

## CSV\_XML\_Importer

Generated by Doxygen 1.9.1



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 reader_support_file.dataframeAndHeaderHandler Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Constructor & Destructor Documentation	5
3.1.2.1 __init__()	5
3.1.3 Member Function Documentation	6
3.1.3.1 find_header_formats()	6
3.1.3.2 ImportFile()	6
3.1.3.3 regex_list_filler()	7
3.1.3.4 regex_tester()	8
3.1.3.5 reset()	8
3.2 csv_xml_importer_gui.gui Class Reference	9
3.2.1 Detailed Description	11
3.2.2 Constructor & Destructor Documentation	11
3.2.2.1 __init__()	11
3.2.3 Member Function Documentation	12
3.2.3.1 ClearAllFilesGUI()	13
3.2.3.2 csvReset()	13
3.2.3.3 ExportAs()	14
3.2.3.4 finalCSVExport()	15
3.2.3.5 finalImporter()	16
3.2.3.6 finalXMLExport()	17
3.2.3.7 ImportAs()	17
3.2.3.8 listboxSelectionChanged()	18
3.2.3.9 OpenFileGUI()	20
3.2.3.10 OpenXSLFile()	20
3.2.3.11 RemoveFileGUI()	21
3.2.3.12 setFileDelimiter()	22
3.2.3.13 setFileEncoding()	23
3.2.3.14 setFileHeader()	23
3.2.3.15 setFileLineTerminator()	24
3.2.3.16 setFileQuotechar()	25
3.2.3.17 setFileQuoting()	26
3.2.3.18 setFileSkipSpaces()	26
3.2.3.19 setXMLHeader()	27
3.2.3.20 setXMLParameter()	28
3.2.3.21 showXMLParameter()	28

3.2.3.22 updatePreview()	29
3.2.3.23 xmlReset()	30
3.3 reader.reader Class Reference	31
3.3.1 Detailed Description	32
3.3.2 Constructor & Destructor Documentation	32
3.3.2.1 __init__()	32
3.3.3 Member Function Documentation	32
3.3.3.1 addToFilesDict()	33
3.3.3.2 addXMLParameter()	33
3.3.3.3 ClearAllFilesFunctionality()	34
3.3.3.4 csvSniffer()	34
3.3.3.5 exportAsCSVFile()	35
3.3.3.6 exportAsXMLFile()	35
3.3.3.7 getXMLParameters()	36
3.3.3.8 giveDataframe()	36
3.3.3.9 importAsDictionary()	37
3.3.3.10 importAsListOfLists()	37
3.3.3.11 importAsNumPyArray()	37
3.3.3.12 importAsPandasDataframe()	37
3.3.3.13 OpenCSVFile()	38
3.3.3.14 OpenXMLFile()	38
3.3.3.15 read_with_init_settings()	39
3.3.3.16 RemoveFilesFunctionality()	40
3.3.3.17 reset()	40
3.3.3.18 update_csv_with_personal_settings()	41
3.3.3.19 update_dataframe()	41
3.4 csv_xml_importer_gui.reader_and_gui_interface Class Reference	43
3.4.1 Detailed Description	44
3.4.2 Constructor & Destructor Documentation	44
3.4.2.1 __init__()	44
3.4.3 Member Function Documentation	44
3.4.3.1 changeXMLParameter()	44
3.4.3.2 chooseXMLParameter()	45
3.4.3.3 ClearAllFilesInterface()	45
3.4.3.4 csvReset()	46
3.4.3.5 finalCSVExportFunctionality()	47
3.4.3.6 finalImporterFunctionality()	47
3.4.3.7 finalXMLExportFunctionality()	48
3.4.3.8 getDataframe()	48
3.4.3.9 getXSLFile()	49
3.4.3.10 RemoveFilesInterface()	50
3.4.3.11 setUserDelimiter()	51

---

3.4.3.12 setUserEncoding()	51
3.4.3.13 setUserHeader()	52
3.4.3.14 setUserLineTerminator()	53
3.4.3.15 setUserQuotechar()	54
3.4.3.16 setUserQuoting()	55
3.4.3.17 setUserSkipSpaces()	55
3.4.3.18 setXMLUserHeader()	56
3.4.3.19 ShowFilesInterface()	57
3.4.3.20 showXMLParameterFunctionality()	57
3.4.3.21 updateDelimiterTextbox()	58
3.4.3.22 updateEncodingTextbox()	58
3.4.3.23 updateHeaderCheckbox()	59
3.4.3.24 updateLineTerminatorTextbox()	60
3.4.3.25 updateQuotecharTextbox()	60
3.4.3.26 updateQuotingRadioButtons()	61
3.4.3.27 updateSkipSpacesCheckbox()	61
3.4.3.28 updateXMLUserHeader()	62
3.4.3.29 updateXSLFileTextbox()	62
3.4.3.30 xmlResetFunctionality()	63
<b>Index</b>	<b>65</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

reader_support_file.dataframeAndHeaderHandler . . . . .	5
reader.reader . . . . .	31
csv_xml_importer_gui.reader_and_gui_interface . . . . .	43
csv_xml_importer_gui.gui . . . . .	9





## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">reader_support_file.dataframeAndHeaderHandler</a> . . . . .	5
<a href="#">csv_xml_importer_gui.gui</a> . . . . .	9
<a href="#">reader.reader</a> . . . . .	31
<a href="#">csv_xml_importer_gui.reader_and_gui_interface</a> . . . . .	43



## Chapter 3

# Class Documentation

### 3.1 reader\_support\_file.dataframeAndHeaderHandler Class Reference

#### Public Member Functions

- def `__init__` (self)
- def `ImportFile` (self, pd.DataFrame new\_dataframe, int column\_amount, bool hasHeader)
- def `reset` (self)
- def `find_header_formats` (self, pd.DataFrame dataframe)
- def `regex_list_filler` (self, list regex\_list, list row)
- def `regex_tester` (self, str elem)

#### Public Attributes

- `main_dataframe`
- `column_amount`
- `default_header`

#### 3.1.1 Detailed Description

This class is responsible for adding the new Dataframes to the Main Dataframe and finding the corresponding headers of the columns of the Main Dataframe through regular expressions

#### 3.1.2 Constructor & Destructor Documentation

##### 3.1.2.1 `__init__()`

```
def reader_support_file.dataframeAndHeaderHandler.__init__ (
    self )
```

The `__init__` function is the constructor of the model class.  
It defines the following attributes for every model object:

`types_dict`: a dictionary, which contains the regular expression, with whom the Dataframe columns will be tested  
`main_dataframe`: the Dataframe in which all converted csv and xml Files will be saved  
`default_header`: a list, containing the column names of the main dataframe the regular expressions found  
`column_amount`: an integer, which saves the amount of columns the main dataframe has, to test if the new import  
`main_dataframe_has_header`: a boolean variable, signaling if the main dataframe already has a header of its own  
`main_dataframe_has_default_header`: a boolean variable, signaling if the main dataframe has a default header fr

### 3.1.3 Member Function Documentation

#### 3.1.3.1 find\_header\_formats()

```
def reader_support_file.dataframeAndHeaderHandler.find_header_formats (
    self,
    pd.DataFrame dataframe )
```

Takes a dataframe and checks the first five rows with regular expressions on their types

Parameter:

dataframe: the dataframe, of which the types will be evaluated

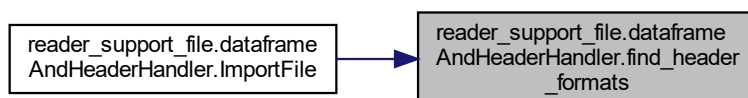
Returns:

key\_list: a list of the types of the columns from the dataframe

Here is the call graph for this function:



Here is the caller graph for this function:



#### 3.1.3.2 ImportFile()

```
def reader_support_file.dataframeAndHeaderHandler.ImportFile (
    self,
    pd.DataFrame new_dataframe,
    int column_amount,
    bool hasHeader )
```

ImportFile adds the latest csv or xml File as a Dataframe, adds it to the main dataframe and tests if the head

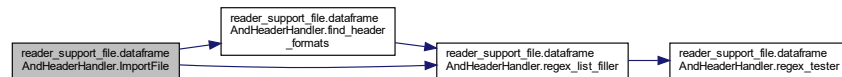
Parameters:

new\_dataframe: the latest csv or xml File converted to a dataframe  
 column\_amount: the amount of columns new\_dataframe has  
 hasHeader: a boolean variable, which shows if new\_dataframe has a header or not

Returns:

main\_dataframe with the latest csv or xml File appended and the latest header

Here is the call graph for this function:



### 3.1.3.3 regex\_list\_filler()

```
def reader_support_file.dataframeAndHeaderHandler.regex_list_filler (
    self,
    list regex_list,
    list row )
```

Tests every element of the row on their type

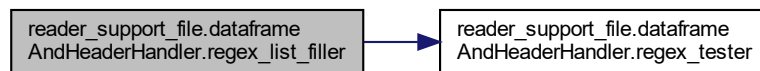
Parameters:

regex\_list: the list, which will contain the types of the rows  
 row: the row of the dataframe whose entries will be tested on regular expressions as a list

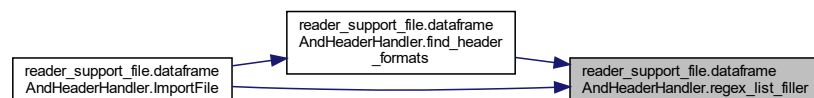
Returns:

regex\_list: the filled list, with the types of the entries of the row

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.1.3.4 regex\_tester()

```
def reader_support_file.dataframeAndHeaderHandler.regex_tester (
    self,
    str elem )
```

This Function tests every element on the regular expressions contained in types\_dict

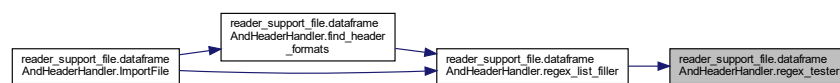
Parameters:

elem: the Element to be tested on every regular expression as a string

Returns:

key or "String": a string, which indicates the type of elem

Here is the caller graph for this function:

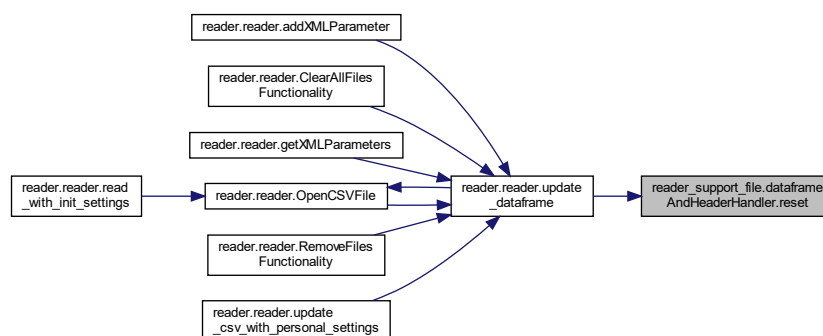


### 3.1.3.5 reset()

```
def reader_support_file.dataframeAndHeaderHandler.reset (
    self )
```

A simple function, which resets the main\_dataframe, the boolean variable main\_dataframe\_has\_header, the column and returns nothing or void

Here is the caller graph for this function:

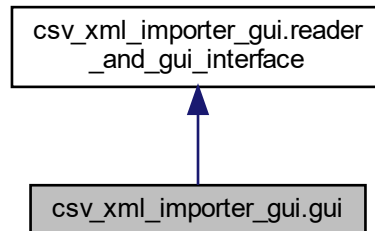


The documentation for this class was generated from the following file:

