
Assign_GUP Documentation

Release 0.g1faac9e.dirty

Pete R. Jemian

Sep 27, 2017

Contents

1	Overview	3
2	Contents	5
3	Indices and tables	41
	Python Module Index	43

Assist in assigning APS GUPs to PRP members

author Pete R. Jemian

email jemian@anl.gov

copyright 2005-2017, UChicago Argonne, LLC

license ANL OPEN SOURCE LICENSE (see *LICENSE*)

docs http://assign_gup.readthedocs.io

git https://github.com/prjemian/assign_gup.git

PyPI https://pypi.python.org/pypi/assign_gup

TODO list https://github.com/prjemian/assign_gup/issues

version 2017.4.0

release 0.g1faac9e.dirty

published Sep 27, 2017

CHAPTER 1

Overview

Assign_GUP is a tool for the Proposal Review Panel of the Advanced Photon Source Small-Angle X-ray Scattering General User Proposal panel. The tool helps the panel chairman to assign the most appropriate reviewers to each panel while balancing the work load across the panel. It replaces a system previously conducted in Excel spreadsheets.

A goal of this software is to automate this process, as much as possible. During the development process of this project, it is expected to discover, identify, and encode the algorithms underlying the proposal assignment process.

The code was first used in the 2011-2 proposal review cycle and was of some, albeit limited, help.

In the 2011-3 cycle, the software was much more useful (although the distribution of panelists amongst the proposals might have contributed) and the entire process was completed within a few hours. Previously, the process would take about a business day.

2.1 Instructions for the Review Panel Chairman

Outline

1. Before the proposal deadline
 - (a) Verify the members of the proposal review panel: *List of Reviewers*
 - (b) Answer questions from review panel members. CC the APS User Office.
2. After the proposal deadline
 - (a) *List of Proposals*
 - (b) Run the Assign_GUP software.
 - i. Create a new project file
 - ii. Import the proposals (from the XML file)
 - iii. Create the list of reviewers or Import from previous .agup file
 - iv. Edit reviewers to assign topic weights
 - v. *Assign topic weights to each proposal*
 - vi. *Assign Reviewers to Proposals*
 - vii. *Update the Assignments on the Web site*
 - viii. Prepare the boilerplate letter
 - ix. *Send emails*

2.1.1 List of Reviewers

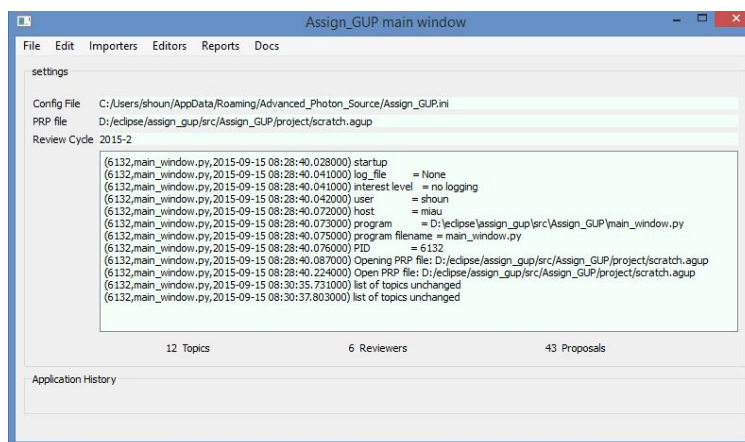


Fig. 2.1: Assign_GUP main window

At least a month before the GUP deadline...

Check the reviewers and verify they all can attend on the review date. (GUP Calendar: http://www.aps.anl.gov/Users/Calendars/GUP_Calendar.htm)

1. Identify panel members whose terms expire soon.
2. Is a temporary reviewer necessary?
3. Identify new candidates to replace expiring panel members. Evaluate panel members strengths in the various topics for review and compare with proposal weights in each topic. This will point out whether the panel has sufficient strength to review proposals on each topic.
4. Confirm the list of reviewers for the upcoming cycle.
5. Verify that each member can attend.
6. Assign values for reviewer strength on each topic. Use a scale from 0 (no strength) to 1.0 (high strength), with 0.1 precision.

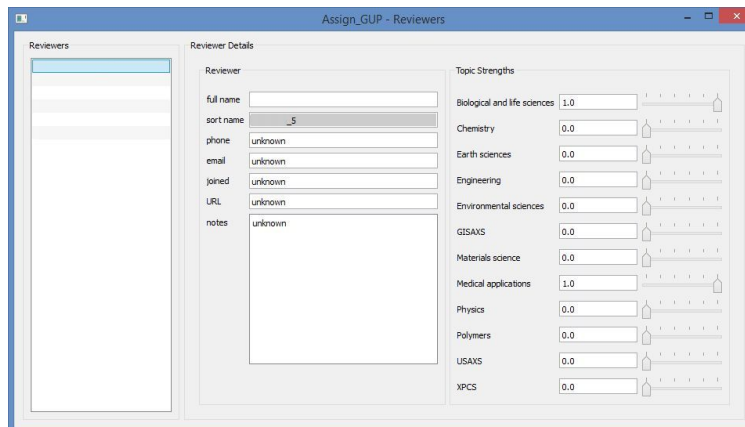


Fig. 2.2: Editor: Reviewers

Note: In various example screen images shown in this documentation, the details of specific individuals or proposals

have been removed or obscured to preserve anonymity.

2.1.2 List of Proposals

Be prompt!

Download the list in XML version of the proposals just after the proposal deadline. Assignments should be received within just a few days. Assigned reviewers must have sufficient to make their reviews.

1. Login to your panel's proposal review web site, and click the "Download the list in XML" link. Each panel has a slightly different URL. Consult the following table or contact the APS User Office:

panel	URL
small-angle	https://beam.aps.anl.gov/pls/apsweb/gup0008.panel_start_page?i_attrib=246B2411-PNL5
spectroscopy	https://beam.aps.anl.gov/pls/apsweb/gup0008.panel_start_page?i_attrib=246B2412-PNL6
proposal system	https://www1.aps.anl.gov/Users-Information/About-Proposals/Apply-for-Time

You might also consider using the "Print All Proposals" link to generate a single PDF file containing all the proposals and attachments. Any problem attachments will be noted on separate pages at the end of the document. Be sure to save that on-screen image to a file or it will go away when you close your browser.

2. When that page appears, it will look unformatted and not like XML. Don't worry, underneath the page is XML but is rendered poorly.
3. In your browser, choose "view source" (Control-U on Firefox) and save to a file such as raw-proposals-2015-2.xml (use the name of the proper review cycle).
4. This raw file may have characters that break the later XML processing. That's why I save it as raw and then make a working copy next.
5. Copy this file to proposals-2015-2.xml and make one change first. Add a period attribute (if not present) to the root element Review_list with a value of the current review cycle, such as:

```
<Review_list period="2015-2">
```

Be on the lookout for:

- proposals that belong on another panel
- PUP or project proposals (they require additional reviewer work)
- identical or related proposals (may not be the same PI or beam line)
- ineligible reviewer(s) (ineligible since reviewer name is on the proposal)

2.1.3 List of Topics

Central to this process is the creation of a list of topics. These topics represent the various scientific or experimental subjects represented by this suite of proposals. Each of the reviewers on the panel will have a different strength in each of the topics. This strength is represented as a decimal topic value, ranging between zero (no strength) to 1.0 (expert

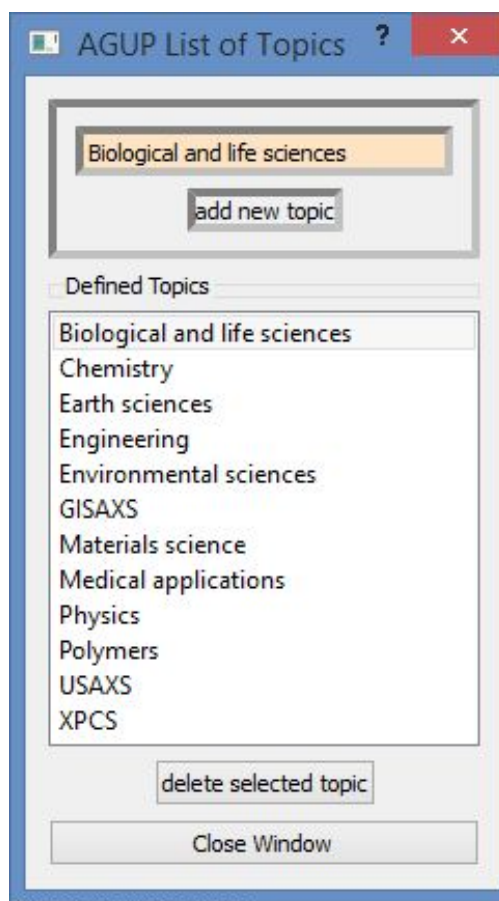


Fig. 2.3: Editor: topics

or confident). The *dot product* of the topic values between any reviewer and proposal can be used as a measure of how appropriate is that reviewer to evaluate that proposal.

Initially, the list of topics is extracted from the proposals from the list of subjects selected for each proposal by the proposal's author.

The list of topics may be edited to remove such nondescript names as “Other” and to add in instrument-specific techniques such as XPCS for which specific expertise may not be so common amongst the panel reviewers.

Once the list of topics has been modified, check both the list of reviewers and the list of proposals that the topic values are assigned properly. (New topics added will be given topic values of 0.0.)

2.1.4 Assign topic weights to each proposal

Similar to the reviewers, each proposal must be assigned a topic weight. The weight (a floating point decimal number) is the relevance of each topic to the proposal. The scale is 0 to 1 where 0 means *not related* and 1 mean *related*.

When the proposals XML file is imported, a list of topics is created from all the subjects as selected by the proposers. Each topic created is assigned a weight of 1.0. **These topic assessments should be verified** since some proposers put down lots of extraneous topics that are weakly related to the specifics of the proposal.

2.1.5 Assign Reviewers to Proposals

Choose a primary and a secondary reviewer for each proposal. Most likely, these will be reviewers with the strongest topic weighting for this proposal.

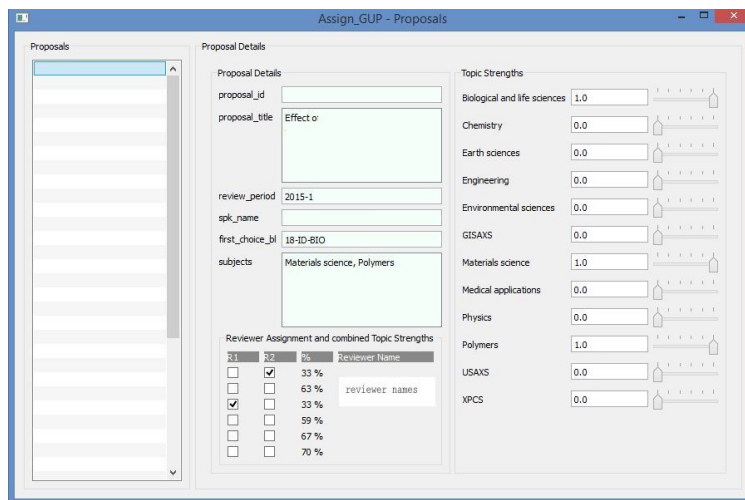


Fig. 2.4: Editor: Proposals

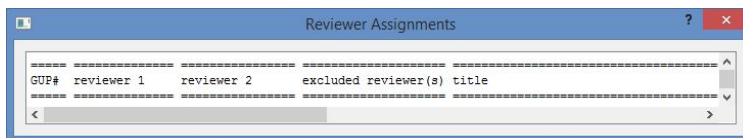
The rules are such that only one reviewer can be a *primary* and a different one can be *secondary*.

