Potku

Class Documentation

1.0.0

Generated by Doxygen 1.8.3.1

Mon May 27 2013 14:27:38

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 runPotku Namespace Reference

Functions

- def **runPotku**

4.1.1 Detailed Description

```
Updated on 23.5.2013

Potku is a graphical user interface for analyzation and
visualization of measurement data collected from a ToF-ERD
telescope. For physics calculations Potku uses external
analyzation components.
Copyright (C) Jarkko Aalto, Timo Konu, Samuli Kärkäinen, Samuli Rahkonen and
Miika Raunio

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program (file named 'LICENCE').
```

Chapter 5

Class Documentation

5.1 Widgets.MatplotlibDepthProfileWidget.MatplotlibDepthProfileWidget.__limit Class Reference

Public Member Functions

- def __init__
- def switch
- def get

Public Attributes

- **limit**

5.1.1 Detailed Description

```
Simple object to control when setting the integration
limits in Depth Profile.
```

5.1.2 Constructor & Destructor Documentation

5.1.2.1 def Widgets.MatplotlibDepthProfileWidget.MatplotlibDepthProfileWidget.__limit.__init__ (*self*)

```
Inits __limit
```

5.1.3 Member Function Documentation

5.1.3.1 def Widgets.MatplotlibDepthProfileWidget.MatplotlibDepthProfileWidget.__limit.get (*self*)

```
Returns the current limit.

Return:
    The current limit a or b.
```

5.1.3.2 def Widgets.MatplotlibDepthProfileWidget.MatplotlibDepthProfileWidget.__limit.switch (*self*)

```
Switches limit between a and b.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Widgets/MatplotlibDepthProfileWidget.py

5.2 Dialogs.AboutDialog.AboutDialog Class Reference

Inherits QDialog.

Public Member Functions

- def __init__
- def closeEvent

Public Attributes

- **ui**
- **x**
- **y**
- **z**
- **color_R**
- **color_G**
- **color_B**

5.2.1 Detailed Description

```
About dialog that shows information about the program itself.
```

5.2.2 Constructor & Destructor Documentation

5.2.2.1 def Dialogs.AboutDialog.AboutDialog.__init__ (*self*)

```
Inits the About Dialog.
```

5.2.3 Member Function Documentation

5.2.3.1 def Dialogs.AboutDialog.AboutDialog.closeEvent (*self, event*)

```
Proper closing.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Dialogs/AboutDialog.py

5.3 Dialogs.CalibrationDialog.CalibrationCurveFittingWidget Class Reference

Inherits QWidget.

Public Member Functions

- def __init__

Public Attributes

- **ui**
- **matplotlib**

5.3.1 Detailed Description

```
Widget class for holding MatplotlibCalibrationCurveFittingWidget.
```

5.3.2 Constructor & Destructor Documentation

5.3.2.1 def Dialogs.CalibrationDialog.CalibrationCurveFittingWidget.__init__ (*self, dialog, cut, tof_calibration, settings, bin_width, column, masses*)

```
Inits widget.

Args:
    dialog: Parent dialog.
    cut: CutFile class object.
    tof_calibration: TOFCalibration class object.
    settings: Settings object
    bin_width: Float representing histogram's bin width.
    column: Integer representing which column number is used.
    masses: Reference to Masses class object.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Dialogs/CalibrationDialog.py

5.4 Dialogs.CalibrationDialog.CalibrationDialog Class Reference

Inherits QDialog.

Public Member Functions

- def __init__
- def remove_selected_points
- def set_calibration_point
- def set_calibration_parameters_to_parent
- def accept_calibration
- def change_current_cut
- def timeout

Public Attributes

- **measurements**
- **settings**
- **cuts**
- **ui**
- **parent_settings_dialog**
- **tof_calibration**
- **cut_file**
- **curveFittingWidget**
- **linearFittingWidget**
- **timer**

5.4.1 Detailed Description

```
A dialog for the time of flight calibration
```

5.4.2 Constructor & Destructor Documentation

5.4.2.1 def Dialogs.CalibrationDialog.CalibrationDialog.__init__ (*self, measurements, settings, masses, parent_settings_dialog =* None)

```
Inits the calibration dialog class

Args:
    measurements: String list representing measurements files.
    settings: Settings object
    masses: Reference to Masses class object.
    parent_settings_dialog: Representing from which dialog this was opened
                    from.
```

5.4.3 Member Function Documentation

5.4.3.1 def Dialogs.CalibrationDialog.CalibrationDialog.accept_calibration (*self*)

```
Accept calibration (parameters).
```

5.4.3.2 def Dialogs.CalibrationDialog.CalibrationDialog.change_current_cut (*self, current_item*)

```
Changes the current cut file drawn to the curve fitting widget.

Args:
    current_item: QtGui.QTreeWidgetItem of CutFile which was selected.
```

5.4.3.3 def Dialogs.CalibrationDialog.CalibrationDialog.remove_selected_points (*self*)

```
Remove selected items from point tree widget
```

5.4.3.4 def Dialogs.CalibrationDialog.CalibrationDialog.set_calibration_parameters_to_parent (*self*)

```
Set calibration parameters to parent dialog's calibration parameters
fields.
```

5.4.3.5 def Dialogs.CalibrationDialog.CalibrationDialog.set_calibration_point (*self, tof*)

```
Set Cut file front edge estimation to specific value.

Args:
    tof: Float representing front edge of linear fit estimation.
```

5.4.3.6 def Dialogs.CalibrationDialog.CalibrationDialog.timeout (*self*)

```
Timeout eventmethod to remove label text.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Dialogs/CalibrationDialog.py

5.5 Dialogs.CalibrationDialog.CalibrationLinearFittingWidget Class Reference

Inherits QWidget.

Public Member Functions

- def __init__

Public Attributes

- **ui**
- **matplotlib**

5.5.1 Detailed Description

```
Widget class for holding MatplotlibCalibrationLinearFittingWidget.
```

5.5.2 Constructor & Destructor Documentation

5.5.2.1 def Dialogs.CalibrationDialog.CalibrationLinearFittingWidget.__init__ (*self, dialog, tof_calibration, old_params*)

```
Inits widget.

Args:
    dialog: Parent dialog.
    tof_calibration: TOFCalibration class object.
    old_params: Old calibration parameters in tuple (slope, offset).
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Dialogs/CalibrationDialog.py

5.6 Modules.CalibrationParameters.CalibrationParameters Class Reference

Public Member Functions

- def __init__
- def show
- def set_settings
- def load_settings
- def save_settings

Public Attributes

- **calibration_settings_filename**
- **config**
- **use_settings**
- **slope**
- **offset**
- **filepath**

5.6.1 Detailed Description

```
MeasuringSettings holds the all project specific measurement unit parameters.
```

5.6.2 Constructor & Destructor Documentation

5.6.2.1 def Modules.CalibrationParameters.CalibrationParameters.__init__ (*self, settings_filepath* = None)

```
Inits MeasuringSettings.

Args:
    settings_filepath: filepath for the settings file to be loaded
```

5.6.3 Member Function Documentation

5.6.3.1 def Modules.CalibrationParameters.CalibrationParameters.load_settings (*self, filepath*)

```
Loads settings' parameters from the given filepath.

Args:
    filepath: Filepath to the settings file.
```

5.6.3.2 def Modules.CalibrationParameters.CalibrationParameters.save_settings (*self, filepath* = None)

```
Saves settings' parameters to the given filepath.

Args:
    filepath: Filepath to the settings file.
```

5.6.3.3 def Modules.CalibrationParameters.CalibrationParameters.set_settings (*self, dialog, used_settings* = None)

```
Takes inputted parameters from the given dialog and sets them to the
corresponding object's parameters

Args:
    dialog: QDialog from which the parameters are taken.
```

5.6.3.4 def Modules.CalibrationParameters.CalibrationParameters.show (*self, dialog*)

```
Shows the measuring parameters in the given measuring settings dialog.

Args:
    dialog: QDialog whose fields are updated with the Calibration
    parameters.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/CalibrationParameters.py

5.7 Modules.UiLogHandlers.customLogHandler Class Reference

Inherits Handler.

Public Member Functions

- def __init__
- def flush
- def emit

Public Attributes

- **log_dialog**
- **formatter**
- **level**

5.7.1 Detailed Description

```
Customloghandler, that handles log messages and emits them to the given
LogWidget's log field.
```

5.7.2 Constructor & Destructor Documentation

5.7.2.1 def Modules.UiLogHandlers.customLogHandler.__init__ (*self, level, formatter, log_dialog*)

```
Initializes the handler.

Args:
    level: The logging level set to this handler.
    formatter: The formatter set to this handler.
    log_dialog: The log dialog, which can add the message to the interface.
```

5.7.3 Member Function Documentation

5.7.3.1 def Modules.UiLogHandlers.customLogHandler.emit (*self, record*)

```
Emits the log message to the destination, which is set when the handler
is initialized.

Args:
    record: The record which will be emitted.
```

5.7.3.2 def Modules.UiLogHandlers.customLogHandler.flush (*self*)

```
Does nothing here, has to be here because this is inherited.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/UiLogHandlers.py

5.8 Modules.CutFile.CutFile Class Reference

Public Member Functions

- def __init__
- def set_info
- def load_file

- def save
- def split
- def copy_info

Public Attributes

- **directory**
- **element**
- **count**
- **is_elem_loss**
- **split_number**
- **split_count**
- **type**
- **weight_factor**
- **energy**
- **detector_angle**
- **element_scatter**
- **data**
- **element_number**

5.8.1 Detailed Description

```
Cut file_path object for when reading cut files is necessary.
```

5.8.2 Constructor & Destructor Documentation

5.8.2.1 def Modules.CutFile.CutFile.__init__ (*self, directory* = None, *elem_loss* = False, *weight_factor* = 1.0, *split_number* = 0, *split_count* = 1)

```
Inits cut file_path object.

Args:
    directory: String representing cut directory.
    elem_loss: Boolean representing whether cut file_path is made from
       elemental losses splits.
    weight_factor: Float representing element weight factor.
    split_number: Integer. Required for Elemental Losses, do not overwrite
          splits.
    split_count: Integer. Required for Elemental Losses, total count of
         splits.
```

5.8.3 Member Function Documentation

5.8.3.1 def Modules.CutFile.CutFile.copy_info (*self, cut_file, data, additional_weight_factor* = 1.0)

```
Copy information from cut file_path object into this.

Args:
    cut_file: CutFile class object.
    data: List of data points.
    additional_weight_factor: Float
```

5.8.3.2 def Modules.CutFile.CutFile.load_file (*self, file*)

```
Load and parse cut file_path.

