>

continues on next page

Table 3.12 – continued from previous page

paintingActive(self)	
palette(self)	
parent(self)	
parentWidget(self)	
physicalDpiX(self)	
physicalDpiY(self)	
pos(self)	
previousInFocusChain(self)	
property(self, name)	
pyqtConfigure(...)	Each keyword argument is either the name of a Qt property or a Qt signal.
raise_(self)	
receivers(self, signal)	
rect(self)	
releaseKeyboard(self)	
releaseMouse(self)	
releaseShortcut(self, id)	
removeAction(self, action)	
removeDockWidget(self, dockwidget)	
removeEventFilter(self, a0)	
removeToolBar(self, toolbar)	
removeToolBarBreak(self, before)	
render(, sourceRegion, flags, ...)	
repaint(-> None -> None)	
resize()	
resizeDocks(self, docks, sizes, orientation)	
resizeEvent(self, a0)	

continues on next page

Table 3.12 – continued from previous page

<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>
<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>

continues on next page

Table 3.12 – continued from previous page

<code>setEnabled(self, a0)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>
<code>setFocus()</code>
<code>setFocusPolicy(self, policy)</code>
<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setForegroundRole(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setHidden(self, hidden)</code>
<code>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>
<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>

continues on next page

Table 3.12 – continued from previous page

<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>
<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>

continues on next page

Table 3.12 – continued from previous page

---

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

---

continues on next page

Table 3.12 – continued from previous page

---

<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>
<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(-&gt; None -&gt; None)</code>
<code>updateGeometry(self)</code>
<code>updateMicroFocus(self)</code>

---

continues on next page

Table 3.12 – continued from previous page

---

updatesEnabled(self)
visibleRegion(self)
whatsThis(self)
wheelEvent(self, a0)
width(self)
widthMM(self)
winId(self)
window(self)
windowFilePath(self)
windowFlags(self)
windowHandle(self)
windowIcon(self)
windowIconText(self)
windowModality(self)
windowOpacity(self)
windowRole(self)
windowState(self)
windowTitle(self)
windowType(self)
x(self)
y(self)

---

## Attributes

AllowNestedDocks	
AllowTabbedDocks	
AnimatedDocks	
DrawChildren	
DrawWindowBackground	
ForceTabbedDocks	
GroupedDragging	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
VerticalTabs	
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
<i>firstPaint</i>	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
iconSizeChanged	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

continues on next page

Table 3.13 – continued from previous page

objectNameChanged	int = ..., arguments: Sequence = ...)	-> PYQT_SIGNAL
staticMetaObject		
tabifiedDockWidgetActivated	int = ..., arguments: Sequence = ...)	-> PYQT_SIGNAL
toolButtonStyleChanged	int = ..., arguments: Sequence = ...)	-> PYQT_SIGNAL
windowIconChanged	int = ..., arguments: Sequence = ...)	-> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments: Sequence = ...)	-> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments: Sequence = ...)	-> PYQT_SIGNAL

**closeEvent**(*self*, *a0*: *QCloseEvent* | *None*)

**firstPaint**

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

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

**Type**

pyqtSignal(\*types, name

**Type**

str = ..., revision

**paintEvent**(*self*, *a0*: *QPaintEvent* | *None*)

### 3.6.18 runmanager.\_\_main\_\_.TabToolButton

**class** runmanager.\_\_main\_\_.TabToolButton(\*args, \*\*kwargs)

Bases: *QToolButton*

**\_\_init\_\_**(\*args, \*\*kwargs)

#### Methods

**\_\_init\_\_**(\*args, \*\*kwargs)

acceptDrops(self)

accessibleDescription(self)

continues on next page

Table 3.14 – continued from previous page

<code>accessibleName(self)</code>
<code>actionEvent(self, a0)</code>
<code>actions(self)</code>
<code>activateWindow(self)</code>
<code>addAction(self, action)</code>
<code>addActions(self, actions)</code>
<code>adjustSize(self)</code>
<code>animateClick(self[, msec])</code>
<code>arrowType(self)</code>
<code>autoExclusive(self)</code>
<code>autoFillBackground(self)</code>
<code>autoRaise(self)</code>
<code>autoRepeat(self)</code>
<code>autoRepeatDelay(self)</code>
<code>autoRepeatInterval(self)</code>
<code>backgroundRole(self)</code>
<code>baseSize(self)</code>
<code>blockSignals(self, b)</code>
<code>changeEvent(self, a0)</code>
<code>checkStateSet(self)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>

continues on next page

Table 3.14 – continued from previous page

<code>clearMask(self)</code>
<code>click(self)</code>
<code>close(self)</code>
<code>closeEvent(self, a0)</code>
<code>colorCount(self)</code>
<code>connectNotify(self, signal)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, a0)</code>
<code>contextMenuPolicy(self)</code>
<code>create(self[, window, initializeWindow, ...])</code>
<code>createWindowContainer(window[, parent, flags])</code>
<code>cursor(self)</code>
<code>customEvent(self, a0)</code>
<code>defaultAction(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>disconnect(-&gt; bool)</code>
<code>disconnectNotify(self, signal)</code>
<code>dragEnterEvent(self, a0)</code>
<code>dragLeaveEvent(self, a0)</code>

continues on next page

Table 3.14 – continued from previous page

<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, e)</code>
<code>eventFilter(self, a0, a1)</code>
<code>find(a0)</code>
<code>findChild(-&gt; QObjectT)</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>
<code>foregroundRole(self)</code>
<code>frameGeometry(self)</code>

continues on next page

Table 3.14 – continued from previous page

<code>frameSize(self)</code>
<code>geometry(self)</code>
<code>getContentsMargins(self)</code>
<code><i>get_correct_position()</i></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>group(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>hitButton(self, pos)</code>
<code>icon(self)</code>
<code>iconSize(self)</code>
<code>inherits(self, classname)</code>
<code>initPainter(self, painter)</code>

continues on next page

Table 3.14 – continued from previous page

<code>initStyleOption(self, option)</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>isCheckedable(self)</code>
<code>isChecked(self)</code>
<code>isDown(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>
<code>isSignalConnected(self, signal)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, a0)</code>
<code>isWidgetType(self)</code>
<code>isWindow(self)</code>