If a reviewer is ineligible to review a proposal (usually because they are named on the proposal team), their name will be grey and the checkboxes to select them as either primary or secondary reviewer will be disabled.

Refer to the summary and analysis grid reports when attempting to balance the number of proposals assigned to each reviewer.

2.1.6 Update the Assignments on the Web site

The assignments report gives a listing of the assigned reviewers for all proposals. Looking at this page, enter the same information into the APS Review web site. There is no upload capability possible for this data, you must enter it in from the web form.



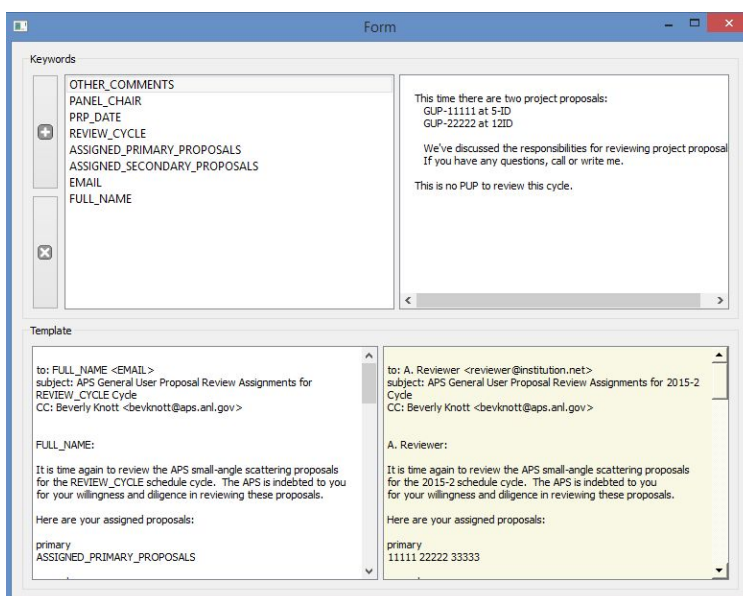
GUP#	reviewer 1	reviewer 2	excluded reviewer(s)	title

Fig. 2.5: Report: Reviewer assignments

2.1.7 Send emails

Send emails to each reviewer listing the proposals on which they are primary or secondary reviewer. CC the APS User Office on each email.

To prepare the template for the emails, open the *Edit Email Template ...* from the *Editors* menu:



Keywords

- OTHER_COMMENTS
- PANEL_CHAIR
- PRP_DATE
- REVIEW_CYCLE
- ASSIGNED_PRIMARY_PROPOSALS
- ASSIGNED_SECONDARY_PROPOSALS
- EMAIL
- FULL_NAME

Template

to: FULL_NAME <EMAIL>
 subject: APS General User Proposal Review Assignments for REVIEW_CYCLE Cycle
 CC: Beverly Knott <bevknott@aps.anl.gov>

FULL_NAME:
 It is time again to review the APS small-angle scattering proposals for the REVIEW_CYCLE schedule cycle. The APS is indebted to you for your willingness and diligence in reviewing these proposals.

Here are your assigned proposals:

primary
 ASSIGNED_PRIMARY_PROPOSALS

This time there are two project proposals:
 GUP-11111 at 5-ID
 GUP-22222 at 12-ID

We've discussed the responsibilities for reviewing project proposal. If you have any questions, call or write me.

This is no PUP to review this cycle.

to: A. Reviewer <reviewer@institution.net>
 subject: APS General User Proposal Review Assignments for 2015-2 Cycle
 CC: Beverly Knott <bevknott@aps.anl.gov>

A. Reviewer:
 It is time again to review the APS small-angle scattering proposals for the 2015-2 schedule cycle. The APS is indebted to you for your willingness and diligence in reviewing these proposals.

Here are your assigned proposals:

primary
 11111 22222 33333

Fig. 2.6: Editor: Email template and substitution keywords

To prepare the email text for each reviewer, select *Email Letters ...* from the *Reports* menu. A new window will appear for each reviewer with details specific to that reviewer. Use this to create an email to each reviewer.

2.2 How to Install Assign_GUP

The basic installation procedure:

1. install Python 2.7
2. install Assign_GUP

2.2.1 Background

This program requires Python 2.7 (not ready for Python 3 yet) and several additional packages. Most of the package dependencies are met by using a Python distribution (provides Python, the basic package suite, and other popular packages).

The major package requirements are:

- *PyQt4* : provides the graphical user interface widgets
- *lxml* : XML support
- *pyRestTable* : presents information in a sensible table format

2.2.2 Python

It is suggested to use the Anaconda Python 2.7 distribution¹ as it contains most of the packages used by this program.

1. use a web browser and visit: <https://www.continuum.io/downloads>
2. select the distribution for your operating system
3. make sure to follow links for *Python 2.7*
4. download the installer file
5. follow the instructions to install on your computer

2.2.3 Assign_GUP

Install this program from the Python Package Index (PyPI) using the *pip* command:

```
pip install pyRestTable
pip install --no-deps Assign_GUP
```

The `--no-deps` option tells *pip* not to download and attempt to build newer versions of other required packages such as *lxml*.

It may be necessary to install the *lxml* package if your distribution does not already have it installed. You can view all the installed packages using this command:

```
pip list
```

The list may have dozens or more items. To install *lxml*:

```
pip install lxml
```

Updating Assign_GUP

To update to a newer version of *Assign_GUP*, use this command:

```
pip install -U --no-deps Assign_GUP
```

The `-U` option tells *pip* to search for and install the latest package update.

¹ Anaconda Python 2.7: <https://www.continuum.io>

Alternative Installation steps

It is possible to install **Assign_GUP** using steps common to Python developers, such as:

```
pip install https://github.com/prjemian/assign_gup
```

or:

```
git clone https://github.com/prjemian/assign_gup.git
cd assign_gup
python ./setup.py install
```

These are not recommended for regular use.

2.3 Assign_GUP package: source code documentation

2.3.1 Module contents

Source code for the Assign_GUP (AGUP) package

2.3.2 Submodules

module: about

show the About box

```
class Assign_GUP.about.InfoBox (parent=None, settings=None)
    Bases: PyQt4.QtGui.QDialog
    a Qt GUI for the About box

    closeEvent (event)
        called when user clicks the big [X] to quit

    doDocsUrl ()
        opening documentation URL in default browser

    doIssuesUrl ()
        opening issues URL in default browser

    doLicense ()
        show the license

    doUrl (url)
        opening URL in default browser
```

module: agup_data

Data model for a review session: proposals, reviewers, topics, and analyses

```
class Assign_GUP.agup_data.AGUP_Data (parent_window=None, config=None)
    Bases: PyQt4.QtCore.QObject
    Complete data for a PRP review session
```


clearAllData ()
clear all data (except for self.settings)

getCycle ()
the review cycle, as defined by the proposals

importEmailTemplate (*xmlFile*)
import the email template support

importProposals (*xmlFile*)
import a Proposals XML file as generated by the APS or proposals from a PRP Project file

importReviewers (*xmlFile*)
import a complete set of reviewers (usually from a previous review cycle's file)
Completely replace the set of reviewers currently in place.

importTopics (*xmlFile*)
import a complete set of Topics (usually from a previous PRP Project file)
Completely replace the set of Topics currently in place.

openPrpFile (*filename*)

setReviewersModel (*rvwrs*)
set the data model for the list of reviewers

write (*filename*)
write this data to an XML file

module: auto_assignment

automatic assignment of reviewers to proposals

class Assign_GUP.auto_assignment.**Auto_Assign** (*agup*)
Bases: object

automatically assign reviewers to proposals

simpleAssignment (): assign the first two reviewers with the highest scores to *unassigned* proposals

getScores (*prop*)
generate scores for this proposal

simpleAssignment ()
assign the first two reviewers with the highest scores to *unassigned* proposals

- no attempt to balance assignment loads in this procedure
- score must be above zero to qualify

module: editor_email_template

edit the template to send emails, include editor for keyword substitutions

Provide tools to:

- manage list of substitution keywords
- manage values of substitution keywords
- display as read-only those substitution keywords defined for each reviewer
- provide editor for email template letter

- show example email with substitutions applied

The email template is stored in the project file.

The substitution keyword dictionary is (at present) stored in the settings file.

```
class Assign_GUP.editor_email_template.Editor (parent, agup, settings=None)
    Bases: PyQt4.QtGui.QWidget

    closeEvent (event)

    doCurrentItemChanged (widget_item)

    doKeywordTextChanged (*args, **kw)

    doMerge ()

    doTemplateTextChanged (*args, **kw)

    onAdd (*args, **kw)
        add keyword substitution

        Keywords are allowed to start with a letter or “_”, then can have numbers also. All letters must be upper-
        case.

        Substitutions may be any text, including any white space.

        Strings must have a minimum length of MINIMUM_KEY_LENGTH (3) characters.

    onDelete (*args, **kw)
        delete the selected key

        check that self.current_key is not None then delete that one and reset to next in list

    restoreSplitterDetails ()
        put the splitters back where they were

    restoreWindowGeometry ()
        put the window back where it was

    saveSplitterDetails ()
        remember where the splitters were

    saveWindowGeometry ()
        remember where the window was

    selectFirstKeyword ()
```

module: email_mvc_view

MVC View for emails

see <http://www.saltycrane.com/blog/2008/01/pyqt-43-simple-qabstractlistmodel/>

see <http://www.saltycrane.com/blog/2007/12/pyqt-43-qtableview-qabstracttablemodel/>

```
class Assign_GUP.email_mvc_view.AGUP_Emails_View (parent=None, agup=None,
                                                settings=None,
                                                hide_email_button=True)

    Bases: PyQt4.QtGui.QWidget

    Show the email to be sent to each Reviewer
```

Parameters

- **parent** (*obj*) – instance of main_window.AGUP_MainWindow or None

- **agup** (*obj*) – instance of agup_data.AGUP_Data
- **settings** (*obj*) – instance of settings.ApplicationQSettings

closeEvent (*event*)

doOpenEmail ()

open the email letter in the email tool (does not work too good now)

index_to_ID (*index*)

convert QListView index to sort_name string

on_item_clicked (*index*)

called when changing the selected Reviewer in the list

restoreWindowGeometry ()

put the window back where it was

saveWindowGeometry ()

remember where the window was

selectFirstListItem ()

selectModelByIndex (*curr*, *prev*)

select Reviewer for email display as referenced by QListView index

Parameters

- **curr** (*index*) – sort_name string of current selected reviewer
- **prev** (*index*) – QListView index of previously selected reviewer

setModel (*model*)

showReviewerEmail ()

select Reviewer for email display as referenced by self.sort_name

class Assign_GUP.email_mvc_view.**Email**

Bases: object

data structure for sending email

send ()

send the email content to the default email tool

Assign_GUP.email_mvc_view.**main** ()

simple starter program to develop this code

module: email_template

Create email notices to each Reviewer describing specific assignments

class Assign_GUP.email_template.**EmailTemplate**

Bases: object

Support the creation of custom emails to each Reviewer from template and fields

It is possible to create and use custom fields. Be sure that the custom fields are *uniquely identifiable* to avoid replacing the wrong text.