Args:
    file: String representing cut file.
```

5.8.3.3 def Modules.CutFile.CutFile.save (*self, element_count* = 0)

```
Save cut file_path.

Saves data points into cut file_path with meta information.

Args:
    element_count: Integer representing which selection was used of total
           count of same element and isotope selection. This is so
           that we do not overwrite first 2H selection with other
           2H selection.
```

5.8.3.4 def Modules.CutFile.CutFile.set_info (*self, selection, data*)

```
Set selection information and data into CutFile.

Args:
    selection: Selection class object.
    data: Lists of data points.
```

5.8.3.5 def Modules.CutFile.CutFile.split (*self, reference_cut, splits* = 10, *save* = True)

```
Splits cut file into X splits based on reference cut.

Args:
    reference_cut: Cut file (of heavy element) which is used split.
    splits: Integer determining how many splits is cut splitted to.
    save: Boolean deciding whether or not to save splits.

Return:
    Returns a list containing lists of the cut's splits' values.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/CutFile.py

5.9 Modules.DepthFiles.DepthFiles Class Reference

Inherits object.

Public Member Functions

- def __init__
- def create_depth_files

Public Attributes

- **bin_dir**
- **command_win**
- **command_unix**

5.9.1 Detailed Description

```
DepthFiles handles calling the external programs to create depth files.
```

5.9.2 Constructor & Destructor Documentation

5.9.2.1 def Modules.DepthFiles.DepthFiles.__init__ (*self, filepaths, outputpath*)

```
Inits DepthFiles

Args:
    filepaths: Full paths of cutfiles to be used.
    outputpath: Full path of where depth files are to be created.
```

5.9.3 Member Function Documentation

5.9.3.1 def Modules.DepthFiles.DepthFiles.create_depth_files (*self*)

```
Generate the files necessary for drawing the depth profile
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/DepthFiles.py

5.10 Modules.DepthProfileSettings.DepthProfileSettings Class Reference

Public Member Functions

- def __init__
- def show
- def set_settings
- def load_settings
- def save_settings

Public Attributes

- **depth_profile_settings_filename**
- **config**
- **use_settings**
- **depth_step_for_stopping**
- **depth_step_for_output**
- **depths_for_concentration_from**
- **depths_for_concentration_to**
- **filepath**

5.10.1 Detailed Description

```
DepthProfileSettings holds the all project specific measurement unit parameters.
```

5.10.2 Constructor & Destructor Documentation

5.10.2.1 def Modules.DepthProfileSettings.DepthProfileSettings.__init__ (*self, settings_filepath* = None)

```
Inits DepthProfileSettings.

Args:
    settings_filepath: filepath for the settings file to be loaded
```

5.10.3 Member Function Documentation

5.10.3.1 def Modules.DepthProfileSettings.DepthProfileSettings.load_settings (*self, filepath*)

```
Loads settings' parameters from the given filepath.

Args:
    filepath: Filepath to the settings file.
```

5.10.3.2 def Modules.DepthProfileSettings.DepthProfileSettings.save_settings (*self, filepath* = None)

```
Saves settings' parameters to the given filepath.

Args:
    filepath: Filepath to the settings file.
```

5.10.3.3 def Modules.DepthProfileSettings.DepthProfileSettings.set_settings (*self, dialog, used_settings* = None)

```
Takes inputted parameters from the given dialog and sets them to the
corresponding object's parameters

Args:
    dialog: QDialog from which the parameters are taken.
```

5.10.3.4 def Modules.DepthProfileSettings.DepthProfileSettings.show (*self, dialog*)

```
Shows the measuring parameters in the given measuring settings dialog.

Args:
    dialog: QDialog whose fields are updated with the depth profile
    parameters.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/DepthProfileSettings.py

5.11 Dialogs.DepthProfileDialog.DepthProfileWidget Class Reference

Inherits QWidget.

Public Member Functions

- def __init__
- def delete

Public Attributes

- **parent**
- **ui**
- **matplotlib**

5.11.1 Detailed Description

```
Depth Profile widget which is added to measurement tab.
```

5.11.2 Constructor & Destructor Documentation

5.11.2.1 def Dialogs.DepthProfileDialog.DepthProfileWidget.__init__ (*self, parent, output_dir, elements, x_units*)

```
Inits widget.

Args:
    parent: MeasurementTabWidget
    output_dir: Directory in which the depth files are located.
    elements: A list of Element objects that are used in depth profile.
    x_units: Units to be used for x-axis of depth profile.
```

5.11.3 Member Function Documentation

5.11.3.1 def Dialogs.DepthProfileDialog.DepthProfileWidget.delete (*self*)

```
Delete variables and do clean up.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Dialogs/DepthProfileDialog.py

5.12 Modules.ElementLosses.ElementLosses Class Reference

Public Member Functions

- def __init__
- def count_element_cuts
- def save_splits

Public Attributes

- **directory_cuts**
- **directory_elemloss**
- **partition_count**
- **checked_cuts**
- **progress_bar**
- **reference_cut_file**
- **reference_key**
- **cut_splits**
- **reference_cut**

5.12.1 Detailed Description

```
Element Losses class.
```

5.12.2 Constructor & Destructor Documentation

5.12.2.1 def Modules.ElementLosses.ElementLosses.__init__ (*self, directory_cuts, directory_elemloss, reference_cut_file, checked_cuts, partition_count, progress_bar* = **Null** ())

```
Inits Element Losses class.

Args:
```

```
    directory_cuts: String representing cut file directory.
    directory_elemloss: String representing elemental losses directory.
    reference_cut_file: String representing reference cut file.
    checked_cuts: String list of cut files to be graphed.
    partition_count: Integer representing split count.
    progress_bar: QtGui.QProgressBar or Null() if not given.
```

5.12.3 Member Function Documentation

5.12.3.1 def Modules.ElementLosses.ElementLosses.count_element_cuts (*self, save_splits =* `False`)

```
Count data points in splits based on reference file.

Args:
    save_splits: Boolean representing whether to save element losses splits.

Return:
    Returns dictionary of elements and their counts within splits.
```

5.12.3.2 def Modules.ElementLosses.ElementLosses.save_splits (*self*)

```
Save element splits as new cut files.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/ElementLosses.py

5.13 Dialogs.ElementLossesDialog.ElementLossesDialog Class Reference

Inherits QDialog.

Public Member Functions

- def __init__

Public Attributes

- **parent**
- **cuts**
- **ui**

5.13.1 Detailed Description

```
Class to handle element losses dialogs.
```

5.13.2 Constructor & Destructor Documentation

5.13.2.1 def Dialogs.ElementLossesDialog.ElementLossesDialog.__init__ (*self, parent*)

```
Inits element losses class.

 Args:
    parent: MeasurementTabWidget
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Dialogs/ElementLossesDialog.py

5.14 Modules.ElementLosses.ElementLossesSplitHolder Class Reference

Public Member Functions

- def __init__
- def count
- def get_keys
- def get_cut
- def get_splits
- def add_splits

5.14.1 Detailed Description

```
Element Losses Split Holder class to hold information of cuts' splits.
```

5.14.2 Constructor & Destructor Documentation

5.14.2.1 def Modules.ElementLosses.ElementLossesSplitHolder.__init__ (*self*)

```
Inits the class
```

5.14.3 Member Function Documentation

5.14.3.1 def Modules.ElementLosses.ElementLossesSplitHolder.add_splits (*self, key, cut, splits*)

```
Add splits to a cut file
```

5.14.3.2 def Modules.ElementLosses.ElementLossesSplitHolder.count (*self*)

```
Get count of splits.

Return:
    Returns count of cut files splitted.
```

5.14.3.3 def Modules.ElementLosses.ElementLossesSplitHolder.get_cut (*self, key*)

```
Get cut file used to make splits.
```

5.14.3.4 def Modules.ElementLosses.ElementLossesSplitHolder.get_keys (*self*)

```
Get keys of splits.

Return:
    Returns all keys that are currently used.
```

5.14.3.5 def Modules.ElementLosses.ElementLossesSplitHolder.get_splits (*self, key*)

```
Get splits of a cut file.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/ElementLosses.py

5.15 Dialogs.ElementLossesDialog.ElementLossesWidget Class Reference

Inherits QWidget.

Public Member Functions

- def __init__
- def delete

Public Attributes

- **parent**
- **progress_bar**
- **ui**
- **losses**
- **matplotlib**

5.15.1 Detailed Description

```
Element losses widget which is added to measurement tab.
```

5.15.2 Constructor & Destructor Documentation

5.15.2.1 def Dialogs.ElementLossesDialog.ElementLossesWidget.__init__ (*self, parent, reference_cut_file, checked_cuts, partition_count, y_scale*)

```
Inits widget.

Args:
    parent: MeasurementTabWidget
    reference_cut_file: String representing reference cut file.
    checked_cuts: String list representing cut files.
    partition_count: Integer representing how many splits cut files
             are divided to.
    y_scale: Integer flag representing how Y axis is scaled.
```

5.15.3 Member Function Documentation

5.15.3.1 def Dialogs.ElementLossesDialog.ElementLossesWidget.delete (*self*)

```
Delete variables and do clean up.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Dialogs/ElementLossesDialog.py

5.16 Dialogs.ElementSelectionDialog.ElementSelectionDialog Class Reference

Inherits QDialog.

Public Member Functions

- def __init__

Public Attributes

- **ui**
- **element**

5.16.1 Detailed Description

```
ElementSelectionDialog opens a periodic table from which user can select
an element.
```

5.16.2 Constructor & Destructor Documentation

5.16.2.1 def Dialogs.ElementSelectionDialog.ElementSelectionDialog.__init__ (*self*)

```
Inits the ElementSelection class
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Dialogs/ElementSelectionDialog.py

5.17 Modules.EnergySpectrum.EnergySpectrum Class Reference

Public Member Functions

- def __init__
- def calculate_spectrum

Public Attributes

- **cut_files**
- **spectrum_width**
- **progress_bar**
- **tof_listed_files**

5.17.1 Detailed Description

5.17.2 Constructor & Destructor Documentation

5.17.2.1 def Modules.EnergySpectrum.EnergySpectrum.__init__ (*self, cut_files, spectrum_width, progress_bar* = **Null** ())

```
Inits energy spectrum

Args:
    cut_files: String list of cut files.
    spectrum_width: Float representing energy spectrum graph width.
    progress_bar: QtGui.QProgressBar for GUI (Null class object otherwise).
```

5.17.3 Member Function Documentation

5.17.3.1 def Modules.EnergySpectrum.EnergySpectrum.calculate_spectrum (*self*)

```
Calculate energy spectrum data from cut files.

Returns list of cut files
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/EnergySpectrum.py

5.18 Dialogs.EnergySpectrumDialog.EnergySpectrumWidget Class Reference

Inherits QWidget.

Public Member Functions

- def __init__
- def delete

Public Attributes

- **parent**
- **progress_bar**
- **ui**
- **energy_spectrum**
- **matplotlib**

5.18.1 Detailed Description

```
Energy spectrum widget which is added to measurement tab.
```

5.18.2 Constructor & Destructor Documentation

5.18.2.1 def Dialogs.EnergySpectrumDialog.EnergySpectrumWidget.__init__ (*self, parent, use_cuts, width*)

```
Inits widget.