continues on next page

Table 3.14 – continued from previous page

<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>keyPressEvent(self, e)</code>
<code>keyReleaseEvent(self, e)</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>menu(self)</code>
<code>metaObject(self)</code>
<code>metric(self, a0)</code>

continues on next page

Table 3.14 – continued from previous page

<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, e)</code>
<code>mousePressEvent(self, a0)</code>
<code>mouseReleaseEvent(self, a0)</code>
<code>move()</code>
<code><i>moveEvent</i>(self, a0)</code>
<code>moveToThread(self, thread)</code>
<code>nativeEvent(self, eventType, message)</code>
<code>nativeParentWidget(self)</code>
<code>nextCheckState(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><i>paintEvent</i>(self, a0)</code>
<code>paintingActive(self)</code>
<code>palette(self)</code>
<code>parent(self)</code>
<code>parentWidget(self)</code>

continues on next page

Table 3.14 – continued from previous page

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

continues on next page

Table 3.14 – continued from previous page

<code>setAccessibleDescription(self, description)</code>
<code>setAccessibleName(self, name)</code>
<code>setArrowType(self, type)</code>
<code>setAttribute(self, attribute[, on])</code>
<code>setAutoExclusive(self, a0)</code>
<code>setAutoFillBackground(self, enabled)</code>
<code>setAutoRaise(self, enable)</code>
<code>setAutoRepeat(self, a0)</code>
<code>setAutoRepeatDelay(self, a0)</code>
<code>setAutoRepeatInterval(self, a0)</code>
<code>setBackgroundRole(self, a0)</code>
<code>setBaseSize()</code>
<code>setCheckable(self, a0)</code>
<code>setChecked(self, a0)</code>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCursor(self, a0)</code>
<code>setDefaultAction(self, a0)</code>
<code>setDisabled(self, a0)</code>
<code>setDown(self, a0)</code>
<code>setEnabled(self, a0)</code>
<code>setFixedHeight(self, h)</code>
<code>setFixedSize()</code>
<code>setFixedWidth(self, w)</code>
<code>setFocus()</code>
<code>setFocusPolicy(self, policy)</code>

continues on next page

Table 3.14 – continued from previous page

---

<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setForegroundColor(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>
<code>setHidden(self, hidden)</code>
<code>setIcon(self, icon)</code>
<code>setIconSize(self, size)</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>setMenu(self, menu)</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>setPopupMode(self, mode)</code>
<code>setProperty(self, name, value)</code>

---

continues on next page

Table 3.14 – continued from previous page

<code>setShortcut(self, key)</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>setText(self, text)</code>
<code>setToolButtonStyle(self, style)</code>
<code>setToolTip(self, a0)</code>
<code>setToolTipDuration(self, msec)</code>
<code>setUpdatesEnabled(self, enable)</code>
<code>setVisible(self, visible)</code>
<code>setWhatsThis(self, a0)</code>
<code>setWindowFilePath(self, filePath)</code>
<code>setWindowFlag(self, a0[, on])</code>
<code>setWindowFlags(self, type)</code>
<code>setWindowIcon(self, icon)</code>
<code>setWindowIconText(self, a0)</code>
<code>setWindowModality(self, windowModality)</code>
<code>setWindowModified(self, a0)</code>
<code>setWindowOpacity(self, level)</code>
<code>setWindowRole(self, a0)</code>

continues on next page

Table 3.14 – continued from previous page

---

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

---

continues on next page

Table 3.14 – continued from previous page

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

continues on next page

Table 3.14 – continued from previous page

<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

DelayedPopup			
DrawChildren			
DrawWindowBackground			
IgnoreMask			
InstantPopup			
MenuButtonPopup			
PdmDepth			
PdmDevicePixelRatio			
PdmDevicePixelRatioScaled			
PdmDpiX			
PdmDpiY			
PdmHeight			
PdmHeightMM			
PdmNumColors			
PdmPhysicalDpiX			
PdmPhysicalDpiY			
PdmWidth			
PdmWidthMM			
clicked	int = ..., arguments:	Sequence = ...)	->
customContextMenuRequested	int = ..., arguments:	Sequence = ...)	->
destroyed	int = ..., arguments:	Sequence = ...)	->
objectNameChanged	int = ..., arguments:	Sequence = ...)	->
pressed	int = ..., arguments:	Sequence = ...)	->
released	int = ..., arguments:	Sequence = ...)	->
staticMetaObject			
toggled	int = ..., arguments:	Sequence = ...)	->
triggered	int = ..., arguments:	Sequence = ...)	->
<b>3.6. runmanager.__main__</b>			<b>161</b>
windowIconChanged	int = ..., arguments:	Sequence = ...)	->
windowIconTextChanged	int = ..., arguments:	Sequence = ...)	->

```
get_correct_position()
moveEvent(self, a0: QMoveEvent | None)
paintEvent(self, a0: QPaintEvent | None)
```

### 3.6.19 runmanager.\_\_main\_\_.TableView

```
class runmanager.__main__.TableView(parent=None)
```

Bases: *ItemView*, *QTableView*

TableView version of our customised ItemView

```
__init__(parent=None)
```

#### Methods

```
__init__([parent])
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)
```

continues on next page

Table 3.15 – continued from previous page

<code>changeEvent(self, a0)</code>
<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>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>

continues on next page

Table 3.15 – continued from previous page

---

<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, destroySubWin- dows])</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>dirtyRegionOffset(self)</code>
<code>disconnect(-&gt; 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>

---

continues on next page

Table 3.15 – continued from previous page

<code>dragMoveEvent(self, e)</code>
<code>drawFrame(self, a0)</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>
<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(-&gt; QObjectT)</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>

continues on next page

Table 3.15 – continued from previous page

---

<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>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>