importXml (*filename*)

Parameters **filename** (*str*) – name of XML file with email template and keywords

mail_merge (***kw_dict*)

create one email with a mail merge of self.keyword_dict and kw_dict into self.email_template

suggest defining at least these four keywords, to be applied for each reviewer during mail merge step (values filled in programmatically):

```
FULL_NAME = 'Ima Reviewer',  
EMAIL = 'reviewer@institution.net',  
ASSIGNED_PRIMARY_PROPOSALS = '11111 22222 33333',  
ASSIGNED_SECONDARY_PROPOSALS = '44444 55555 66666',
```

mass_merge (*reviewers, **kw*)

create emails for all Reviewers, return as a dictionary by sort_name

Parameters **reviewers** (*obj*) – instance of revu_mvc_data.AGUP_Reviewers_List

writeXmlNode (*root_node*)

write email template data to the XML document

Parameters **root_node** (*obj*) – XML node to contain this data

module: event_filters

event filters for certain MVC widgets such as QListView

class Assign_GUP.event_filters.**ArrowKeysEventFilter**

Bases: PyQt4.QtCore.QObject

watches for ArrowUp and ArrowDown (navigator keys) to change selection in QtCore.QAbstractListModel

eventFilter (*listView, event*)

custom event filter

module: general_mvc_model

Generic MVC Model for AGUP

class Assign_GUP.general_mvc_model.**AGUP_MVC_Model** (*data_object=[], headerdata=None, parent=None*)

Bases: PyQt4.QtCore.QAbstractListModel

Generic MVC model for AGUP

This is an adapter for the actual data object

columnCount (*parent*)

data (*index, role*)

rowCount (*parent*)

module: history

document program history events in a log file

class Assign_GUP.history.**Logger** (*log_file=None, level=20, statusbar=None, history_widget=None, minor_details=False*)

Bases: object

use Python logging package to record program history

Parameters

- **log_file** (*str*) – name of file to store history
- **level** (*enum*) – logging interest level (default=logging.INFO, no logs = -1)
- **statusbar** (*obj*) – QStatusBar instance in main window
- **history_widget** (*str*) – QPlainTextEdit instance in main window
- **minor_details** (*bool*) – Include minor details in the logs?

add (*message*, *major_status=True*)

log a message or report from the application

Parameters

- **message** (*str*) – words to be logged
- **major_status** (*bool*) – major (True) or minor (False) status of this message

first_logs ()

`Assign_GUP.history.addLog (message=", major=True)`

put this message in the logs, note whether if major (True)

`Assign_GUP.history.logMinorDetails (choice)`

choose to record (True) or not record (False) minor details in the logs

module: main_window

Main code runs the GUI.

class `Assign_GUP.main_window.AGUP_MainWindow`

Bases: `PyQt4.QtGui.QMainWindow`

Creates a Qt GUI for the main window

adjustMainWindowTitle ()

mark if main window is dirty

indicate in main window title when there are unsaved modifications (i.e., when `self.cannotProceed()` is True)

cannotProceed ()

advise if the application has unsaved changes and should not do the proposed action

closeEvent (*event*)

called when user clicks the big [X] to quit

closeSubwindows ()

close all other windows created by this code

confirmAbandonChangesNotOk ()

Ask user to save changes before exit or opening another project.

Return True if application should *NOT* exit.

confirmEditTopics ()

confirm before proceeding

doAgupInfo (**args*, ***kw*)

describe this application and where to get more info

doAnalysis_gridReport ()
show a table with dotProducts for each reviewer against each proposal *and* assignments

doAssignmentsReport ()
show a read-only text page with assignments for each proposal

doAutomatedAssignment ()
make automated assignments of reviewers to proposals

doClose (*args, **kw)
called when user chooses exit (or quit), or from closeEvent()

doEditEmailTemplate ()
edit the template to send emails, include editor for keyword substitutions

doEditProposals ()
edit the list of Proposals

doEditReviewers ()
edit the list of Reviewers

doEditTopics ()
Create Window to edit list of Topics

doImportProposals ()
import the proposal file as downloaded from the APS web site

doImportReviewers ()
copy the list of Reviewers into this project from another PRP Project file

doImportTopics ()
copy the list of Topics from another PRP file into this project

doLettersReport ()
prepare the email form letters to each reviewer with their assignments

doNewPrpFile ()
clear the data in self.agup

doOpenPrpFile ()
open an existing PRP file

doRecalc ()

doResetDefaultSettings ()
user requested to reset the settings to their default values

Note: does not write to the rcfile

doSave ()
save the self.agup data to the known project file name

doSaveAs ()
save the self.agup data to the data file name selected from a dialog box

You may choose any file name and extension that you prefer. It is strongly suggested you choose the default file extension, to identify AGUP PRP Project files more easily on disk. Multiple projects files, perhaps for different review cycles, can be saved in the same directory. Or you can save each project file in a different directory as you choose.

By default, the file extension will be **.agup**, indicating that this is an AGUP PRP Project file. The extensions **.prp** or **.xml** may be used as alternatives. Each of these describes a file with *exactly the same file format*, an XML document.

doSummaryReport ()
this report is helpful to balance proposal assignments
show a read-only text page with how many primary and secondary proposals assigned to each reviewer

doUnassignProposals ()
Remove ALL assignments of reviewers to proposals

getOpenFileName (*args, **items)
convenience wrapper for Qt open file dialog

getReviewCycleText ()

importProposals (filename)
read a proposals XML file and set the model accordingly

importReviewers (filename)
read Reviewers from a PRP Project file and set the model accordingly

importTopics (filename)
read Topics from an AGUP PRP Project file and set the model accordingly

onAssignmentsChanged ()
called when a reviewer assignment checkbox has been changed

onTemplateChanged ()

onTopicValuesChanged (*args, **kw)
called when a proposal or reviewer topic value has been changed

openPrpFile (filename)
choose the XML file with data for this PRP review

requestConfirmation (message, informative_text, buttons=None, default_button=None)
request confirmation from the user about something

restoreWindowGeometry ()
put the window back where it was

saveWindowGeometry ()
remember where the window was

setIndicators ()
show the numbers of topics, reviewers, and proposals, also the cycle

setNumProposalsWidget (number)

setNumReviewersWidget (number)

setNumTopicsWidget (number)

setPrpFileText (text)

setRcFileText (text)

setReviewCycleText (text)

Assign_GUP.main_window.**main** ()
start the program

Assign_GUP.main_window.**process_command_line** ()
support command-line options such as `--help` and `--version`

module: plainTextEdit

show text in a GUI window

```
class Assign_GUP.plainTextEdit.TextWindow (parent=None, title='window title', text="", settings=None)
```

Bases: PyQt4.QtGui.QWidget

show text in a GUI window that remembers its geometry, based on supplied window title

Parameters

- **parent** (*obj*) – instance of QWidget
- **title** (*str*) – to be used as the window title (and settings group name)
- **text** (*str*) – window content
- **settings** (*obj*) – instance of settings.ApplicationQSettings

closeEvent (*event*)

moveEvent (*event*)

restoreWindowGeometry ()
put the window back where it was

saveWindowGeometry ()
remember where the window was

setText (*text*)

setTitle (*title*)

settingsGroupName ()
need a group name in the settings file to save the window geometry, based on window title

module: prop_mvc_data

Proposals: underlying data class for the MVC model

```
class Assign_GUP.prop_mvc_data.AGUP_Proposals_List (agup)  
Bases: PyQt4.QtCore.QObject
```

the list of all proposals

addTopic (*key, value=None*)
add a new topic key and initial value to all proposals

addTopics (*key_list*)
add several topics at once (with default values)

exists (*prop_id*)
given ID string, does proposal exist?

getByIndex (*index*)
given index in sorted list of proposals, return indexed proposal
note: index is *not* the proposal ID number

getProposal (*prop_id*)
return proposal selected by ID string

importXml (*filename*)

Parameters *filename* (*str*) – name of XML file with proposals

removeTopic (*key*)

remove an existing topic key from all proposals

removeTopics (*key_list*)

remove several topics at once

setTopicValue (*prop_id*, *topic*, *value*)

set the topic value on a proposal identified by GUP ID

writeXmlNode (*specified_node*)

write Proposals' data to a specified node in the XML document

Parameters *specified_node* (*obj*) – XML node to contain this data

module: prop_mvc_view

MVC View for proposals

see <http://www.saltycrane.com/blog/2008/01/pyqt-43-simple-qabstractlistmodel/>

see <http://www.saltycrane.com/blog/2007/12/pyqt-43-qtableview-qabstracttablemodel/>

class Assign_GUP.prop_mvc_view.**AGUP_Proposals_View** (*parent=None*, *agup=None*, *settings=None*)

Bases: PyQt4.QtGui.QWidget

Manage the list of proposals, including assignments of topic weights and reviewers

Parameters

- **parent** (*obj*) – instance of main_window.AGUP_MainWindow or None
- **agup** (*obj*) – instance of agup_data.AGUP_Data
- **settings** (*obj*) – instance of settings.ApplicationQSettings

closeEvent (*event*)

in response to user requesting the window be closed

details_modified ()

OK to select a different proposal now?

editProposal (*prop_id*, *prev_prop_index*)

select Proposal for editing as referenced by ID number

index_to_ID (*index*)

convert QListView index to GUP ID string

isProposalListModified ()

onAssignmentsChanged ()

called when a reviewer assignment checkbox has been changed

onTopicValueChanged (*prop_id*, *topic*, *value*)

called when user changed a topic value in the details panel

on_item_clicked (*index*)

called when changing the selected Proposal in the list

recalc ()

recalculate dot products

restoreWindowGeometry ()
put the window back where it was

saveWindowGeometry ()
remember where the window was

selectFirstListItem ()

selectModelByIndex (*curr*, *prev*)
select Proposal for editing as referenced by QListView index

Parameters

- **curr** (*index*) – GUP ID string of current selected proposal
- **prev** (*index*) – QListView index of previously selected proposal

setModel (*model*)

update ()

module: prop_revu_grid

widgets of one Reviewer of one Proposal instance

ProposalReviewerRow

adds one row of widgets for possible Reviewer of Proposal

Method	Description
<i>getAssignment</i> ()	return which type of reviewer this is (0, 1, 2)
<i>setAssignment</i> ()	define which type of reviewer this is (0, 1, 2)
<i>setValue</i> ()	set dotProduct value as percentage
<i>setEnabled</i> ()	enable/disable the checkboxes based on Reviewer eligibility
<i>setNumberAssigned</i> ()	show the number of assigned proposals with given role (1 2)
<i>dotProduct</i> ()	compute and set widget with dot product of reviewer & proposal topics

ReviewerAssignmentGridLayout

QGridLayout of possible Reviewers of Proposal

Method	Description
<i>addReviewer</i> ()	add controls for one Reviewer to the grid
<i>addReviewers</i> ()	add list of Reviewers to the grid
<i>setEnabled</i> ()	enable/disable one Reviewer
<i>setProposal</i> ()	specify the Proposal associated with this grid
<i>setAssignment</i> ()	define the type for a named Reviewer
<i>onCheck</i> ()	ensure only one reviewer is either primary or secondary
<i>setValue</i> ()	set dotProduct value of a named Reviewer as percentage

class Assign_GUP.prop_revu_grid.**ProposalReviewerRow** (*parent, layout, reviewer, proposal*)