Args:
    parent: MeasurementTabWidget
    use_cuts: String list representing CutFiles
    width: Float representing Energy Spectrum histogram's bin width.
```

5.18.3 Member Function Documentation

5.18.3.1 def Dialogs.EnergySpectrumDialog.EnergySpectrumWidget.delete (*self*)

```
Delete variables and do clean up.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Dialogs/EnergySpectrumDialog.py

5.19 Modules.GlobalSettings.GlobalSettings Class Reference

Public Member Functions

- def __init__
- def save_config
- def get_project_directory
- def set_project_directory
- def get_project_directory_last_open
- def set_project_directory_last_open
- def get_element_colors
- def get_element_color
- def set_element_color

5.19.1 Detailed Description

```
Global settings class to handle software settings.
```

5.19.2 Constructor & Destructor Documentation

5.19.2.1 def Modules.GlobalSettings.GlobalSettings.__init__ (*self*)

```
Inits GLobalSettings class.
```

5.19.3 Member Function Documentation

5.19.3.1 def Modules.GlobalSettings.GlobalSettings.get_element_color (*self, element*)

```
Get a specific element's color.

Args:
    element: String representing element name.
```

5.19.3.2 def Modules.GlobalSettings.GlobalSettings.get_element_colors (*self*)

```
Get all elements' colors.
```

5.19.3.3 def Modules.GlobalSettings.GlobalSettings.get_project_directory (*self*)

```
Get default project directory.
```

5.19.3.4 def Modules.GlobalSettings.GlobalSettings.get_project_directory_last_open (*self*)

```
Get directory where last project was opened.
```

5.19.3.5 def Modules.GlobalSettings.GlobalSettings.save_config (*self*)

```
Save current global settings.
```

5.19.3.6 def Modules.GlobalSettings.GlobalSettings.set_element_color (*self, element, color*)

```
Set default color for an element.

Args:
    element: String representing element.
    color: String representing color.
```

5.19.3.7 def Modules.GlobalSettings.GlobalSettings.set_project_directory (*self, directory*)

```
Save default project directory.

Args:
    directory: String representing folder where projects will be saved
    by default.
```

5.19.3.8 def Modules.GlobalSettings.GlobalSettings.set_project_directory_last_open (*self, directory*)

```
Save last opened project directory.

Args:
    directory: String representing project folder.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/GlobalSettings.py

5.20 Modules.IconManager.IconManager Class Reference

Public Member Functions

- def __init__
- def get_icon
- def set_icon

5.20.1 Detailed Description

```
Icon manager class to handle all icons for the program.
```

5.20.2 Constructor & Destructor Documentation

5.20.2.1 def Modules.IconManager.IconManager.__init__ (*self*)

```
Inits IconManager class.
```

5.20.3 Member Function Documentation

5.20.3.1 def Modules.IconManager.IconManager.get_icon (*self, icon_name*)

```
Get specific icon.

Args:
    icon_name: String representing icon file name.

Return:
    Returns QtGui.QIcon of icon_name and empty icon if not found.
```

5.20.3.2 def Modules.IconManager.IconManager.set_icon (*self, target, icon_name, size* = (20, 20))

```
Set icon (icon_name) to target.

Args:
    target: QtGui element that has icon. (setIcon method)
    icon_name: String representing filename of the icon.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/IconManager.py

5.21 Modules.InputValidator.InputValidator Class Reference

Inherits QDoubleValidator.

Public Member Functions

- def __init__
- def validate

5.21.1 Detailed Description

```
Validator to check the validity of user inputs.

Accepts double values with scientific notation (i.e. 0.232, 12.5e-12) and turns
empty input to 0.0 and commas (,) to points (.).
```

5.21.2 Constructor & Destructor Documentation

5.21.2.1 def Modules.InputValidator.InputValidator.__init__ (*self, bottom* = float_info.min, *top* = float_info.max, *decimals* = float_info.dig, *parent* = None)

```
Initiates the class.

Args:
    bottom: Float minimum value.
    top: Float maximum value.
    decimals: Integer representing decimals.
    parent: Parent object.
```

5.21.3 Member Function Documentation

5.21.3.1 def Modules.InputValidator.InputValidator.validate (*self, input_value, pos*)

```
Validates the given input. Overrides the QDoubleValidator's validate
function.

Args:
    input_value: User given string to be validated.
    pos: Cursor position (if required).
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/InputValidator.py

5.22 Widgets.LogWidget.LogWidget Class Reference

Inherits QWidget.

Public Member Functions

- def __init__
- def add_text
- def add_error
- def closeEvent
- def minimize_window

Public Attributes

- **want_to_close**
- **ui**

5.22.1 Detailed Description

```
Log widget which displays the log. This widget handles the loghandlers emits.
```

5.22.2 Constructor & Destructor Documentation

5.22.2.1 def Widgets.LogWidget.LogWidget.__init__ (*self*)

```
Initializes the loghandler widget.
```

5.22.3 Member Function Documentation

5.22.3.1 def Widgets.LogWidget.LogWidget.add_error (*self, message*)

```
Adds the specified message to the error field.

Args:
    message: the message which will be displayed.
```

5.22.3.2 def Widgets.LogWidget.LogWidget.add_text (*self, message*)

```
Adds the specified message to the log field.

Args:
    message: the message which will be displayed.
```

5.22.3.3 def Widgets.LogWidget.LogWidget.closeEvent (*self, evnt*)

```
Event which happens when the windows is closing.

Instead of closing, minimize the window. This is because the disabling of
the close button isn't implemented yet.

Args:
    envt: Close event
```

5.22.3.4 def Widgets.LogWidget.LogWidget.minimize_window (*self*)

```
Minimize the window.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Widgets/LogWidget.py

5.23 Modules.Masses.Masses Class Reference

Public Member Functions

- def __init__
- def load_isotopes
- def get_standard_isotope
- def get_most_common_isotope

Public Attributes

- **isotopes**

5.23.1 Detailed Description

```
Masses class handles all element isotopes' masses.
```

5.23.2 Constructor & Destructor Documentation

5.23.2.1 def Modules.Masses.Masses.__init__ (*self, filepath*)

```
Inits Masses object

Args:
    filepath: String representing filepath to masses.dat
```

5.23.3 Member Function Documentation

5.23.3.1 def Modules.Masses.Masses.get_most_common_isotope (*self, element*)

```
Get the most common isotope for an element.

Args:
    element: String representing element.

Return:
    Returns the most common isotope for the element (int)
    and the propability (commonness) of the isotope (float)
    as a tuple(int, float).
```

5.23.3.2 def Modules.Masses.Masses.get_standard_isotope (*self, element*)

```
Calculate standard element weight.

Args:
    element: String representing element.

Return:
    Returns standard weight of given element (float).
```

5.23.3.3 def Modules.Masses.Masses.load_isotopes (*self, element, combobox, current_isotope* = None)

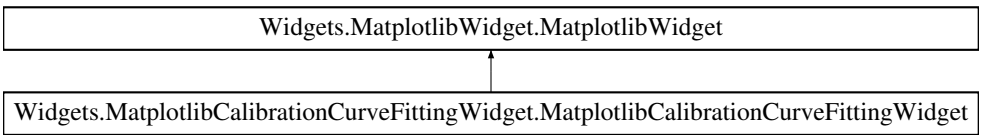
```
Load isotopes into given combobox.

Args:
    element: String representing selected element of which
        isotopes are loaded.
    combobox: QComboBox to which items are added.
    current_isotope: Current isotope to select it on combobox by default
             (string).
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/Masses.py

5.24 Widgets.MatplotlibCalibrationCurveFittingWidget.MatplotlibCalibrationCurveFittingWidget Class Reference

Inheritance diagram for Widgets.MatplotlibCalibrationCurveFittingWidget.MatplotlibCalibrationCurveFittingWidget:



Public Member Functions

- def __init__
- def onclick
- def set_calibration_point_externally
- def change_cut

- def change_bin_width
- def on_draw
- def toggle_clicks

Public Attributes

- **dialog**
- **settings**
- **cut**
- **masses**
- **cut_standard_mass**
- **cut_standard_scatter_mass**
- **bin_width**
- **use_column**
- **tof_histogram**
- **tof_calibration_point**
- **selection_given_manually**
- **selected_tof**
- **selectButton**

5.24.1 Detailed Description

```
Energy spectrum widget
```

5.24.2 Constructor & Destructor Documentation

5.24.2.1 def Widgets.MatplotlibCalibrationCurveFittingWidget.MatplotlibCalibrationCurveFittingWidget.__init__ (*self, parent, settings, tof_calibration, cut, masses, bin_width* = 2.0, *column* = 1, *dialog* = None)

```
Inits Energy Spectrum widget.

Args:
    parent: CalibrationCurveFittingWidget
    settings: Settings class object.
    tof_calibration: TOFCalibration class object.
    cut: CutFile class object.
    masses: Reference to element masses object of main program.
    bin_width: Histograms bin width
    column: Which column of the CutFile's data is used to create a
    histogram.
    dialog: parent's parent dialog.
```

5.24.3 Member Function Documentation

5.24.3.1 def Widgets.MatplotlibCalibrationCurveFittingWidget.MatplotlibCalibrationCurveFittingWidget.change_bin_width (*self, bin_width*)

```
Change histogram bin width.

Args:
    bin_width: Float representing graph bin width.
```

5.24.3.2 def Widgets.MatplotlibCalibrationCurveFittingWidget.MatplotlibCalibrationCurveFittingWidget.change_cut (*self, cut*)

```
Changes the cut file to be drawn and analyzed
```

5.24.3.3 def Widgets.MatplotlibCalibrationCurveFittingWidget.MatplotlibCalibrationCurveFittingWidget.on_draw (*self*)

```
Draw method for matplotlib.
```

5.24.3.4 def Widgets.MatplotlibCalibrationCurveFittingWidget.MatplotlibCalibrationCurveFittingWidget.onclick (*self, event*)

```
Handles clicks on the graph

Args:
    event: Mouse click event.
```

5.24.3.5 def Widgets.MatplotlibCalibrationCurveFittingWidget.MatplotlibCalibrationCurveFittingWidget.set_calibration_point_externally (*self, tof*)

```
Set calibration point.