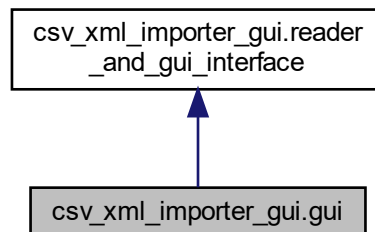
- C:/Users/Philip/Desktop/Studienarbeit\_Python\_Philip\_Bartmann\_62442491/src/reader\_support\_file.py

## 3.2 csv\_xml\_importer\_gui.gui Class Reference

Inheritance diagram for csv\_xml\_importer\_gui.gui:



Collaboration diagram for csv\_xml\_importer\_gui.gui:



### Public Member Functions

- def `__init__` (self)
- def `updatePreview` (self)
- def `OpenFileGUI` (self)
- def `RemoveFileGUI` (self)
- def `ClearAllFilesGUI` (self)
- def `setFileEncoding` (self, return\_event)
- def `setFileDelimiter` (self, return\_event)
- def `setFileQuotechar` (self, return\_event)
- def `setFileHeader` (self)
- def `setFileSkipSpaces` (self)
- def `setFileLineTerminator` (self, return\_event)
- def `setFileQuoting` (self)
- def `csvReset` (self)
- def `OpenXSLFile` (self)

- def [showXMLParameter](#) (self, selection\_event)
- def [setXMLParameter](#) (self, return\_event)
- def [setXMLHeader](#) (self)
- def [xmlReset](#) (self)
- def [listboxSelectionChanged](#) (self, select\_event)
- def [ImportAs](#) (self)
- def [finalImporter](#) (self)
- def [ExportAs](#) (self)
- def [finalCSVExport](#) (self)
- def [finalXMLExport](#) (self)

## Public Attributes

- root
- Importer\_Labelframe
- Configurator\_Labelframe
- preview\_table\_Labelframe
- csv\_configurator\_frame
- listbox
- encodings\_textbox\_label
- encoding\_textbox
- delimiter\_textbox\_label
- delimiter\_textbox
- quotechar\_textbox\_label
- quotechar\_textbox
- header\_var
- header\_checkbox\_label
- header\_checkbox
- skip\_spaces\_var
- skip\_spaces\_checkbox\_label
- skip\_spaces\_checkbox
- line\_terminator\_textbox\_label
- line\_terminator\_textbox
- quoting\_var
- quoting\_radiobuttons\_label
- quote\_minimal\_button
- quote\_all\_button
- quote\_non\_numeric\_button
- quote\_none\_button
- csv\_reset\_button
- xml\_konfigurator\_frame
- xsl\_stylesheet\_textbox\_label
- xsl\_stylesheet\_textbox
- button\_add\_xsl\_File
- valid\_xsl\_file
- xml\_parameters\_listbox\_label
- xml\_parameter\_listbox
- xml\_parameters\_textbox
- xml\_header\_var
- xml\_header\_checkbox\_label
- xml\_header\_checkbox
- xml\_reset\_button
- button\_addFile
- button\_removeFile



- `button_removeAllFiles`
- `preview_table`
- `import_export_buttons_frame`
- `button_import_as`
- `button_export_as`
- `button_exit`
- `filename`
- `import_var`
- `csv_export_encoding`
- `csv_export_delimiter`
- `csv_export_quotechar`
- `csv_export_line_terminator`
- `xml_export_encoding`

### 3.2.1 Detailed Description

This class is responsible for the GUI and to call the methods from the interface class

### 3.2.2 Constructor & Destructor Documentation

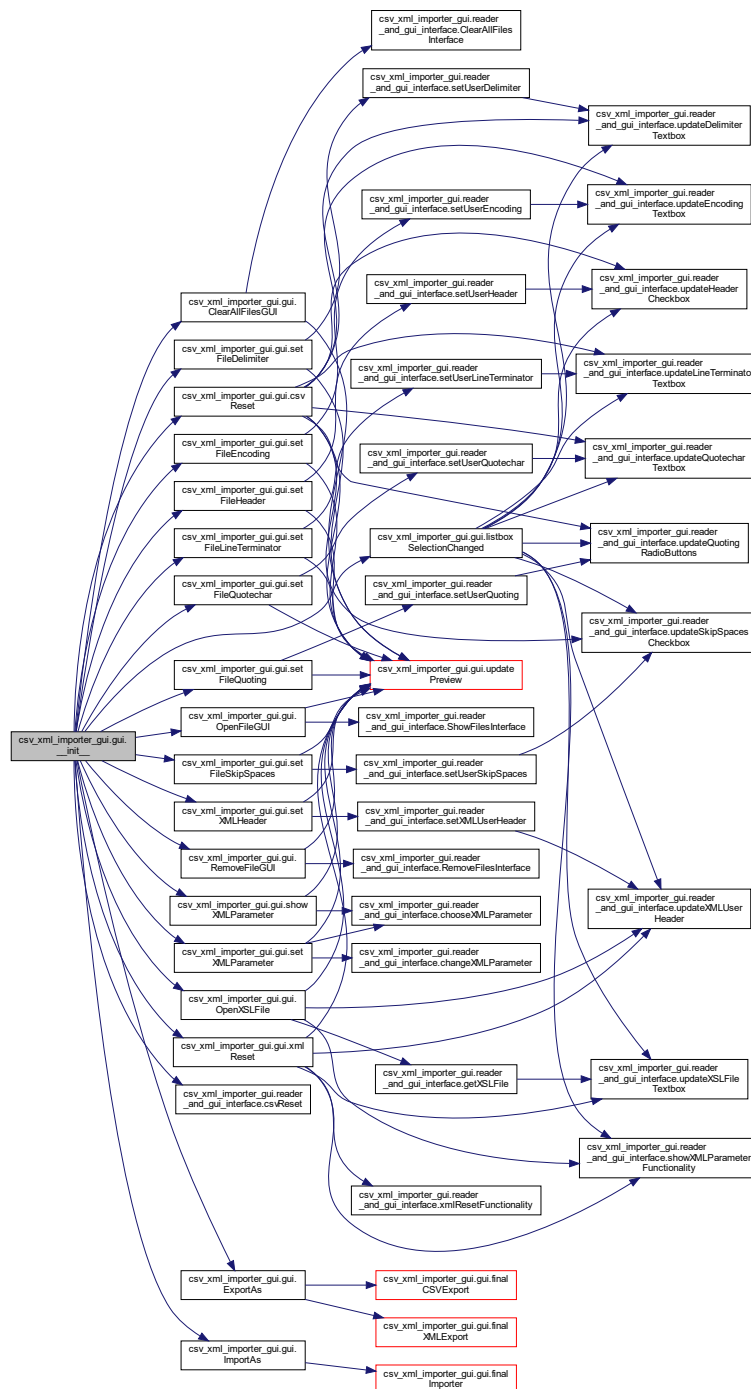
#### 3.2.2.1 `__init__()`

```
def csv_xml_importer_gui.gui.__init__ (
    self )
```

The constructor of the class gui.  
It defines the Elements of the gui and a reader\_and\_gui\_interface

Reimplemented from [csv\\_xml\\_importer\\_gui.reader\\_and\\_gui\\_interface](#).

Here is the call graph for this function:



### 3.2.3 Member Function Documentation

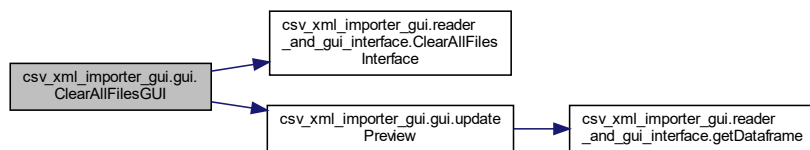
### 3.2.3.1 ClearAllFilesGUI()

```
def csv_xml_importer_gui.gui.ClearAllFilesGUI (
    self )
```

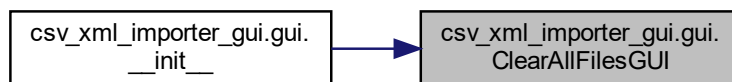
Removes all csv and xml files from the gui and opened\_files\_dict

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



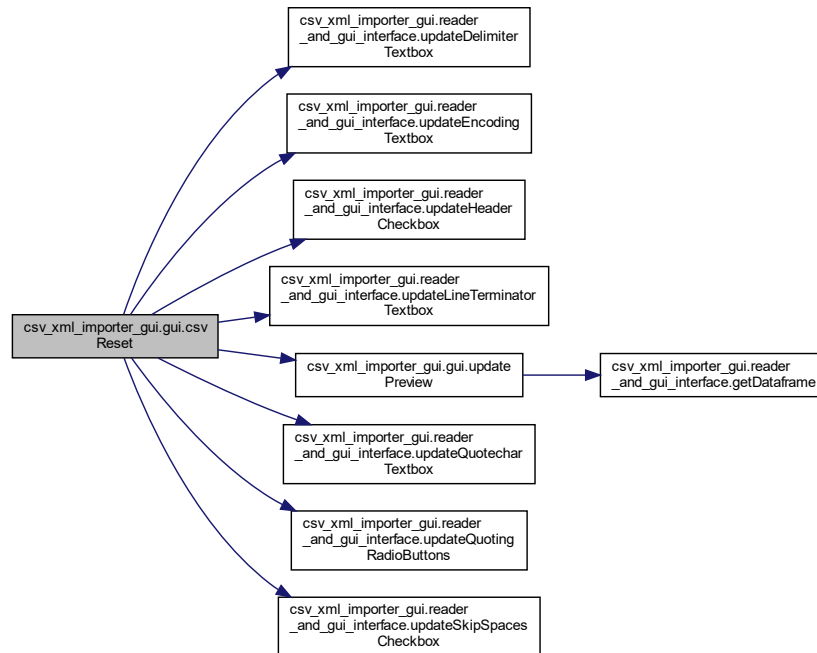
### 3.2.3.2 csvReset()

```
def csv_xml_importer_gui.gui.csvReset (
    self )
```

Resets every csv parameter of the selected file to its default sniffed value

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



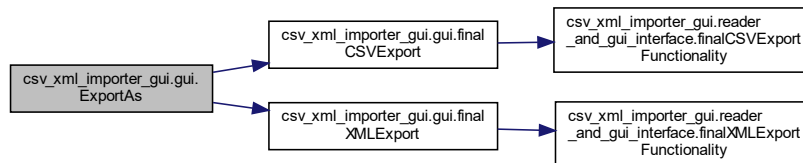
### 3.2.3.3 ExportAs()

```
def csv_xml_importer_gui.gui.ExportAs (
    self )
```

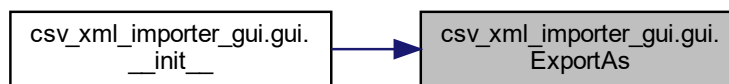
This method is called, if the "Export As..." button is clicked and defines a toplevel window, to let the user decide if he wants to export the dataframe as a csv or xml file and with what parameters

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.2.3.4 finalCSVExport()

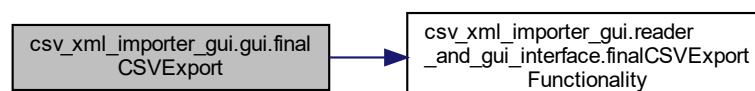
```
def csv_xml_importer_gui.gui.finalCSVExport (
    self )
```

exports the selected files as a csv file to the filepath the user gives, with the given parameters of the top

Returns:

nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



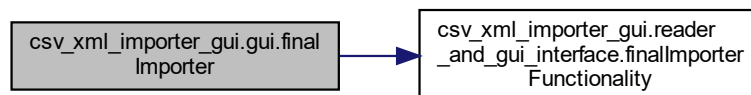
### 3.2.3.5 finalImporter()

```
def csv_xml_importer_gui.gui.finalImporter (
    self )
```

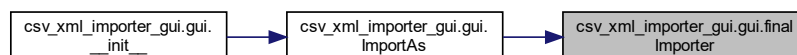
gets the radiobutton value of the "Convert to..." Toplevel window and converts the dataframe to the selected c

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



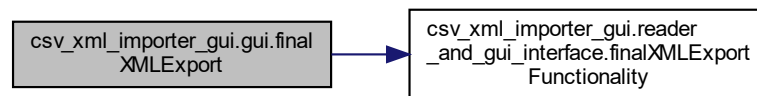
### 3.2.3.6 finalXMLExport()

```
def csv_xml_importer_gui.gui.finalXMLExport (
    self )
```

exports the selected files as a xml file to the filepath the user gives, with the given parameters of the topl

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.2.3.7 ImportAs()

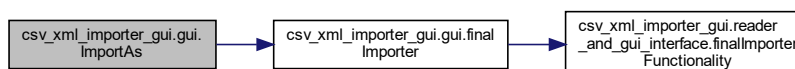
```
def csv_xml_importer_gui.gui.ImportAs (
    self )
```

This function will be called if the "Convert to..." Button is clicked.