Bases: PyQt4.QtCore.QObject

Adds a row of widgets to an existing grid layout for one Reviewer of one Proposal instance

dotProduct ()

dot product of Proposal and Reviewer topic strengths, $\vec{p} \cdot \vec{r}$

Computes $\vec{p} \cdot \vec{r}$ where:

- \vec{p} is array of topic value strengths for Proposal
- \vec{r} is array of topic value strengths for Reviewer

getAssignment ()

report which type of reviewer this is

returns	description
0	unassigned
1	primary reviewer (#1)
2	secondary reviewer (#2)

onCheckBoxClick (*widget*)

either check box was clicked

rowCheck (*checkbox*)

ensure at most one checkbox (primary or secondary) is checked for this reviewer

Parameters **checkbox** (*obj*) – instance of QCheckBox

setAssignment (*code*)

define which type of reviewer this is

code	description
0	unassigned
1	primary reviewer (#1)
2	secondary reviewer (#2)

Parameters **code** (*int*) – integer code (0 | 1 | 2)

setNumberAssigned (*number, role*)

show the number of assigned proposals with given role

setProposal (*proposal*)

define the proposal to be used with this row

setValue (*percentage*)

set the percentage value

Parameters **percentage** (*int*) – dot product of reviewer and proposal topic strengths

update ()

class Assign_GUP.prop_revu_grid.**ReviewerAssignmentGridLayout** (*parent, agup*)

Bases: PyQt4.QtGui.QGridLayout

display and manage the assignment checkboxes and reported percentages for each reviewer on this proposal

addReviewer (*rvwr*)

add this reviewer object for display

addReviewers (*reviewers*)

add a list of reviewers

onCheck (*row_widget*)

ensure only one reviewer is either primary or secondary

setAssignment (*sort_name, code*)

define which type of reviewer this is

Parameters

- **sort_name** (*str*) – reviewer’s identifying key
- **code** (*int*) – integer code (0 | 1 | 2)

setEnabled (*sort_name, state=True*)

enable (True) or disable (False) a Reviewer identified by sort_name

All eligible Reviewers are enabled. Reviewers become ineligible when they are named as part of the Proposal team.

setProposal (*proposal*)

declare which proposal is associated with this grid

setReviewersValues (*reviewers*)

set the widget values for all Reviewers

setValue (*sort_name, percentage*)

set the percentage value

update ()

updateNumbersOfAssignedProposals ()

update the numbers of assigned proposals shown for each reviewer and each role

`Assign_GUP.prop_revu_grid.developer_main()`

create QGroupBox + QGridLayout and run the panel

R1	#1	R2	#2	%	Reviewer Name
[x]	1	[]	0	100%	Joe User
[]	0	[x]	1	80%	Second Reviewer
[]	0	[]	0	22%	Some Panel Member
[]	0	[]	0	1%	Another Panel Member

module: proposal

Data for one General User Proposal

class `Assign_GUP.proposal.AGUP_Proposal_Data` (*xmlParentNode=None, xmlFile=None*)

Bases: `object`

A single General User Proposal

addTopic (*topic, value=None*)

declare a new topic and give it an initial value (default value=0.0)

topic must not exist or `KeyError` exception will be raised

addTopics (*topics_list*)

declare several new topics and give them all default values

each topic must not exist or `KeyError` exception will be raised

getAssignedReviewers ()
return a list of assigned reviewers for this proposal

getExcludedReviewers (*reviewers*)
return a list of excluded reviewers for this proposal

Parameters *reviewers* (*obj*) – list of all available reviewers

getKey (*key*)
get the item from self.db identified by *key*

getSubjects ()
return the list of subjects as specified in the Proposal

getTopic (*topic*)
return the value of the named topic

getTopicList ()
return a list of all topics

hasTopic (*topic*)
does the named topic exist?

importXml (*proposal_node*)
Fill the class variables with values from the XML node

Parameters *proposal_node* – lxml node of the Proposal

removeTopic (*key*)
remove the named topic

removeTopics (*key_list*)
remove several topics at once

setAssignedReviewer (*reviewer*, *role=None*)
assign a reviewer to this proposal as primary (*role=1*) or secondary (*role=2*) or None (unassigned)

setTopic (*topic*, *value=None*)
set value of an existing topic

topic must exist or `KeyError` exception will be raised

writeXmlNode (*specified_node*)
write this Proposal's data to a specified node in the XML document

Parameters *specified_node* (*obj*) – XML node to contain this data

module: proposal_details

QtGui widget to edit one Proposal instance

AGUP_ProposalDetails

adds one row of widgets for possible Reviewer of Proposal

Method	Description
<code>addReviewer()</code>	add a Reviewer to the panel

```
class Assign_GUP.proposal_details.AGUP_ProposalDetails (parent=None, settings=None, agup=None)  
    Bases: PyQt4.QtGui.QWidget  
    QtGui widget to edit one Proposal instance  
addReviewers (reviewers)  
        Add the list of Reviewers to the details panel  
addTopic (topic, value)  
addTopics (topic_list)  
clear ()  
getProposalId ()  
onAssignmentsChanged ()  
        called when a reviewer assignment checkbox has been changed  
onTopicValueChanged (topic)  
restoreSplitterDetails ()  
saveSplitterDetails ()  
setAll (prop_id, title, period, speaker, choice, subjects)  
setFirstChoiceBl (value)  
setProposalId (value)  
setProposalTitle (value)  
setReviewPeriod (value)  
setSpkName (value)  
setSubjects (value)  
setTopic (key, value)  
setupProposal (proposal)  
        install proposal data in the editor's widgets  
update ()
```

module: qt_utils

Helper routines for Qt

```
Assign_GUP.qt_utils.setButtonBackground (widget, color='#efefef')
```

```
Assign_GUP.qt_utils.setWidgetBackground (widget, color='#efefef')  
    change the background color of a Qt widget
```

Parameters **color** (*str*) – specified as name (mintcream), hex RGB (#dea)

module: report_analysis_grid

show a table with dotProducts for each reviewer against each proposal *and* assignments

```
class Assign_GUP.report_analysis_grid.Report (parent, agup, settings)  
    Bases: Assign_GUP.plainTextEdit.TextWindow
```

```

makeText ()
    generate the text of the panel

update ()

```

module: report_assignments

show a read-only text page with assignments for each proposal

GUP#	reviewer 1	reviewer 2	excluded reviewer(s)	title
11111	A Reviewer	Ima Reviewer		Study of stuff

```

class Assign_GUP.report_assignments.Report (parent, agup, settings)
    Bases: Assign_GUP.plainTextEdit.TextWindow

    makeText ()
        generate the text of the panel

    update ()

```

module: report_summary

show a read-only text page with how many primary and secondary proposals assigned to each reviewer

```

total number of proposals: #
primary proposals per reviewer: #.#

Overall topic strength: TBA

Primary assignments:
reviewer1  ##: ##### ##### #####
reviewer2  ##: ##### ##### #####
reviewer3  ##: ##### ##### #####

Secondary assignments:
reviewer1  ##: ##### ##### #####
reviewer2  ##: ##### ##### #####
reviewer3  ##: ##### ##### #####

Unassigned proposals: #

```

```

class Assign_GUP.report_summary.Report (parent, agup, settings)
    Bases: Assign_GUP.plainTextEdit.TextWindow

    makeText ()
        generate the text of the panel

    update ()

```

module: resources

(internal) support for items in resources folder, such as forms defined in .ui files

```

Assign_GUP.resources.get_forms_path (path=None)
    identify our resources directory

```

`Assign_GUP.resources.loadUi(ui_file, baseinstance=None, **kw)`
load a .ui file for use in building a GUI

Wraps `uic.loadUi()` with code that finds our program's *resources* directory.

See <http://nullege.com/codes/search/PyQt4.uic.loadUi>

See <http://bitesofcode.blogspot.ca/2011/10/comparison-of-loading-techniques.html>

inspired by: [http://stackoverflow.com/questions/14892713/how-do-you-load-ui-files-onto-python-classes-with-pyside?](http://stackoverflow.com/questions/14892713/how-do-you-load-ui-files-onto-python-classes-with-pyside?lq=1)
`lq=1`

Basic Procedure

1. Use Qt Designer to create a .ui file.
2. Create a python class of the same type as the widget you created in the .ui file.
3. When initializing the python class, use `uic` to dynamically load the .ui file onto the class.

Here is an example from this code:

```
1 from PyQt4 import QtGui
2 import resources
3
4 UI_FILE = 'plainTextEdit.ui'
5
6 class TextWindow(QtGui.QDialog, form_class):
7
8     def __init__(self, title, text):
9         QtGui.QDialog.__init__(self, parent)
10        resources.loadUi(UI_FILE, baseinstance=self)
11        self.setWindowTitle(title)
12        self.plainTextEdit.setPlainText(text)
13
14 import sys
15 app = QtGui.QApplication(sys.argv)
16 win = TextWindow('the title', __doc__)
17 win.show()
18 sys.exit(app.exec_())
```

`Assign_GUP.resources.resource_file(filename, dir=None)`
absolute path to file in resources directory

module: reviewer

Data for one Reviewer of General User Proposals

class `Assign_GUP.reviewer.AGUP_Reviewer_Data(xmlParentNode=None, xmlFile=None)`
Bases: `object`

A Reviewer of General User Proposals

addTopic (*topic*, *value=None*)

declare a new topic and give it an initial value

topic must not exist or `KeyError` exception will be raised

addTopics (*topics_list*)
declare several new topics and give them all default values
each topic must not exist or `KeyError` exception will be raised

getAssignments (*proposals, role*)
find all proposals assigned to this reviewer with the given role

getFullName ()

getKey (*key*)

getSortName ()

getTopic (*topic*)
return the value of the named topic

getTopicList ()
return a list of all topics

importXml (*reviewer*)
Fill the class variables with values from the XML node
Parameters **reviewer** – lxml node node of the Reviewer

removeTopic (*key*)
remove the named topic

removeTopics (*key_list*)
remove several topics at once

setKey (*key, value*)
save a value to a known key
example:

```
self.setKey('full_name', 'Pete Jemian')
```

setTopic (*topic, value=None*)
set value of an existing topic
topic must exist or `KeyError` exception will be raised

writeXmlNode (*specified_node*)
write this Reviewer's data to a specified node in the XML document
Parameters **specified_node** (*obj*) – XML node to contain this data

module: reviewer_details

QtGui widget to edit one Reviewer instance

```
class Assign_GUP_reviewer_details.AGUP_ReviewerDetails (parent=None,          set-
                                                         tings=None)
    Bases: PyQt4.QtGui.QWidget
    QtGui widget to edit one Reviewer instance
    addTopic (topic, value)
    addTopics (topic_list)
    getEmail ()
    getFullName ()
```

```
getJoined()  
getNotes()  
getPhone()  
getSortName()  
getUrl()  
onTopicValueChanged(topic)  
restoreSplitterDetails()  
saveSplitterDetails()  
setEmail(value)  
setFullName(value)  
setJoined(value)  
setNotes(value)  
setPhone(value)  
setSortName(value)  
setTopic(key, value)  
setUrl(value)
```

module: revu_mvc_data

Reviewers: underlying data class for the MVC model

```
class Assign_GUP.revu_mvc_data.AGUP_Reviewers_List(agup)  
    Bases: PyQt4.QtCore.QObject  
    the list of all reviewers  
    addTopic(key, value=None)  
        add a new topic key and initial value to all reviewers  
    addTopics(key_list)  
        add several topics at once (with default values)  
    exists(sort_name)  
        given sort_name string, does reviewer exist?  
    getByFullName(full_name)  
        return reviewer selected by full_name string  
    getByIndex(index)  
    getReviewer(sort_name)  
        return reviewer selected by sort_name string  
    getReviewerAssignments(sort_name, role=None)  
        find all proposals assigned to the named reviewer
```