Args:
    tof: Integer representing x axis value Time of Flight [Channel].
```

5.24.3.6 def Widgets.MatplotlibCalibrationCurveFittingWidget.MatplotlibCalibrationCurveFittingWidget.toggle_clicks (*self*)

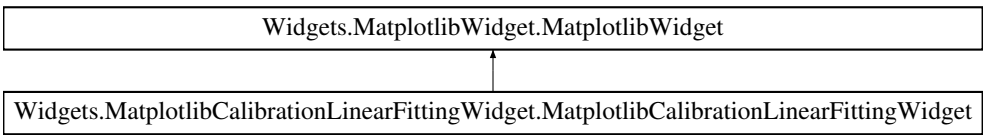
```
Toggle between manual ToF channel (x axis) selection.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Widgets/MatplotlibCalibrationCurveFittingWidget.py

5.25 Widgets.MatplotlibCalibrationLinearFittingWidget.MatplotlibCalibrationLinearFittingWidget Class Reference

Inheritance diagram for Widgets.MatplotlibCalibrationLinearFittingWidget.MatplotlibCalibrationLinearFittingWidget:



Public Member Functions

- def __init__
- def on_draw

Public Attributes

- **dialog**
- **old_params**
- **tof_calibration**
- **enable_selection_tool**

5.25.1 Detailed Description

```
Energy spectrum widget
```

5.25.2 Constructor & Destructor Documentation

5.25.2.1 def Widgets.MatplotlibCalibrationLinearFittingWidget.MatplotlibCalibrationLinearFittingWidget.__init__ (*self, parent, tof_calibration, dialog* = None, *old_params* = None)

```
Inits Energy Spectrum widget.

Args:
    parent: CalibrationCurveFittingWidget
    tof_calibration: TOFCalibration class object.
    dialog: parent's parent dialog.
    old_params: tuple of parameters (x0, A, k)
```

5.25.3 Member Function Documentation

5.25.3.1 def Widgets.MatplotlibCalibrationLinearFittingWidget.MatplotlibCalibrationLinearFittingWidget.on_draw (*self*)

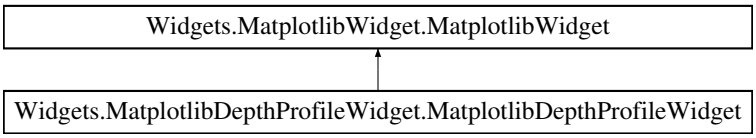
```
Draw method for matplotlib.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Widgets/MatplotlibCalibrationLinearFittingWidget.py

5.26 Widgets.MatplotlibDepthProfileWidget.MatplotlibDepthProfileWidget Class Reference

Inheritance diagram for Widgets.MatplotlibDepthProfileWidget.MatplotlibDepthProfileWidget:



Classes

- class __limit

Public Member Functions

- def __init__
- def onclick
- def on_draw

Public Attributes

- **x_units**
- **draw_legend**

- **elements**
- **depth_dir**
- **depth_files**
- **read_files**
- **rel_files**
- **hyb_files**
- **selection_colors**
- **icon_manager**
- **lim_a**
- **lim_b**
- **lim_icons**
- **lim_mode**
- **limButton**
- **modeButton**
- **viewButton**

5.26.1 Detailed Description

```
Depth profile widget
```

5.26.2 Constructor & Destructor Documentation

5.26.2.1 def Widgets.MatplotlibDepthProfileWidget.MatplotlibDepthProfileWidget.__init__ (*self, parent, depth_dir, elements, x_units = 'nm', legend =* `True`)

```
Inits depth profile widget.

Args:
    depth_dir: Directory where depth files are located.
    elements: List of Element objects.
    x_units: Unit to be used as x-axis.
    legend: Boolean of whether to show the legend.
```

5.26.3 Member Function Documentation

5.26.3.1 def Widgets.MatplotlibDepthProfileWidget.MatplotlibDepthProfileWidget.on_draw (*self*)

```
Draws the depth profile graph
```

5.26.3.2 def Widgets.MatplotlibDepthProfileWidget.MatplotlibDepthProfileWidget.onclick (*self, event*)

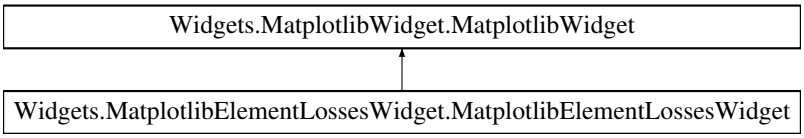
```
Handles clicks on the graph

Args:
    event: A click event on the graph
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Widgets/MatplotlibDepthProfileWidget.py

5.27 Widgets.MatplotlibElementLossesWidget.MatplotlibElementLossesWidget Class Reference

Inheritance diagram for Widgets.MatplotlibElementLossesWidget.MatplotlibElementLossesWidget:



Public Member Functions

- def __init__
- def on_draw

Public Attributes

- **draw_legend**
- **split**
- **y_scale**
- **selection_colors**

5.27.1 Detailed Description

```
Energy spectrum widget
```

5.27.2 Constructor & Destructor Documentation

5.27.2.1 def Widgets.MatplotlibElementLossesWidget.MatplotlibElementLossesWidget.__init__ (*self, parent, split, legend =* `True`, *y_scale =* `0`)

```
Inits Energy Spectrum widget.

Args:
    parent: ElementLossesWidget class object.
    split: List of counted split counts for each element.
    legend: Boolean representing whether to draw legend or not.
```

5.27.3 Member Function Documentation

5.27.3.1 def Widgets.MatplotlibElementLossesWidget.MatplotlibElementLossesWidget.on_draw (*self*)

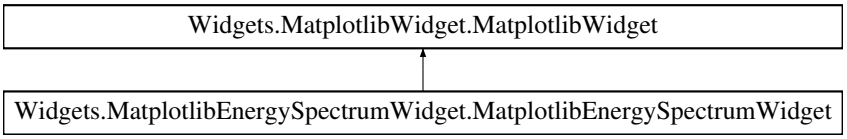
```
Draw method for matplotlib.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Widgets/MatplotlibElementLossesWidget.py

5.28 Widgets.MatplotlibEnergySpectrumWidget.MatplotlibEnergySpectrumWidget Class Reference

Inheritance diagram for Widgets.MatplotlibEnergySpectrumWidget.MatplotlibEnergySpectrumWidget:



Public Member Functions

- def __init__
- def on_draw

Public Attributes

- **draw_legend**
- **histed_files**
- **selection_colors**

5.28.1 Detailed Description

```
Energy spectrum widget
```

5.28.2 Constructor & Destructor Documentation

5.28.2.1 def Widgets.MatplotlibEnergySpectrumWidget.MatplotlibEnergySpectrumWidget.__init__ (*self, parent, histed_files, legend* = `True`)

```
Inits Energy Spectrum widget.

Args:
    parent: EnergySpectrumWidget class object.
    histed_files: List of calculated energy spectrum files.
    legend: Boolean representing whether to draw legend or not.
```

5.28.3 Member Function Documentation

5.28.3.1 def Widgets.MatplotlibEnergySpectrumWidget.MatplotlibEnergySpectrumWidget.on_draw (*self*)

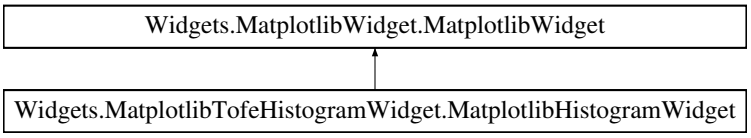
```
Draw method for matplotlib.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Widgets/MatplotlibEnergySpectrumWidget.py

5.29 Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget Class Reference

Inheritance diagram for Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget:



Public Member Functions

- def __init__
- def on_draw
- def on_click
- def graph_settings_dialog
- def selection_settings_dialog
- def load_selections
- def save_cuts
- def enable_element_selection
- def enable_selection_select
- def remove_selected
- def remove_all_selections
- def undo_point
- def show_yourself

Public Attributes

- **icon_manager**
- **invert_Y**
- **invert_X**
- **transpose_axes**
- **bins**
- **name_y_axis**
- **name_x_axis**
- **measurement**
- **color_scheme_selected**
- **elementSelectionButton**
- **elementSelectUndoButton**
- **elementSelectionSelectButton**
- **elementSelectDeleteButton**
- **elementSelectionDeleteButton**

Static Public Attributes

- dictionary **color_scheme**

5.29.1 Detailed Description

```
Matplotlib histogram widget, used to graph "bananas" (ToF-E).
```

5.29.2 Constructor & Destructor Documentation

5.29.2.1 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.__init__ (*self, parent, measurement_data, icon_manager*)

```
Inits histogram widget

Args:
    parent: TofeHistogramWidget class object.
    measurement_data: List of data points.
    icon_manager: IconManager class object.
```

5.29.3 Member Function Documentation

5.29.3.1 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.enable_element_selection (*self*)

```
Enable element selection.
```

5.29.3.2 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.enable_selection_select (*self*)

```
Enable selection selecting tool.
```

5.29.3.3 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.graph_settings_dialog (*self*)

```
Show graph settings dialog.
```

5.29.3.4 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.load_selections (*self*)

```
Show dialog to load selections.
```

5.29.3.5 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.on_click (*self, event*)

```
On click event above graph.
```

5.29.3.6 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.on_draw (*self*)

```
Draw method for matplotlib.
```

5.29.3.7 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.remove_all_selections (*self*)

```
Remove all selections.
```

5.29.3.8 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.remove_selected (*self*)

```
Remove selected selection.
```

5.29.3.9 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.save_cuts (*self*)

```
Save measurement cuts.
```

5.29.3.10 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.selection_settings_dialog (*self*)

```
Show selection settings dialog.
```

5.29.3.11 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.show_yourself (*self, ui*)

```
Show ToF-E histogram settings in ui.
```

5.29.3.12 def Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.undo_point (*self*)

```
Undo last point in open selection.
```

5.29.4 Member Data Documentation

5.29.4.1 dictionary Widgets.MatplotlibTofeHistogramWidget.MatplotlibHistogramWidget.color_scheme [static]

Initial value:

```
1 = {"Default color":"jet",
2                    "Greyscale":"Greys",
3                    "Greyscale (inverted)":"gray"}
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Widgets/MatplotlibTofeHistogramWidget.py

5.30 Widgets.MatplotlibWidget.MatplotlibWidget Class Reference

Inheritance diagram for Widgets.MatplotlibWidget.MatplotlibWidget:

Public Member Functions

- def __init__
- def fork_toolbar_buttons
- def remove_axes_ticks
- def delete

Public Attributes

- **main_frame**
- **dpi**
- **show_axis_ticks**
- **fig**
- **canvas**
- **axes**
- **mpl_toolbar**

5.30.1 Detailed Description

```
Base class for matplotlib widgets
```

5.30.2 Constructor & Destructor Documentation

5.30.2.1 def Widgets.MatplotlibWidget.MatplotlibWidget.__init__ (*self, parent*)

```
Inits matplotlib widget.

