Excel-operator API Documentation

The excel operator is composed by three parts, model to store the data, view to show content and controller to handle all the actions.

1.Model

There are six sub-packages in the Model package.

- Workbook
- Workcell
- Worksheet
- CASbook
- CASsheet
- PSbook
- PSsheet

Folder structure

```
model
├── __init__.py
├── Workbook.py
├── Workcell.py
├── Worksheet.py
├── CASbook.py
├── CASsheet.py
├── PSbook.py
└── PSsheet.py
```

Workbook Package

There is only one class in Workbook package.

- class Workbook(object)

Workbook Class Reference

The Workbook is a abstract class that used to store the instance created by xlwings and openpyxl.

Pubilc Methods

- def __init__(self,workbook = None,app = None)
- def __del__(self)
- def init_book(self,workbook,app)
- def init_model(self)
- def update_model(self)
- def update_sheet_name_model(self)
- def load_sheets(self,sheet_cls,sheets,sheets_wr)
- def load_sheets_name(self,sheets)
- def open(self,path_name)
- def save_as(self,path_name)
- def save(self)

- def recover(self,direction)
- def sheets(self)
- def sheet_name_model(self)
- def workbook(self)
- def workbook_wr(self)
- def workbook_name(self)
- def sheetnames(self)

Private Methods

- _workbook
- _workbook_wr
- _workbook_name
- _sheets
- _sheet_name_model

Detailed Description

The Workbook provides a series of interface that could be used for both PS file and CAS file. If there are some particular methods only for PS file or CAS file, they will be defined in PSbook class and CASbook class.

Method Documentation

_workbook

Store the workbook instance created by openpyxl for reading.

_workbook_wr

Store the workbook instance created by xlwings for writing.

_workbook_name

Store the name of the workbook.

_sheets

A list to store all the sheets object.

_sheet_name_model

A Qt item model object for presenting sheets name.

def __init__(self,workbook = None,app = None)

Create a new Workbook instance with the given **workbook** and **app**.

def __del__(self)

Deletes all sheets hold by the workbook instance and release the workbook instance.

def init_book(self,workbook,app)

Load workbook by xlwings and openpyxl, the xlwings is used to write and the openpyxl is used to read.

def init_model(self)

Initialize the workbook's name and create a new model to hold all the sheet's name.

def update_model(self)

Whenever workbook changed, call update_model() to keep the Workbook model update.
For now, the Workbook only keep a sheet name model.

def update_sheet_name_model(self)

Read the sheet's name and fill the model.

def load_sheets(self,sheet_cls,sheets,sheets_wr)

Called by the child class.
sheet_cls provide a specified class with some special method that could be applied for the sheet.
sheets is used to read.
sheets_wr is used to write.

def load_sheets_name(self,sheets)

This method is used to load every sheet's name to a list.

def open(self,path_name)

Virtual function, only a placeholder that waiting for overwrite by child class.

def save_as(self,path_name)

Save the workbook with the given **path_name** by xlwings.

def save(self)

Virtual function, only a placeholder that waiting for overwrite by child class.

def recover(self,direction)

Virtual function, only a placeholder that waiting for overwrite by child class.

def sheets(self)

Property member, provide a interface to access **_sheets**.

def sheet_name_model(self)

Property member, provide a interface to access **_sheet_name_model**.

def workbook(self)

Property member, provide a interface to access **_workbook**.

def workbook_wr(self)

Property member, provide a interface to access **_workbook_wr**.

def workbook_name(self)

Property member, provide a interface to access **_workbook_name**.

def sheetnames(self)

Property member, provide a interface to access **_workbook.sheetname**.

Workcell Package

The Workcell package contains four classes.

- class Workcell(object)
- class XmlName(Workcell)

- class Header(Workcell)
- class Status(Workcell)

Workcell Class Reference

Public Methods

- def __init__(self, cell = None, sheet_wr = None, header = None, xmlname = None)
- def cell(self)
- def row(self)
- def col(self)
- def column(self)
- def col_letter(self)
- def value(self)
- def value(self, value)
- def header(self)
- def xmlname(self)
- def sheet_wr(self)

Private Methods

- _cell
- _header
- _xmlname
- _sheet_wr

Detailed Description

Workcell is a base class that provides interfaces to access cells via xlwings or openpyxl.

Method Documentation

_cell

Store the cell instance created by openpyxl.

_header

Store the Header object related to this cell.

_xmlname

Store the Xmlname object related to this cell.

_sheet_wr

Store the sheet which this cell belongs.

def __init__(self,cell = None,sheet_wr = None,header = None,xmlname = None)

Create a new Workcell object with the given **cell**, **sheet_wr**, **header**, **xmlname**.

def cell(self)

Property member, provide a interface to access **_cell**.

def row(self)

Property member, provide a interface to access **_cell.row**.

def col(self)

Property member, provide a interface to access **_cell.col_idx**.

def column(self)

Property member, provide a interface to access **_cell.col_idx**.

def col_letter(self)

Property member, provide a interface to access **_cell.column**.

def value(self)

Property member, provide a interface to read **_cell.value**.
It is completed by openpyxl.

def value(self,value)

Property member, provide a interface to write **_sheet_wr.api.Range(self.row,self.column).value**.
It is completed by xlwings.

def header(self)

Property member, provide a interface to access **_header**.

def xmlname(self)

Property member, provide a interface to access **_xmlname**.

def sheet_wr(self)

Property member, provide a interface to access **_sheet_wr**.

Xmlname Class Reference

Inherits **Workcell**.

Public Methods

- def __init__(self,cell = None,sheet_wr = None)
- def get_item_by_header(self,header)

Private Methods

Inherits Workcell.

Detailed Description

Xmlname is a child class of Workcell. It is used to store the xmlname cells.

Method Documentation

def __init__(self,cell = None,sheet_wr = None)

Create a new Xmlname object with given **cell** and **sheet_wr**.

get_item_by_header(self,header)

Return the cell which is located at the cross of the row of xmlanme and the column of header.

Header Class Reference

Inherits **Workcell**.

Public Methods

- def __init__(self,cell = None,sheet_wr = None)
- def get_item_by_xmlname(self,xmlname)

Private Methods

Inherits Workcell.

Detailed Description

Header is a child class of Workcell. It is used to store the header cells.

Method Documentation

def __init__(self,cell = None,sheet_wr = None)

Create a new Header object with given **cell** and **sheet_wr**.

get_item_by_xmlname(self,xmlname)

Return the cell which is located at the cross of the row of xmlanme and the column of header.

Status Class Reference

Inherits **Workcell**.

Public Methods

- def __init__(self,cell = None,sheet_wr = None)
- def get_item_by_xmlname(self,xmlname)

Private Methods

Inherits Workcell.

Detailed Description

Status is a child class of Workcell. It is used to store the status cells.