Parameters

- **sort_name** (*str*) – key for reviewer name
- **role** (*int*) – one of [None, 0, 1, 2], where None or 0 mean get both roles

importXml (*filename*)
Parameters **filename** (*str*) – name of XML file with reviewers

inOrder ()

keyOrder ()

removeReviewer (*full_name*)
remove named reviewer

removeTopic (*key*)
remove an existing topic key from all reviewers

removeTopics (*key_list*)
remove several topics at once

setTopicValue (*sort_name*, *topic*, *value*)
set the topic value on a reviewer identified by *sort_name*

writeXmlNode (*specified_node*)
write Reviewers' data to a specified node in the XML document
Parameters **specified_node** (*obj*) – XML node to contain this data

module: revu_mvc_view

MVC View for reviewers

see <http://www.saltycrane.com/blog/2008/01/pyqt-43-simple-qabstractlistmodel/>

see <http://www.saltycrane.com/blog/2007/12/pyqt-43-qtableview-qabstracttablemodel/>

class Assign_GUP.revu_mvc_view.**AGUP_Reviewers_View** (*parent=None*, *agup=None*, *settings=None*)

Bases: PyQt4.QtGui.QWidget

Manage the list of Reviewers, including assignments of topic weights

Parameters

- **parent** (*obj*) – instance of `main_window.AGUP_MainWindow` or `None`
- **agup** (*obj*) – instance of `agup_data.AGUP_Data`
- **settings** (*obj*) – instance of `settings.ApplicationQSettings`

closeEvent (*event*)

details_modified ()

OK to select a different reviewer now?

editReviewer (*sort_name*, *prev_index*)

select Reviewer for editing as referenced by *sort_name*

index_to_ID (*index*)

convert `QListView` index to *sort_name* string

isReviewerListModified ()

onDetailsModified (**args*)

onTopicValueChanged (*sort_name*, *topic*, *value*)

called when user changed a topic value in the details panel

on_item_clicked (*index*)
called when changing the selected Reviewer in the list

restoreWindowGeometry ()
put the window back where it was

saveReviewerDetails ()
copied Reviewer details from editor panel to main data structure

saveWindowGeometry ()
remember where the window was

selectFirstListItem ()

selectModelByIndex (*curr*, *prev*)
select Reviewer for editing as referenced by QListView index

Parameters

- **curr** (*index*) – sort_name string of current selected reviewer
- **prev** (*index*) – QListView index of previously selected reviewer

setModel (*model*)

`Assign_GUP.revu_mvc_view.main()`
simple starter program to develop this code

module: settings

Support for AGUP program settings

A settings file is used to preserve certain values of the application (such as window positions and full path to the project file). The name of the settings file is given in the main window. Note, the settings file may have the suffix “.ini” on some operating systems. Remove the settings file to clear any settings. There is also a menu item to clear this file and reset it to defaults.

This module uses QSettings (<http://doc.qt.io/qt-4.8/qsettings.html>).

Note: Multi-monitor support : method `restoreWindowGeometry()`

On multi-monitor systems such as laptops, window may be restored to offscreen position. Here is how it happens:

- geo was saved while window was on 2nd screen while docked
- now re-opened on laptop display and window is off-screen

For now, keep the windows on the main screen or learn how to edit the settings file.

```
class Assign_GUP.settings.ApplicationQSettings
    Bases: PyQt4.QtCore.QSettings

    manage and preserve default settings for this application using QSettings

    Use the .ini file format and save under user directory

    getEmailKeywords ()
        return the email substitution keywords as a dictionary

    getKey (key)
        return the Python value (not a QVariant) of key or None if not found
```

Raises **TypeError** – if key is None

getPrpFile ()

getReviewCycle ()

keyExists (*key*)

does the named key exist?

resetDefaults ()

Reset all application settings to default values.

restoreSplitterDetails (*window*)

put the splitter back where it was

Parameters **window** (*obj*) – instance of QWidget

restoreWindowGeometry (*window*, *group=None*)

put the window back where it was

Parameters **window** (*obj*) – instance of QWidget

saveEmailKeywords (*keyword_dict*)

remember the email substitution keywords

saveSplitterDetails (*window*)

remember where the splitter was

Parameters **window** (*obj*) – instance of QWidget

saveWindowGeometry (*window*, *group=None*)

remember where the window was

Parameters **window** (*obj*) – instance of QWidget

setKey (*key*, *value*)

set the value of a configuration key, creates the key if it does not exist

Parameters **key** (*str*) – either *key* or *group/key*

Complement: `self.value(key)` returns value of key

setPrpFile (*prp_file*)

setReviewCycle (*review_cycle*)

updateTimeStamp ()

module: signals

Custom Qt4 signals

signal	args	comments
changed		template editor
checkBoxGridChanged		reviewer assignment
closed		topics_editor
recalc		dotProduct
topicValueChanged	str, str, float	reviewer or proposal

class Assign_GUP.signals.CustomSignals

Bases: PyQt4.QtCore.QObject

custom signals

module: tools

common tools used in various places

for more information on the various codepoints, see <https://docs.python.org/2/library/codecs.html#standard-encodings>

```
Assign_GUP.tools.text_decode(source)
    try decoding source with various known codepoints to unicode
```

```
Assign_GUP.tools.text_encode(source)
    encode source using the default codepoint
```

module: topic_slider

add label, slider, value to a QGridLayout

Coordinate the action of a slider with the topic value:

label	value	slider
bio	0.7	--- --- --- [-] ---
phys	0.2	-- [] -- --- --- ---

widget	type	description
label	QLabel	mnemonic name (no white space)
value	QLineEdit	string with floating point value: 0 <= value <= 1.0
slider	QSlider	graphical adjustment of value

These three widgets will be added to the *parent* widget, assumed to be on the same row of a QGridLayout.

A *topic* (known here as *label*) is some scientific area of interest to the PRP. Such as, for the SAXS review panel, some of the proposals are for XPCS (X-ray Photon Correlation Spectroscopy).

Each proposal will have a strength value assigned for each topic, indicating how important that topic is to the proposed experiment.

Each reviewer will have a strength value assigned for each topic, indicating the strength of that reviewer in the particular topic.

The strength values will be constrained to the range [0 .. 1]. A QValidator() object will be used to color the background of the QLineEdit to indicate whether or not the entered text is acceptable. The value of the slider will update with acceptable values from the text entry. The background color indicates acceptable or invalid input. The slider will be updated for acceptable values. The validator will also constrain the input to a precision of 2 decimal places.

```
class Assign_GUP.topic_slider.AGUP_TopicSlider (parent, row, label, value)
```

```
    Bases: PyQt4.QtCore.QObject
```

```
    add topic, slider, value_entry to a QGridLayout
```

```
    getSliderValue ()
```

```
    getValue ()
```

```
    onSliderChange (value)
```

```
        update the QLineEdit when the slider is changed
```

```
    onValueChange (value)
```

```
        update the QSlider when the value is changed
```

setSliderValue (*value*)

set value of the slider indicating strength of this topic (0:low, 100: high)

Parameters **value** (*int*) – $0 \leq \text{value} \leq 100$

This routine sets the text value.

setValue (*value*)

set strength of this topic (0:low, 1.0: high)

Parameters **value** (*int*) – $0 \leq \text{value} \leq 100$

This routine sets the slider value.

module: topics

Support AGUP topics

class Assign_GUP.topics.Topics

Bases: object

manage the list of AGUP topics (known here as *key*)

add (*key*, *value*=0.0)

define a new topic (known here as *key*)

addTopics (*key_list*)

add several topics at once (with default values)

Parameters **key_list** (*[str]*) – list of topics (strings) to be added

clearAll ()

remove all keys from the list of topics

compare (*other_topics_object*)

compare topics in self.topics with the other_topics_object, return True if identical

compares sorted list of topics between each object

Parameters **other_topics_object** (*obj*) – instance of Topics()

diff (*other_topics_object*)

differences in list of topics between self.topics and other_topics_object

Comparison assumes that self.topics is the final result. Returned result shows topics added and removed from *other_topics_object* to obtain current list.

Parameters **other_topics_object** (*obj*) – instance of Topics()

Returns ([],[]) first list is topics added, second list is topics removed

dotProduct (*other*)

normalized dot product of Proposal (*self*) and Reviewer (*other*) topic strengths, $\vec{p} \cdot \vec{r}$

Parameters **other** (*obj*) – instance of Topics()

Returns $\sum \vec{p} \cdot \vec{r} / \sum \vec{p}$

- \vec{p} is array of topic value strengths for Proposal
- \vec{r} is array of topic value strengths for Reviewer

exists (*key*)

Is *key* already known?

get (*key*)
return value of an existing topic (known here as *key*)
topic must exist or `KeyError` exception will be raised

getTopicList ()
return a list of all topics

importXml (*xmlFile*, *read_values=True*)

Parameters

- **filename** (*str*) – name of XML file with Topics
- **read_values** (*bool*) – import topic values?

importXmlTopics (*parent_node*, *read_values=True*)
make this common code segment reusable

Parameters

- **parent_node** (*obj*) – XML parent node
- **read_values** (*bool*) – import topic values?

inOrder ()

remove (*key*)
remove the named topic

Parameters **key** (*str*) – topic to be removed

removeTopics (*key_list*)
remove several topics at once

Parameters **key_list** (*[str]*) – list of topics (strings) to be removed

set (*key*, *value*)
set value of an existing topic (known here as *key*)
topic must exist or `KeyError` exception will be raised

valueOrder ()
sort by topic values

writeXml (*specified_node*, *write_values=True*)
write Topics' data to a specified node in the XML document

Parameters

- **specified_node** (*obj*) – XML node to contain this data
- **read_values** (*bool*) – write topic values?