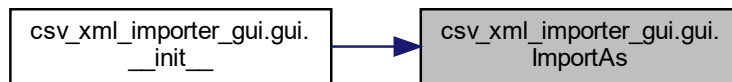
if defines a toplevel window to let the user decide into what data type the dataframe should be converted to. The available data types are: dictionary, list of lists, numpy array and dataframe

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



#### 3.2.3.8 listBoxSelectionChanged()

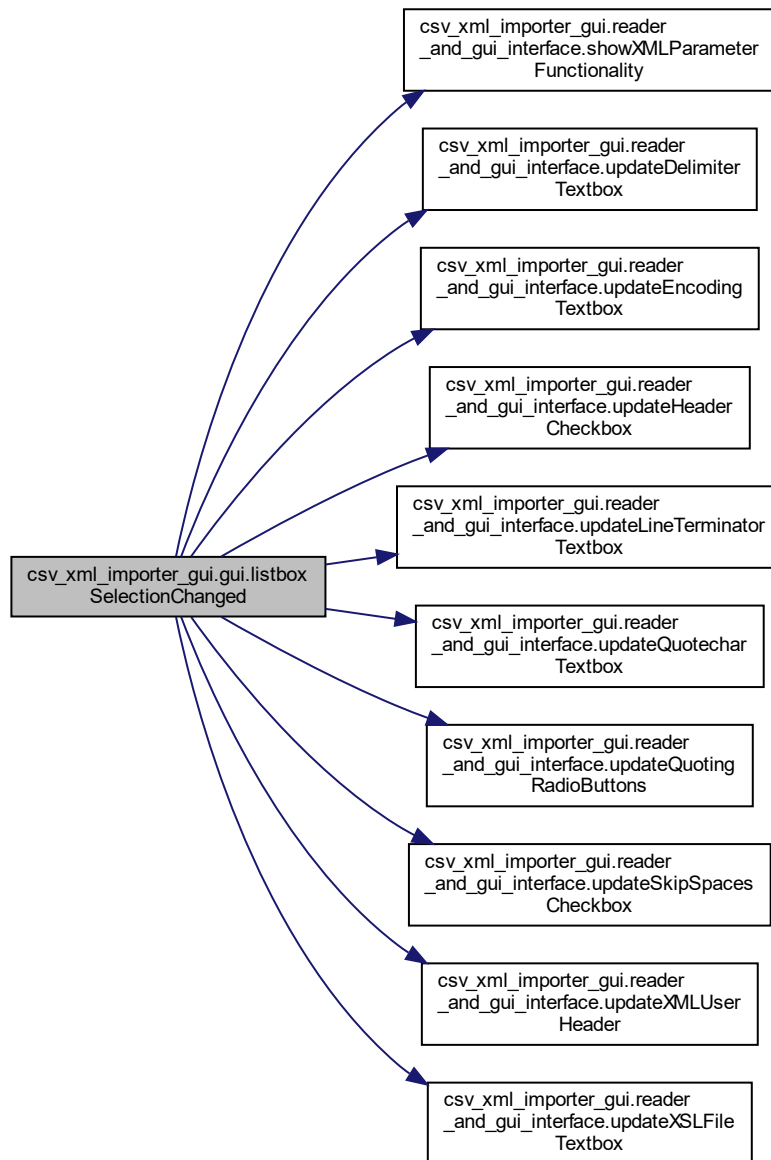
```
def csv_xml_importer_gui.gui.listBoxSelectionChanged (
    self,
    select_event )
```

This method will be called everytime the user selects a entry in the main listbox for the files. It sets the according configurator to the values of its parameters

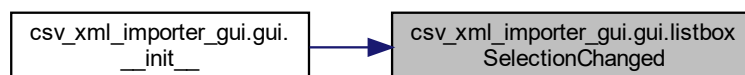
Returns:  
nothing or void



Here is the call graph for this function:



Here is the caller graph for this function:



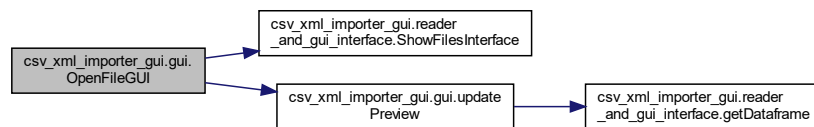
### 3.2.3.9 OpenFileGUI()

```
def csv_xml_importer_gui.gui.OpenFileGUI (
    self )
```

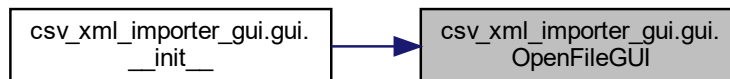
Opens and shows csv and xml files in the gui

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



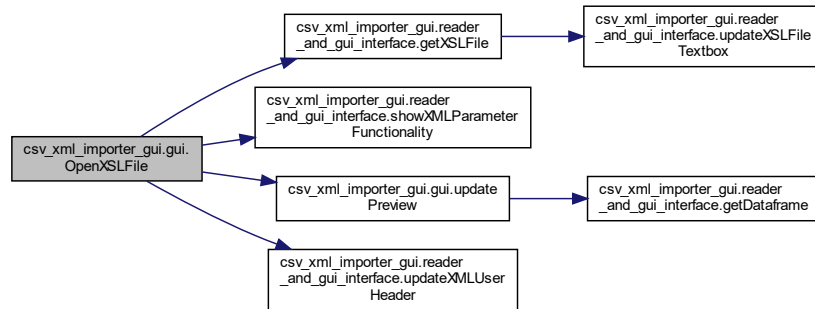
### 3.2.3.10 OpenXSLFile()

```
def csv_xml_importer_gui.gui.OpenXSLFile (
    self )
```

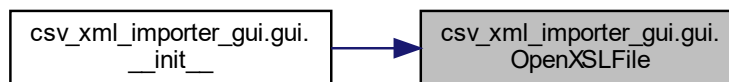
asks the user to choose a xsl Stylesheet. If a valid Stylesheet is chosen, all other elements of the xml Configurator will be made responsive, except for the textbox, where the filepath of the xsl file is stored in, to prevent the user from changing the filepath without the file dialog

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



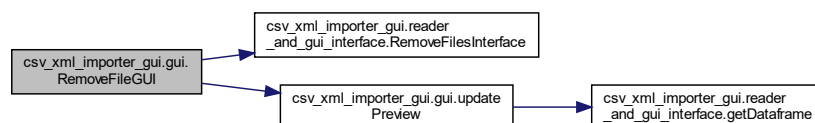
### 3.2.3.11 RemoveFileGUI()

```
def csv_xml_importer_gui.gui.RemoveFileGUI (
    self )
```

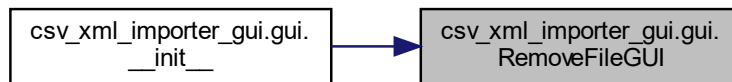
Removes csv and xml files from the gui and opened\_files\_dict

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



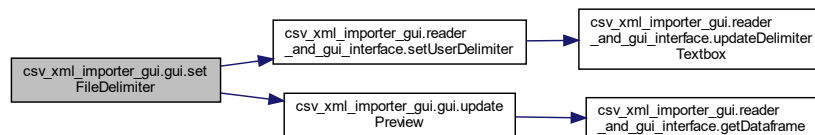
### 3.2.3.12 setFileDelimiter()

```
def csv_xml_importer_gui.gui.setFileDelimiter (
    self,
    return_event )
```

sets the delimiter of the selected csv file in the listbox to the delimiter, which is given in the delimiter textbox. If its not possible to set the delimiter the default delimiter from the file will be taken. The Textbox is confirmed by pressing enter

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



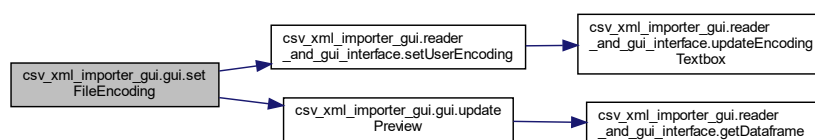
### 3.2.3.13 setFileEncoding()

```
def csv_xml_importer_gui.gui.setFileEncoding (
    self,
    return_event )
```

sets the encoding of the selected csv file in the listbox to the encoding, which is given in the encoding textbox. If its not possible to set the encoding the default encoding from the file will be taken. The Textbox is confirmed by pressing enter

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



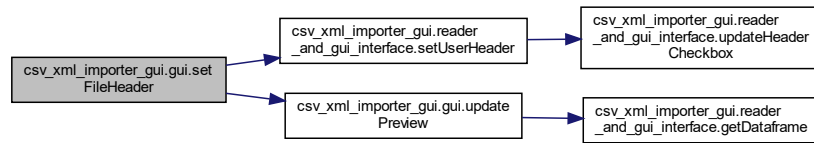
### 3.2.3.14 setFileHeader()

```
def csv_xml_importer_gui.gui.setFileHeader (
    self )
```

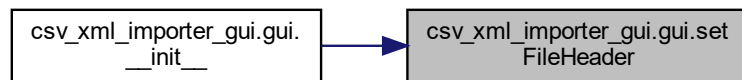
gets the value of the checkbox responsible for the csv header and activates or deactivates the header

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.2.3.15 setFileLineTerminator()

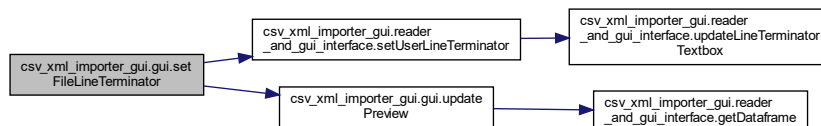
```

def csv_xml_importer_gui.gui.setFileLineTerminator (
    self,
    return_event )
  
```

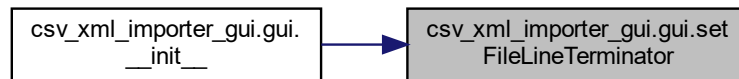
sets the character used for line termination of the selected csv file in the listbox to the character, which is entered in the lineTerminator textbox. If it's not possible to set the character the default line terminator character from the file is used. The Textbox is confirmed by pressing enter