---

continues on next page

Table 3.15 – continued from previous page

<code>hasTabletTracking(self)</code>
<code>height(self)</code>
<code>heightForWidth(self, a0)</code>
<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>indexOfAt(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>

continues on next page

Table 3.15 – continued from previous page

<code>installEventFilter(self, a0)</code>	
<code>isActiveWindow(self)</code>	
<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(-&gt; Optional[QAbstractItemDelegate])</code>	Op-

continues on next page

Table 3.15 – continued from previous page

<code>itemDelegateForColumn(self, column)</code>
<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>leaveEvent(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 3.15 – 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>mouseDoubleClickEvent(self, e)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, e)</code>
<code>mousePressEvent(self, e)</code>
<code>mouseReleaseEvent(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><i>on_column_resized</i>(col)</code>
<code>openPersistentEditor(self, index)</code>
<code>overrideWindowFlags(self, type)</code>
<code>overrideWindowState(self, state)</code>

continues on next page

Table 3.15 – 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(-&gt; None -&gt; None)</code>	
<code>reset(self)</code>	
<code>resetHorizontalScrollMode(self)</code>	
<code>resetVerticalScrollMode(self)</code>	
<code>resize()</code>	

continues on next page

Table 3.15 – continued from previous page

---

<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>
<code>scrollToBottom(self)</code>
<code>scrollToTop(self)</code>

---

continues on next page

Table 3.15 – continued from previous page

<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>
<code>setContentsMargins()</code>
<code>setContextMenuPolicy(self, policy)</code>
<code>setCornerButtonEnabled(self, enable)</code>

continues on next page

Table 3.15 – continued from previous page

---

<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>
<code>setFrameShape(self, a0)</code>
<code>setFrameStyle(self, a0)</code>
<code>setGeometry()</code>
<code>setGraphicsEffect(self, effect)</code>

---

continues on next page

Table 3.15 – continued from previous page

<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>
<code>setMinimumHeight(self, minh)</code>
<code>setMinimumSize()</code>
<code>setMinimumWidth(self, minw)</code>
<code>setModel(self, model)</code>
<code>setMouseTracking(self, enable)</code>

continues on next page

Table 3.15 – continued from previous page

---

<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>
<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>

---

continues on next page

Table 3.15 – continued from previous page

<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>
<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>

continues on next page

Table 3.15 – continued from previous page

<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>
<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>

continues on next page

Table 3.15 – continued from previous page

<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>
<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>

continues on next page

Table 3.15 – continued from previous page

---

<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>
<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>

---

continues on next page

Table 3.15 – continued from previous page

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

### Attributes

AboveItem
AdjustIgnored
AdjustToContents
AdjustToContentsOnFirstShow
AllEditTriggers
AnimatingState
AnyKeyPressed
BelowItem
Box
COLOR_HIGHLIGHT
CollapsingState
ContiguousSelection
CurrentChanged
DoubleClicked
DragDrop
DragOnly
DragSelectingState
DraggingState

continues on next page

Table 3.16 – continued from previous page

DrawChildren
DrawWindowBackground
DropOnly
EditKeyPressed
EditingState
EnsureVisible
ExpandingState
ExtendedSelection
HLine
IgnoreMask
InternalMove
MoveDown
MoveEnd
MoveHome
MoveLeft
MoveNext
MovePageDown
MovePageUp
MovePrevious
MoveRight
MoveUp
MultiSelection
NoDragDrop
NoEditTriggers
NoFrame
NoSelection

continues on next page

Table 3.16 – continued from previous page

NoState
OnItem
OnViewport
Panel
PdmDepth
PdmDevicePixelRatio
PdmDevicePixelRatioScaled
PdmDpiX
PdmDpiY
PdmHeight
PdmHeightMM
PdmNumColors
PdmPhysicalDpiX
PdmPhysicalDpiY
PdmWidth
PdmWidthMM
Plain
PositionAtBottom
PositionAtCenter
PositionAtTop
Raised
ScrollPerItem
ScrollPerPixel
SelectColumns
SelectItems
SelectRows

continues on next page

Table 3.16 – continued from previous page

SelectedClicked			
Shadow_Mask			
Shape_Mask			
SingleSelection			
StyledPanel			
Sunken			
VLine			
WinPanel			
activated	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
clicked	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
customContextMenuRequested	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
destroyed	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
doubleClicked	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
doubleLeftClicked	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
entered	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
iconSizeChanged	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
leftClicked	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
objectNameChanged	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
pressed	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
staticMetaObject			
viewportEntered	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
windowIconChanged	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
windowIconTextChanged	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL
windowTitleChanged	int = ..., arguments:	Sequence = ...)	-> PYQT_SIGNAL

`on_column_resized(col)`

### 3.6.20 runmanager.\_\_main\_\_.TreeView

**class** runmanager.\_\_main\_\_.TreeView(*parent=None*)

Bases: *ItemView*, *QTreeView*

Treeview version of our customised ItemView

**\_\_init\_\_**(*parent=None*)

#### Methods

<code>__init__([parent])</code>
<code>acceptDrops(self)</code>
<code>accessibleDescription(self)</code>
<code>accessibleName(self)</code>
<code>actionEvent(self, a0)</code>
<code>actions(self)</code>
<code>activateWindow(self)</code>
<code>addAction(self, action)</code>
<code>addActions(self, actions)</code>
<code>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>

continues on next page

Table 3.17 – continued from previous page

<code>childAt(-&gt; Optional[QWidget])</code>
<code>childEvent(self, a0)</code>
<code>children(self)</code>
<code>childrenRect(self)</code>
<code>childrenRegion(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>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>
<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>

continues on next page

Table 3.17 – continued from previous page

<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>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>dirtyRegionOffset(self)</code>
<code>disconnect(-&gt; 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>

continues on next page

Table 3.17 – continued from previous page

<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>
<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(-&gt; QObjectT)</code>
<code>findChildren(...)</code>

continues on next page

Table 3.17 – continued from previous page

<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>
<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>

continues on next page

Table 3.17 – continued from previous page

<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>
<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>

continues on next page

Table 3.17 – continued from previous page

<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>
<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>

continues on next page

Table 3.17 – continued from previous page

<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>	
<code>isWindow(self)</code>	
<code>isWindowModified(self)</code>	
<code>isWindowType(self)</code>	
<code>itemDelegate(-&gt; Optional[QAbstractItemDelegate])</code>	Op-
<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>leaveEvent(self, a0)</code>	
<code>lineWidth(self)</code>	
<code>locale(self)</code>	
<code>logicalDpiX(self)</code>	

continues on next page

Table 3.17 – continued from previous page

<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>
<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>mouseDoubleClickEvent(self, e)</code>
<code>mouseGrabber()</code>
<code>mouseMoveEvent(self, event)</code>
<code>mousePressEvent(self, e)</code>
<code>mouseReleaseEvent(self, event)</code>