Method Documentation

def __init__(self,cell = None,sheet_wr = None)

Create a new Header object with given **cell** and **sheet_wr**.

get_item_by_xmlname(self,xmlname)

Return the cell which is located at the cross of the row of xmlanme and the column of Status.

Worksheet Package

The Worksheet package contains four classes.

- class Worksheet(object)
- class QPreviewItem(QStandardItem)
- class QComparisonItem(QStandardItem)

- class QHeaderItem(QStandardItem)

Worksheet Class Reference

Public Methods

- def __init__(self,sheet = None,sheet_wr = None)
- def __del__(self)
- def init_sheet(self)
- def init_model(self)
- def update_model(self)
- def search_by_value(self,value)
- def search_header_by_value(self,value)
- def search_xmlname_by_value(self,value)
- def xml_names(self)
- def xml_names_value(self)
- def headers(self)
- def headers_value(self)
- def select_all_headers(self)
- def unselect_all_headers(self)
- def cell(self,row,col)
- def cell_value(self,row,col)
- def cell_wr(self,row,col)
- def cell_wr_value(self,row,col)
- def xmlname(self)
- def preview_model(self)
- def header_model(self)
- def header_list(self)
- def xml_name_model(self)
- def extended_preview_model(self)
- def checked_headers(self)
- def worksheet(self)
- def rows(self)
- def cols(self)
- def max_row(self)
- def min_row(self)
- def max_col(self)
- def min_col(self)

Private Methods

- _worksheet
- _worksheet_wr
- _xmlname
- _header_model

Detailed Description

Worksheet is a abstract class to define the general methods of worksheet that could be applied on both PS file and CAS file. If there are some particular methods only for PS file or CAS file, they will be defined in PSsheet class and CASsheet class.

Method Documentation

_worksheet

Store the worksheet object created by openpyxl for reading.

_worksheet_wr

Store the worksheet object created by xlwings for writing.

_xmlname

Store the 'xmlname' cell object. If there is no 'xmlname', the value will be 'None'.

_header_model

A Qt item model for storing the all the headers.

def __init__(self,sheet = None,sheet_wr = None)

Create a new Worksheet object with the given **sheet** and **sheet_wr**.

def __del__(self)

Release all the **_worksheet** and **_worksheet_wr**.

def init_sheet(self)

Initialize the **_xmlname**.

def init_model(self)

Create a Qt item model to store all the headers.

def update_model(self)

Reload the **_header_model** and keep the model update.

def search_by_value(self,value)

Search and return the Workcell object by the value of targe cell.

def search_header_by_value(self,value)

Search and return the Header object by the value of target cell.

def search_xmlname_by_value(self,value)

Search and return the Xmlname object by the value of target cell.

def xml_names(self)

Return a list of Xmlname object which contains all the xmlnames in this worksheet.

def xml_names_value(self)

Return a list of xmlname string of all the xmlnames in this worksheet.

def headers(self)

Return a list of Header object which contains all the headers in this worksheet.

def headers_value(self)

Return a list of header string of all the headers in this worksheet.

def select_all_headers(self)

Set the states of all the items of **_header_model** to **Qt.Checked**.

def unselect_all_headers(self)

Set the states of all the items of **_header_model** to **Qt.Unchecked**.

def cell(self,row,col)

Return the specified cell with the given **row** and **col** via openpyxl.

def cell_value(self,row,col)

Retrun the value of specified cell with the given **row** and **col** via openpyxl.

def cell_wr(self,row,col)

Return the specified cell with the given **row** and **col** via xlwings.

def cell_wr_value(self,row,col)

Return the value of specified cell with the given **row** and **col** via xlwings.

def xmlname(self)

Property member, provide a interface to access the **_xmlname**.

def preview_model(self)

Property member, provide a interface to access the **_preview_model**.

def header_model(self)

Property member, provide a interface to access the **_header_model**.

def header_list(self)

Property member. Return a list of header string of all the headers in this worksheet. If there is not a 'xmlname' cell, return a blank list instead.

def xml_name_model(self)

Property member, provide a interface to access the **_xml_name_model**.

def extended_preview_model(self)

Property member, provide a interface to access the **_extended_preview_model**.

def checked_headers(self)

Return a list of all the checked item in **_header_model**.

def worksheet(self)

Property member, provide a interface to access **_worksheet**.

def rows(self)

Property member, provide a interface to access **_worksheet.rows**.

def cols(self)

Property member, provide a interface to access **_worksheet.columns**.

def max_row(self)

Property member, provide a interface to access **_worksheet.max_row**.

def min_row(self)

Property member, provide a interface to access **_worksheet.min_row**.

def max_col(self)

Property member, provide a interface to access **_worksheet.max_column**.

def min_col(self)

Property member, provide a interface to access **_worksheet.min_column**.

QPreviewItem Class Reference

Inherits **QStandardItem**.

Public Methods

- def __init__(self,cell)
- def cell(self)
- def value(self)
- def row(self)
- def col(self)
- def col_letter(self)

Private Methods

- _cell

Detailed Description

QPreviewItem inherits from QStandardItem.

Method Documentation

_cell

Store the Workcell object.

def init(self,cell)

Create a new QPreviewItem object with the given **cell**.

def cell(self)

Property member, provide a interface to access the **_cell**.

def value(self)

Property member, provide a interface to access the **_cell.value**.

def row(self)

Property member, provide a interface to access the **_cell.row**.

def col(self)

Property member, provide a interface to access the **_cell.col**.

def col_letter(self)

Property member, provide a interface to access the **_cell.col_letter**.

QComparisonItem Class Reference

Inherits **QStandardItem**.

Public Methods

- def __init__(self,cell)
- def cell(self)
- def value(self)
- def row(self)
- def col(self)
- def col_letter(self)

Private Methods

- _cell

Detailed Description

QComparisonItem inherits from QStandardItem.

Method Documentation

_cell

Store the Workcell object.

def init(self,cell)

Create a new QComparisonItem object with the given **cell**.

def cell(self)

Property member, provide a interface to access the **_cell**.

def value(self)

Property member, provide a interface to access the **_cell.value**.

def row(self)

Property member, provide a interface to access the **_cell.row**.

def col(self)

Property member, provide a interface to access the **_cell.col**.

def col_letter(self)

Property member, provide a interface to access the **_cell.col_letter**.

QHeaderItem Class Reference

Inherits **QStandardItem**.

Public Methods

- def __init__(self,cell)
- def get_item_by_xmlname(self,xmlanme)
- def cell(self)
- def value(self)
- def row(self)
- def col(self)
- def col_letter(self)

Private Methods

- _cell

Detailed Description

QHeaderItem inherits from QStandardItem.