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



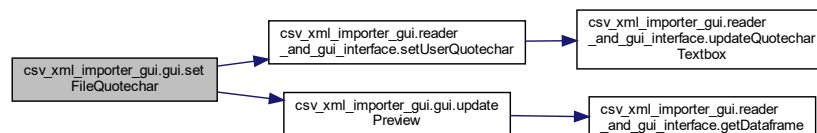
### 3.2.3.16 setFileQuotechar()

```
def csv_xml_importer_gui.gui.setFileQuotechar (
    self,
    return_event )
```

sets the character used for quoting of the selected csv file in the listbox to the character, which is given in quotechar textbox. If its not possible to set the character the default quoting character from the file will be used. The Textbox is confirmed by pressing enter

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



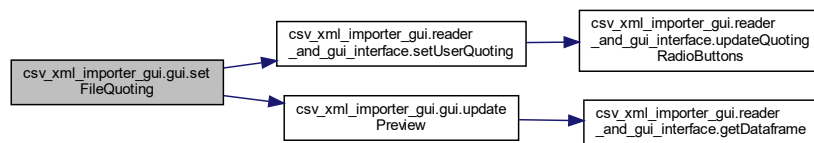
### 3.2.3.17 setFileQuoting()

```
def csv_xml_importer_gui.gui.setFileQuoting (
    self )
```

gets the value of the selected radio button and sets the quoting of the csv files to one of the four possible minimal, all, non numeric and none

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.2.3.18 setFileSkipSpaces()

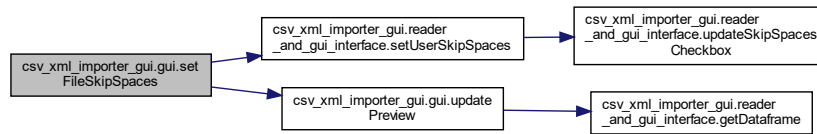
```
def csv_xml_importer_gui.gui.setFileSkipSpaces (
    self )
```

gets the value of the checkbox responsible for skipping initial spaces for csv files and skips or doesn't skip initial spaces

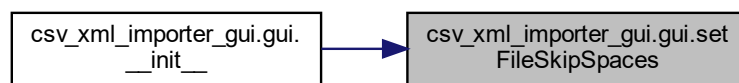
Returns:  
nothing or void



Here is the call graph for this function:



Here is the caller graph for this function:



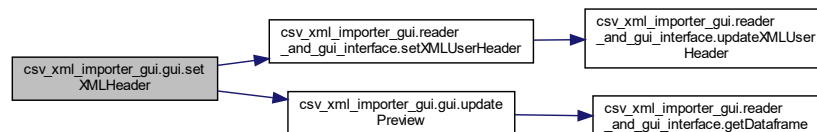
### 3.2.3.19 setXMLHeader()

```
def csv_xml_importer_gui.gui.setXMLHeader (
    self )
```

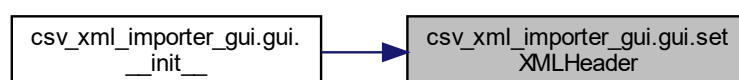
gets the value of the xml header checkbox and activates or deactivates the header accordingly

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



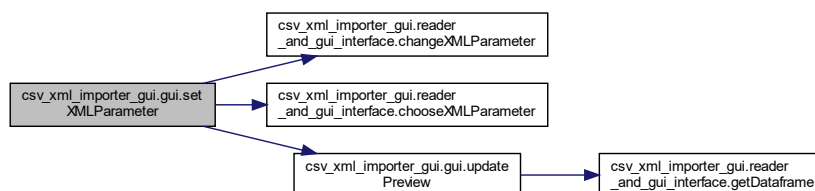
### 3.2.3.20 setXMLParameter()

```
def csv_xml_importer_gui.gui.setXMLParameter (
    self,
    return_event )
```

lets the user choose a xml parameter from the listbox. The value of the parameter will be shown in the textbox below in the gui to let the user change it

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



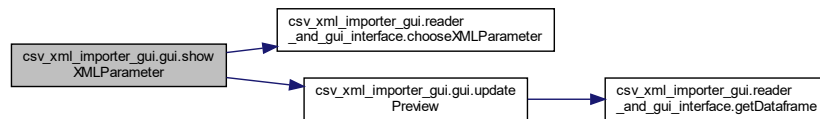
### 3.2.3.21 showXMLParameter()

```
def csv_xml_importer_gui.gui.showXMLParameter (
    self,
    selection_event )
```

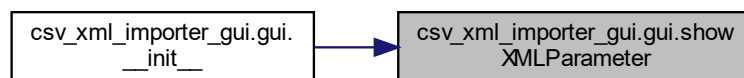
shows the found xml Parameters from the stylesheet in the listbox for the xml Parameters

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



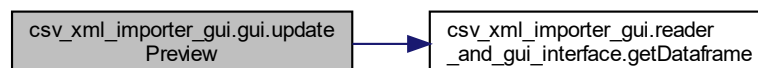
### 3.2.3.22 updatePreview()

```
def csv_xml_importer_gui.gui.updatePreview (
    self )
```

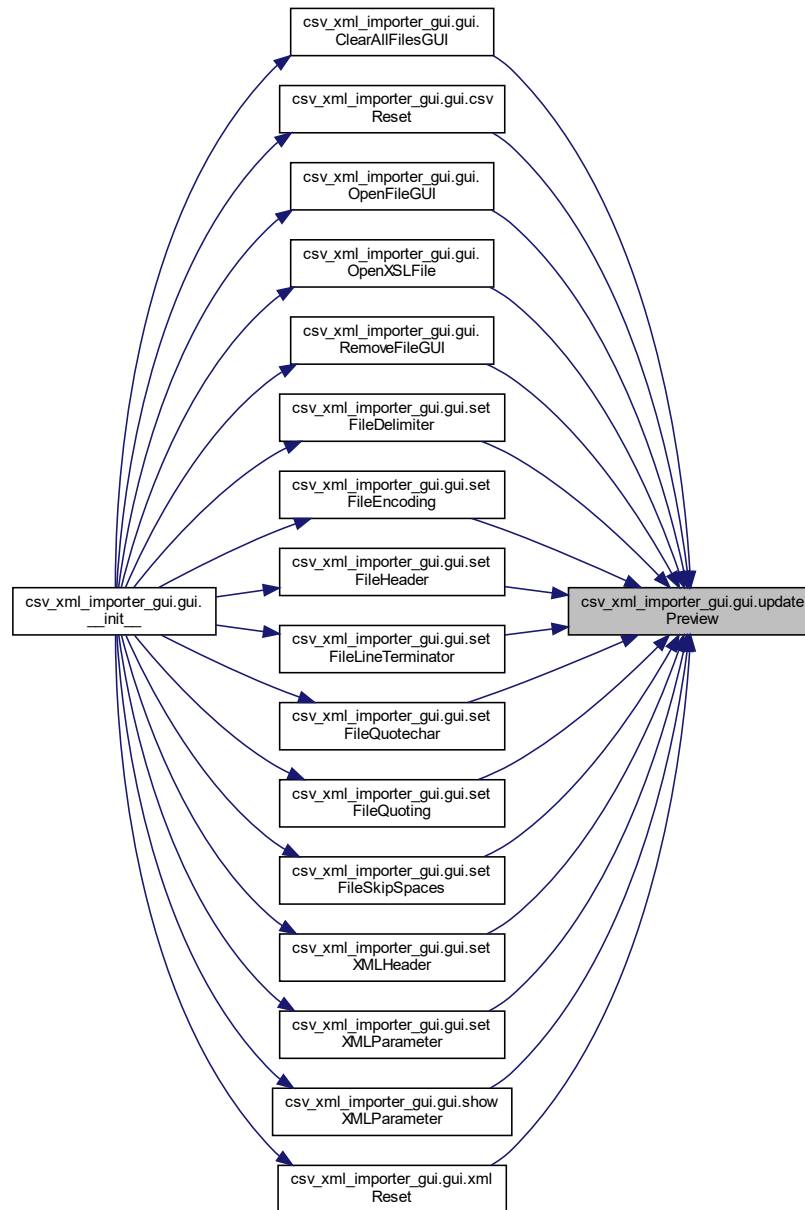
A simple Function to update the pandastable preview table

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



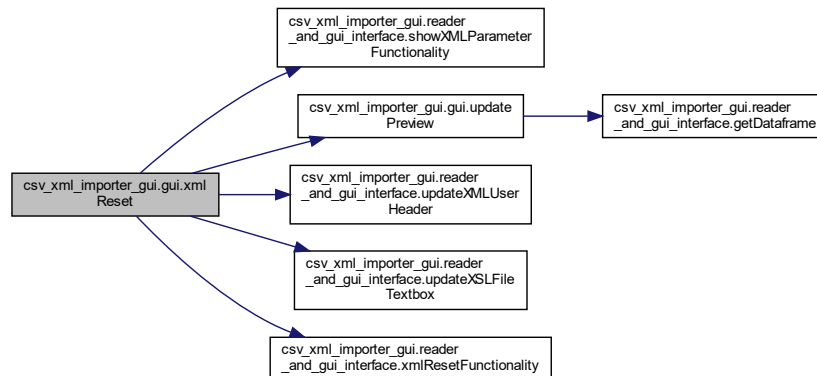
### 3.2.3.23 xmlReset()

```
def csv_xml_importer_gui.gui.xmlReset (
    self )
```

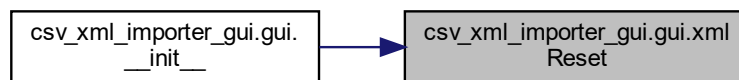
resets every parameter from the chosen xml file and resets the header

Returns:  
nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



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

- C:/Users/Philip/Desktop/Studienarbeit\_Python\_Philip\_Bartmann\_62442491/src/csv\_xml\_importer\_gui.py

## 3.3 reader.reader Class Reference

### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self)
- def [giveDataframe](#) (self)
- def [addToFilesDict](#) (self, str filename)
- def [csvSniffer](#) (self, str filename)
- def [read\\_with\\_init\\_settings](#) (self, str filename)
- def [update\\_dataframe](#) (self)
- def [update\\_csv\\_with\\_personal\\_settings](#) (self, str filename, bool wantHeader=None, str encoding=None, str Delimiter=None, str Quotechar=None, bool skipInitSpace=None, str lineTerminator=None, int quoting=None)
- def [OpenCSVFile](#) (self, str filename)
- def [getXMLParameters](#) (self, str filename, str xsl\_file)
- def [addXMLParameter](#) (self, str filename, str param=None, str value=None, bool wantHeader=None)
- def [OpenXMLFile](#) (self, str filename)
- def [reset](#) (self)

- def [RemoveFilesFunctionality](#) (self, str elem\_name)
- def [ClearAllFilesFunctionality](#) (self)
- def [importAsDictionary](#) (self)
- def [importAsListOfLists](#) (self)
- def [importAsNumPyArray](#) (self)
- def [importAsPandasDataframe](#) (self)
- def [exportAsCSVFile](#) (self, str exported\_file\_path, str encoding="UTF-8", str delimiter=";", str quotechar="\"", str line\_terminator="\r\n")
- def [exportAsXMLFile](#) (self, str exported\_file\_path, str encoding="UTF-8")

## Public Attributes

- **main\_dataframe**
- **importer**
- **column\_amount**

### 3.3.1 Detailed Description

The class reader converts csv and xml Files to Dataframes, lets the user change the parameters of the Files and Files as either a csv or a xml File or import as a Dictionary, a Numpy Array, a List of Lists or a Dataframe

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 `__init__()`

```
def reader.reader.__init__ (
    self )
```

The constructor of the class reader.