continues on next page

Table 3.17 – continued from previous page

<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>	
<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>	

continues on next page

Table 3.17 – continued from previous page

<code>rect(self)</code>
<code>reexpand(self)</code>
<code>releaseKeyboard(self)</code>
<code>releaseMouse(self)</code>
<code>releaseShortcut(self, id)</code>
<code>removeAction(self, action)</code>
<code>removeEventFilter(self, a0)</code>
<code>render(, sourceRegion, flags, ...)</code>
<code>repaint(-&gt; None -&gt; None)</code>
<code>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>

continues on next page

Table 3.17 – continued from previous page

<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>
<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>

continues on next page

Table 3.17 – continued from previous page

<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>
<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>

continues on next page

Table 3.17 – continued from previous page

<code>setFocusPolicy(self, policy)</code>
<code>setFocusProxy(self, a0)</code>
<code>setFont(self, a0)</code>
<code>setForegroundRole(self, a0)</code>
<code>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>

continues on next page

Table 3.17 – continued from previous page

<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>
<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 3.17 – 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>
<code>setWindowFlags(self, type)</code>

continues on next page

Table 3.17 – continued from previous page

<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>
<code>sizeHintForIndex(self, index)</code>
<code>sizeHintForRow(self, row)</code>

continues on next page

Table 3.17 – continued from previous page

<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>
<code>unsetCursor(self)</code>
<code>unsetLayoutDirection(self)</code>
<code>unsetLocale(self)</code>

continues on next page

Table 3.17 – continued from previous page

update()
updateEditorData(self)
updateEditorGeometries(self)
updateGeometries(self)
updateGeometry(self)
updateMicroFocus(self)
updatesEnabled(self)
verticalOffset(self)
verticalScrollBar(self)
verticalScrollBarPolicy(self)
verticalScrollMode(self)
verticalScrollbarAction(self, action)
verticalScrollbarValueChanged(self, value)
viewOptions(self)
viewport(self)
viewportEvent(self, event)
viewportMargins(self)
viewportSizeHint(self)
visibleRegion(self)
visualRect(self, index)
visualRegionForSelection(self, selection)
whatsThis(self)
wheelEvent(self, a0)
width(self)
widthMM(self)
winId(self)

continues on next page

Table 3.17 – 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 3.18 – continued from previous page

Box
COLOR_HIGHLIGHT
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 3.18 – 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
PdmWidthMM

continues on next page

Table 3.18 – continued from previous page

Plain	
PositionAtBottom	
PositionAtCenter	
PositionAtTop	
Raised	
ScrollPerItem	
ScrollPerPixel	
SelectColumns	
SelectItems	
SelectRows	
SelectedClicked	
Shadow_Mask	
Shape_Mask	
SingleSelection	
StyledPanel	
Sunken	
VLine	
WinPanel	
activated	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
clicked	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
collapsed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
customContextMenuRequested	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
destroyed	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
doubleClicked	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
doubleLeftClicked	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL
entered	int = ..., arguments: Sequence = ...) -> PYQT_SIGNAL

continues on next page

Table 3.18 – continued from previous page

expanded	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
iconSizeChanged	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
leftClicked	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
objectNameChanged	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
pressed	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
staticMetaObject		
viewportEntered	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
windowIconChanged	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
windowIconTextChanged	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	
windowTitleChanged	int = ..., arguments: Sequence = ...)	->
	PYQT_SIGNAL	

## LABSCRIPT SUITE COMPONENTS

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

Table 4.1: Python libraries

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

Table 4.2: Graphical applications

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



## PYTHON MODULE INDEX

### r

runmanager, 17  
runmanager.\_\_main\_\_, 35  
runmanager.batch\_compiler, 35  
runmanager.functions, 28  
runmanager.globals\_diff, 35  
runmanager.remote, 28



## Symbols

- `__init__()` (*runmanager.TraceDictionary* method), 26
  - `__init__()` (*runmanager.\_\_main\_\_.AlternatingColorModel* method), 37
  - `__init__()` (*runmanager.\_\_main\_\_.Editor* method), 45
  - `__init__()` (*runmanager.\_\_main\_\_.FingerTabBarWidget* method), 64
  - `__init__()` (*runmanager.\_\_main\_\_.FingerTabWidget* method), 80
  - `__init__()` (*runmanager.\_\_main\_\_.GroupTab* method), 96
  - `__init__()` (*runmanager.\_\_main\_\_.ItemDelegate* method), 100
  - `__init__()` (*runmanager.\_\_main\_\_.ItemView* method), 104
  - `__init__()` (*runmanager.\_\_main\_\_.PoppedOutOutputBoxWindow* method), 105
  - `__init__()` (*runmanager.\_\_main\_\_.RemoteServer* method), 120
  - `__init__()` (*runmanager.\_\_main\_\_.RunManager* method), 122
  - `__init__()` (*runmanager.\_\_main\_\_.RunmanagerMainWindow* method), 131
  - `__init__()` (*runmanager.\_\_main\_\_.TabToolButton* method), 146
  - `__init__()` (*runmanager.\_\_main\_\_.TableView* method), 162
  - `__init__()` (*runmanager.\_\_main\_\_.TreeView* method), 185
  - `__init__()` (*runmanager.batch\_compiler.BatchProcessor* method), 35
  - `__init__()` (*runmanager.remote.Client* method), 32
- ## A
- `abort()` (*in module runmanager.remote*), 29
  - `abort()` (*runmanager.remote.Client* method), 33
  - `add_expansion_groups()` (*in module runmanager*), 19
  - `add_item_to_axes_model()` (*runmanager.\_\_main\_\_.RunManager* method), 127
  - `addTab()` (*runmanager.\_\_main\_\_.FingerTabWidget* method), 95
- ## AlternatingColorModel
- (*class in runmanager.\_\_main\_\_*), 37
- ## AXES\_COL\_LENGTH
- (*runmanager.\_\_main\_\_.RunManager* attribute), 126
- ## AXES\_COL\_NAME
- (*runmanager.\_\_main\_\_.RunManager* attribute), 126
- ## AXES\_COL\_SHUFFLE
- (*runmanager.\_\_main\_\_.RunManager* attribute), 126
- ## AXES\_ROLE\_NAME
- (*runmanager.\_\_main\_\_.RunManager* attribute), 126
- ## B
- ## BatchProcessor
- (*class in runmanager.batch\_compiler*), 35
- ## C
- ## change\_global\_expansion()
- (*runmanager.\_\_main\_\_.GroupTab* method), 99
- ## change\_global\_units()
- (*runmanager.\_\_main\_\_.GroupTab* method), 99
- ## change\_global\_value()
- (*runmanager.\_\_main\_\_.GroupTab* method), 99
- ## check\_child\_exited()
- (*runmanager.\_\_main\_\_.RunManager* method), 127
- ## check\_for\_boolean\_values()
- (*runmanager.\_\_main\_\_.GroupTab* method), 99
- ## check\_output\_folder\_update()
- (*runmanager.\_\_main\_\_.RunManager* method), 127
- ## Client
- (*class in runmanager.remote*), 32
- ## close()
- (*runmanager.\_\_main\_\_.GroupTab* method), 99
- ## close\_current\_tab()
- (*runmanager.\_\_main\_\_.RunManager* method), 127
- ## close\_globals\_file()
- (*runmanager.\_\_main\_\_.RunManager* method), 127
- ## close\_group()
- (*runmanager.\_\_main\_\_.RunManager* method), 127
- ## closeEvent()
- (*runmanager.\_\_main\_\_.PoppedOutOutputBoxWindow* method), 120
- ## closeEvent()
- (*runmanager.\_\_main\_\_.RunmanagerMainWindow* method), 146