Method Documentation

_cell

Store the Workcell object.

def init(self,cell)

Create a new QHeaderItem object with the given **cell**.

def get_item_by_xmlname(self,xmlname)

Return the cell object which is located at the cross of the row of xmlname and the column of **_cell**.

def cell(self)

Property member, provide a interface to access the **_cell**.

def value(self)

Property member, provide a interface to access the **_cell.value**.

def row(self)

Property member, provide a interface to access the **_cell.row**.

def col(self)

Property member, provide a interface to access the **_cell.col**.

def col_letter(self)

Property member, provide a interface to access the **_cell.col_letter**.

CASbook package

The CASbook package contains one class.

- class CASbook(Workbook)

CASbook Class Reference

Inherits **Workbook**.

Public Methods

- def __init__(self,file_name = None,app = None)
- def init_cas_book(self)

Private Methods

Inherits from Workbook class.

Detailed Description

The only difference between CASbook and PSbook is about the **load_sheets**.
load_sheets is used to create specified Worksheet objects for each sheet and load them into a list.
For CASbook, the specified Worksheet object should be CASsheet.

Method Documentation

def __init__(self,file_name = None,app = None)

Create a new CASbook objects with the given **file_name** and **app**.

def init_cas_book(self)

Create CASsheet objects for each worksheet and load them into a list by **load_sheets**.

CASsheet package

The CASsheet package contains one class.

- class CASsheet(Worksheet)

CASsheet Class Reference

Inherits **Worksheet**.

Public Methods

- def __init__(self,sheet = None,sheet_wr = None)
- def init_cas_sheet(self)
- def cell(self,row,col)

Private Methods

- _subject_matter
- _container_name

Detailed Description

While initializing CASsheet object, it will search for two headers, 'Subject Matter Functional Area' and 'Container Name Technical Specification'.

Method Documentation

_subject_matter

Store the **QHeaderItem** object as the searching result of cell 'Subject Matter Functional Area'.

_container_name

Store the **QHeaderItem** object as the searching result of cell 'Container Name Technical Specification'.

def __init__(self,sheet = None,sheet_wr = None)

Create a new CASsheet object with the given **sheet** and **sheet_wr**.

def init_cas_sheet(self)

If there is a 'xmlname' cell in this worksheet, search for 'Subject Matter Functional Area' and 'Container Name Technical Specification' and store the result.

def cell(self,row,col)

Return the cell object via xlwings.

PSbook package

The PSbook package contains one class.

- class PSbook(Workbook)

PSbook Class Reference

Inherits **Workbook**.

Public Methods

- def __init__(self,file_name = None,app = None)
- def init_ps_book(self)

Private Methods

Inherits from Workbook class.

Detailed Description

The only difference between CASbook and PSbook is about the **load_sheets**.
load_sheets is used to create specified Worksheet objects for each sheet and load them into a list.
For PSbook, the specified Worksheet object should be PSsheet.

Method Documentation

def __init__(self,file_name = None,app = None)

Create a new PSbook objects with the given **file_name** and **app**.

def init_cas_book(self)

Create PSsheet objects for each worksheet and load them into a list by **load_sheets**.

PSsheet package

The PSsheet package contains one class.

- class PSsheet(Worksheet)

PSsheet Class Reference

Inherits **Worksheet**.

Public Methods

- def __init__(self,sheet = None,sheet_wr = None)
- def __del__(self)
- def init_ps_sheet(self)
- def init_ps_model(self)
- def update_model(self)

- def status(self)
- def cell(self,row,col)
- def auto_fit(self,cols)
- def add_row(self,start_pos,offset,orientation)
- def delete_row(self,start_pos,offset)
- def lock_row(self,row,status)
- def lock_sheet(self)
- def unlock_sheet(self)
- def unlock_all_cells(self)
- def extended_preview_model(self)
- def extended_preview_model_list(self)
- def preview_model(self)

Private Methods

- _status
- _subject_matter
- _container_name
- _preview_model
- _preview_model_list
- _extended_preview_model
- _extended_preview_model_list

Detailed Description

PSsheet class provides more interfaces than Worksheet.The class hold two data model, **_preview_model** for the four columns preview window and **_extended_preview_model** for the full content preview mode.It also allow user to append, delete and lock rows in the worksheet.

Method Documentation

_status

Store the **QHeaderItem** object as the searching result of cell 'Status(POR,INIT,PREV)'.

_subject_matter

Store the **QHeaderItem** object as the searching result of cell 'Subject Matter Functional Area'.

_container_name

Store the **QHeaderItem** object as the searching result of cell 'Container Name Technical Specification'.

_preview_model

Store the data model for the content of four columns preview window.

_preview_model_list

Convert **_preview_model** to list for transmitting between multiprocessing.

_extended_preview_model

Store the data model for the full content of preview window.

_extended_preview_model_list

def __init__(self,sheet = None,sheet_wr = None)

Create a new PSsheet object with the given **sheet** and **sheet_wr**.
Initialize the **_preview_model**, **_preview_model_list**, **_extended_preview_model** and

_extended_preview_model_list.
Search for all the headers and constract the **_preview_model**.

def __del__(self)

Release the **_preview_model** and **_extended_preview_model**.

def init_ps_sheet(self)

Search for headers that

def init_ps_model(self)

Constract the data model for four columns preview window.

def update_model(self)

Re-initialize the **_preview_model** and keep the content update.

def status(self)

Return a list of Status object which contains all the status items in this worksheet.

def cell(self,row,col)

Return the cell object for reading via openpyxl.

def auto_fit(self,cols)

Adjust the columns width automatically with the given **cols**. **cols** should be a collection of the columns you want to adjust.

def add_row(self,start_pos,offset,orientation)

Insert several rows below the specified row with the given **start_pos**, **offset**, **orientation**.
start_pos indicates the position of the row that you want to insert below.
offset represents the number of rows you want to insert.
orientation represents the direction of insertion.

def delete_row(self,start_pos,offset)

Delete several rows from the specified row with the given **start_pos**, **offset**.
start_pos indicates the start position of the row that you want to delete.
offset represents the number of rows you want to delete.

def lock_row(self,row,status)

Set the row's protection mode with the given **row** and **status**.
row represents the row you want to handle.
status represents the target status you want to set. For example, **True** stands for locked and **False** stands for unlocked.

def lock_sheet(self)

Set the worksheet's protection mode to locked.

def unlock_sheet(self)

Set the worksheet's protection mode to unlocked.

def unlock_all_cells(self)

Set the protection mode of all the cells in this worksheet to unlocked.

def extended_preview_model(self)

Work through the whole worksheet and constrcut a data model for the full content preview window. The construction of data model would only be executed for the first time. The following calls would return the result of the first call.

def extended_preview_model_list(self)

Convert **_extended_preview_model** to string list to adapter the multiproccessing communication on Windows.

def preview_model(self)

Property member, provide a interface to access the **_preview_model_list**.

2.View

There are four sub-packages in the Model package.

