easyInterface

Release 0.0.5

Simon Ward

Feb 18, 2020

CONTENTS:

CHAPTER

ONE

EASYINTERFACE CRYSPY CALCULATOR

class easyInterface.Diffraction.Calculators.**CryspyCalculator**(*main_rcif_path=None*)

asCifDict()

...

Return type dict

getPhases()

Set phases (sample model tab in GUI)

Return type Phases

refine()

refinement ...

setExperiments(*experiments*)

Set experiments (Experimental data tab in GUI)

setObjFromProjectDicts(*phases*, *experiments*)

Set all the cryspy parameters from project dictionary

setPhases(*phases*)

Set phases (sample model tab in GUI)

CHAPTER

TWO

EASYINTERFACE INTERFACE

class easyInterface.Diffraction.Interface.**CalculatorInterface**(*calculator*)

Interface to calculators in the *easyInterface.Diffraction.Calculator* class.

addExperimentDefinition(*exp_path*)

Add an experiment to be simulated from a cif file. Note that this will not have any crystallographic phases associated with it.

Parameters **exp_path** (str) – Path to a experiment file (*.cif*)

addPhase(*phase*)

Add a new phases from a cif file to the list of existing crystal phases.

Parameters **phase** (Phase) –

addPhaseDefinition(*phase_path*)

Add a new phases from a cif file to the list of existing crystal phases.

Parameters **phase_path** (str) – Path to a phase definition file (*.cif*)

Example:

```
interface = CalculatorInterface(calculator)
phase_path = '~/Experiments/new_phase.cif'
interface.addPhaseDefinition(phase_path)
```

asCifDict()

...

Return type dict

asDict()

Return data dict.

Return type dict

experimentsCount()

Returns number of experiments in the project.

Return type int

experimentsIds()

Returns labels of the experiments in the project.

Return type list

property final_chi_square

Calculates the final chi squared of the simulation. Where the final chi squared is the chi squared divided by the number of data points.

Return type `float`

Returns Final chi squared

getPhase(*phase_name*)
Returns a phase from the project dictionary by name if one is supplied. If the phase name is none then all phases are returned. If the phase name does not exist KeyError is thrown. :type phase_name: `Optional[str]` :param phase_name: Name of the phase to de returned or None for all phases :rtype: `Phase` :return: Copy of the project dictionaries phase object with name phase_name :raises KeyError:

phasesCount()
Returns number of phases in the project.

Return type `int`

phasesIds()
Returns labels of the phases in the project.

Return type `list`

refine()
refinement ...

Return type `dict`

setExperiment(*experiment*)
Set phases (sample model tab in GUI)

setExperimentDefinition(*exp_path*)
Parse the relevant phases file and update the corresponding model

setExperiments(*experiments=None*)
Set experiments (Experimental data tab in GUI)

setPhase(*phase*)
Set phases (sample model tab in GUI)

setPhaseDefinition(*phase_path*)
Parse a phases cif file and replace existing crystal phases

Parameters **phase_path** (`str`) – Path to new phase definition file (*.cif*)

Example:

```
interface = CalculatorInterface(calculator)
phase_path = '~/Experiments/phases.cif'
interface.setPhaseDefinition(phase_path)
```

setPhases(*phases=None*)
Set phases (sample model tab in GUI)

setProjectFromCalculator()
Sets the project dictionary from the calculator given on initialisation. Calling this function will regenerate the project dictionary and changes may be lost.

CHAPTER

THREE

EASYINTERFACES PROJECT DICTIONARY

class easyInterface.Diffraction.Interface.**ProjectDict**(*interface*, *app*, *calculator*, *info*, *phases*, *experiments*, *calculations*)

This class deals with the creation and modification of the main project dictionary.

classmethod default()

Create a default and empty project dictionary

Return type *ProjectDict*

Returns Default project dictionary with undo/redo functionality

classmethod fromPars(*experiments*, *phases*, *calculations*)

Create a main project dictionary from phases and experiments.

Parameters

- **experiments** (Union[Experiments, Experiment, List[Experiment]]) – A collection of experiments to be compared to calculations
- **phases** (Union[Phases, Phase, List[Phase]]) – A Collection of crystallographic phases to be calculated

Return type *ProjectDict*

Returns Project dictionary with undo/redo

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

INDEX