COLOR\_BOOL\_OFF (*runmanager.\_\_main\_\_.GroupTab* attribute), 98  
 COLOR\_BOOL\_ON (*runmanager.\_\_main\_\_.GroupTab* attribute), 98  
 COLOR\_ERROR (*runmanager.\_\_main\_\_.GroupTab* attribute), 98  
 COLOR\_HIGHLIGHT (*runmanager.\_\_main\_\_.ItemView* attribute), 104  
 COLOR\_OK (*runmanager.\_\_main\_\_.GroupTab* attribute), 98  
 compile() (*runmanager.batch\_compiler.BatchProcessor* method), 35  
 compile\_labscript() (in module *runmanager*), 19  
 compile\_labscript\_async() (in module *runmanager*), 19  
 compile\_labscript\_with\_globals\_files() (in module *runmanager*), 19  
 compile\_labscript\_with\_globals\_files\_async() (in module *runmanager*), 20  
 compile\_loop() (*runmanager.\_\_main\_\_.RunManager* method), 127  
 compile\_multishot\_async() (in module *runmanager*), 20  
 complete\_change\_global\_value() (*runmanager.\_\_main\_\_.GroupTab* method), 99  
 composite\_colors() (in module *runmanager.\_\_main\_\_*), 36  
 connect\_signals() (*runmanager.\_\_main\_\_.GroupTab* method), 99  
 connect\_signals() (*runmanager.\_\_main\_\_.RunManager* method), 127  
 copy\_group() (in module *runmanager*), 20  
 copy\_group() (*runmanager.\_\_main\_\_.RunManager* method), 127  
 createEditor() (*runmanager.\_\_main\_\_.ItemDelegate* method), 103

## D

data() (*runmanager.\_\_main\_\_.AlternatingColorModel* method), 44  
 delete\_global() (in module *runmanager*), 20  
 delete\_global() (*runmanager.\_\_main\_\_.GroupTab* method), 99  
 delete\_group() (in module *runmanager*), 20  
 delete\_group() (*runmanager.\_\_main\_\_.RunManager* method), 127  
 dict\_diff() (in module *runmanager*), 21  
 do\_model\_sort() (*runmanager.\_\_main\_\_.GroupTab* method), 99  
 do\_model\_sort() (*runmanager.\_\_main\_\_.RunManager* method), 127  
 doubleLeftClicked (*runmanager.\_\_main\_\_.ItemView* attribute), 104  
 drop\_times() (in module *runmanager.functions*), 28

## E

Editor (class in *runmanager.\_\_main\_\_*), 45  
 engage() (in module *runmanager.remote*), 29  
 engage() (*runmanager.remote.Client* method), 33  
 error\_dialog() (in module *runmanager.\_\_main\_\_*), 36  
 error\_in\_globals() (in module *runmanager.remote*), 29  
 error\_in\_globals() (*runmanager.remote.Client* method), 33  
 evaluate\_globals() (in module *runmanager*), 21  
 eventFilter() (*runmanager.\_\_main\_\_.ItemDelegate* method), 103  
 expand\_globals() (in module *runmanager*), 21  
 ExpansionError, 28  
 EXTRA\_COL\_WIDTH (*runmanager.\_\_main\_\_.ItemDelegate* attribute), 103  
 EXTRA\_ROW\_HEIGHT (*runmanager.\_\_main\_\_.ItemDelegate* attribute), 103

## F

find\_comments() (in module *runmanager*), 21  
 FingerTabBarWidget (class in *runmanager.\_\_main\_\_*), 64  
 FingerTabWidget (class in *runmanager.\_\_main\_\_*), 80  
 first() (in module *runmanager.functions*), 28  
 firstPaint (*runmanager.\_\_main\_\_.RunManagerMainWindow* attribute), 146  
 flatten\_globals() (in module *runmanager*), 21

## G

get\_active\_groups() (*runmanager.\_\_main\_\_.RunManager* method), 127  
 get\_all\_groups() (in module *runmanager*), 21  
 get\_bgbrush() (*runmanager.\_\_main\_\_.AlternatingColorModel* method), 45  
 get\_correct\_position() (*runmanager.\_\_main\_\_.TabToolButton* method), 162  
 get\_default\_output\_folder() (*runmanager.\_\_main\_\_.RunManager* method), 127  
 get\_expansion() (in module *runmanager*), 22  
 get\_global\_item\_by\_name() (*runmanager.\_\_main\_\_.GroupTab* method), 99  
 get\_globals() (in module *runmanager*), 22  
 get\_globals() (in module *runmanager.remote*), 30  
 get\_globals() (*runmanager.remote.Client* method), 33  
 get\_globalslist() (in module *runmanager*), 22  
 get\_group\_item\_by\_name() (*runmanager.\_\_main\_\_.RunManager* method), 127  
 get\_grouplist() (in module *runmanager*), 22  
 get\_labscript\_file() (in module *runmanager.remote*), 30