Args:
    parent: parent class object.
```

5.30.3 Member Function Documentation

5.30.3.1 def Widgets.MatplotlibWidget.MatplotlibWidget.delete (*self*)

```
Delete matplotlib objects.
```

5.30.3.2 def Widgets.MatplotlibWidget.MatplotlibWidget.fork_toolbar_buttons (*self*)

```
Remove figure options & subplot config that might not work properly.
```

5.30.3.3 def Widgets.MatplotlibWidget.MatplotlibWidget.remove_axes_ticks (*self*)

```
Remove ticks from axes.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Widgets/MatplotlibWidget.py

5.31 Modules.Measurement.Measurement Class Reference

Public Member Functions

- def __init__
- def load_data
- def set_loggers
- def remove_and_close_log
- def set_axes
- def add_point
- def undo_point
- def purge_selection
- def remove_all
- def draw_selection
- def end_open_selection
- def selection_select
- def selection_count
- def reset_select
- def remove_selected
- def save_cuts

- def get_cut_files
- def fill_cuts_treewidget
- def load_selection
- def generate_tof_in

Public Attributes

- **measurement_file**
- **measurement_name**
- **directory**
- **directory_cuts**
- **directory_elemloss**
- **project**
- **data**
- **tab_id**
- **measurement_settings**
- **statusbar**
- **selector**
- **color_scheme**
- **defaultlog**
- **errorlog**

5.31.1 Detailed Description

```
Measurement class to handle one measurement data.
```

5.31.2 Constructor & Destructor Documentation

5.31.2.1 def Modules.Measurement.Measurement.__init__ (*self, measurement_file, project, tab_id*)

```
Inits measurement.

Args:
    measurement_file: String representing path to measurement file.
    project: Project class object.
    tab_id: Integer representing tab identifier for measurement.
```

5.31.3 Member Function Documentation

5.31.3.1 def Modules.Measurement.Measurement.add_point (*self, point, canvas*)

```
Add point into selection or create new selection if first or all closed.

Args:
    point: Point (x, y) to be added to selection.
    canvas: matplotlib's FigureCanvas where selections are drawn.

Return:
    1: When point closes open selection and allows new selection to
be made.
    0: When point was added to open selection.
    -1: When new selection is not allowed and there are no selections.
```

5.31.3.2 def Modules.Measurement.Measurement.draw_selection (*self*)

```
Draw all selections in measurement.
```

5.31.3.3 def Modules.Measurement.Measurement.end_open_selection (*self, canvas*)

```
End last open selection.

Ends last open selection. If selection is open, it will show dialog to
select element information and draws into canvas before opening the dialog.

Args:
    canvas: Matplotlib's FigureCanvas

Return:
    1: If selection closed
    0: Otherwise
```

5.31.3.4 def Modules.Measurement.Measurement.fill_cuts_treewidget (*self, treewidget, use_elemloss* = `False`)

```
Fill QTreeWidget with cut files.

Args:
    treewidget: QtGui.QTreeWidget, where cut files are added to.
    elemloss: Boolean representing whether to add elemental losses or not.
```

5.31.3.5 def Modules.Measurement.Measurement.generate_tof_in (*self*)

```
Generate tof.in file for external programs.

Generates tof.in file for measurement to be used in external programs
(tof_list, erd_depth).
```

5.31.3.6 def Modules.Measurement.Measurement.get_cut_files (*self*)

```
Get cut files from a measurement.

Return:
    Returns a list of cut files in measurement.
```

5.31.3.7 def Modules.Measurement.Measurement.load_data (*self*)

```
Loads measurement data from filepath
```

5.31.3.8 def Modules.Measurement.Measurement.load_selection (*self, filename*)

```
Load selections from a file_path.

Removes all current selections and loads selections from given filename.

Args:
    filename: String representing (full) directory to selection file_path.
```

5.31.3.9 def Modules.Measurement.Measurement.purge_selection (*self*)

```
Purges (removes) all open selections and allows new selection to be made.
```

5.31.3.10 def Modules.Measurement.Measurement.remove_all (*self*)

```
Remove all selections in selector.
```

5.31.3.11 def Modules.Measurement.Measurement.remove_and_close_log (*self, log_filehandler*)

```
Closes the log file and removes it from the logger.

Args:
    log_filehandler: Log's filehandler.
```

5.31.3.12 def Modules.Measurement.Measurement.remove_selected (*self*)

```
Remove selection

Removes currently selected selection.
```

5.31.3.13 def Modules.Measurement.Measurement.reset_select (*self*)

```
Reset selection to None.

Resets current selection to None and resets colors of all selections
to their default values.
```

5.31.3.14 def Modules.Measurement.Measurement.save_cuts (*self*)

```
Save cut files

Saves data points within selections into cut files.
```

5.31.3.15 def Modules.Measurement.Measurement.selection_count (*self*)

```
Get count of selections.

Return:
    Returns the count of selections in selector object.
```

5.31.3.16 def Modules.Measurement.Measurement.selection_select (*self, cursorpoint, highlight* = True)

```
Select a selection based on point.

Args:
    point: Point (x, y) which is clicked on the graph to select selection.
    highlight: Boolean to determine whether to highlight just this
       selection.

Return:
    1: If point is within selection.
    0: If point is not within selection.
```

5.31.3.17 def Modules.Measurement.Measurement.set_axes (*self, axes*)

```
Set axes information to selector within measurement.

Sets axes information to selector to add selection points. Since
previously when creating measurement old selection could not be checked.
Now is time to check for it, while data is still "loading".

Args:
    axes: Matplotlib FigureCanvas's subplot
```

5.31.3.18 def Modules.Measurement.Measurement.set_loggers (*self*)

```
Sets the loggers for this specified measurement.

The logs will be displayed in the measurements folder.
After this, the measurement logger can be called from anywhere of the
program, using logging.getLogger([measurement_name]).
```

5.31.3.19 def Modules.Measurement.Measurement.undo_point (*self*)

```
Undo last point in open selection.

Undo last point in open (last) selection. If there are no selections,
do nothing.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/Measurement.py

5.32 Widgets.MeasurementInfoWidget.MeasurementInfoWidget Class Reference

Inherits QWidget.

Public Member Functions

- def **__init__**

Public Attributes

- **ui**

5.32.1 Detailed Description

```
Class for creating an info tab widget
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Widgets/MeasurementInfoWidget.py

5.33 Modules.Measurement.Measurements Class Reference

Public Member Functions

- def __init__
- def is_empty
- def **get_key_value**
- def add_measurement_file
- def remove_by_tab_id

Public Attributes

- **project**
- **measurements**
- **measuring_unit_settings**
- **default_settings**

5.33.1 Detailed Description

```
Measurements class handles multiple measurements.
```

5.33.2 Constructor & Destructor Documentation

5.33.2.1 def Modules.Measurement.Measurements.__init__ (*self, project*)

```
Inits measurements class.

Args:
    project: Project class object.
```

5.33.3 Member Function Documentation

5.33.3.1 def Modules.Measurement.Measurements.add_measurement_file (*self, measurement_file, tab_id*)

```
Add a new file to measurements.

Args:
    measurement_filepath: String representing file containing measurement
                  data.
    tab_id: Integer representing identifier for measurement's tab.

Return:
    Returns new measurement or None if it wasn't added
```

5.33.3.2 def Modules.Measurement.Measurements.is_empty (*self*)

```
Check if there are any measurements.

Return:
    Returns True if there are no measurements currently in the
    measurements object.
```

5.33.3.3 def Modules.Measurement.Measurements.remove_by_tab_id (*self, tab_id*)

```
Removes measurement from measurements by tab id

Args:
    tab_id: Integer representing tab identifier.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/Measurement.py

5.34 Widgets.MeasurementTabWidget.MeasurementTabWidget Class Reference

Inherits QWidget.

Public Member Functions

- def __init__
- def hide_panel
- def measurement_save_cuts
- def open_measuring_unit_settings
- def open_depth_profile_settings
- def open_calibration_settings
- def open_depth_profile
- def open_energy_spectrum
- def open_element_losses
- def add_widget
- def del_widget
- def add_histogram
- def add_log
- def add_UI_logger

Public Attributes

- **tab_id**
- **ui**
- **measurement**
- **icon_manager**
- **histogram**
- **elemental_losses_widget**
- **energy_spectrum_widget**
- **depth_profile_widget**
- **panel_shown**
- **log**

5.34.1 Detailed Description

```
Tab widget where measurement stuff is added.
```

5.34.2 Constructor & Destructor Documentation

5.34.2.1 def Widgets.MeasurementTabWidget.MeasurementTabWidget.__init__ (*self, tab_id, measurement, icon_manager*)

```
Init measurement tab class.

Args:
    tab_id: Integer representing ID of the tabwidget.
    measurement: Measurement class object.
    icon_manager: IconManager class object.
```

5.34.3 Member Function Documentation

5.34.3.1 def Widgets.MeasurementTabWidget.MeasurementTabWidget.add_histogram (*self*)

```
Adds ToF-E histogram into tab if it doesn't have one already.
```

5.34.3.2 def Widgets.MeasurementTabWidget.MeasurementTabWidget.add_log (*self*)

```
Add the measurement log to measurement tab widget.

Checks also if there's already some logging for this measurement and appends
the text field of the user interface with this log.
```

5.34.3.3 def Widgets.MeasurementTabWidget.MeasurementTabWidget.add_UI_logger (*self, log_widget*)

```
Adds handlers to measurement logger so the logger can log the events to
the user interface too.

log_widget specifies which ui element will handle the logging. That should
be the one which is added to this MeasurementTabWidget.
```

5.34.3.4 def Widgets.MeasurementTabWidget.MeasurementTabWidget.add_widget (*self, widget, minimized =* `None`, *has_close_button =* `True`, *icon =* `None`)

```
Adds a new widget to current (measurement) tab.

Args:
    widget: QWidget to be added into measurement tab widget.
    minimized: Boolean representing if widget should be minimized.
    icon: QtGui.QIcon for the subwindow.
```

5.34.3.5 def Widgets.MeasurementTabWidget.MeasurementTabWidget.del_widget (*self, widget*)

```
Delete a widget from current (measurement) tab.