`Assign_GUP.topics.checkTopicValueRange` (*value*)
topic values must be 0..1 inclusive: standardize this check

Parameters **value** (*float*) – topic value to be checked

`Assign_GUP.topics.diffLists` (*new_list*, *old_list*)
differences between two lists, return tuple([items added], [items removed])
assumes each list had only unique entries, no redundancies

Parameters

- **new_list** (*[str]*) – new list of strings to be compared

- **old_list** (*[str]*) – old list of strings to be compared

`Assign_GUP.topics.sortListUnique` (*the_list*)
sort list and eliminate redundant items

- make a dictionary with each list item
- redundancies will be overwritten

Parameters **the_list** (*[str]*) – list of strings to be sorted

`Assign_GUP.topics.synchronizeTopics` (*a_list, b_list*)
make the topic names in each list be the same

- assumes each topics list had only unique entries, no redundancies
- modifies objects in place

Parameters

- **a_list** (*obj*) – instance of Topics()
- **b_list** (*obj*) – instance of Topics()

module: topics_editor

GUI to edit the list of topics

class `Assign_GUP.topics_editor.AGUP_TopicsEditor` (*parent=None, topics_list=None, settings=None*)

Bases: `PyQt4.QtGui.QDialog`

add topic, slider, value_entry to a `QGridLayout`

closeEvent (*event*)

getTopicList ()

when all editing is complete, call this method to get the final list

onAdd (**args, **kw*)

add the text in the entry box to the list of topics

onCloseButton (*event*)

onDelete (**args*)

remove the selected item from the list of topics

restoreWindowGeometry ()

put the window back where it was

saveWindowGeometry ()

remember where the window was

module: xml_utility

XML utility methods

exception `Assign_GUP.xml_utility.IncorrectXmlRootTag`

Bases: `lxml.etree.DocumentInvalid`

the root tag of the XML file is incorrect

exception Assign_GUP.xml_utility.InvalidWithXmlSchema

Bases: lxml.etree.DocumentInvalid

error while validating against the XML Schema

exception Assign_GUP.xml_utility.XmlSyntaxError

Bases: lxml.etree.XMLSyntaxError

Xml Syntax error

Assign_GUP.xml_utility.getXmlText (parent, tag, default=None)

Read the text content of an XML node

Parameters

- **reviewer** – lxml node node
- **default** – default value is no node text

Returns node text or None

Assign_GUP.xml_utility.readValidXmlDoc (filename, expected_root_tag, XML_Schema_file,
alt_root_tag="", alt_schema=None)

Common code to read an XML file, validate it with an XML Schema, and return the XML doc object

Parameters **XML_Schema_file** (str) – name of XML Schema file (local to package directory)

Assign_GUP.xml_utility.validate (xml_tree, XSD_Schema_file)

validate an XML document tree against an XML Schema file

Parameters

- **xml_tree** (obj) – instance of etree._ElementTree
- **XSD_Schema_file** (str) – name of XSD Schema file (local to package directory)

2.4 History of Source Code Changes

2017.4.0

- #72 build for RTD in conda virtual environment
- #25 pick up GitHub release info using python-versioneer

2017.3.1 fixes (issue #69) resolve conflicting reviewer assignments on XML import

2017.3.0 fixes (issue #67) call setModel() after importing reviewers

2016.1025.0 standardize versioning kit with pyRestTable

2016.0312.0 fixes (issue #58) excluded reviewers are not auto-assigned, fixes (issue #63) save changes before opening another project

2016.0311.1 (fixes #57) codepoint conflicts resolved

2016.0309.1 (fixes #50) window objects need to be None after disposal

2016.0309.0 (fixes #51) topic strengths not assigned automatically

2016.0308.4 (fixes #55) unhandled non-ASCII character(s)

2016.0209.0 added missing imports

2016.0208.0 (#33) validate number entry in topic slider

2016.0205.0 (#44) reports update after auto-assign

2016.0203.2 fixes (issue #36)

2016.0203.1 fixes (issue #21)

2016.0203.0 fixes (issue #19), fixes (issue #32), fixes (issue #35), fixes (issue #41), fixes (issue #42)

2016.0201.1 provide an “un-assign all proposals” capability (issue #37)

2016.0201.0 announce that auto_assign has completed (issue #36)

2016.0114.1 described program installation

2016.0114.0 added simple auto-assignment of proposals (issue #20), report window focus problem fixed

2016.0113.4 remove search for legacy .xsd file

2016.0113.3 assignments report window re-displays on demand now

2016.0113.2 resolved errors when building the documentation due to versioneer

2016.0113.1 #24 apparently fixed by previous work

2016.0113.0 added *versioneer* support, LICENSE GUI now displays (#24 fixed also)

2015.0921.0 first release of new **Assign_GUP** application

2015.09.rc1 first pre-release using PyQt4, resolving issues 1-9, provides every capability in previous wxPython version

2013.11.0 last release using wxPython

2.5 Source Code License

```

1 Copyright (c) 2011–2016, UChicago Argonne, LLC
2
3 All Rights Reserved
4
5 Assign_GUP
6
7 Advanced Photon Source, Argonne National Laboratory
8
9
10 OPEN SOURCE LICENSE
11
12 Redistribution and use in source and binary forms, with or without
13 modification, are permitted provided that the following conditions are met:
14
15 1. Redistributions of source code must retain the above copyright notice,
16    this list of conditions and the following disclaimer. Software changes,
17    modifications, or derivative works, should be noted with comments and
18    the author and organization's name.
19
20 2. Redistributions in binary form must reproduce the above copyright notice,
21    this list of conditions and the following disclaimer in the documentation
22    and/or other materials provided with the distribution.
23
24 3. Neither the names of UChicago Argonne, LLC or the Department of Energy
25    nor the names of its contributors may be used to endorse or promote
26    products derived from this software without specific prior written
27    permission.
28

```

```
29 4. The software and the end-user documentation included with the
30 redistribution, if any, must include the following acknowledgment:
31
32 "This product includes software produced by UChicago Argonne, LLC
33 under Contract No. DE-AC02-06CH11357 with the Department of Energy."
34
35 *****
36
37 DISCLAIMER
38
39 THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND.
40
41 Neither the United States GOVERNMENT, nor the United States Department
42 of Energy, NOR UChicago Argonne, LLC, nor any of their employees, makes
43 any warranty, express or implied, or assumes any legal liability or
44 responsibility for the accuracy, completeness, or usefulness of any
45 information, data, apparatus, product, or process disclosed, or
46 represents that its use would not infringe privately owned rights.
47
48 *****
```

CHAPTER 3

Indices and tables

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

a

- Assign_GUP, 12
- Assign_GUP.about, 12
- Assign_GUP.agup_data, 12
- Assign_GUP.auto_assignment, 13
- Assign_GUP.editor_email_template, 13
- Assign_GUP.email_mvc_view, 14
- Assign_GUP.email_template, 15
- Assign_GUP.event_filters, 16
- Assign_GUP.general_mvc_model, 16
- Assign_GUP.history, 16
- Assign_GUP.main_window, 17
- Assign_GUP.plainTextEdit, 20
- Assign_GUP.prop_mvc_data, 20
- Assign_GUP.prop_mvc_view, 21
- Assign_GUP.prop_revu_grid, 22
- Assign_GUP.proposal, 24
- Assign_GUP.proposal_details, 25
- Assign_GUP.qt_utils, 26
- Assign_GUP.report_analysis_grid, 26
- Assign_GUP.report_assignments, 27
- Assign_GUP.report_summary, 27
- Assign_GUP.resources, 27
- Assign_GUP.reviewer, 28
- Assign_GUP.reviewer_details, 29
- Assign_GUP.revu_mvc_data, 30
- Assign_GUP.revu_mvc_view, 31
- Assign_GUP.settings, 32
- Assign_GUP.signals, 33
- Assign_GUP.tools, 34
- Assign_GUP.topic_slider, 34
- Assign_GUP.topics, 35
- Assign_GUP.topics_editor, 37
- Assign_GUP.xml_utility, 37

A

- add() (Assign_GUP.history.Logger method), 17
- add() (Assign_GUP.topics.Topics method), 35
- addLog() (in module Assign_GUP.history), 17
- addReviewer() (Assign_GUP.prop_revu_grid.ReviewerAssignmentGridLayout method), 23
- addReviewers() (Assign_GUP.prop_revu_grid.ReviewerAssignmentGridLayout method), 23
- addReviewers() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26
- addTopic() (Assign_GUP.prop_mvc_data.AGUP_Proposals_List method), 20
- addTopic() (Assign_GUP.proposal.AGUP_Proposal_Data method), 24
- addTopic() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26
- addTopic() (Assign_GUP.reviewer.AGUP_Reviewer_Data method), 28
- addTopic() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 29
- addTopic() (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List method), 30
- addTopics() (Assign_GUP.prop_mvc_data.AGUP_Proposals_List method), 20
- addTopics() (Assign_GUP.proposal.AGUP_Proposal_Data method), 24
- addTopics() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26
- addTopics() (Assign_GUP.reviewer.AGUP_Reviewer_Data method), 28
- addTopics() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 29
- addTopics() (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List method), 30
- addTopics() (Assign_GUP.topics.Topics method), 35
- adjustMainWindowTitle() (Assign_GUP.main_window.AGUP_MainWindow method), 17
- AGUP_Data (class in Assign_GUP.agup_data), 12
- AGUP_Emails_View (class in Assign_GUP.email_mvc_view), 14
- AGUP_MainWindow (class in Assign_GUP.main_window), 17
- AGUP_MVC_Model (class in Assign_GUP.general_mvc_model), 16
- AGUP_Proposal_Data (class in Assign_GUP.proposal), 24
- AGUP_ProposalDetails (class in Assign_GUP.proposal_details), 25
- AGUP_Proposals_List (class in Assign_GUP.prop_mvc_data), 20
- AGUP_Proposals_View (class in Assign_GUP.prop_mvc_view), 21
- AGUP_Reviewer_Data (class in Assign_GUP.reviewer), 28
- AGUP_ReviewerDetails (class in Assign_GUP.reviewer_details), 29
- AGUP_Reviewers_List (class in Assign_GUP.revu_mvc_data), 30
- AGUP_Reviewers_View (class in Assign_GUP.revu_mvc_view), 31
- AGUP_TopicsEditor (class in Assign_GUP.topics_editor), 37
- AGUP_TopicSlider (class in Assign_GUP.topic_slider), 34
- ApplicationQSettings (class in Assign_GUP.settings), 32
- ArrowKeysEventFilter (class in Assign_GUP.event_filters), 16
- Assign_GUP (module), 12
- Assign_GUP.about (module), 12
- Assign_GUP.agup_data (module), 12
- Assign_GUP.auto_assignment (module), 13
- Assign_GUP.editor_email_template (module), 13
- Assign_GUP.email_mvc_view (module), 14
- Assign_GUP.email_template (module), 15
- Assign_GUP.event_filters (module), 16
- Assign_GUP.general_mvc_model (module), 16
- Assign_GUP.history (module), 16
- Assign_GUP.main_window (module), 17

Assign_GUP.plainTextEdit (module), 20
Assign_GUP.prop_mvc_data (module), 20
Assign_GUP.prop_mvc_view (module), 21
Assign_GUP.prop_revu_grid (module), 22
Assign_GUP.proposal (module), 24
Assign_GUP.proposal_details (module), 25
Assign_GUP.qt_utils (module), 26
Assign_GUP.report_analysis_grid (module), 26
Assign_GUP.report_assignments (module), 27
Assign_GUP.report_summary (module), 27
Assign_GUP.resources (module), 27
Assign_GUP.viewer (module), 28
Assign_GUP.viewer_details (module), 29
Assign_GUP.revu_mvc_data (module), 30
Assign_GUP.revu_mvc_view (module), 31
Assign_GUP.settings (module), 32
Assign_GUP.signals (module), 33
Assign_GUP.tools (module), 34
Assign_GUP.topic_slider (module), 34
Assign_GUP.topics (module), 35
Assign_GUP.topics_editor (module), 37
Assign_GUP.xml_utility (module), 37
Auto_Assign (class in Assign_GUP.auto_assignment), 13