- get\_labscript\_file() (*runmanager.remote.Client method*), 33  
 get\_run\_shots() (*in module runmanager.remote*), 30  
 get\_run\_shots() (*runmanager.remote.Client method*), 34  
 get\_save\_data() (*runmanager.\_\_main\_\_.RunManager method*), 127  
 get\_shot\_globals() (*in module runmanager*), 22  
 get\_shot\_output\_folder() (*in module runmanager.remote*), 30  
 get\_shot\_output\_folder() (*runmanager.remote.Client method*), 34  
 get\_shuffle() (*in module runmanager.remote*), 30  
 get\_shuffle() (*runmanager.remote.Client method*), 34  
 get\_units() (*in module runmanager*), 22  
 get\_value() (*in module runmanager*), 22  
 get\_version() (*in module runmanager.remote*), 30  
 get\_version() (*runmanager.remote.Client method*), 34  
 get\_view\_shots() (*in module runmanager.remote*), 30  
 get\_view\_shots() (*runmanager.remote.Client method*), 34  
 globals\_changed() (*runmanager.\_\_main\_\_.GroupTab method*), 99  
 globals\_changed() (*runmanager.\_\_main\_\_.RunManager method*), 127  
 GLOBALS\_COL\_DELETE (*runmanager.\_\_main\_\_.GroupTab attribute*), 98  
 GLOBALS\_COL\_EXPANSION (*runmanager.\_\_main\_\_.GroupTab attribute*), 98  
 GLOBALS\_COL\_NAME (*runmanager.\_\_main\_\_.GroupTab attribute*), 98  
 GLOBALS\_COL\_UNITS (*runmanager.\_\_main\_\_.GroupTab attribute*), 98  
 GLOBALS\_COL\_VALUE (*runmanager.\_\_main\_\_.GroupTab attribute*), 98  
 globals\_diff\_groups() (*in module runmanager*), 23  
 globals\_diff\_shots() (*in module runmanager*), 23  
 GLOBALS\_DUMMY\_ROW\_TEXT (*runmanager.\_\_main\_\_.GroupTab attribute*), 98  
 GLOBALS\_ROLE\_IS\_BOOL (*runmanager.\_\_main\_\_.GroupTab attribute*), 98  
 GLOBALS\_ROLE\_IS\_DUMMY\_ROW (*runmanager.\_\_main\_\_.GroupTab attribute*), 98  
 GLOBALS\_ROLE\_PREVIOUS\_TEXT (*runmanager.\_\_main\_\_.GroupTab attribute*), 98  
 GLOBALS\_ROLE\_SORT\_DATA (*runmanager.\_\_main\_\_.GroupTab attribute*), 99  
 GROUPS\_COL\_ACTIVE (*runmanager.\_\_main\_\_.RunManager attribute*), 126  
 GROUPS\_COL\_DELETE (*runmanager.\_\_main\_\_.RunManager attribute*), 126  
 GROUPS\_COL\_NAME (*runmanager.\_\_main\_\_.RunManager attribute*), 126  
 GROUPS\_COL\_OPENCLOSE (*runmanager.\_\_main\_\_.RunManager attribute*), 126  
 GROUPS\_DUMMY\_ROW\_TEXT (*runmanager.\_\_main\_\_.RunManager attribute*), 126  
 GROUPS\_ROLE\_GROUP\_IS\_OPEN (*runmanager.\_\_main\_\_.RunManager attribute*), 127  
 GROUPS\_ROLE\_IS\_DUMMY\_ROW (*runmanager.\_\_main\_\_.RunManager attribute*), 127  
 GROUPS\_ROLE\_PREVIOUS\_NAME (*runmanager.\_\_main\_\_.RunManager attribute*), 127  
 GROUPS\_ROLE\_SORT\_DATA (*runmanager.\_\_main\_\_.RunManager attribute*), 127  
 GroupTab (*class in runmanager.\_\_main\_\_*), 96  
 guess\_expansion\_modes() (*runmanager.\_\_main\_\_.RunManager method*), 127  
 guess\_expansion\_type() (*in module runmanager*), 23
- ## H
- handle\_abort() (*runmanager.\_\_main\_\_.RemoteServer method*), 121  
 handle\_engage() (*runmanager.\_\_main\_\_.RemoteServer method*), 121  
 handle\_error\_in\_globals() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_get\_globals() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_get\_labscript\_file() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_get\_run\_shots() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_get\_shot\_output\_folder() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_get\_shuffle() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_get\_view\_shots() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_is\_output\_folder\_default() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_n\_shots() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_reset\_shot\_output\_folder() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_set\_globals() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_set\_labscript\_file() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_set\_run\_shots() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_set\_shot\_output\_folder() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_set\_shuffle() (*runmanager.\_\_main\_\_.RemoteServer method*), 122  
 handle\_set\_view\_shots() (*runmanager.\_\_main\_\_.RemoteServer method*), 122