- ExtendedPreview
- ExtendedPreviewUI
- Window
- WindowUI

Folder structure

```
view
├── ExtendedPreview.py
├── ExtendedPreview.ui
├── ExtendedPreviewUI.py
├── icon.png
├── __init__.py
├── resource.qrc
├── resource_rc.py
├── tool.png
├── Window.py
├── Window.ui
└── WindowUI.py
```

ExtendedPreview Package

There is only one class in ExtendedPreview package.

- class ExtendedPreview(QMainWindow)

ExtendedPreview Class Reference

Inherits **QMainWindow**.

Public Methods

- def __init__(self, model = None)
- def init_Form(self)
- def init_model(self, model)
- def update_extended_preview(self, array)

Private Methods

- ui

Detailed Description

ExtendedPreivew is used to create the extended preview graphic user interface. There is a data model that contains all the necessary message for presenting full content of the worksheet.

Method Documentation

ui

Store the ExtendedPreviewUI object.

def init(self,model = None)

Create a new ExtendedPreview object with the given **model**.

def init_Form(self)

Initialize the Form UI object.

def init_model(self,model)

Initialize the data model for presenting full content of the worksheet.

def update_extended_preview(self,array)

Refresh the **extended_preview** data model to keep the content update.

ExtendedPreviewUI Package

There is only one class in ExtendedPreviewUI package.

- class Ui_Form(object)

ExtendedPreviewUI.Ui_Form Class Reference

Generated by pyuic4.

Public Methods

- def setupUi(self, Form)
- def retranslateUi(self, Form)

Private Methods

- extended_preview

Detailed Description

Please refer to Qt official documentation.

Method Documentation

Please refer to Qt official documentation.

Window Package

There are only one class in Window package.

- class Window(QMainWindow)

Window Class Reference

Inherits **QMainWindow**.

Public Methods

- def __init__(self)
- def init_Window(self)
- def bind_open_cas(self,func)
- def bind_open_ps(self,func)
- def bind_save_cas(self,func)
- def bind_save_ps(self,func)
- def bind_saveas_cas(self,func)
- def bind_saveas_ps(self,func)
- def bind_select_cas_sheet(self,func)
- def bind_select_ps_sheet(self,func)
- def bind_select_preview(self,func)
- def bind_sync_ps_to_cas(self,func)
- def bind_sync_cas_to_ps(self,func)
- def bind_sync_select_all_ps_headers(self,func)
- def bind_sync_select_all_cas_headers(self,func)
- def bind_comparison_start(self,func)
- def bind_comparison_delete(self,func)
- def bind_comparison_append(self,func)
- def bind_comparison_select_all_delete(self,func)
- def bind_comparison_select_all_append(self,func)
- def bind_preview_add(self,func)
- def bind_preview_delete(self,func)
- def bind_preview_lock(self,func)
- def bind_undo_cas(self,func)
- def bind_undo_ps(self,func)
- def bind_select_extended_preview(self,func)
- def bind_ps_header_changed(self,func)
- def bind_cas_header_changed(self,func)
- def bind_comparison_append_list_changed(self,func)
- def bind_comparison_delete_list_changed(self,func)
- def update_cas_file(self,filename)
- def update_ps_file(self,filename)
- def update_cas_sheets(self,sheetnames)
- def update_ps_sheets(self,sheetnames)
- def update_preview(self,itemss)
- def update_ps_header(self,headers)
- def update_cas_header(self,headers)
- def update_ps_header_selected(self,idx)
- def update_cas_header_selected(self,idx)
- def update_comparison_delete_list(self,deletes)
- def update_comparison_append_list(self,appends)
- def update_message(self,model)
- def update_msg(self,model)
- def update_selected_cell(self,model)
- def update_progressBar(self,model)
- def open_file_confirm(self)
- def pop_up_message(self,msg)

Private Methods

- ui

Detailed Description

Window is used to create the main program graphic user interface.

Method Documentation

ui

Store the ExtendedPreviewUI object.

def __init__(self)

Create a new Window object.

def init_Window(self)

Create WindowUI object and setup the Ui.
Set the window state to **Qt.WindowMaximized**.

def bind_open_cas(self,func)

Bind UI event to the given **func**.

def bind_open_ps(self,func)

Bind UI event to the given **func**.

def bind_save_cas(self,func)

Bind UI event to the given **func**.

def bind_save_ps(self,func)

Bind UI event to the given **func**.

def bind_saveas_cas(self,func)

Bind UI event to the given **func**.

def bind_saveas_ps(self,func)

Bind UI event to the given **func**.

def bind_select_cas_sheet(self,func)

Bind UI event to the given **func**.

def bind_select_ps_sheet(self,func)

Bind UI event to the given **func**.

def bind_select_preview(self,func)

Bind UI event to the given **func**.

def bind_sync_ps_to_cas(self,func)

Bind UI event to the given **func**.

def bind_sync_cas_to_ps(self,func)

Bind UI event to the given **func**.

def bind_sync_select_all_ps_headers(self,func)

Bind UI event to the given **func**.

def bind_sync_select_all_cas_headers(self,func)

Bind UI event to the given **func**.

def bind_comparison_start(self,func)

Bind UI event to the given **func**.

def bind_comparison_delete(self,func)

Bind UI event to the given **func**.

def bind_comparison_append(self,func)

Bind UI event to the given **func**.

def bind_comparison_select_all_delete(self,func)

Bind UI event to the given **func**.

def bind_comparison_select_all_append(self,func)

Bind UI event to the given **func**.

def bind_preview_add(self,func)

Bind UI event to the given **func**.

def bind_preview_delete(self,func)

Bind UI event to the given **func**.

def bind_preview_lock(self,func)

Bind UI event to the given **func**.

def bind_undo_cas(self,func)

Bind UI event to the given **func**.

def bind_undo_ps(self,func)

Bind UI event to the given **func**.

def bind_select_extended_preview(self,func)

Bind UI event to the given **func**.

def bind_ps_header_changed(self,func)

Bind UI event to the given **func**.

def bind_cas_header_changed(self,func)

Bind UI event to the given **func**.

def bind_comparison_append_list_changed(self,func)

Bind UI event to the given **func**.

def bind_comparison_delete_list_changed(self,func)

Bind UI event to the given **func**.

def update_cas_file(self,filename)

Refresh UI with the given **filename**.

def update_ps_file(self,filename)

Refresh UI with the given **filename**.

def update_cas_sheets(self,sheetnames)

Refresh UI with the given **sheetnames**.

def update_ps_sheets(self,sheetnames)

Refresh UI with the given **sheetnames**.

def update_preview(self,itemss)

Refresh UI with the given **itemss**.

def update_ps_header(self,headers)

Refresh UI with the given **headers**.

def update_cas_header(self,headers)

Refresh UI with the given **headers**.

def update_ps_header_selected(self,idx)

Refresh UI with the given **idx**.

def update_cas_header_selected(self,idx)

Refresh UI with the given **idx**.

def update_comparison_delete_list(self,deletes)

Refresh UI with the given **deletes**.

def update_comparison_append_list(self,appends)

Refresh UI with the given **appends**.

def update_message(self,model)

Refresh UI with the given **model**.

def update_msg(self,model)

Refresh UI with the given **model**.

def update_selected_cell(self,model)

Refresh UI with the given **model**.

def update_progressBar(self,model)

Refresh UI with the given **model**.

def open_file_confirm(self)

Open a file selection dialog for open file confirm.

def pop_up_message(self,msg)

Pop up a warning with the given **msg**.