C

cannotProceed() (Assign_GUP.main_window.AGUP_MainWindow method), 17
checkTopicValueRange() (in module Assign_GUP.topics), 36
clear() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26
clearAll() (Assign_GUP.topics.Topics method), 35
clearAllData() (Assign_GUP.agup_data.AGUP_Data method), 12
closeEvent() (Assign_GUP.about.InfoBox method), 12
closeEvent() (Assign_GUP.editor_email_template.Editor method), 14
closeEvent() (Assign_GUP.email_mvc_view.AGUP_Emails_View method), 15
closeEvent() (Assign_GUP.main_window.AGUP_MainWindow method), 17
closeEvent() (Assign_GUP.plainTextEdit.TextWindow method), 20
closeEvent() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View method), 21
closeEvent() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View method), 31
closeEvent() (Assign_GUP.topics_editor.AGUP_TopicsEditor method), 37
closeSubwindows() (Assign_GUP.main_window.AGUP_MainWindow method), 17
columnCount() (Assign_GUP.general_mvc_model.AGUP_MVC_Model method), 16

compare() (Assign_GUP.topics.Topics method), 35
confirmAbandonChangesNotOk() (Assign_GUP.main_window.AGUP_MainWindow method), 17
confirmEditTopics() (Assign_GUP.main_window.AGUP_MainWindow method), 17
CustomSignals (class in Assign_GUP.signals), 33

D

data() (Assign_GUP.general_mvc_model.AGUP_MVC_Model method), 16
details_modified() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View method), 21
details_modified() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View method), 31
developer_main() (in module Assign_GUP.prop_revu_grid), 24
diff() (Assign_GUP.topics.Topics method), 35
diffLists() (in module Assign_GUP.topics), 36
doAgupInfo() (Assign_GUP.main_window.AGUP_MainWindow method), 17
doAnalysis_gridReport() (Assign_GUP.main_window.AGUP_MainWindow method), 17
doAssignmentsReport() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doAutomatedAssignment() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doClose() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doCurrentItemChanged() (Assign_GUP.editor_email_template.Editor method), 14
doDocsUrl() (Assign_GUP.about.InfoBox method), 12
doEditEmailTemplate() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doEditProposals() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doEditReviewers() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doEditTopics() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doImportProposals() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doImportReviewers() (Assign_GUP.main_window.AGUP_MainWindow method), 18

method), 18
doImportTopics() (Assign_GUP.main_window.AGUP_MainWindow method), 30
method), 18
doIssuesUrl() (Assign_GUP.about.InfoBox method), 12
doKeywordTextChanged() (Assign_GUP.editor_email_template.Editor method), 14
doLettersReport() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doLicense() (Assign_GUP.about.InfoBox method), 12
doMerge() (Assign_GUP.editor_email_template.Editor method), 14
doNewPrpFile() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doOpenEmail() (Assign_GUP.email_mvc_view.AGUP_Emails_View method), 15
doOpenPrpFile() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doRecalc() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doResetDefaultSettings() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doSave() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doSaveAs() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doSummaryReport() (Assign_GUP.main_window.AGUP_MainWindow method), 18
doTemplateTextChanged() (Assign_GUP.editor_email_template.Editor method), 14
dotProduct() (Assign_GUP.prop_revu_grid.ProposalReviewerRow method), 23
dotProduct() (Assign_GUP.topics.Topics method), 35
doUnassignProposals() (Assign_GUP.main_window.AGUP_MainWindow method), 19
doUrl() (Assign_GUP.about.InfoBox method), 12

E
Editor (class in Assign_GUP.editor_email_template), 14
editProposal() (Assign_GUP.prop_mvc_view.AGUP_Proposals_List method), 21
editReviewer() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_List method), 31
Email (class in Assign_GUP.email_mvc_view), 15
EmailTemplate (class in Assign_GUP.email_template), 15
eventFilter() (Assign_GUP.event_filters.ArrowKeysEventFilter method), 16
exists() (Assign_GUP.prop_mvc_data.AGUP_Proposals_List method), 20
exists() (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List method), 31
exists() (Assign_GUP.topics.Topics method), 35

F
first_logs() (Assign_GUP.history.Logger method), 17

G
get() (Assign_GUP.topics.Topics method), 35
get_forms_path() (in module Assign_GUP.resources), 27
getAssignedReviewers() (Assign_GUP.proposal.AGUP_Proposal_Data method), 24
getAssignment() (Assign_GUP.prop_revu_grid.ProposalReviewerRow method), 23
getAssignments() (Assign_GUP.reviewer.AGUP_Reviewer_Data method), 29
getByFullName() (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List method), 30
getByIndex() (Assign_GUP.prop_mvc_data.AGUP_Proposals_List method), 20
getByIndex() (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List method), 30
getCycle() (Assign_GUP.agup_data.AGUP_Data method), 13
getEmail() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 29
getEmailKeywords() (Assign_GUP.settings.ApplicationQSettings method), 32
getExcludedReviewers() (Assign_GUP.proposal.AGUP_Proposal_Data method), 25
getFullName() (Assign_GUP.reviewer.AGUP_Reviewer_Data method), 29
getFullName() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 29
getJoined() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 29
getKey() (Assign_GUP.proposal.AGUP_Proposal_Data method), 25
getKey() (Assign_GUP.reviewer.AGUP_Reviewer_Data method), 29
getKey() (Assign_GUP.settings.ApplicationQSettings method), 32
getNotes() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 30
getOpenFileName() (Assign_GUP.main_window.AGUP_MainWindow method), 19
getPhone() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 30
getProposal() (Assign_GUP.prop_mvc_data.AGUP_Proposals_List method), 20

[getProposalId\(\)](#) (Assign_GUP.proposal_details.AGUP_Proposal_Details method), 26
[getPrpFile\(\)](#) (Assign_GUP.settings.ApplicationQSettings importReviewers() (Assign_GUP.agup_data.AGUP_Data method), 13
 method), 33
[getReviewCycle\(\)](#) (Assign_GUP.settings.ApplicationQSettings method), 19
 method), 33
[getReviewCycleText\(\)](#) (Assign_GUP.main_window.AGUP_MainWindow importTopics() (Assign_GUP.agup_data.AGUP_Data method), 13
 method), 19
[getReviewer\(\)](#) (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List importTopics() (Assign_GUP.main_window.AGUP_MainWindow method), 19
 method), 30
[getReviewerAssignments\(\)](#) (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List importXml() (Assign_GUP.email_template.EmailTemplate method), 15
 method), 30
[getScores\(\)](#) (Assign_GUP.auto_assignment.Auto_Assign importXml() (Assign_GUP.prop_mvc_data.AGUP_Proposals_List method), 20
 method), 13
[getSliderValue\(\)](#) (Assign_GUP.topic_slider.AGUP_TopicSlider importXml() (Assign_GUP.proposal.AGUP_Proposal_Data method), 25
 method), 34
[getSortName\(\)](#) (Assign_GUP.reviewer.AGUP_Reviewer_Data importXml() (Assign_GUP.reviewer.AGUP_Reviewer_Data method), 29
 method), 29
[getSortName\(\)](#) (Assign_GUP.reviewer_details.AGUP_Reviewer_Details importXml() (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List method), 30
 method), 30
[getSubjects\(\)](#) (Assign_GUP.proposal.AGUP_Proposal_Data IncorrectXmlRootTag, 37
 method), 25
[getTopic\(\)](#) (Assign_GUP.proposal.AGUP_Proposal_Data index_to_ID() (Assign_GUP.email_mvc_view.AGUP_Emails_View method), 15
 method), 25
[getTopic\(\)](#) (Assign_GUP.reviewer.AGUP_Reviewer_Data index_to_ID() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View method), 21
 method), 29
[getTopicList\(\)](#) (Assign_GUP.proposal.AGUP_Proposal_Data index_to_ID() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View method), 31
 method), 25
[getTopicList\(\)](#) (Assign_GUP.reviewer.AGUP_Reviewer_Data InfoBox (class in Assign_GUP.about), 12
 method), 29
[getTopicList\(\)](#) (Assign_GUP.topics.Topics method), 36
[getTopicList\(\)](#) (Assign_GUP.topics_editor.AGUP_TopicsEditor inOrder() (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List method), 31
 method), 37
[getUrl\(\)](#) (Assign_GUP.reviewer_details.AGUP_Reviewer_Details inOrder() (Assign_GUP.topics.Topics method), 36
 method), 30
[getValue\(\)](#) (Assign_GUP.topic_slider.AGUP_TopicSlider InvalidWithXmlSchema, 37
 method), 34
[getXmlText\(\)](#) (in module Assign_GUP.xml_utility), 38
 isProposalListModified() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View method), 21
 isReviewerListModified() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View method), 31

H

[hasTopic\(\)](#) (Assign_GUP.proposal.AGUP_Proposal_Data method), 25

I

[importEmailTemplate\(\)](#) (Assign_GUP.agup_data.AGUP_Data method), 13
[importProposals\(\)](#) (Assign_GUP.agup_data.AGUP_Data method), 13
[importProposals\(\)](#) (Assign_GUP.main_window.AGUP_MainWindow method), 19

K

[keyExists\(\)](#) (Assign_GUP.settings.ApplicationQSettings method), 33
[keyOrder\(\)](#) (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List method), 31

L

[loadUi\(\)](#) (in module Assign_GUP.resources), 27
[Logger](#) (class in Assign_GUP.history), 16
[logMinorDetails\(\)](#) (in module Assign_GUP.history), 17