- handler() (*runmanager.\_\_main\_\_.RemoteServer method*), 122
- I
- indexOfPos() (*runmanager.\_\_main\_\_.FingerTabBarWidget method*), 80
- is\_output\_folder\_default() (*in module runmanager.remote*), 31
- is\_output\_folder\_default() (*runmanager.remote.Client method*), 34
- is\_valid\_hdf5\_group\_name() (*in module runmanager*), 23
- is\_valid\_python\_identifier() (*in module runmanager*), 23
- isMovable() (*runmanager.\_\_main\_\_.FingerTabBarWidget method*), 80
- ItemDelegate (*class in runmanager.\_\_main\_\_*), 100
- ItemView (*class in runmanager.\_\_main\_\_*), 104
- iterator\_to\_tuple() (*in module runmanager*), 23
- K
- keyPressEvent() (*runmanager.\_\_main\_\_.ItemView method*), 105
- L
- leaveEvent() (*runmanager.\_\_main\_\_.ItemView method*), 105
- leftClicked (*runmanager.\_\_main\_\_.ItemView attribute*), 105
- load\_configuration() (*runmanager.\_\_main\_\_.RunManager method*), 128
- log\_if\_global() (*in module runmanager.\_\_main\_\_*), 36
- M
- mainloop() (*runmanager.batch\_compiler.BatchProcessor method*), 35
- make\_global\_row() (*runmanager.\_\_main\_\_.GroupTab method*), 99
- make\_group\_row() (*runmanager.\_\_main\_\_.RunManager method*), 128
- make\_h5\_files() (*runmanager.\_\_main\_\_.RunManager method*), 128
- make\_run\_file\_from\_globals\_files() (*in module runmanager*), 24
- make\_run\_files() (*in module runmanager*), 24
- make\_single\_run\_file() (*in module runmanager*), 24
- MIN\_ROW\_HEIGHT (*runmanager.\_\_main\_\_.ItemDelegate attribute*), 103
- module
- runmanager, 17
  - runmanager.\_\_main\_\_, 35
  - runmanager.batch\_compiler, 35
  - runmanager.functions, 28
  - runmanager.globals\_diff, 35
  - runmanager.remote, 28
- mouseDoubleClickEvent() (*runmanager.\_\_main\_\_.ItemView method*), 105
- mousePressEvent() (*runmanager.\_\_main\_\_.FingerTabBarWidget method*), 80
- mousePressEvent() (*runmanager.\_\_main\_\_.ItemView method*), 105
- mouseReleaseEvent() (*runmanager.\_\_main\_\_.FingerTabBarWidget method*), 80
- mouseReleaseEvent() (*runmanager.\_\_main\_\_.ItemView method*), 105
- moveCursor() (*runmanager.\_\_main\_\_.ItemView method*), 105
- moveEvent() (*runmanager.\_\_main\_\_.TabToolButton method*), 162
- N
- n\_shots() (*in module runmanager.remote*), 31
- n\_shots() (*runmanager.remote.Client method*), 34
- nested() (*in module runmanager.\_\_main\_\_*), 36
- new\_global() (*in module runmanager*), 24
- new\_global() (*runmanager.\_\_main\_\_.GroupTab method*), 99
- new\_globals\_file() (*in module runmanager*), 24
- new\_group() (*in module runmanager*), 25
- new\_group() (*runmanager.\_\_main\_\_.RunManager method*), 128
- new\_sequence\_details() (*in module runmanager*), 25
- next\_sequence\_index() (*in module runmanager*), 25
- O
- on\_abort\_clicked() (*runmanager.\_\_main\_\_.RunManager method*), 128
- on\_axes\_check\_selected\_triggered() (*runmanager.\_\_main\_\_.RunManager method*), 128
- on\_axes\_item\_changed() (*runmanager.\_\_main\_\_.RunManager method*), 128
- on\_axes\_uncheck\_selected\_triggered() (*runmanager.\_\_main\_\_.RunManager method*), 128
- on\_axis\_down\_clicked() (*runmanager.\_\_main\_\_.RunManager method*), 128
- on\_axis\_to\_bottom\_clicked() (*runmanager.\_\_main\_\_.RunManager method*), 128
- on\_axis\_to\_top\_clicked() (*runmanager.\_\_main\_\_.RunManager method*), 128
- on\_axis\_up\_clicked() (*runmanager.\_\_main\_\_.RunManager method*), 128
- on\_close\_event() (*runmanager.\_\_main\_\_.RunManager method*), 128

- on\_column\_resized() (runmanager. \_\_main\_\_. TableView method), 184  
 on\_diff\_globals\_file\_clicked() (runmanager. \_\_main\_\_. RunManager method), 128  
 on\_edit\_labscrip\_file\_clicked() (runmanager. \_\_main\_\_. RunManager method), 128  
 on\_engage\_clicked() (runmanager. \_\_main\_\_. RunManager method), 128  
 on\_globals\_delete\_selected\_triggered() (runmanager. \_\_main\_\_. GroupTab method), 99  
 on\_globals\_model\_expansion\_changed() (runmanager. \_\_main\_\_. GroupTab method), 99  
 on\_globals\_model\_item\_changed() (runmanager. \_\_main\_\_. GroupTab method), 99  
 on\_globals\_model\_name\_changed() (runmanager. \_\_main\_\_. GroupTab method), 99  
 on\_globals\_model\_units\_changed() (runmanager. \_\_main\_\_. GroupTab method), 99  
 on\_globals\_model\_value\_changed() (runmanager. \_\_main\_\_. GroupTab method), 99  
 on\_globals\_set\_selected\_bools\_triggered() (runmanager. \_\_main\_\_. GroupTab method), 99  
 on\_groups\_close\_selected\_files\_triggered() (runmanager. \_\_main\_\_. RunManager method), 128  
 on\_groups\_close\_selected\_groups\_triggered() (runmanager. \_\_main\_\_. RunManager method), 128  
 on\_groups\_copy\_selected\_groups\_triggered() (runmanager. \_\_main\_\_. RunManager method), 128  
 on\_groups\_delete\_selected\_triggered() (runmanager. \_\_main\_\_. RunManager method), 128  
 on\_groups\_model\_active\_changed() (runmanager. \_\_main\_\_. RunManager method), 128  
 on\_groups\_model\_item\_changed() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_groups\_model\_name\_changed() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_groups\_model\_openclose\_changed() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_groups\_open\_selected\_triggered() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_groups\_set\_selection\_active\_triggered() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_labscrip\_file\_text\_changed() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_load\_configuration\_triggered() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_master\_shuffle\_clicked() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_new\_globals\_file\_clicked() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_open\_globals\_file\_clicked() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_output\_popout\_button\_clicked() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_reset\_shot\_output\_folder\_clicked() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_restart\_subprocess\_clicked() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_revert\_configuration\_triggered() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_save\_configuration\_as\_triggered() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_save\_configuration\_triggered() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_select\_labscrip\_file\_clicked() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_select\_shot\_output\_folder\_clicked() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_shot\_output\_folder\_text\_changed() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_tabCloseRequested() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_tableView\_globals\_context\_menu\_requested() (runmanager. \_\_main\_\_. GroupTab method), 99  
 on\_tableView\_globals\_leftClicked() (runmanager. \_\_main\_\_. GroupTab method), 99  
 on\_treeView\_axes\_context\_menu\_requested() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_treeView\_groups\_context\_menu\_requested() (runmanager. \_\_main\_\_. RunManager method), 129  
 on\_treeView\_groups\_doubleLeftClicked() (runmanager. \_\_main\_\_. RunManager method), 130  
 on\_treeView\_groups\_leftClicked() (runmanager. \_\_main\_\_. RunManager method), 130  
 open\_globals\_file() (runmanager. \_\_main\_\_. RunManager method), 130  
 open\_group() (runmanager. \_\_main\_\_. RunManager method), 130
- ## P
- paint() (runmanager. \_\_main\_\_. ItemDelegate method), 103  
 paintEvent() (runmanager. \_\_main\_\_. FingerTabBarWidget method), 80