Args:
    widget: QWidget to be removed.
```

5.34.3.6 def Widgets.MeasurementTabWidget.MeasurementTabWidget.hide_panel (*self, enable_hide =* `None`)

```
Sets the frame (including all the tool buttons) visible.

Args:
    enable_hide: If True, sets the frame visible and vice versa.
         If not given, sets the frame visible or hidden
         depending its previous state.
```

5.34.3.7 def Widgets.MeasurementTabWidget.MeasurementTabWidget.measurement_save_cuts (*self*)

```
Save measurement selections to cut files.
```

5.34.3.8 def Widgets.MeasurementTabWidget.MeasurementTabWidget.open_calibration_settings (*self*)

```
Opens calibration settings dialog.
```

5.34.3.9 def Widgets.MeasurementTabWidget.MeasurementTabWidget.open_depth_profile (*self, parent*)

```
Opens depth profile dialog.

Args:
    parent: MeasurementTabWidget
```

5.34.3.10 def Widgets.MeasurementTabWidget.MeasurementTabWidget.open_depth_profile_settings (*self*)

```
Opens depth profile settings dialog.
```

5.34.3.11 def Widgets.MeasurementTabWidget.MeasurementTabWidget.open_element_losses (*self, parent*)

```
Opens element losses dialog.

Args:
    parent: MeasurementTabWidget
```

5.34.3.12 def Widgets.MeasurementTabWidget.MeasurementTabWidget.open_energy_spectrum (*self, parent*)

```
Opens energy spectrum dialog.

Args:
    parent: MeasurementTabWidget
```

5.34.3.13 def Widgets.MeasurementTabWidget.MeasurementTabWidget.open_measuring_unit_settings (*self*)

```
Opens measurement settings dialog.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Widgets/MeasurementTabWidget.py

5.35 Modules.MeasuringSettings.MeasuringSettings Class Reference

Public Member Functions

- def __init__
- def show
- def set_settings
- def load_settings
- def save_settings

Public Attributes

- **measuring_unit_settings_filename**
- **config**
- **use_settings**
- **element**
- **energy**
- **detector_angle**
- **target_angle**
- **time_of_flight_lenght**
- **carbon_foil_thickness**
- **target_density**
- **filepath**

5.35.1 Detailed Description

```
MeasuringSettings holds the all project specific measurement unit parameters.
```

5.35.2 Constructor & Destructor Documentation

5.35.2.1 def Modules.MeasuringSettings.MeasuringSettings.__init__ (*self, settings_filepath* = None)

```
Inits MeasuringSettings.

Args:
    settings_filepath: filepath for the settings file to be loaded.
```

5.35.3 Member Function Documentation

5.35.3.1 def Modules.MeasuringSettings.MeasuringSettings.load_settings (*self, filepath*)

```
Loads settings' parameters from the given filepath.

Args:
    filepath: Filepath to the settings file.
```

5.35.3.2 def Modules.MeasuringSettings.MeasuringSettings.save_settings (*self, filepath* = None)

```
Saves settings' parameters to the given filepath.

Args:
    filepath: Filepath to the settings file.
```

5.35.3.3 def Modules.MeasuringSettings.MeasuringSettings.set_settings (*self, dialog, used_settings* = None)

```
Takes inputted parameters from the given dialog and sets them to the
corresponding object's parameters

Args:
    dialog: Measuring Settings QDialog from which the parameters are taken.
```

5.35.3.4 def Modules.MeasuringSettings.MeasuringSettings.show (*self, dialog*)

```
Shows the measuring parameters in the given measuring settings dialog.

Args:
    dialog: Measuring Settings QDialog whose fields are updated with the
    MeasuringSettings parameters.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/MeasuringSettings.py

5.36 Modules.Null.Null Class Reference

Public Member Functions

- def **__init__**
- def **__call__**
- def **__getattr__**
- def **__setattr__**
- def **__delattr__**
- def **__repr__**
- def **__str__**

5.36.1 Detailed Description

```
A class for implementing Null objects.

This class ignores all parameters passed when constructing or
calling instances and traps all attribute and method requests.
Instances of it always (and reliably) do 'nothing'.

The code might benefit from implementing some further special
Python methods depending on the context in which its instances
are used. Especially when comparing and coercing Null objects
the respective methods' implementation will depend very much
on the environment and, hence, these special methods are not
provided here.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/Null.py

5.37 potku.Potku Class Reference

Inherits QMainWindow.

Public Member Functions

- def __init__
- def open_about_dialog
- def hide_panel
- def current_measurement_save_cuts
- def current_measurement_analyze_elemental_losses
- def current_measurement_create_energy_spectrum

- def current_measurement_create_depth_profile
- def delete_selections
- def focus_selected_tab
- def remove_tab
- def open_project_settings
- def open_global_settings
- def open_new_measurement
- def make_new_project
- def open_project

Public Attributes

- **ui**
- **title**
- **icon_manager**
- **settings**
- **project**
- **masses**
- **measurement_tab_widgets**
- **tab_id**
- **panel_shown**

5.37.1 Detailed Description

```
Potku is main window class.
```

5.37.2 Constructor & Destructor Documentation

5.37.2.1 def potku.Potku.__init__ (*self*)

```
Init main window for Potku.
```

5.37.3 Member Function Documentation

5.37.3.1 def potku.Potku.current_measurement_analyze_elemental_losses (*self*)

```
Opens the element losses analyzation tool for the current open
measurement tab widget.
```

5.37.3.2 def potku.Potku.current_measurement_create_depth_profile (*self*)

```
Opens the depth profile analyzation tool for the current open
measurement tab widget.
```

5.37.3.3 def potku.Potku.current_measurement_create_energy_spectrum (*self*)

```
Opens the energy spectrum analyzation tool for the current open
measurement tab widget.
```

5.37.3.4 def potku.Potku.current_measurement_save_cuts (*self*)

```
Saves the current open measurement tab widget's selected cuts
to cut files.
```

5.37.3.5 def potku.Potku.delete_selections (*self*)

```
Deletes the selected tree widget items.
```

5.37.3.6 def potku.Potku.focus_selected_tab (*self, clicked_item*)

```
Focus to selected tab (in tree widget) and if it isn't open, open it.

Args:
    clicked_item: TreeWidgetItem with tab_id attribute (int) that connects
    the item to the corresponding MeasurementTabWidget
```

5.37.3.7 def potku.Potku.hide_panel (*self, enable_hide* = None)

```
Sets the frame (including measurement navigation view, global
settings and project settings buttons) visible.

Args:
    enable_hide: If True, sets the frame visible and vice versa.
    If not given, sets the frame visible or hidden depending its
    previous state.
```

5.37.3.8 def potku.Potku.make_new_project (*self*)

```
Opens a dialog for creating a new project.
```

5.37.3.9 def potku.Potku.open_about_dialog (*self*)

```
Show Potku program about dialog.
```

5.37.3.10 def potku.Potku.open_global_settings (*self*)

```
Opens global settings dialog.
```

5.37.3.11 def potku.Potku.open_new_measurement (*self*)

```
Opens file an open dialog and if filename is given opens new measurement
from it.
```

5.37.3.12 def potku.Potku.open_project (*self*)

```
Shows a dialog to open a project.
```

5.37.3.13 def potku.Potku.open_project_settings (*self*)

```
Opens project settings dialog.
```

5.37.3.14 def potku.Potku.remove_tab (*self, tab_index*)

```
Remove tab which's close button has been pressed.

Args:
    tab_index: Integer representing index of the current tab
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/potku.py

5.38 Modules.Project.Project Class Reference

Public Member Functions

- def __init__
- def get_measurements_files
- def load
- def save

Public Attributes

- **directory**
- **project_name**
- **settings**
- **global_settings**
- **masses**
- **statusbar**
- **measurements**
- **project_file**

5.38.1 Detailed Description

```
Project class to handle all measurements.
```

5.38.2 Constructor & Destructor Documentation

5.38.2.1 def Modules.Project.Project.__init__ (*self, directory, masses, statusbar, global_settings*)

```
Inits Project class.

Args:
    directory: String representing project directory
    masses: Reference to Masses (class) object.
    statusbar: QtGui.QMainWindow's QStatusBar
    global_settings: Reference to GlobalSettings object (of the program)
```

5.38.3 Member Function Documentation

5.38.3.1 def Modules.Project.Project.get_measurements_files (*self*)

```
Get measurements files inside project folder.
```

5.38.3.2 def Modules.Project.Project.load (*self*)

```
Load project
```

5.38.3.3 def Modules.Project.Project.save (*self*)

```
Save project
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/Project.py

5.39 Dialogs.ProjectNewDialog.ProjectNewDialog Class Reference

Inherits QDialog.

Public Member Functions

- def __init__

Public Attributes

- **parent**
- **folder**
- **directory**
- **ui**
- **default_directory_used**
- **name**

5.39.1 Detailed Description

```
Dialog creating a new project.
```

5.39.2 Constructor & Destructor Documentation

5.39.2.1 def Dialogs.ProjectNewDialog.ProjectNewDialog.__init__ (*self, parent*)

```
Inits energy spectrum dialog.

Args:
    parent: Ibasoft class object.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Dialogs/ProjectNewDialog.py

5.40 Modules.Selection.Selection Class Reference

Public Member Functions

- def __init__
- def add_point
- def undo_last
- def get_points
- def get_first
- def get_last
- def count
- def end_selection
- def delete
- def draw
- def set_color
- def reset_color
- def save_string
- def transpose
- def point_inside

Public Attributes

- **id**
- **masses**
- **element_colormap**
- **settings**
- **default_color**
- **type**
- **element**
- **weight_factor**
- **element_scatter**
- **is_closed**
- **points**
- **axes**
- **axes_limits**

Static Public Attributes

- string **LINE_STYLE** = '-'
- string **LINE_MARKER** = 'o'
- float **LINE_MARKER_SIZE** = 3.0
- int **GLOBAL_ID** = 0

5.40.1 Detailed Description

```
Selection object which knows all selection points.
```

5.40.2 Constructor & Destructor Documentation

5.40.2.1 def Modules.Selection.Selection.__init__ (*self, axes, masses, element_colormap, settings, element* = None, *isotope* = None, *element_type* = "ERD", *color* = None, *points* = None, *scatter* = None, *weight_factor* = 1, *transposed* = False)

```
Inits Selection class.