M

[mail_merge\(\)](#) (Assign_GUP.email_template.EmailTemplate

- method), 15
 - main() (in module Assign_GUP.email_mvc_view), 15
 - main() (in module Assign_GUP.main_window), 19
 - main() (in module Assign_GUP.revu_mvc_view), 32
 - makeText() (Assign_GUP.report_analysis_grid.Report method), 26
 - makeText() (Assign_GUP.report_assignments.Report method), 27
 - makeText() (Assign_GUP.report_summary.Report method), 27
 - mass_merge() (Assign_GUP.email_template.EmailTemplate method), 16
 - moveEvent() (Assign_GUP.plainTextEdit.TextWindow method), 20
- O**
- on_item_clicked() (Assign_GUP.email_mvc_view.AGUP_Emails_View method), 15
 - on_item_clicked() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View method), 21
 - on_item_clicked() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View method), 31
 - onAdd() (Assign_GUP.editor_email_template.Editor method), 14
 - onAdd() (Assign_GUP.topics_editor.AGUP_TopicsEditor method), 37
 - onAssignmentsChanged() (Assign_GUP.main_window.AGUP_MainWindow method), 19
 - onAssignmentsChanged() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View method), 21
 - onAssignmentsChanged() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26
 - onCheck() (Assign_GUP.prop_revu_grid.ReviewerAssignmentGrid method), 24
 - onCheckBoxClick() (Assign_GUP.prop_revu_grid.ProposalReviewerRow method), 23
 - onCloseButton() (Assign_GUP.topics_editor.AGUP_TopicsEditor method), 37
 - onDelete() (Assign_GUP.editor_email_template.Editor method), 14
 - onDelete() (Assign_GUP.topics_editor.AGUP_TopicsEditor method), 37
 - onDetailsModified() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View method), 31
 - onSliderChange() (Assign_GUP.topic_slider.AGUP_TopicSlider method), 34
 - onTemplateChanged() (Assign_GUP.main_window.AGUP_MainWindow method), 19
 - onTopicValueChanged() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View method), 21
 - onTopicValueChanged() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26
 - onTopicValueChanged() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 30
 - onTopicValueChanged() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View method), 31
 - onTopicValuesChanged() (Assign_GUP.main_window.AGUP_MainWindow method), 19
 - onValueChange() (Assign_GUP.topic_slider.AGUP_TopicSlider method), 34
 - openPrpFile() (Assign_GUP.agup_data.AGUP_Data method), 13
 - openPrpFile() (Assign_GUP.main_window.AGUP_MainWindow method), 19
- P**
- process_command_line() (in module Assign_GUP.main_window), 19
 - ProposalReviewerRow (class in Assign_GUP.prop_revu_grid), 22
- R**
- readValidXmlDoc() (in module Assign_GUP.xml_utility), 38
 - recalc() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View method), 21
 - remove() (Assign_GUP.topics.Topics method), 36
 - removeReviewer() (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List method), 31
 - removeTopic() (Assign_GUP.prop_mvc_data.AGUP_Proposals_List method), 21
 - removeTopic() (Assign_GUP.proposal.AGUP_Proposal_Data method), 25
 - removeTopic() (Assign_GUP.reviewer.AGUP_Reviewer_Data method), 29
 - removeTopic() (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List method), 31
 - removeTopics() (Assign_GUP.prop_mvc_data.AGUP_Proposals_List method), 21
 - removeTopics() (Assign_GUP.proposal.AGUP_Proposal_Data method), 25
 - removeTopics() (Assign_GUP.reviewer.AGUP_Reviewer_Data method), 29

removeTopics() (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List
 method), 31
 removeTopics() (Assign_GUP.topics.Topics method), 36
 Report (class in Assign_GUP.report_analysis_grid), 26
 Report (class in Assign_GUP.report_assignments), 27
 Report (class in Assign_GUP.report_summary), 27
 requestConfirmation() (Assign_GUP.main_window.AGUP_MainWindow
 method), 19
 resetDefaults() (Assign_GUP.settings.ApplicationQSettings
 method), 33
 resource_file() (in module Assign_GUP.resources), 28
 restoreSplitterDetails() (Assign_GUP.editor_email_template.Editor
 method), 14
 restoreSplitterDetails() (Assign_GUP.proposal_details.AGUP_ProposalDetails
 method), 26
 restoreSplitterDetails() (Assign_GUP.viewer_details.AGUP_ReviewerDetails
 method), 30
 restoreSplitterDetails() (Assign_GUP.settings.ApplicationQSettings
 method), 33
 restoreWindowGeometry() (Assign_GUP.editor_email_template.Editor
 method), 14
 restoreWindowGeometry() (Assign_GUP.email_mvc_view.AGUP_Emails_View
 method), 15
 restoreWindowGeometry() (Assign_GUP.main_window.AGUP_MainWindow
 method), 19
 restoreWindowGeometry() (Assign_GUP.plainTextEdit.TextWindow method),
 20
 restoreWindowGeometry() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View
 method), 21
 restoreWindowGeometry() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View
 method), 32
 restoreWindowGeometry() (Assign_GUP.settings.ApplicationQSettings
 method), 33
 restoreWindowGeometry() (Assign_GUP.topics_editor.AGUP_TopicsEditor
 method), 37
 ReviewerAssignmentGridLayout (class in As-
 sign_GUP.prop_revu_grid), 23
 rowCheck() (Assign_GUP.prop_revu_grid.ProposalReviewerRow
 method), 23
 rowCount() (Assign_GUP.general_mvc_model.AGUP_MVC_Model
 method), 16
 saveEmailKeywords() (Assign_GUP.settings.ApplicationQSettings
 method), 33
 saveReviewerDetails() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View
 method), 32
 saveSplitterDetails() (Assign_GUP.editor_email_template.Editor
 method), 14
 saveSplitterDetails() (Assign_GUP.proposal_details.AGUP_ProposalDetails
 method), 26
 saveSplitterDetails() (Assign_GUP.viewer_details.AGUP_ReviewerDetails
 method), 30
 saveSplitterDetails() (Assign_GUP.settings.ApplicationQSettings
 method), 33
 saveWindowGeometry() (Assign_GUP.editor_email_template.Editor
 method), 14
 saveWindowGeometry() (Assign_GUP.email_mvc_view.AGUP_Emails_View
 method), 15
 saveWindowGeometry() (Assign_GUP.main_window.AGUP_MainWindow
 method), 19
 saveWindowGeometry() (Assign_GUP.plainTextEdit.TextWindow method),
 20
 saveWindowGeometry() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View
 method), 22
 saveWindowGeometry() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View
 method), 32
 saveWindowGeometry() (Assign_GUP.settings.ApplicationQSettings
 method), 33
 saveWindowGeometry() (Assign_GUP.topics_editor.AGUP_TopicsEditor
 method), 37
 selectFirstKeyword() (Assign_GUP.editor_email_template.Editor
 method), 14
 selectFirstListItem() (Assign_GUP.email_mvc_view.AGUP_Emails_View
 method), 15
 selectFirstListItem() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View
 method), 22
 selectFirstListItem() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View
 method), 32

method), 32

selectModelByIndex() (Assign_GUP.email_mvc_view.AGUP_Emails_View method), 15

selectModelByIndex() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View method), 22

selectModelByIndex() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View method), 32

send() (Assign_GUP.email_mvc_view.Email method), 15

set() (Assign_GUP.topics.Topics method), 36

setAll() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26

setAssignedReviewer() (Assign_GUP.proposal.AGUP_Proposal_Data method), 25

setAssignment() (Assign_GUP.prop_revu_grid.ProposalReviewerRow method), 23

setAssignment() (Assign_GUP.prop_revu_grid.ReviewerAssignmentGridLayout method), 24

setButtonBackground() (in module Assign_GUP.qt_utils), 26

setEmail() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 30

setEnabled() (Assign_GUP.prop_revu_grid.ReviewerAssignmentGridLayout method), 24

setFirstChoiceBl() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26

setFullName() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 30

setIndicators() (Assign_GUP.main_window.AGUP_MainWindow method), 19

setJoined() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 30

setKey() (Assign_GUP.reviewer.AGUP_Reviewer_Data method), 29

setKey() (Assign_GUP.settings.ApplicationQSettings method), 33

setModel() (Assign_GUP.email_mvc_view.AGUP_Emails_View method), 15

setModel() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View method), 22

setModel() (Assign_GUP.revu_mvc_view.AGUP_Reviewers_View method), 32

setNotes() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 30

setNumberAssigned() (Assign_GUP.prop_revu_grid.ProposalReviewerRow method), 23

setNumProposalsWidget() (Assign_GUP.main_window.AGUP_MainWindow method), 19

setNumReviewersWidget() (Assign_GUP.main_window.AGUP_MainWindow method), 19

setNumTopicsWidget() (Assign_GUP.main_window.AGUP_MainWindow method), 19

setPhone() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 30

setProposal() (Assign_GUP.prop_revu_grid.ProposalReviewerRow method), 23

setProposal() (Assign_GUP.prop_revu_grid.ReviewerAssignmentGridLayout method), 24

setProposalId() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26

setProposalTitle() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26

setPrpFile() (Assign_GUP.settings.ApplicationQSettings method), 33

setPrpFileText() (Assign_GUP.main_window.AGUP_MainWindow method), 19

setRcFileText() (Assign_GUP.main_window.AGUP_MainWindow method), 19

setReviewCycle() (Assign_GUP.settings.ApplicationQSettings method), 33

setReviewCycleText() (Assign_GUP.main_window.AGUP_MainWindow method), 19

setReviewersModel() (Assign_GUP.agup_data.AGUP_Data method), 13

setReviewersValues() (Assign_GUP.prop_revu_grid.ReviewerAssignmentGridLayout method), 24

setReviewPeriod() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26

setSliderValue() (Assign_GUP.topic_slider.AGUP_TopicSlider method), 34

setSortName() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 30

setSpkName() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26

setSubjects() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26

setText() (Assign_GUP.plainTextEdit.TextWindow method), 20

setTopicsGroupName() (Assign_GUP.plainTextEdit.TextWindow method), 20

setTitle() (Assign_GUP.plainTextEdit.TextWindow method), 20

setTopic() (Assign_GUP.proposal.AGUP_Proposal_Data method), 25

setTopic() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 25

method), 26
setTopic() (Assign_GUP.reviewer.AGUP_Reviewer_Data method), 29
setTopic() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 30
setTopicValue() (Assign_GUP.prop_mvc_data.AGUP_Proposals_List method), 21
setTopicValue() (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List method), 31
setupProposal() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26
setUrl() (Assign_GUP.reviewer_details.AGUP_ReviewerDetails method), 30
setValue() (Assign_GUP.prop_revu_grid.ProposalReviewerRow method), 23
setValue() (Assign_GUP.prop_revu_grid.ReviewerAssignmentGridLayout method), 24
setValue() (Assign_GUP.topic_slider.AGUP_TopicSlider method), 35
setWidgetBackground() (in module Assign_GUP.qt_utils), 26
showReviewerEmail() (Assign_GUP.email_mvc_view.AGUP_Emails_View method), 15
simpleAssignment() (Assign_GUP.auto_assignment.Auto_Assign method), 13
sortListUnique() (in module Assign_GUP.topics), 37
synchronizeTopics() (in module Assign_GUP.topics), 37

updateTimeStamp() (Assign_GUP.settings.ApplicationQSettings method), 33
validate() (in module Assign_GUP.xml_utility), 38
valueOrder() (Assign_GUP.topics.Topics method), 36

W
write() (Assign_GUP.agup_data.AGUP_Data method), 13
writeXml() (Assign_GUP.topics.Topics method), 36
writeXmlNode() (Assign_GUP.email_template.EmailTemplate method), 16
writeXmlNode() (Assign_GUP.prop_mvc_data.AGUP_Proposals_List method), 21
writeXmlNode() (Assign_GUP.proposal.AGUP_Proposal_Data method), 25
writeXmlNode() (Assign_GUP.reviewer.AGUP_Reviewer_Data method), 29
writeXmlNode() (Assign_GUP.revu_mvc_data.AGUP_Reviewers_List method), 31

X
XmlSyntaxError, 38

T

text_decode() (in module Assign_GUP.tools), 34
text_encode() (in module Assign_GUP.tools), 34
TextWindow (class in Assign_GUP.plainTextEdit), 20
Topics (class in Assign_GUP.topics), 35

U

update() (Assign_GUP.prop_mvc_view.AGUP_Proposals_View method), 22
update() (Assign_GUP.prop_revu_grid.ProposalReviewerRow method), 23
update() (Assign_GUP.prop_revu_grid.ReviewerAssignmentGridLayout method), 24
update() (Assign_GUP.proposal_details.AGUP_ProposalDetails method), 26
update() (Assign_GUP.report_analysis_grid.Report method), 27
update() (Assign_GUP.report_assignments.Report method), 27
update() (Assign_GUP.report_summary.Report method), 27
updateNumbersOfAssignedProposals() (Assign_GUP.prop_revu_grid.ReviewerAssignmentGridLayout method), 24