- paintEvent() (runmanager.\_\_main\_\_.RunmanagerMainWindow method), 146
- paintEvent() (runmanager.\_\_main\_\_.TabToolButton method), 162
- parse\_globals() (runmanager.\_\_main\_\_.RunManager method), 130
- PoppedOutOutputBoxWindow (class in runmanager.\_\_main\_\_), 105
- populate\_model() (runmanager.\_\_main\_\_.GroupTab method), 99
- preparse\_globals() (runmanager.\_\_main\_\_.RunManager method), 130
- preparse\_globals\_loop() (runmanager.\_\_main\_\_.RunManager method), 130
- ## Q
- quadspace() (in module runmanager.functions), 28
- question\_dialog() (in module runmanager.\_\_main\_\_), 36
- ## R
- RemoteServer (class in runmanager.\_\_main\_\_), 120
- remove\_comments\_and\_tokenify() (in module runmanager), 25
- rename\_global() (in module runmanager), 25
- rename\_global() (runmanager.\_\_main\_\_.GroupTab method), 99
- rename\_group() (in module runmanager), 25
- rename\_group() (runmanager.\_\_main\_\_.RunManager method), 130
- request() (runmanager.remote.Client method), 34
- reset\_shot\_output\_folder() (in module runmanager.remote), 31
- reset\_shot\_output\_folder() (runmanager.remote.Client method), 34
- resizeEvent() (runmanager.\_\_main\_\_.Editor method), 64
- rollover\_shot\_output\_folder() (runmanager.\_\_main\_\_.RunManager method), 130
- runmanager  
module, 17
- RunManager (class in runmanager.\_\_main\_\_), 122
- runmanager.\_\_main\_\_  
module, 35
- runmanager.batch\_compiler  
module, 35
- runmanager.functions  
module, 28
- runmanager.globals\_diff  
module, 35
- runmanager.remote  
module, 28
- RunmanagerMainWindow (class in runmanager.\_\_main\_\_), 131
- ## S
- save\_configuration() (runmanager.\_\_main\_\_.RunManager method), 130
- say\_hello() (in module runmanager.remote), 31
- say\_hello() (runmanager.remote.Client method), 34
- scroll\_view\_to\_row\_if\_current() (in module runmanager.\_\_main\_\_), 37
- send\_to\_BLACS() (runmanager.\_\_main\_\_.RunManager method), 130
- send\_to\_runviewer() (runmanager.\_\_main\_\_.RunManager method), 130
- set\_expansion() (in module runmanager), 25
- set\_file\_and\_group\_name() (runmanager.\_\_main\_\_.GroupTab method), 99
- set\_globals() (in module runmanager.remote), 31
- set\_globals() (runmanager.remote.Client method), 34
- set\_labscript\_file() (in module runmanager.remote), 31
- set\_labscript\_file() (runmanager.remote.Client method), 34
- set\_run\_shots() (in module runmanager.remote), 31
- set\_run\_shots() (runmanager.remote.Client method), 34
- set\_shot\_output\_folder() (in module runmanager.remote), 32
- set\_shot\_output\_folder() (runmanager.remote.Client method), 34
- set\_shuffle() (in module runmanager.remote), 32
- set\_shuffle() (runmanager.remote.Client method), 34
- set\_tab\_icon() (runmanager.\_\_main\_\_.GroupTab method), 100
- set\_units() (in module runmanager), 26
- set\_value() (in module runmanager), 26
- set\_view\_shots() (in module runmanager.remote), 32
- set\_view\_shots() (runmanager.remote.Client method), 34
- setEditorData() (runmanager.\_\_main\_\_.ItemDelegate method), 103
- setModelData() (runmanager.\_\_main\_\_.ItemDelegate method), 103
- setMovable() (runmanager.\_\_main\_\_.FingerTabBarWidget method), 80
- setTabButton() (runmanager.\_\_main\_\_.FingerTabBarWidget method), 80
- setTabClosable() (runmanager.\_\_main\_\_.FingerTabWidget method), 95
- setup\_axes\_tab() (runmanager.\_\_main\_\_.RunManager method), 130

setup\_config() (*runmanager.\_\_main\_\_.RunManager method*), 130  
 setup\_groups\_tab() (*runmanager.\_\_main\_\_.RunManager method*), 130  
 sizeHint() (*runmanager.\_\_main\_\_.ItemDelegate method*), 103  
 start\_trace() (*runmanager.TraceDictionary method*), 27  
 stop\_trace() (*runmanager.TraceDictionary method*), 27  
 switch\_tabs() (*runmanager.\_\_main\_\_.RunManager method*), 130

## T

tabLayoutChange() (*runmanager.\_\_main\_\_.FingerTabBarWidget method*), 80  
 TableView (*class in runmanager.\_\_main\_\_*), 162  
 tabSizeHint() (*runmanager.\_\_main\_\_.FingerTabBarWidget method*), 80  
 TabToolButton (*class in runmanager.\_\_main\_\_*), 146  
 TraceDictionary (*class in runmanager*), 26  
 TreeView (*class in runmanager.\_\_main\_\_*), 185

## U

update\_axes\_indentation() (*runmanager.\_\_main\_\_.RunManager method*), 130  
 update\_axes\_tab() (*runmanager.\_\_main\_\_.RunManager method*), 130  
 update\_global\_shuffle\_state() (*runmanager.\_\_main\_\_.RunManager method*), 130  
 update\_parse\_indication() (*runmanager.\_\_main\_\_.GroupTab method*), 100  
 update\_size() (*runmanager.\_\_main\_\_.Editor method*), 64  
 update\_tabs\_parsing\_indication() (*runmanager.\_\_main\_\_.RunManager method*), 130

## W

wait\_until\_preparse\_complete() (*runmanager.\_\_main\_\_.RunManager method*), 130