WindowUI Package

There is only one class in WindowUI package.

- class Ui_MainWindow(object)

WindowUI.Ui_MainWindow Class Reference

Generated by pyuic4.

Public Methods

- def setupUi(self, MainWindow)
- def retranslateUi(self, MainWindow)

Private Methods

- centralwidget
- gridLayout_7
- horizontalLayout
- verticalLayout
- gridLayout
- label
- name_cas
- sheets_cas
- open_cas
- save_cas
- saveas_cas
- undo_cas
- label_2
- name_ps
- sheets_ps
- open_ps
- save_ps
- saveas_ps
- undo_ps
- gridLayout_3
- label_5
- comparison_start
- label_7
- label_8
- comparison_delete_list
- comparison_append_list
- comparison_delete
- comparison_select_all_append
- comparison_append
- comparison_select_all_delete
- gridLayout_6
- label_3
- label_4
- label_6
- ps_header
- cas_header
- sync_ps_to_cas

- sync_select_all_cas
- sync_cas_to_ps
- sync_select_all_ps
- verticalLayout_2
- gridLayout_4
- preview
- preview_add
- preview_delete
- preview_lock
- extended_preview
- gridLayout_5
- label_9
- selected_row
- label_10
- selected_col
- progressBar
- msg
- menubar
- statusbar

Detailed Description

Please refer to Qt official documentation.

Method Documentation

Please refer to Qt official documentation.

3.Controller

There are two sub-packages in the Controller package.

- FileStack
- MainController

Folder structure

```
controller/
├── FileStack.py
├── __init__.py
└── MainController.py
```

FileStack Package

There are three classes in FileStack package.

- class FileStack(object)
- class CasPack(object)
- class PsPack(object)

FileStack Class Reference

Public Methods

- def __init__(self,max_depth = 10)
- def push(self,pack)
- def pop(self)
- def is_stack_full(self)
- def is_stack_empty(self)
- def fileStack(self)
- def len(self)
- def currentFile(self)

Private Methods

- _max_depth
- _file_stack
- _current_depth

Detailed Description

FileStack is used to store the files for the recovery action.
The maximum number of stored files could be set. If the stack is full, the oldest file will be discard to store the lastest file.

Method Documentation

_max_depth

Store the maximun number of stored files.

_file_stack

Store the list of stored file.

_current_depth

Store the current number of stored files.

def __init__(self,max_depth = 10)

Create a new FileStack object and initialize the file list for storing files.
The default maximun number of stored files is 10.

def push(self,pack)

Push a file pack into the **_file_stack**.
If the stack is full, the oldest file will be discarded to store the lastest file.

def pop(self)

Pop out the lastest file.
If stack is empty, return 'None'.

def is_stack_full(self)

Return **True** if the stack is full, else return **False**.

def is_stack_empty(self)

Return **True** if the stack is empty, else return **False**.

def fileStack(self)

Property member, provide a interface to access **_file_stack**.

def len(self)

Property member, return the current length of **_file_stack**.

def currentFile(self)

Property member, provide a interface to access the lastest file in stack.

CasPack Class Reference

Public Methods

- def __init__(self,action = None,fileName = None)
- def action(self)
- def file_name(self)

Private Methods

- _action
- _file_name

Detailed Description

CasPack is a collection that contains all the nessecery informations of a CAS file.

Method Documentation

_action

Store the description of last action applied on **_file_name**.

_file_name

Store the name of the stored file.

def init(self,action = None,fileName = None)

Create a new CasPack object with the given **action** and **fileName**.

def action(self)

Property member, provide a interface to access the **_action**.

def file_name(self)

Property member, provide a interface to access the **_file_name**.

PsPack Class Reference

Public Methods

- def __init__(self,action = None,fileName = None)
- def action(self)
- def file_name(self)

Private Methods

- _action
- _file_name

Detailed Description

PsPack is a collection that contains all the nessecery informations of a PS file.

Method Documentation

_action

Store the description of last action applied on **_file_name**.

_file_name

Store the name of the stored file.

def init(self,action = None,fileName = None)

Create a new PsPack object with the given **action** and **fileName**.

def action(self)

Property member, provide a interface to access the **_action**.

def file_name(self)

Property member, provide a interface to access the **_file_name**.

MainController Package

There are four classes in MainController package.

- class MainControllerUI(QObject)
- class MainControllerUILoop(QThread)
- class MainController(object)
- class QComparisonItem(QStandardItem)

MainControllerUI Class Reference

MainControllerUI is used to hold the instance of the GUI and handle all the actions triggered by user. Inherits **QObject**.

Public Methods

- def __init__(self,queue_wr=None,queue_rd=None)
- def run(self)
- def __del__(self)
- def init_GUI(self)
- def init_worker(self)
- def bind_GUI_event(self)
- def open_cas(self)
- def open_ps(self)
- def save_cas(self)
- def save_ps(self)
- def saveas_cas(self)
- def saveas_ps(self)
- def select_cas_sheet(self,index)
- def select_ps_sheet(self,index)
- def select_preview(self,index)
- def select_sync_ps_to_cas(self)
- def select_sync_cas_to_ps(self)
- def select_sync_select_all_ps_headers(self,state)

- def select_sync_select_all_cas_headers(self,state)
- def comparison_start(self)
- def comparison_delete(self)
- def comparison_append(self)
- def comparison_select_all_delete(self,state)
- def comparison_select_all_append(self,state)
- def preview_add(self)
- def preview_delete(self)
- def preview_lock(self)
- def undo_cas(self)
- def undo_ps(self)
- def select_extended_preview(self)
- def ps_header_changed(self,index)
- def cas_header_changed(self,index)
- def comparison_append_list_changed(self,index)
- def comparison_delete_list_changed(self,index)
- def set_CASbook_modified(self,state)
- def set_PSbook_modified(self,state)
- def animation_progressBar(self,model)
- def bind_worker_event(self,worker)
- def show_GUI(self)

Private Methods

- _application
- _progressBar_status
- _CASbook_modified
- _PSbook_modified
- _queue_wr
- _queue_rd
- _status

Detailed Description

Before a MainControllerUI object start to deal with the actions triggered by user, it spawned a child thread to receive messages from **_queue_rd**.

Method Documentation

_application

Store the Qt application object.

_progressBar_status

Store the status of progress bar.