Args:
    axes: Matplotlib FigureCanvas's subplot
    masses: Reference to element masses object of main program.
    element_colormap: Default colors for new element selections.
    settings: Measurement's settings to which selector belongs.
      (for selection dialog)
    element: String representing element
    isotope: String representing isotope
    element_type: "ERD" or "RBS"
    color: String representing color for the element
    points: String list representing points in selection.
    "X1, X2, X3;Y1, Y2, Y3"
    scatter: String representing scatter element.
    weight_factor: Weight factor for the element.
    transposed: Boolean representing if axes are transposed.
```

5.40.3 Member Function Documentation

5.40.3.1 def Modules.Selection.Selection.add_point (*self, point*)

```
Adds a point to selection.

Adds a point to selection. If selection is closed, do nothing.

Args:
    point: Point (x, y) to be added to selection.

Return:
    0: Point was added.
    -1: If selection is closed.
```

5.40.3.2 def Modules.Selection.Selection.count (*self*)

```
Get count of points in selection

Return
    Integer: Count of points in selection
```

5.40.3.3 def Modules.Selection.Selection.delete (*self*)

```
Delete this selection.
```

5.40.3.4 def Modules.Selection.Selection.draw (*self*)

```
Draw selection points into graph (matplotlib) axes
```

5.40.3.5 def Modules.Selection.Selection.end_selection (*self, canvas* = None)

```
End selection.
```

```
Ends selection. If selection is open and canvas is not None, it will
show dialog to select element information and draws into canvas
before opening the dialog.

Args:
    canvas: Matplotlib's FigureCanvas or None when we don't want
    to new selection window. None, when loading selections
    so we do not want to open new selection settings dialog.

Return:
    True: Selection was completed
    False: Selection settings was not set (cancel button)
```

5.40.3.6 def Modules.Selection.Selection.get_first (*self*)

```
Get first point in selection

Return:
    None: If no point in selection
    (x, y): Otherwise
```

5.40.3.7 def Modules.Selection.Selection.get_last (*self*)

```
Get last point in selection

Return:
    None: If no point in selection
    (x, y): Otherwise
```

5.40.3.8 def Modules.Selection.Selection.get_points (*self*)

```
Get points in selection

Get points in selection in list. Format: ((x1,y1), (x2,y2), ...).
If no points, empty list is returned

Return:
   ((x1, y1), (x2, y2), ...)
```

5.40.3.9 def Modules.Selection.Selection.point_inside (*self, point*)

```
Check if point is inside selection.

Args:
    point: [X, Y] representing a point.

Return:
    Returns True if point is within selection. False otherwise.
```

5.40.3.10 def Modules.Selection.Selection.reset_color (*self*)

```
Reset selection color to default color.
```

5.40.3.11 def Modules.Selection.Selection.save_string (*self, is_transposed*)

```
Get selection in string format for selectiong file save.

Args:
    is_transposed: Boolean representing if axes are transposed.

Return:
    String representing current selection object.
```

5.40.3.12 def Modules.Selection.Selection.set_color (*self, color*)

```
Set selection color

Args:
    color: String representing color.
   Format is whatever QtGui.QColor(string) understands.
```

5.40.3.13 def Modules.Selection.Selection.transpose (*self, transpose*)

```
Transpose selection points.

Args:
    transpose: Boolean representing whether to transpose selection points.
```

5.40.3.14 def Modules.Selection.Selection.undo_last (*self*)

```
Undo last point in selection.

Return:
    1: If selection is closed or there are no points in selection.
    0: If everything is ok.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/Selection.py

5.41 Dialogs.SelectionDialog.SelectionSettingsDialog Class Reference

Inherits QDialog.

Public Member Functions

- def __init__

Public Attributes

- **selection**
- **settings**
- **element_colormap**
- **ui**
- **isOk**
- **color**

5.41.1 Detailed Description

```
Selection Settings dialog handles showing settings for selection made in
measurement (in matplotlib graph).
```

5.41.2 Constructor & Destructor Documentation

5.41.2.1 def Dialogs.SelectionDialog.SelectionSettingsDialog.__init__ (*self, selection*)

```
Inits selection settings dialog.

Args:
    selection: Selection class object.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Dialogs/SelectionDialog.py

5.42 Modules.Selection.Selector Class Reference

Public Member Functions

- def __init__
- def count
- def is_empty
- def get_at
- def get_selected
- def add_point
- def undo_point
- def purge
- def remove_selected
- def remove_all
- def distance
- def draw
- def end_open_selection
- def select
- def reset_select
- def reset_colors
- def get_colors
- def grey_out_except
- def auto_save
- def load
- def update_axes_limits
- def transpose

Public Attributes

- **element_colormap**
- **settings**
- **measurement_name**
- **directory**
- **selection_file**
- **selections**
- **new_selection_is_allowed**

- **is_transposed**
- **looseness**
- **axes**
- **axes_limits**
- **selected_id**
- **draw_legend**
- **masses**

5.42.1 Detailed Description

```
Selector objects handles all selections within measurement.
```

5.42.2 Constructor & Destructor Documentation

5.42.2.1 def Modules.Selection.Selector.__init__ (*self, directory, measurement_name, masses, element_colormap, settings*)

```
Inits Selector.

Inits Selector object.

Args:
    filepath: String representing filepath of measurement data (ascii file).
    masses: Reference to element masses object of main program.
    element_colormap: Default colors for new element selections.
    settings: Measurement's settings to which selector belongs.
      (for selection dialog)
```

5.42.3 Member Function Documentation

5.42.3.1 def Modules.Selection.Selector.add_point (*self, point, canvas*)

```
Adds a new point.

Adds a new point to last selection. If new selection is allowed, create
a new selection to which point is added. If point is in close proximity
of first point in (last) Selection, then close selection and allow new
selection to be made.

Args:
    point: Point (x, y) to be added to selection.
    canvas: matplotlib's FigureCanvas where selections are drawn.

Return:
    1: When point closes open selection and allows new selection to
be made.
    0: When point was added to open selection.
    -1: When new selection is not allowed and there are no selections.
```

5.42.3.2 def Modules.Selection.Selector.auto_save (*self*)

```
Save all selections into a file.
```

5.42.3.3 def Modules.Selection.Selector.count (*self*)

```
Get count of selections.

Return:
    Returns the count of selections in selector object.
```

5.42.3.4 def Modules.Selection.Selector.distance (*self, p0, p1*)

```
Distance between points

Calculates and returns distance between two points.

Args:
    p0: Point A
    p1: Point B

Return:
    Distance (float) between two points.
```

5.42.3.5 def Modules.Selection.Selector.draw (*self*)

```
Draw selections.

Issue draw to all selections in selector.
```

5.42.3.6 def Modules.Selection.Selector.end_open_selection (*self, canvas*)

```
End last open selection.

Ends last open selection. If selection is open, it will show dialog to
select element information and draws into canvas before opening the dialog.

Args:
    canvas: Matplotlib's FigureCanvas

Return:
    1: If selection closed
    0: Otherwise
```

5.42.3.7 def Modules.Selection.Selector.get_at (*self, index*)

```
Get selection at index.

Args:
    index: Integer of index we want to get from selections.

Return:
    Returns Selection at said index. If index is out of range, returns None.
```

5.42.3.8 def Modules.Selection.Selector.get_colors (*self*)

```
Get colors of each selection in selector.

Return:
    Returns dictionary of all element selections and their colors.
```

5.42.3.9 def Modules.Selection.Selector.get_selected (*self*)

```
Get currently selected selection.

Return:
    Returns Selection of selected Selection on matplotlib graph. If none
    selected, returns None.
```

5.42.3.10 def Modules.Selection.Selector.grey_out_except (*self, selected_id*)

```
Grey out all selections except selected one.

Sets all selections' colors to grey except selected, which is set to red.

Args:
    selected_id: Integer of selected selection id
```

5.42.3.11 def Modules.Selection.Selector.is_empty (*self*)

```
Check if no selections.

Return:
    Returns True if no selections.
```

5.42.3.12 def Modules.Selection.Selector.load (*self, filename*)

```
Load selections from a file.

Removes all current selections and loads selections from given filename.

Args:
    filename: String representing (full) path to selection file.
```

5.42.3.13 def Modules.Selection.Selector.purge (*self*)

```
Purges (removes) all open selections and allows new selection to be made.
```

5.42.3.14 def Modules.Selection.Selector.remove_all (*self*)

```
Remove all selections in selector.
```

5.42.3.15 def Modules.Selection.Selector.remove_selected (*self*)

```
Remove selected selection.

Removes selected selection if one is selected. Otherwise do nothing.
```

5.42.3.16 def Modules.Selection.Selector.reset_colors (*self*)

```
Reset selection colors.

Reset all selections' colors to their default values.
```

5.42.3.17 def Modules.Selection.Selector.reset_select (*self*)

```
Reset selection to None.

Resets current selection to None and resets colors of all selections
to their default values.
```

5.42.3.18 def Modules.Selection.Selector.select (*self, point, highlight* = `True`)

```
Select a selection based on point.

Args:
    point: Point (x, y) which is clicked on the graph to select selection.
    highlight: Boolean to determine whether to highlight just this
       selection.

Return:
    1: If point is within selection.
    0: If point is not within selection.
```

5.42.3.19 def Modules.Selection.Selector.transpose (*self, is_transposed*)

```
Transpose graph axes.

Args:
    is_transposed: Boolean representing whether axes are transposed.
```

5.42.3.20 def Modules.Selection.Selector.undo_point (*self*)

```
Undo last point in open (last) selection.