If defines the following attributes for every instance of the class:

`opened_files_dict`: This dictionary is keeping track of every File and its Parameters by defining the key as the value as a dictionary of the parameters

`multiple_files_counter`: a counter for every file that has been read multiple times and changes the name of e.g. `testFile.csv`, `testFile.csv_0`

`main_dataframe`: holds the main dataframe, therefore all the files appended to one dataframe

`column_amount`: the amount of columns main\_dataframe has

`importer`: a instance of the class `dataframeAndHeaderHandler` to call the methods of this class for every File

### 3.3.3 Member Function Documentation

### 3.3.3.1 addToFilesDict()

```
def reader.reader.addToFilesDict (
    self,
    str filename )
```

adds the given filename to opened\_files\_dict as a key and gives each filename a value of a dictionary for the

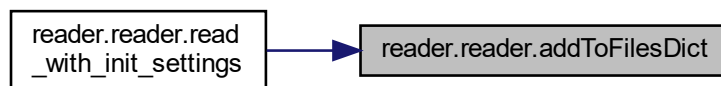
Parameters:

filename: a string, specifying the absolute or relative filepath of the file

Returns:

opened\_files\_dict: the updated dictionary with the new entry

Here is the caller graph for this function:



### 3.3.3.2 addXMLParameter()

```
def reader.reader.addXMLParameter (
    self,
    str filename,
    str param = None,
    str value = None,
    bool wantHeader = None )
```

Changes the values of the Parameters read from the xsl Stylesheet to the ones specified by the user

Parameters:

filename: a string, specifying the absolute or relative filepath of the file

param: the parameter of the stylesheet, which will be changed

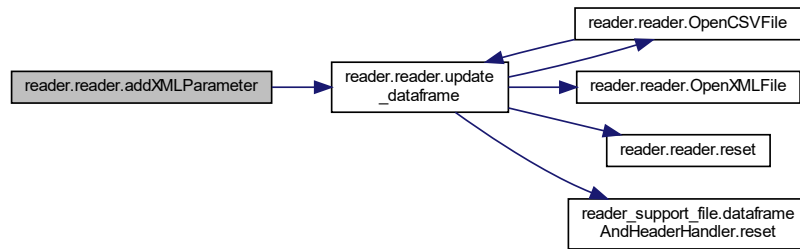
value: the new value of param, given by the user

wantHeader: a boolean variable, which signals if the user wants a header or not

Returns:

nothing or void

Here is the call graph for this function:

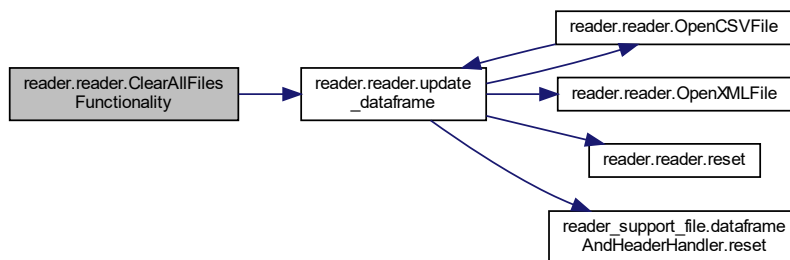


### 3.3.3.3 ClearAllFilesFunctionality()

```
def reader.reader.ClearAllFilesFunctionality (
    self )
```

Empties the whole `opened_files_dict` and updates `main_dataframe`

Here is the call graph for this function:



### 3.3.3.4 csvSniffer()

```
def reader.reader.csvSniffer (
    self,
    str filename )
```



This sniffer is using the sniffer of the csv module to get the most probable parameters of the given file. It also sniffs if the file has a header or not.

Parameters:

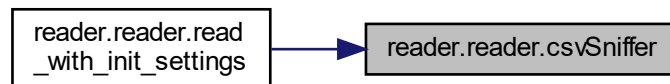
filename: a string, specifying the absolute or relative filepath of the file

Returns:

has\_header: a boolean variable, which signals if the file has a header or not

dialect: the sniffed dialect of the file, e.g. separator, line terminator, quoting,...

Here is the caller graph for this function:



### 3.3.3.5 exportAsCSVFile()

```
def reader.reader.exportAsCSVFile (
    self,
    str exported_file_path,
    str encoding = "UTF-8",
    str delimiter = ",",
    str quotechar = "\"",
    str line_terminator = "\r\n" )
```

Converts main\_dataframe to a csv File with the given parameters, which can be configured by the user and exported.

Parameters:

exported\_file\_path: the absolute or relative filepath, the csv File shall be exported to

encoding: the encoding the csv File will be encoded in

delimiter: the delimiter or separator used for the csv file

quotechar: the character used for quoting

line\_terminator: the character, which will be used for starting a new line in the csv File

Returns:

nothing or void

### 3.3.3.6 exportAsXMLFile()

```
def reader.reader.exportAsXMLFile (
    self,
    str exported_file_path,
    str encoding = "UTF-8" )
```

Converts main\_dataframe to a xml file with the given encoding and exports it to the given filepath

Parameters:

exported\_file\_path: the absolute or relative filepath, the csv File shall be exported to  
encoding: the encoding the csv File will be encoded in

Returns:

nothing or void

### 3.3.3.7 getXMLParameters()

```
def reader.reader.getXMLParameters (
    self,
    str filename,
    str xsl_file )
```

Searches the given xsl Stylesheet for parameters, gives those parameters to the specified xml File as a dictio

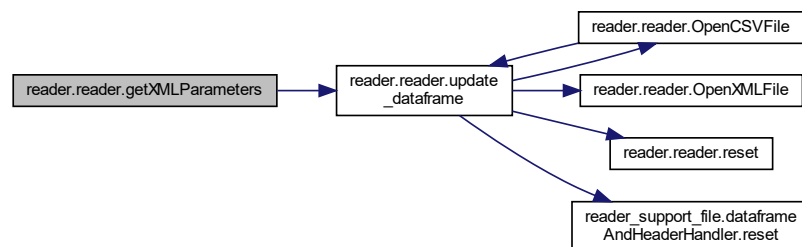
Parameters:

filename: a string, specifying the absolute or relative filepath of the file  
xsl\_file: a string, specifying the absolute or relative filepath of the xsl Stylesheet

Returns:

nothing or void

Here is the call graph for this function:



### 3.3.3.8 giveDataframe()

```
def reader.reader.giveDataframe (
    self )
```

A simple Function for giving main\_dataframe to another class

Returns:

main\_dataframe: the main dataframe

### 3.3.3.9 importAsDictionary()

```
def reader.reader.importAsDictionary (
    self )

imports main_dataframe as a Dictionary

Returns:
    main_dataframe converted to a Dictionary
```

### 3.3.3.10 importAsListOfLists()

```
def reader.reader.importAsListOfLists (
    self )

imports main_dataframe as a List of Lists

Returns:
    main_dataframe converted to a List of Lists
```

### 3.3.3.11 importAsNumPyArray()

```
def reader.reader.importAsNumPyArray (
    self )

imports main_dataframe as a Numpy Array

Returns:
    main_dataframe converted to a Numpy Array
```

### 3.3.3.12 importAsPandasDataframe()

```
def reader.reader.importAsPandasDataframe (
    self )

returns main_dataframe without conversion as its already a dataframe

Returns:
    main_dataframe
```

### 3.3.3.13 OpenCSVFile()

```
def reader.reader.OpenCSVFile (
    self,
    str filename )
```

Opens the given csv file with the specified parameters, converts it to a dataframe and adds this dataframe to

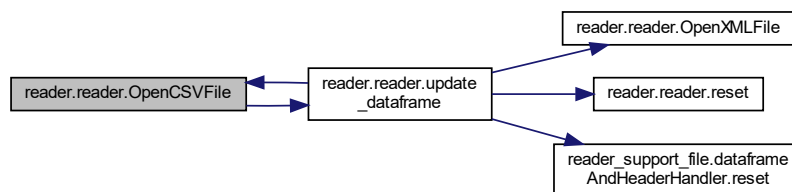
Parameters:

filename: a string, specifying the absolute or relative filepath of the file

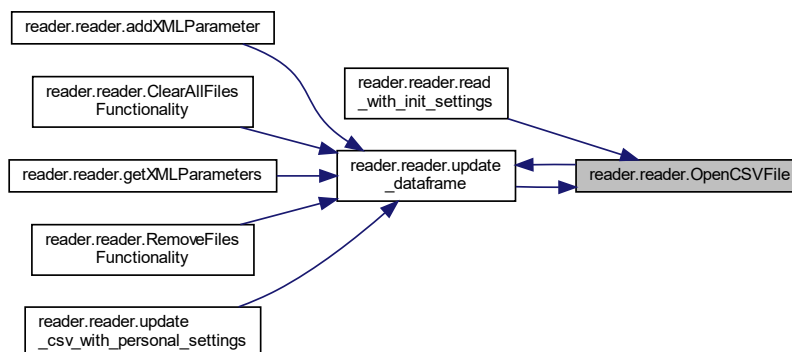
Returns:

main\_dataframe: the updated main\_dataframe with the appended csv File

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.3.3.14 OpenXMLFile()

```
def reader.reader.OpenXMLFile (
    self,
    str filename )
```

Opens the given xml File with the parameters of the xsl Stylesheet and converts it to a dataframe to append it

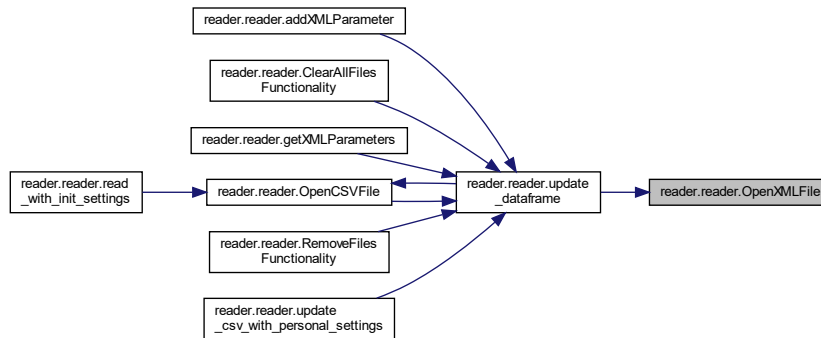
Parameters:

filename: a string, specifying the absolute or relative filepath of the file

Returns:

main\_dataframe: the updated main\_dataframe with the appended xml File

Here is the caller graph for this function:



### 3.3.3.15 read\_with\_init\_settings()

```
def reader.reader.read_with_init_settings (
    self,
    str filename )
```

This function takes a csv File, adds it to opened\_files\_dict, sniffs its dialect and if it has a header, gives the filename in opened\_files\_dict its corresponding parameter values and calls the method OpenCSVFile

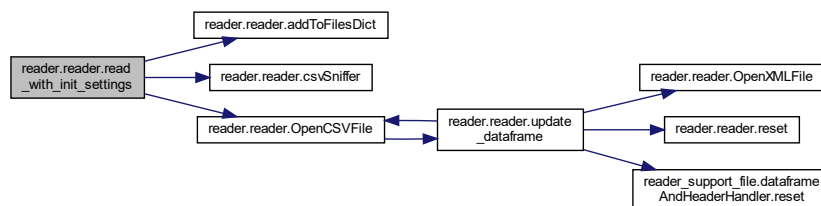
Parameters:

filename: a string, specifying the absolute or relative filepath of the file

Returns:

nothing or void

Here is the call graph for this function:



### 3.3.3.16 RemoveFilesFunctionality()

```
def reader.reader.RemoveFilesFunctionality (
    self,
    str elem_name )
```

Removes elem\_name from openend\_files\_dict and updates main\_dataframe

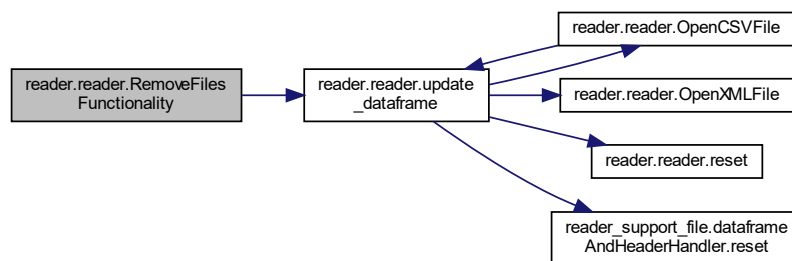
Parameters:

elem\_name: a string, specifying the absolute or relative filepath of the file, which must be in opened\_files

Returns:

nothing or void

Here is the call graph for this function:

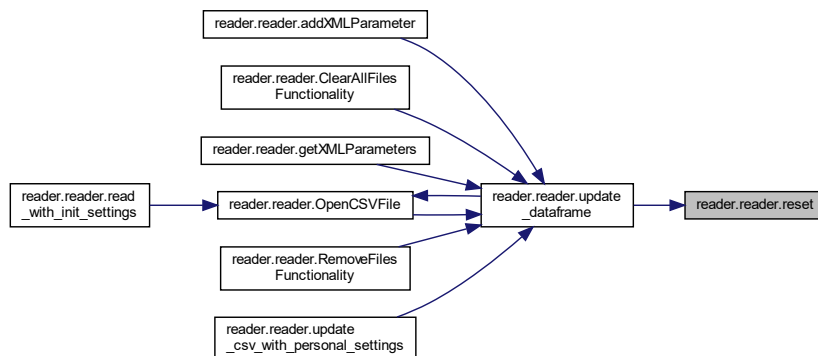


### 3.3.3.17 reset()

```
def reader.reader.reset (
    self )
```

A simple Function which resets main\_dataframe to a empty dataframe and the column\_amount to 0

Here is the caller graph for this function:



### 3.3.3.18 update\_csv\_with\_personal\_settings()

```
def reader.reader.update_csv_with_personal_settings (
    self,
    str filename,
    bool wantHeader = None,
    str encoding = None,
    str Delimiter = None,
    str Quotechar = None,
    bool skipInitSpace = None,
    str lineTerminator = None,
    int quoting = None )
```

This Function updates the parameters of the given csv file with the values the user wants, e.g. encoding, delimiter, etc. It also updates main\_dataframe

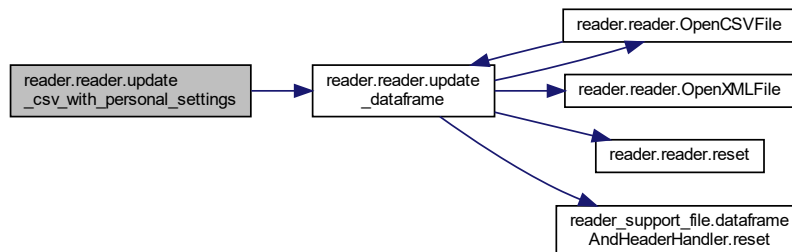
Parameters:

filename: a string, specifying the absolute or relative filepath of the file  
 wantHeader: a boolean value, signaling if the user wants a header or not  
 encoding: the encoding given by the user  
 delimiter: the delimiter specified by the user  
 Quotechar: the character, which will be used for quoting the entries of the csv file  
 skipInitSpaces: a boolean variable, signaling if the user wants to delete spaces at the beginning of the entries  
 lineTerminator: the character which will be used to start a new line  
 quoting: a integer, which corresponds to a specific quoting style, 0 means minimal quoting, 1 means all entries are quoted, 2 means all non numeric entries will be quoted and 3 means nothing will be quoted

Returns:

nothing or void

Here is the call graph for this function:



### 3.3.3.19 update\_dataframe()

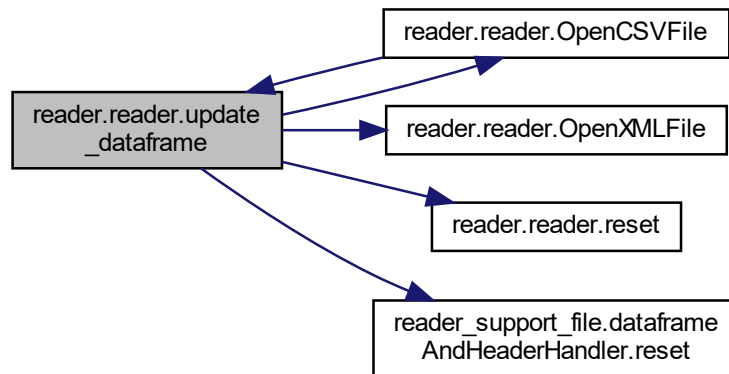
```
def reader.reader.update_dataframe (
    self )
```

A simple Function which resets the importer and the attributes of the instance of this class to read every file anew if any parameter of any file has changed

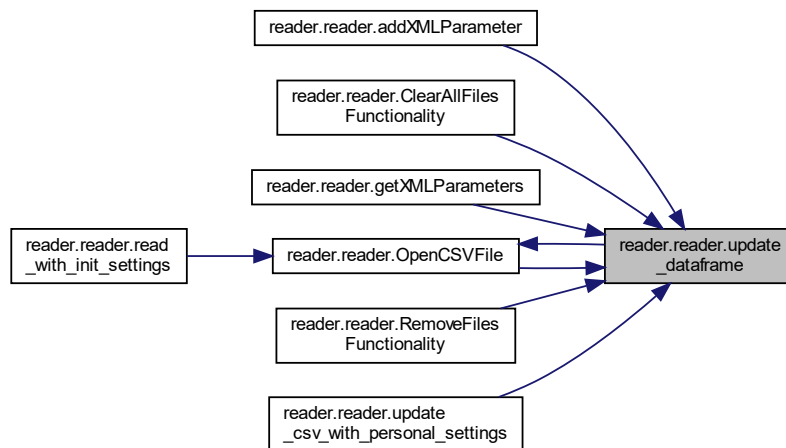
Returns:

nothing or void

Here is the call graph for this function:



Here is the caller graph for this function:



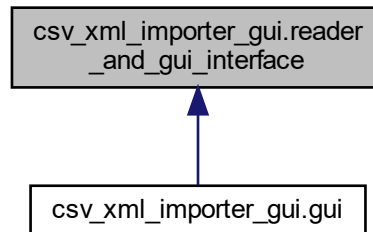
The documentation for this class was generated from the following file:

- `C:/Users/Philip/Desktop/Studienarbeit_Python_Philip_Bartmann_62442491/src/reader.py`



### 3.4 csv\_xml\_importer\_gui.reader\_and\_gui\_interface Class Reference

Inheritance diagram for csv\_xml\_importer\_gui.reader\_and\_gui\_interface:



#### Public Member Functions

- def `__init__` (self)
- def `getDataframe` (self)
- def `ShowFilesInterface` (self, tk.Listbox listbox)
- def `RemoveFilesInterface` (self, tk.Listbox listbox)
- def `ClearAllFilesInterface` (self, tk.Listbox listbox)
- def `setUserEncoding` (self, tk.Listbox listbox, tk.Entry encoding\_textbox, str wanted\_encoding)
- def `updateEncodingTextbox` (self, tk.Entry encodings\_textbox, str selected\_file)
- def `setUserDelimiter` (self, tk.Listbox listbox, tk.Entry delimiter\_textbox, str wanted\_delimiter)
- def `updateDelimiterTextbox` (self, tk.Entry delimiter\_textbox, str selected\_file)
- def `setUserQuotechar` (self, tk.Listbox listbox, tk.Entry quotechar\_textbox, str wanted\_quotechar)
- def `updateQuotecharTextbox` (self, tk.Entry quotechar\_textbox, str selected\_file)
- def `setUserHeader` (self, tk.Listbox listbox, header\_checkbox\_value)
- def `updateHeaderCheckbox` (self, tk.IntVar header\_checkbox\_value, str selected\_file)
- def `setUserSkipSpaces` (self, tk.Listbox listbox, tk.IntVar skip\_spaces\_checkbox\_value)
- def `updateSkipSpacesCheckbox` (self, tk.IntVar skip\_spaces\_checkbox\_value, str selected\_file)
- def `setUserLineTerminator` (self, tk.Listbox listbox, tk.Entry line\_terminator\_textbox, str wanted\_line\_terminator)
- def `updateLineTerminatorTextbox` (self, tk.Entry line\_terminator\_textbox, str selected\_file)
- def `setUserQuoting` (self, tk.Listbox listbox, tk.IntVar quoting\_var)
- def `updateQuotingRadioButtons` (self, tk.IntVar quoting\_var, str selected\_file)
- def `csvReset` (self, tk.Listbox listbox)
- def `getXSLFile` (self, tk.Listbox listbox, tk.Entry xsl\_textbox)
- def `updateXSLFileTextbox` (self, tk.Entry xsl\_textbox, str selected\_file)
- def `showXMLParameterFunctionality` (self, tk.Listbox listbox, tk.Listbox parameter\_listbox, str filename)
- def `chooseXMLParameter` (self, tk.Listbox listbox, tk.Listbox parameter\_listbox, tk.Entry xml\_parameters\_textbox, str filename)
- def `changeXMLParameter` (self, tk.Listbox listbox, tk.Listbox parameter\_listbox, tk.Entry parameters\_textbox, str filename)
- def `setXMLUserHeader` (self, tk.Listbox listbox, tk.IntVar xml\_header\_var)
- def `updateXMLUserHeader` (self, tk.IntVar xml\_header\_var, tk.Listbox listbox)
- def `xmlResetFunctionality` (self, tk.Listbox listbox)
- def `finalImporterFunctionality` (self, tk.IntVar import\_var\_value)
- def `finalCSVExportFunctionality` (self, str encoding, str delimiter, str quotechar, str line\_terminator)
- def `finalXMLExportFunctionality` (self, str encoding)

## Public Attributes

- `reader`
- `main_dataframe`

### 3.4.1 Detailed Description

This class acts as an interface between instances of the class `reader` and instances of the class `gui`, to separate these two classes, so that `reader` can be operated as an API and the class `gui` just takes the methods of `reader` and realises them in a `gui`

### 3.4.2 Constructor & Destructor Documentation

#### 3.4.2.1 `__init__()`

```
def csv_xml_importer_gui.reader_and_gui_interface.__init__ (
    self )
```

The constructor of the class `reader_and_gui_interface`.  
It defines the following attributes for every instance of the class:

```
reader: a instance of the class reader, to have the methods and attributes of reader available for the gui
index: an integer, which counts the Elements of the main Listbox in the gui, where the filenames are stored
main_dataframe: the main dataframe, to which all opened files are appended to, if they are compatible
```

Reimplemented in [csv\\_xml\\_importer\\_gui.gui](#).