_CASbook_modified

It is used to indicate the status of CAS file.
If CAS file has been modified, the value should be **True**, otherwise it should be **False**.

_PSbook_modified

It is used to indicate the status of PS file.
If PS file has been modified, the value should be **True**, otherwise it should be **False**.

_queue_wr

To communicate with the background data handler process. Write only.

_queue_rd

To communicate with the background data handler process. Read only.

_status

Indicate the status of this process.
If the object is runing, the value should be **True**, otherwise it should be **False**.

def __init__(self,queue_wr=None,queue_rd=None)

Create a new MainControllerUI object with the given **queue_wr** and **queue_rd**.
Initialize all the status.

def run(self)

Start the graphic user interface.

def __del__(self)

Safety exit the child thread.

def init_GUI(self)

Create application object, window object and extended preview window object.
Mount GUI event to it own functions.

def init_worker(self)

Start a child thread for handling messages from data handler process.

def bind_GUI_event(self)

Mount GUI event to it own functions.

def open_cas(self)

Open a file selection dialog for open cas actions.

def open_ps(self)

Open a file selection dialog for open ps actions.

def save_cas(self)

Send 'save_cas' to **_queue_wr**.

def save_ps(self)

Send 'save_ps' to **_queue_wr**.

def saveas_cas(self)

Open a file selection dialog for saveas cas actions and send 'saveas_cas' to **_queue_wr**.

def saveas_ps(self)

Open a file selection dialog for saveas ps actions and send 'saveas_ps' to **_queue_wr**.

def select_cas_sheet(self,index)

Send 'select_cas_sheet' and **index** to **_queue_wr**.

def select_ps_sheet(self,index)

Send 'select_ps_sheet' and **index** to **_queue_wr**.

def select_preview(self,index)

Send 'select_preview' and **index** to **_queue_wr**.

def select_sync_ps_to_cas(self)

Send 'select_sync_ps_to_cas' to **_queue_wr**.

def select_sync_cas_to_ps(self)

Send 'select_sync_cas_to_ps' to **_queue_wr**.

def select_sync_select_all_ps_headers(self,state)

Send 'select_sync_select_all_ps_headers' and **state** to **_queue_wr**.

def select_sync_select_all_cas_headers(self,state)

Send 'select_sync_select_all_cas_headers' and **state** to **_queue_wr**.

def comparison_start(self)

Send 'comparison_start' to **_queue_wr**.

def comparison_delete(self)

Send 'comparison_delete' to **_queue_wr**.

def comparison_append(self)

Send 'comparison_append' to **_queue_wr**.

def comparison_select_all_delete(self,state)

Send 'comparison_select_all_delete' and **state** to **_queue_wr**.

def comparison_select_all_append(self,state)

Send 'comparison_select_all_append' and **state** to **_queue_wr**.

def preview_add(self)

Send 'preview_add' to **_queue_wr**.

def preview_delete(self)

Send 'preview_delete' to **_queue_wr**.

def preview_lock(self)

Send 'preview_lock' to **_queue_wr**.

def undo_cas(self)

Send 'undo_cas' to **_queue_wr**.

def undo_ps(self)

Send 'undo_ps' to **_queue_wr**.

def select_extended_preview(self)

Send 'select_extended_preview' to **_queue_wr**.

def ps_header_changed(self,index)

Send 'ps_header_changed' and **index** to **_queue_wr**.

def cas_header_changed(self,index)

Send 'cas_header_changed' and **index** to **_queue_wr**.

def comparison_append_list_changed(self,index)

Send 'comparison_append_list_changed' and **index** to **_queue_wr**.

def comparison_delete_list_changed(self,index)

Send 'comparison_delete_list_changed' and **index** to **_queue_wr**.

def set_CASbook_modified(self,state)

Set **_CASbook_modified** to the given **state**.

def set_PSbook_modified(self,state)

Set **_PSbook_modified** to the given **state**.

def animation_progressBar(self,model)

Refresh progress bar with the given **model**.

def bind_worker_event(self,worker)

Bind child thread's signals to the slots of the main thread.
Please refer to the Qt official documentation.

def show_GUI(self)

Show GUI on screen.

MainControllerUILoop Class Reference

MainControllerUILoop acts as a worker that handles messages for MainControllerUI. Inherits **QThread**.

Public Methods

- def __init__(self,queue_rd=None,parent=None)
- def run(self)
- def stop(self)

Private Methods

- _status
- _queue_rd
- signal_refresh_cas_book_name
- signal_refresh_ps_book_name
- signal_refresh_cas_sheet_name
- signal_refresh_ps_sheet_name
- signal_refresh_preview
- signal_refresh_ps_header
- signal_refresh_cas_header
- signal_refresh_comparison_delete_list
- signal_refresh_comparison_append_list
- signal_refresh_message
- signal_refresh_msg
- signal_refresh_warning

- signal_refresh_selected_cell
- signal_refresh_progressBar
- signal_animation_progressBar
- signal_refresh_ps_header_selected
- signal_refresh_cas_header_selected
- signal_refresh_extended_preview
- signal_set_CASbook_modified
- signal_set_PSbook_modified

Detailed Description

The main task of **MainControllerUILoop** is to receive messages from **MainController** and forward the message by signal-slot connection to **MainControllerUI**.

Method Documentation

_status

Indicate the status of this process.
If the object is runing, the value should be **True**, otherwise it should be **False**.

_queue_rd

To communicate with the background data handler process. Read only.

signal_refresh_cas_book_name

Qt signal, to trigger **refresh_cas_book_name** in **MainControllerUI**.

signal_refresh_ps_book_name

Qt signal, to trigger **refresh_ps_book_name** in **MainControllerUI**.

signal_refresh_cas_sheet_name

Qt signal, to trigger **refresh_cas_sheet_name** in **MainControllerUI**.

signal_refresh_ps_sheet_name

Qt signal, to trigger **refresh_ps_sheet_name** in **MainControllerUI**.

signal_refresh_preview

Qt signal, to trigger **refresh_preview** in **MainControllerUI**.

signal_refresh_ps_header

Qt signal, to trigger **refresh_ps_header** in **MainControllerUI**.

signal_refresh_cas_header

Qt signal, to trigger **refresh_cas_header** in **MainControllerUI**.

signal_refresh_comparison_delete_list

Qt signal, to trigger **refresh_comparison_delete_list** in **MainControllerUI**.

signal_refresh_comparison_append_list

Qt signal, to trigger **refresh_comparison_append_list** in **MainControllerUI**.

signal_refresh_message

Qt signal, to trigger **refresh_message** in **MainControllerUI**.

signal_refresh_msg

Qt signal, to trigger **refresh_msg** in **MainControllerUI**.

signal_refresh_warning

Qt signal, to trigger **refresh_warning** in **MainControllerUI**.