Undo last point in open (last) selection. If there are no selections,
do nothing.
```

5.42.3.21 def Modules.Selection.Selector.update_axes_limits (*self*)

```
Update selector's axes limits based on all points in all selections.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/Selection.py

5.43 Modules.Settings.Settings Class Reference

Public Member Functions

- def __init__
- def get_measurement_settings

Public Attributes

- **project_settings**
- **measuring_unit_settings**
- **calibration_settings**
- **depth_profile_settings**

5.43.1 Detailed Description

```
Settings class to handle settings of project and measurement.
```

5.43.2 Constructor & Destructor Documentation

5.43.2.1 def Modules.Settings.Settings.__init__ (*self*, *directory* = None, *project_settings* = None)

```
Inits Settings class.

Args:
    directory: String representing directory for settings.
    project_settings: Settings class object of project.
```

5.43.3 Member Function Documentation

5.43.3.1 def Modules.Settings.Settings.get_measurement_settings (*self*)

```
Get the measurement specific settings.

Get currently used settings by measurement. If measurement uses project
settings (by default), it will return project's settings instead.

Returns:
    Settings object that has all the references to settings that a
    measurement uses.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/Settings.py

5.44 Modules.Calibration.TOFCalibration Class Reference

Public Member Functions

- def __init__
- def add_point
- def remove_point
- def point_exists
- def get_points
- def linear_function
- def fit_linear_function
- def get_linear_fit_points
- def get_fit_parameters

Public Attributes

- **slope**
- **offset**
- **tof_points**

5.44.1 Detailed Description

```
Class for holding list of TOFCalibrationPoints and creating a linear fit
of their values.
```

5.44.2 Constructor & Destructor Documentation

5.44.2.1 def Modules.Calibration.TOFCalibration.__init__ (*self*)

```
Inits the class
```

5.44.3 Member Function Documentation

5.44.3.1 def Modules.Calibration.TOFCalibration.add_point (*self, tof_calibration_point*)

```
Adds a TOFCalibrationPoint to ToF Calibration

Args:
    tof_calibration_point: TOFCalibrationPoint class object.
```

5.44.3.2 def Modules.Calibration.TOFCalibration.fit_linear_function (*self, x, y, guess_a, guess_b*)

```
Fits a linear function to the given data.
a*x + b

Args:
    x: data's x axis as a list
    y: data's y axis as a list
    guess_x0: Guess for the a's value
    guess_A: Guess for the b's value

Returns:
    tuple(a, b) of parameters of a fitted linear function.
```

5.44.3.3 def Modules.Calibration.TOFCalibration.get_fit_parameters (*self*)

```
Get fit parameters.

Return:
    Returns Slope and Offset of calibration.
```

5.44.3.4 def Modules.Calibration.TOFCalibration.get_linear_fit_points (*self, params, x_min, x_max, points_in_range*)

```
Generates points from the linear function with given range and
number of points.

Args:
    params: tuple of parameters (x0, A, k)
    x_min:
    x_max:
    points_in_range:

Returns:
    tuple(x_values, y_values) of generated lists of axis data (x and y axis)
```

5.44.3.5 def Modules.Calibration.TOFCalibration.get_points (*self*)

```
Returns TOFCalibrationPoints that have the point_used property set True.

Return:
    tuple(x,y, name) of lists containing used points for the linear fit.
```

5.44.3.6 def Modules.Calibration.TOFCalibration.linear_function (*self, x, params*)

```
The function used for linear fit. Takes the function parameters as a "namedtuple" or "tuple".
a*x + b

Args:
    params: namedtuple or tuple that brings the used parameters ("a b")

Return:
    Returns linear function value from the given point x.
```

5.44.3.7 def Modules.Calibration.TOFCalibration.point_exists (*self, tof_calibration_point*)

```
Check if point exists in ToF Calibration.

Args:
    tof_calibration_point: TOFCalibrationPoint class object.

Return:
    Returns True if point exists. False otherwise.
```

5.44.3.8 def Modules.Calibration.TOFCalibration.remove_point (*self, tof_calibration_point*)

```
Removes a TOFCalibrationPoint from ToF Calibration

Args:
    tof_calibration_point: TOFCalibrationPoint class object.

Return:
    Returns True if point was removed. False otherwise.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/Calibration.py

5.45 Modules.Calibration.TOFCalibrationHistogram Class Reference

Public Member Functions

- def __init__
- def get_error_function_parameters
- def error_function
- def fit_error_function
- def find_middle
- def get_curve_fit_points

Public Attributes

- **cut**
- **bin_width**
- **use_column**
- **histogram_x**
- **histogram_y**

5.45.1 Detailed Description

```
Class for creating a histogram based on a cut file data. Can make a curve
fit to histogram's front edge.
```

5.45.2 Constructor & Destructor Documentation

5.45.2.1 def Modules.Calibration.TOFCalibrationHistogram.__init__ (*self, cut, bin_width, use_column* = 1)

```
Inits the class.

Args:
```

```
    cut: CutFile that is used to make a histogram.
    bin_width: Created histograms bin width
    use_column: Which column of the CutFile's data is used to create a
    histogram.
```

5.45.3 Member Function Documentation

5.45.3.1 def Modules.Calibration.TOFCalibrationHistogram.error_function (*self, x, params*)

```
The function used for fit.

Takes the function parameters as a "namedtuple" or "tuple".
A * (erf((x - x0) / k) + 1) / 2

Args:
    x: Float representing value on X axis.
    params: namedtuple or tuple that brings the used parameters ("x0 A k").

Return:
    Returns calculated error function value for x.
```

5.45.3.2 def Modules.Calibration.TOFCalibrationHistogram.find_middle (*self*)

```
Finds the point at x axis that is somewhere in the middle of the histogram.
This is very inaccurate way.

Return:
    The value at the histogram's x axis that is somewhere in the middle of
    the top of the graph.
```

5.45.3.3 def Modules.Calibration.TOFCalibrationHistogram.fit_error_function (*self, x, y, guess_x0, guess_A, guess_k*)

```
Fits a error function to the given data.

Args:
    x: data's x axis a list
    y: data's y axis a list
    guess_x0: Guess for the x_0's value
    guess_A: Guess for the A's value
    guess_k: Guess for the k's value

Return:
    tuple(x0, A, k) of parameters of a fitted error function.
```

5.45.3.4 def Modules.Calibration.TOFCalibrationHistogram.get_curve_fit_points (*self, params, points_in_range*)

```
Generates points from the error function with the histogram's range

Args:
    params: tuple of parameters (x0, A, k)

Return:
    tuple(xp, pxp) of generated lists of axis data (x and y axis)
```

5.45.3.5 def Modules.Calibration.TOFCalibrationHistogram.get_error_function_parameters (*self, end_of_front_edge, start_of_front_edge =* 0)

```
Get the parameters of the fitted curve. Parameters are used to specify
the range where the curve fit is made.
```

```
Args:
    end_of_front_edge: End of the histogram's range in x axis.
    start_of_front_edge: Start of the histogram's range in x axis.

Return:
    Tuple of fit function parameters (x0, A, k).
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/Calibration.py

5.46 Modules.Calibration.TOFCalibrationPoint Class Reference

Public Member Functions

- def __init__
- def get_tof_channel
- def get_tof_seconds
- def get_name
- def calculate_time_of_flight
- def get_point

Public Attributes

- **cut**
- **type**
- **point_used**
- **masses**
- **recoiled_mass**
- **beam_mass**
- **beam_energy**
- **lenght**
- **target_angle**
- **scatter_element_mass_kg**
- **carbon_thickness**
- **stopping_energy**
- **time_of_flight_channel**
- **time_of_flight_seconds**

5.46.1 Detailed Description

```
Class for the calculation of a theoretical time of flight.
```

5.46.2 Constructor & Destructor Documentation

5.46.2.1 def Modules.Calibration.TOFCalibrationPoint.__init__ (*self, time_of_flight, cut, masses, settings*)

```
Inits the class.

Args:
    time_of_flight:
    cut: CutFile class object.
    masses: Reference to Masses class object.
    settings: Settings class object.
```

5.46.3 Member Function Documentation

5.46.3.1 def Modules.Calibration.TOFCalibrationPoint.calculate_time_of_flight (*self*)

```
Calculates the time of flight.
    In case of ERD use:

    t = l/(sqrt( 2 * (k * E_I0 - dE_RT1) / M_R))
    where:

    E_I0 = beam energy
    dE_RT1 = stopping energy of the recoiled particle
    M_R = mass of the recoiled particle
    M_I = mass of the scattered particle
    k = kinetic factor, which is (4 * M_I * M_R * cos(a)^2) / (M_I + M_R)^2

    In case of RBS use:

    t = l/(sqrt( 2 * (k * E_I0 - dE_IT1) / M_R))
    where:

    dE_RT1 = stopping energy of the scattered particle
    M_R = mass of the recoiled particle
    M_I = mass of the scattered particle
    k = kinetic factor, which is (sqrt(( M_R^2 - M_I^2 * cos(a)^2) + M_I *
    cos(a)) / (M_I + M_R))^2


    Return:
Calculated time of flight as float. None if the cut file's type is
not either ERD or RBS.
```

5.46.3.2 def Modules.Calibration.TOFCalibrationPoint.get_name (*self*)

```
Get name of the used CutFile.

Return:
    Returns name of the used CutFile.
```

5.46.3.3 def Modules.Calibration.TOFCalibrationPoint.get_point (*self*)

```
Get TOFCalibrationPoint values in tuple.

Return:
    Returns TOFCalibrationPoint values in tuple.
```

5.46.3.4 def Modules.Calibration.TOFCalibrationPoint.get_tof_channel (*self*)

```
Get Time of Flight channel.

Return:
    Returns Time of Flight channel.
```

5.46.3.5 def Modules.Calibration.TOFCalibrationPoint.get_tof_seconds (*self*)

```
Get Time of Flight seconds.

Return:
    Returns Time of Flight seconds.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Modules/Calibration.py

5.47 Dialogs.GraphSettingsDialog.TofeGraphSettingsWidget Class Reference

Inherits QDialog.

Public Member Functions

- def __init__
- def accept_settings

Public Attributes

- **parent**
- **ui**

5.47.1 Detailed Description

5.47.2 Constructor & Destructor Documentation

5.47.2.1 def Dialogs.GraphSettingsDialog.TofeGraphSettingsWidget.__init__ (*self, parent*)

```
Inits ToF-E graph histogram graph settings dialog.

Args:
    parent: MatplotlibHistogramWidget which settings are being changed.
```

5.47.3 Member Function Documentation

5.47.3.1 def Dialogs.GraphSettingsDialog.TofeGraphSettingsWidget.accept_settings (*self*)

```
Accept changed settings and save them.
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Dialogs/GraphSettingsDialog.py

5.48 Widgets.TofeHistogramWidget.TofeHistogramWidget Class Reference

Inherits QWidget.

Public Member Functions

- def __init__
- def set_cut_button_enabled

Public Attributes

- **ui**
- **matplotlib**
- **measurement**

5.48.1 Detailed Description

```
HistogramWidget

Used to draw ToF-E Histograms
```

5.48.2 Constructor & Destructor Documentation

5.48.2.1 def Widgets.TofeHistogramWidget.TofeHistogramWidget.__init__ (*self, measurement, icon_manager*)

```
Inits TofeHistogramWidget widget.

Args:
    measurement: Measurement class object.
    icon_manager: IconManager class object.
```

5.48.3 Member Function Documentation

5.48.3.1 def Widgets.TofeHistogramWidget.TofeHistogramWidget.set_cut_button_enabled (*self, selections* = None)

```
Enables save cuts button if the given selections list's lenght is not 0.
Otherwise disable.

Args:
    selections: list of Selection objects
```

The documentation for this class was generated from the following file:

- C:/MyTemp/ibasoft/ibasoft/Widgets/TofeHistogramWidget.py

Index