### 3.4.3 Member Function Documentation

#### 3.4.3.1 `changeXMLParameter()`

```
def csv_xml_importer_gui.reader_and_gui_interface.changeXMLParameter (
    self,
    tk.Listbox listbox,
    tk.Listbox parameter_listbox,
    tk.Entry parameters_textbox,
    str filename )
```

lets the user change the content of the Parameters textbox to set a personal value for the parameter

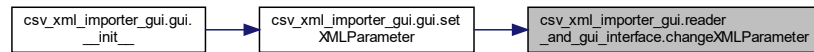
Parameters:

```
listbox: the main listbox at the top of the gui window, to show which files have been selected and in which
         they have been selected
parameter_listbox: the listbox for the xsl Parameters found in the xsl Stylesheet
parameters_textbox: a textbox below parameter_listbox in the gui, to write an change the parameter value
filename: the selected file, whose parameters shall be written in parameter_listbox
```

Returns:

```
nothing or void
```

Here is the caller graph for this function:



### 3.4.3.2 chooseXMLParameter()

```

def csv_xml_importer_gui.reader_and_gui_interface.chooseXMLParameter (
    self,
    tk.Listbox listbox,
    tk.Listbox parameter_listbox,
    tk.Entry xml_parameters_textbox,
    str filename )
  
```

This function lets the user choose a parameter from parameter\_listbox, and the corresponding value of the parameter will be written in xml\_parameters\_textbox

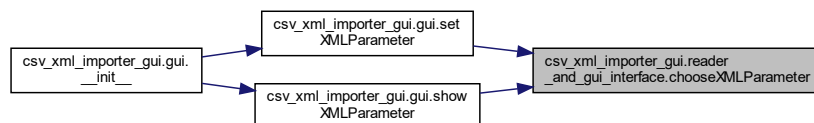
Parameters:

listbox: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected  
 parameter\_listbox: the listbox for the xsl Parameters found in the xsl Stylesheet  
 xml\_parameters\_textbox: a textbox below parameter\_listbox in the gui, to write an change the parameter value  
 filename: the selected file, whose parameters shall be written in parameter\_listbox

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.3 ClearAllFilesInterface()

```

def csv_xml_importer_gui.reader_and_gui_interface.ClearAllFilesInterface (
    self,
    tk.Listbox listbox )
  
```

Clears the whole listbox and opened\_files\_dict of entries

Parameters:

listbox: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected

Returns:

nothing or void

Here is the caller graph for this function:



#### 3.4.3.4 csvReset()

```
def csv_xml_importer_gui.reader_and_gui_interface.csvReset (
    self,
    tk.Listbox listbox )
```

Resets the selected csv File to have its default values for its Parameters again through sniffing from the file

Parameters:

listbox: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected

Returns:

filename: the selected file from the main listbox

Here is the caller graph for this function:



### 3.4.3.5 finalCSVExportFunctionality()

```
def csv_xml_importer_gui.reader_and_gui_interface.finalCSVExportFunctionality (
    self,
    str encoding,
    str delimiter,
    str quotechar,
    str line_terminator )
```

exports the chosen files as a csv file

Parameters:

encoding: the encoding of the csv file  
 delimiter: the delimiter of the csv file  
 quotechar: the character, which shall be used for quoting  
 line\_terminator: the character, which shall be used for ending lines

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.6 finalImporterFunctionality()

```
def csv_xml_importer_gui.reader_and_gui_interface.finalImporterFunctionality (
    self,
    tk.IntVar import_var_value )
```

imports the chosen files according to a value from the radiobuttons

Parameters:

import\_var\_value: the value, which decides what data type the dataframe will be converted to  
 1: Convert to a dictionary  
 2: Convert to a List of Lists  
 3: Convert to a Numpy Array  
 4: Convert to a Dataframe

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.7 finalXMLExportFunctionality()

```
def csv_xml_importer_gui.reader_and_gui_interface.finalXMLExportFunctionality (
    self,
    str encoding )
```

exports the chosen files as a xml file

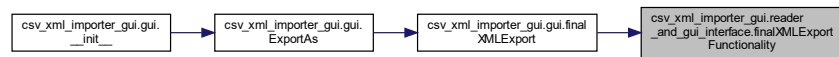
Parameters:

encoding: the encoding of the xml file

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.8 getDataframe()

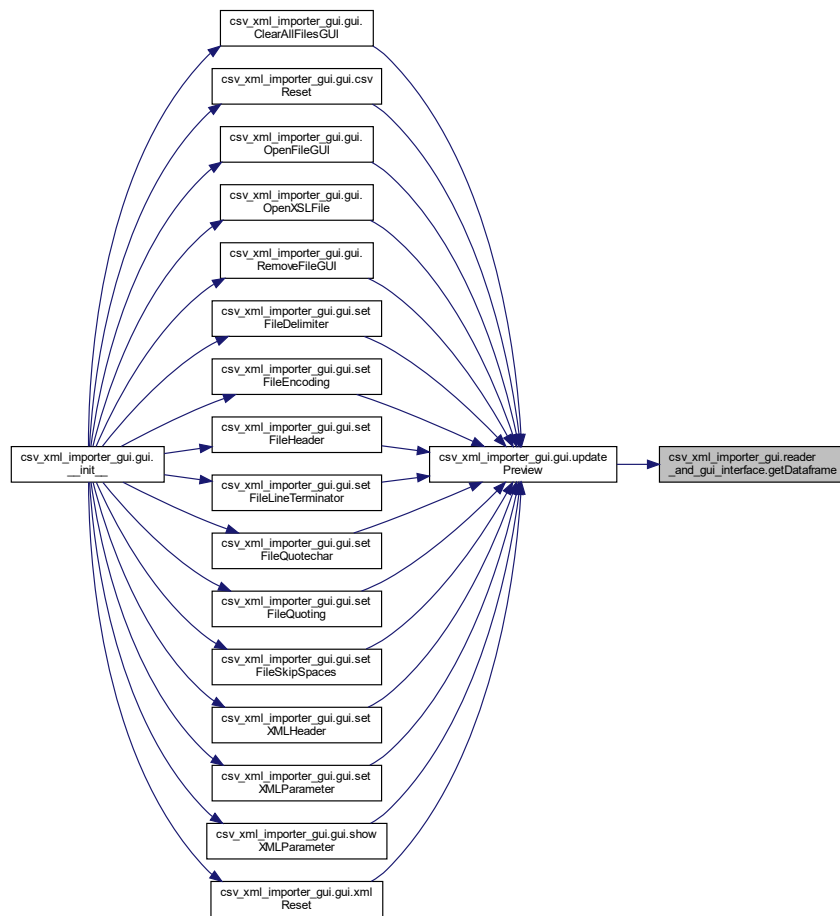
```
def csv_xml_importer_gui.reader_and_gui_interface.getDataframe (
    self )
```

A simple function, which gets main\_dataframe from the reader object and puts it in main\_dataframe of this object, to update pandastable

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.9 getXSLFile()

```
def csv_xml_importer_gui.reader_and_gui_interface.getXSLFile (
    self,
    tk.Listbox listbox,
    tk.Entry xsl_textbox )
```

reads xsl Stylesheet from file dialog

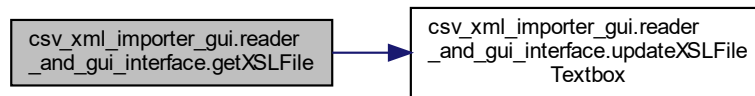
Parameters:

listbox: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected  
xsl\_textbox: The textbox the relative or absolute filepath of the xsl Stylesheet will be written in

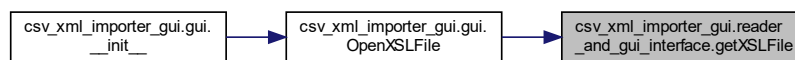
Returns:

True or False to check if the xsl Stylesheet can be used for the xml or not

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.4.3.10 RemoveFilesInterface()

```
def csv_xml_importer_gui.reader_and_gui_interface.RemoveFilesInterface (
    self,
    tk.Listbox listbox )
```

Removes Files from the main listbox and from the opened\_files\_dict dictionary

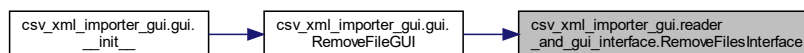
Parameters:

`listbox`: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected

Returns:

nothing or void

Here is the caller graph for this function:





### 3.4.3.11 setUserDelimiter()

```
def csv_xml_importer_gui.reader_and_gui_interface.setUserDelimiter (
    self,
    tk.Listbox listbox,
    tk.Entry delimiter_textbox,
    str wanted_delimiter )
```

Sets the delimiter or separator of the file selected in the listbox to the string the user typed into the text  
the delimiter

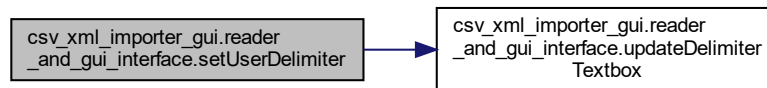
**Parameters:**

listbox: the main listbox at the top of the gui window, to show which files have been selected and in which  
they have been selected  
delimiter\_textbox: a Entry widget for the user to type in the wanted delimiter  
wanted\_delimiter: the string the user typed into delimiter\_textbox, which will be used as a delimiter

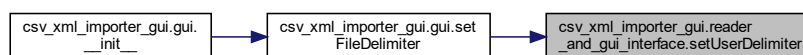
**Returns:**

filename: the selected file from the main listbox

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.4.3.12 setUserEncoding()

```
def csv_xml_importer_gui.reader_and_gui_interface.setUserEncoding (
    self,
    tk.Listbox listbox,
    tk.Entry encoding_textbox,
    str wanted_encoding )
```

This Function takes the string, the user typed in the textbox for encoding for this file and tries to set the encoding of the file to this string.  
If its not a valid encoding, the default sniffed encoding of the file will be used

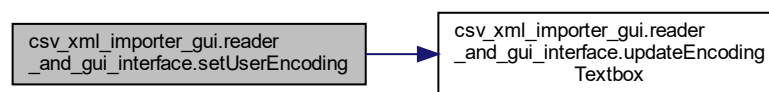
Parameters:

listbox: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected  
encoding\_textbox: a Entry widget from tkinter, in which the user can type in the wanted encoding  
wanted\_encoding: the string the user typed into encoding\_textbox

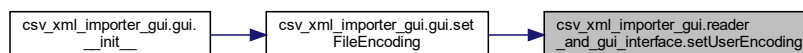
Returns:

filename: the selected filename from the listbox

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.4.3.13 setUserHeader()

```
def csv_xml_importer_gui.reader_and_gui_interface.setUserHeader (
    self,
    tk.Listbox listbox,
    header_checkbox_value )
```

Takes the boolean value of the checkbox for the header and sets the Header of the file accordingly

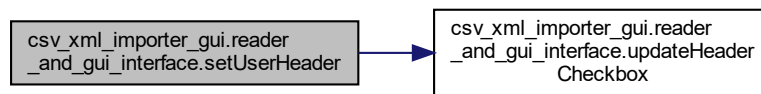
Parameters:

listbox: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected  
header\_checkbox\_value: the boolean value of the header checkbox

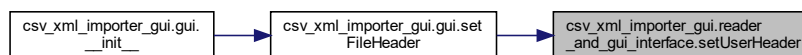
Returns:

filename: the selected file from the main listbox

Here is the call graph for this function:



Here is the caller graph for this function:



#### 3.4.3.14 setUserLineTerminator()

```

def csv_xml_importer_gui.reader_and_gui_interface.setUserLineTerminator (
    self,
    tk.Listbox listbox,
    tk.Entry line_terminator_textbox,
    str wanted_line_terminator )
  
```

Sets the character, which will be used for the end of a line of the file selected in the listbox to the string the user typed into the textbox for the Line Terminator