signal_refresh_selected_cell

Qt signal, to trigger **refresh_selected_cell** in **MainControllerUI**.

signal_refresh_progressBar

Qt signal, to trigger **refresh_progressBar** in **MainControllerUI**.

signal_animation_progressBar

Qt signal, to trigger **animation_progressBar** in **MainControllerUI**.

signal_refresh_ps_header_selected

Qt signal, to trigger **refresh_ps_header_selected** in **MainControllerUI**.

signal_refresh_cas_header_selected

Qt signal, to trigger **refresh_cas_header_selected** in **MainControllerUI**.

signal_refresh_extended_preview

Qt signal, to trigger **refresh_extended_preview** in **MainControllerUI**.

signal_set_CASbook_modified

Qt signal, to trigger **set_CASbook_modified** in **MainControllerUI**.

signal_set_PSbook_modified

Qt signal, to trigger **set_PSbook_modified** in **MainControllerUI**.

def init(self,queue_rd=None,parent=None)

Create a new **MainControllerUILoop** object with the given **queue_rd** and **parent**.

def run(self)

Keep receiving messages from **_queue_rd**.

def stop(self)

Stop the main loop.

MainController Class Reference

MainController is mainly responsible for the background operations on excel file.

Public Methods

- def __init__(self, queue_wr=None, queue_rd=None)
- def run(self)
- def stop(self)
- def __del__(self)
- def init_logging(self)
- def init_tmp_directory(self)

- def init_model(self)
- def init_file_stack(self)
- def start_xlwings_app(self)
- def open_cas_by_name(self,filename)
- def open_cas_by_bytesio(self,bytesio)
- def open_ps_by_name(self,filename)
- def open_ps_by_bytesio(self,bytesio)
- def save_cas(self)
- def save_ps(self)
- def saveas_cas(self,fileName)
- def saveas_ps(self,fileName)
- def select_cas_sheet(self,sheet_idx)
- def select_ps_sheet(self,sheet_idx)
- def select_preview(self,row,column)
- def select_sync_select_all_ps_headers(self,state)
- def select_sync_select_all_cas_headers(self,state)
- def select_sync_ps_to_cas(self)
- def select_sync_cas_to_ps(self)
- def comparison_start(self)
- def comparison_delete(self)
- def comparison_append(self)
- def comparison_select_all_delete(self,state)
- def comparison_select_all_append(self,state)
- def checked_delete(self)
- def checked_delete_count(self)
- def checked_append(self)
- def checked_append_count(self)
- def preview_add(self)
- def preview_delete(self)
- def preview_lock(self)
- def ps_header_changed(self,row,state)
- def cas_header_changed(self,row,state)
- def comparison_append_list_changed(self,row,state)
- def comparison_delete_list_changed(self,row,state)
- def recover_ps_sheet_selected(self)
- def recover_cas_sheet_selected(self)
- def store_ps_file(self,action)
- def store_ps_file_without_open(self,action)
- def store_cas_file(self,action)
- def store_cas_file_without_open(self,action)
- def copy_cas(self,filename)
- def copy_ps(self,filename)
- def undo_ps(self)
- def undo_cas(self)
- def select_extended_preview(self)
- def CASbook_modified(self)
- def CASbook_modified(self,value)
- def PSbook_modified(self)
- def PSbook_modified(self,value)
- def refresh_cas_book_name(self,model)
- def refresh_ps_book_name(self,model)
- def refresh_cas_sheet_name(self,model)
- def refresh_ps_sheet_name(self,model)
- def refresh_preview(self,model)

- def refresh_ps_header(self,model)
- def refresh_cas_header(self,model)
- def refresh_comparison_delete_list(self,model)
- def refresh_comparison_append_list(self,model)
- def refresh_msg(self,model)
- def refresh_warning(self,model)
- def refresh_selected_cell(self,model)
- def refresh_progressBar(self,model)
- def animation_progressBar(self,model)
- def refresh_ps_header_selected(self,model)
- def refresh_cas_header_selected(self,model)
- def refresh_extended_preview(self,model)

Private Methods

- _status
- _queue_wr
- _queue_rd
- _xw_app
- _xw_app_2
- _PSbook
- _PSbook_name
- _PSbook_sheets
- _PSbook_current_sheet
- _PSbook_current_sheet_idx
- _PSbook_current_sheet_name
- _PSbook_autosave_flag
- _PSbook_modified
- _CASbook
- _CASbook_name
- _CASbook_sheets
- _CASbook_current_sheet
- _CASbook_current_sheet_idx
- _CASbook_current_sheet_name
- _CASbook_autosave_flag
- _CASbook_modified
- _PSstack
- _CASstack
- _progressBar_status

Detailed Description

Method Documentation

_status

Indicate the status of this process.
If the object is runing, the value should be **True**, otherwise it should be **False**.

_queue_wr

To communicate with the GUI process. Write only.

_queue_rd

To communicate with the GUI process. Read only.

_xw_app

Excel app 1, created by xlwings.

_xw_app_2

Excel app 2, created by xlwings.

_PSbook

Store the **PSbook** object.

_PSbook_name

Store the name of **PSbook** object.

_PSbook_sheets

Store the sheets of **PSbook** object.

_PSbook_current_sheet

Store the current sheet of **PSbook** object.

_PSbook_current_sheet_idx

Store the index of the current sheet of **PSbook** object.

_PSbook_current_sheet_name

Store the name of the current sheet of **PSbook** object.

_PSbook_autosave_flag

Indicate if the next step should be autosave.

_PSbook_modified

Indicate if the **PSbook** has been modified.

_CASbook

Store the **CASbook** object.

_CASbook_name

Store the name of **CASbook** object.

_CASbook_sheets

Store the sheets of **CASbook** object.

_CASbook_current_sheet

Store the current sheet of **CASbook** object.

_CASbook_current_sheet_idx

Store the index of the current sheet of **CASbook** object.

_CASbook_current_sheet_name

Store the name of the current sheet of **CASbook** object.

_CASbook_autosave_flag

Indicate if the next step should be autosave.

_CASbook_modified

Indicate if the **CASbook** has been modified.

_PSstack

Store the **FileStack** object for the PS file.

_CASstack

Store the **FileStack** object for the CAS file.

_progressBar_status

Store the status of progress bar.

def __init__(self,queue_wr=None,queue_rd=None)

Create a new **MainController** with the given **queue_wr** and **queue_rd**.

def run(self)

Initialize all the parameter.
Initialize the logging system.
Initialize the recovery system.
Start xlwings app.
Initialize the data model.
Run the main loop.

def stop(self)

Stop the main loop.

def __del__(self)

Release PS workbook object and CAS workbook object.
Quit xlwings applications.
Remove the temporary directory.

def init_logging(self)

Initialize the logging system.

def init_tmp_directory(self)

Create temporary directory.

def init_model(self)

Initialize the data model for comparison.

def init_file_stack(self)

Initialize the FileStack object.

def start_xlwings_app(self)

Start xlwings application.

def open_cas_by_name(self,filename)

Open cas file by the given **filename**.
Initialize the data model and keep the content update.

def open_cas_by_bytesio(self,bytesio)

An optional way to open a cas file.

def open_ps_by_name(self,filename)

Open ps file by the given **filename**.
Initialize the data model and keep the content update.

def open_ps_by_bytesio(self,bytesio)

An optional way to open a ps file.

def save_cas(self)

Save cas file.

def save_ps(self)

Save ps file.

def saveas_cas(self,fileName)

Save cas file with the given **fileName**.

def saveas_ps(self,fileName)

Save ps file with the given **fileName**.

def select_cas_sheet(self,sheet_idx)

Switch current worksheet with the given **sheet_idx**.

def select_ps_sheet(self,sheet_idx)

Switch current worksheet with the given **sheet_idx**.

def select_preview(self,row,column)

Update the **_preview_selected_cell** with the given **row** and **column**.

def select_sync_select_all_ps_headers(self,state)

Update the data model status of ps current sheet with the given **state**.

def select_sync_select_all_cas_headers(self,state)

Update the data model status of cas current sheet with the given **state**.

def select_sync_ps_to_cas(self)

Sync the specified columns from ps sheet to cas sheet.

def select_sync_cas_to_ps(self)

Sync the specified columns from cas sheet to ps sheet.

def comparison_start(self)

Compare the xmlnames in cas sheet and ps sheet. Send messages to show difference on GUI.

def comparison_delete(self)

Delete the checked xmlnames in ps sheet.

def comparison_append(self)

Append the checked xmlnames below the **_preview_selected_cell** in ps sheet.

def comparison_select_all_delete(self,state)

Update the checked state of comparison delete list with the given **state**.

def comparison_select_all_append(self,state)

Update the checked state of comparison append list with the given **state**.

def checked_delete(self)

Return a list of the checked items in comparison delete list.

def checked_delete_count(self)

Return the number of the checked items in comparison delete list.

def checked_append(self)

Return a list of the checked items in comparison append list.

def checked_append_count(self)

Return the number of the checked items in comparison append list.

def preview_add(self)

Append a blank row below the **_preview_selected_cell**.

def preview_delete(self)

Delete the whole row of the **_preview_selected_cell**.

def preview_lock(self)

Lock the rows whose 'status' equals 'POR'.

def ps_header_changed(self,row,state)

Keep ps header data model and the data hold by GUI in sync.

def cas_header_changed(self,row,state)

Keep cas header data model and the data hold by GUI in sync.

def comparison_append_list_changed(self,row,state)

Keep the data model of comparison append item list and the data hold by GUI in sync.

def comparison_delete_list_changed(self,row,state)

Keep the data model of comparison delete item list and the data hold by GUI in sync.

def recover_ps_sheet_selected(self)

Recover the current sheet of ps file to the last status in order to keep the GUI showing continuously.

def recover_cas_sheet_selected(self)

Recover the current sheet of cas file to the last status in order to keep the GUI showing continuously.

def store_ps_file(self,action)

Store the ps file for the file recovery and reopen it to keep content update.

def store_ps_file_without_open(self,action)

Store the ps file for the file recovery.

def store_cas_file(self,action)

Store the cas file for the file recovery and reopen it to keep content update.

def store_cas_file_without_open(self,action)

Store the cas file for the file recovery.

def copy_cas(self,filename)

Copy cas file right after you select the cas file in GUI.
Divide the modifications and the source file to avoid misoperations.

def copy_ps(self,filename)

Copy ps file right after you select the ps file in GUI.
Divide the modifications and the source file to avoid misoperations.

def undo_ps(self)

Revert the last action applied on ps file.

def undo_cas(self)

Revert the last action applied on cas file.

def select_extended_preview(self)

Open the extended preview window.

def CASbook_modified(self)

Property member, provide a interface to accesss the **_CASbook_modified**.

def CASbook_modified(self,value)

Property member, provide a interface to write the **_CASbook_modified**.

def PSbook_modified(self)

Property member, provide a interface to accesss the **_PSbook_modified**.

def PSbook_modified(self,value)

Property member, provide a interface to write the **_PSbook_modified**.

def refresh_cas_book_name(self,model)

Send 'refresh_cas_book_name' and the given **model** to **_queue_wr**.

def refresh_ps_book_name(self,model)

Send 'refresh_ps_book_name' and the given **model** to **_queue_wr**.

def refresh_cas_sheet_name(self,model)

Send 'refresh_cas_sheet_name' and the given **model** to **_queue_wr**.

def refresh_ps_sheet_name(self,model)

Send 'refresh_ps_sheet_name' and the given **model** to **_queue_wr**.

def refresh_preview(self,model)

Send 'refresh_preview' and the given **model** to **_queue_wr**.

def refresh_ps_header(self,model)

Send 'refresh_ps_header' and the given **model** to **_queue_wr**.

def refresh_cas_header(self,model)

Send 'refresh_cas_header' and the given **model** to **_queue_wr**.

def refresh_comparison_delete_list(self,model)

Send 'refresh_comparison_delete_list' and the given **model** to **_queue_wr**.

def refresh_comparison_append_list(self,model)

Send 'refresh_comparison_append_list' and the given **model** to **_queue_wr**.

def refresh_msg(self,model)

Send 'refresh_msg' and the given **model** to **_queue_wr**.
Record the given **model** into logging file.

def refresh_warning(self,model)

Send 'refresh_warning' and the given **model** to **_queue_wr**.

def refresh_selected_cell(self,model)

Send 'refresh_selected_cell' and the given **model** to **_queue_wr**.

def refresh_progressBar(self,model)

Send 'refresh_progressBar' and the given **model** to **_queue_wr**.

def animation_progressBar(self,model)

Send 'animation_progressBar' and the given **model** to **_queue_wr**.

def refresh_ps_header_selected(self,model)

Send 'refresh_ps_header_selected' and the given **model** to **_queue_wr**.

def refresh_cas_header_selected(self,model)

Send 'refresh_cas_header_selected' and the given **model** to **_queue_wr**.

def refresh_extended_preview(self,model)

Send 'refresh_extended_preview' and the given **model** to **_queue_wr**.

QComparisonItem Class Reference

Inherits **QStandardItem**.

Public Methods

- def __init__(self,cell)
- def cell(self)
- def value(self)
- def col_letter(self)

Private Methods

- _cell

Detailed Description

To store the items of the result of comparison.

Method Documentation

_cell

Store the cell object.

def __init__(self,cell)

Create a new **QComparisonItem** with the given **cell**.

def cell(self)

Property member, provide a interface to access the **_cell**.

def value(self)

Property member, provide a interface to access the **_cell.value**.

def col_letter(self)

Property member, provide a interface to access the **_cell.col_letter**.