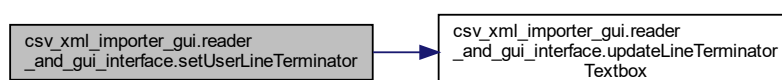
Parameters:

listbox: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected  
 line\_terminator\_textbox: a Entry widget for the user to type in the wanted Line Terminator  
 wanted\_line\_terminator: the character the user typed into line\_terminator\_textbox, which will be used as the line terminator

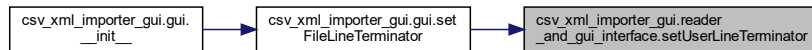
Returns:

filename: the selected file from the main listbox

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.4.3.15 setUserQuotechar()

```

def csv_xml_importer_gui.reader_and_gui_interface.setUserQuotechar (
    self,
    tk.Listbox listbox,
    tk.Entry quotechar_textbox,
    str wanted_quotechar )
  
```

Sets the character, which will be used for the quoting of the file selected in the listbox to the string the user typed into the textbox for the quotchar

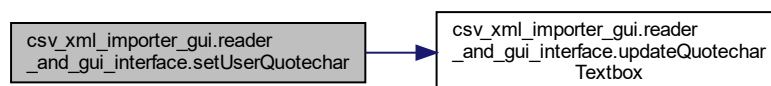
Parameters:

listbox: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected  
 quotechar\_textbox: a Entry widget for the user to type in the wanted quotechar  
 wanted\_quotechar: the character the user typed into quotechar\_textbox, which will be used as the quotechar

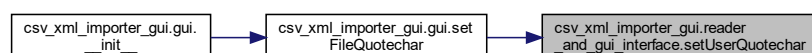
Returns:

filename: the selected file from the main listbox

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.4.3.16 setUserQuoting()

```
def csv_xml_importer_gui.reader_and_gui_interface.setUserQuoting (
    self,
    tk.Listbox listbox,
    tk.IntVar quoting_var )
```

Sets the quoting style of the File according to the selected radio Button in the gui.  
 There are four different quoting styles to choose:  
 0: Minimal quoting  
 1: Everything will be quoted  
 2: All non numeric fields of the dataframe will be quoted  
 3: nothing will be quoted

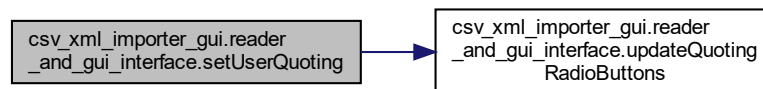
**Parameters:**

listbox: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected  
 quoting\_var: the integer variable, which has one of the four above mentioned values to set the quoting style

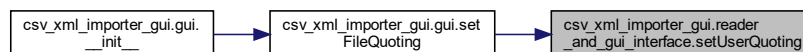
**Returns:**

filename: the selected file from the main listbox

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.4.3.17 setUserSkipSpaces()

```
def csv_xml_importer_gui.reader_and_gui_interface.setUserSkipSpaces (
    self,
    tk.Listbox listbox,
    tk.IntVar skip_spaces_checkbox_value )
```

Takes the boolean value of the `skip_spaces_checkbox` and skips the spaces at the beginning of a entry of the data if there are any

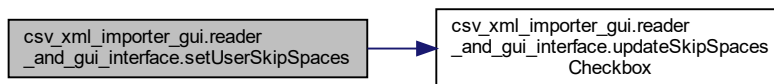
Parameters:

`listbox`: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected  
`skip_spaces_checkbox_value`: the boolean value of the `skip_spaces_checkbox`

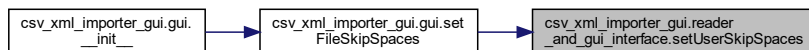
Returns:

`filename`: the selected file from the main listbox

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.4.3.18 setXMLUserHeader()

```
def csv_xml_importer_gui.reader_and_gui_interface.setXMLUserHeader (
    self,
    tk.Listbox listbox,
    tk.IntVar xml_header_var )
```

gets the value of the checkbox for the xml header of the selected file and sets the header accordingly

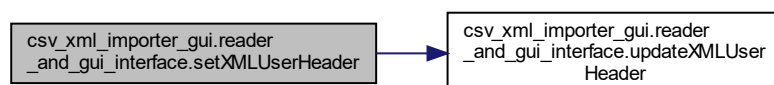
Parameters:

`listbox`: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected  
`xml_header_var`: the variable, which holds the selected value of the checkbox for the header

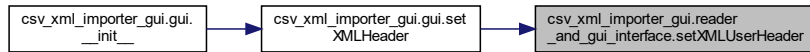
Returns:

`filename`: the selected file from the main listbox

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.4.3.19 ShowFilesInterface()

```
def csv_xml_importer_gui.reader_and_gui_interface.ShowFilesInterface (
    self,
    tk.Listbox listbox )
```

ShowFilesInterface asks the user to select the file or files he wants to see, opens them with the method `read_with_init_settings` of the reader object, but only if they are csv or xml files, and inserts them in the listbox for files in the gui

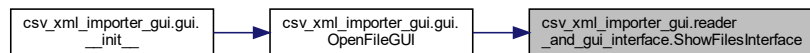
Parameters:

listbox: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.20 showXMLParameterFunctionality()

```
def csv_xml_importer_gui.reader_and_gui_interface.showXMLParameterFunctionality (
    self,
    tk.Listbox listbox,
    tk.Listbox parameter_listbox,
    str filename )
```

writes the found xsl Parameters in the listbox in the xml configurator labelframe

Parameters:

listbox: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected

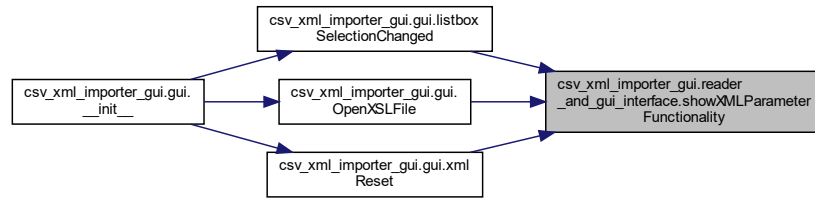
parameter\_listbox: the listbox for the xsl Parameters found in the xsl Stylesheet

filename: the selected file, whose parameters shall be written in parameter\_listbox

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.21 updateDelimiterTextbox()

```

def csv_xml_importer_gui.reader_and_gui_interface.updateDelimiterTextbox (
    self,
    tk.Entry delimiter_textbox,
    str selected_file )
  
```

Updates the Textbox the Delimiter is placed in, to show the user which Delimiter is currently used

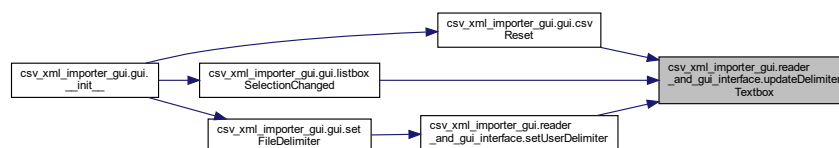
Parameters:

delimiter\_textbox: a Entry widget for the user to type in the wanted delimiter  
 selected\_file: the file, which has been selected in the main listbox

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.22 updateEncodingTextbox()

```

def csv_xml_importer_gui.reader_and_gui_interface.updateEncodingTextbox (
    self,
    tk.Entry encodings_textbox,
    str selected_file )
  
```



updates the textbox for encoding, so that it shows the encoding used to decode the file

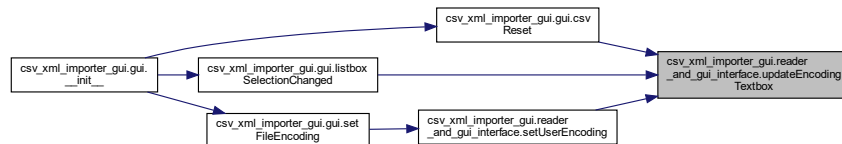
Parameters:

encodings\_textbox: a Entry widget from tkinter, in which the user can type in the wanted encoding  
 selected\_file: the file, which has been selected in the main listbox

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.23 updateHeaderCheckbox()

```

def csv_xml_importer_gui.reader_and_gui_interface.updateHeaderCheckbox (
    self,
    tk.IntVar header_checkbox_value,
    str selected_file )
  
```

updates the checkbox in the gui to show if the header is set or not

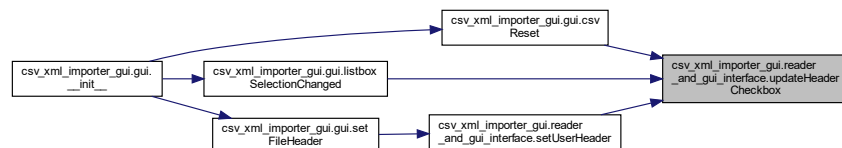
Parameters:

header\_checkbox\_value: the value of the checkbox for the header  
 selected\_file: the file, which has been selected in the main listbox

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.24 updateLineTerminatorTextbox()

```
def csv_xml_importer_gui.reader_and_gui_interface.updateLineTerminatorTextbox (
    self,
    tk.Entry line_terminator_textbox,
    str selected_file )
```

Updates the Textbox the character for the Line Terminator of the file is placed in, to show the user which character is currently used

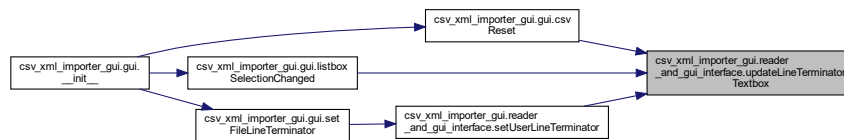
Parameters:

line\_terminator\_textbox: a Entry widget for the user to type in the wanted Line Terminator  
selected\_file: the file, which has been selected in the main listbox

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.25 updateQuotecharTextbox()

```
def csv_xml_importer_gui.reader_and_gui_interface.updateQuotecharTextbox (
    self,
    tk.Entry quotechar_textbox,
    str selected_file )
```

Updates the Textbox the character for the quoting of the file is placed in, to show the user which character is currently used

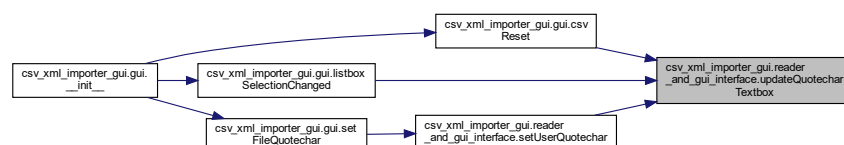
Parameters:

quotechar\_textbox: a Entry widget for the user to type in the wanted quotechar  
selected\_file: the file, which has been selected in the main listbox

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.26 updateQuotingRadioButtons()

```
def csv_xml_importer_gui.reader_and_gui_interface.updateQuotingRadioButtons (
    self,
    tk.IntVar quoting_var,
    str selected_file )
```

updates the variable for the quoting style, so the right radio button is selected in the gui

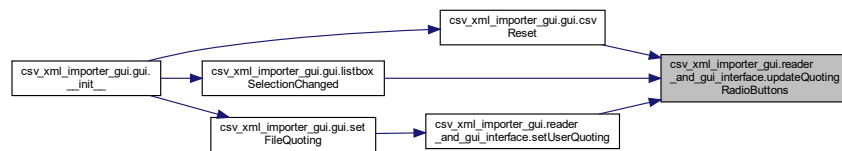
Parameters:

quoting\_var: the integer variable, which has one of the four above mentioned values to set the quoting style  
 selected\_file: the file, which has been selected in the main listbox

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.27 updateSkipSpacesCheckbox()

```
def csv_xml_importer_gui.reader_and_gui_interface.updateSkipSpacesCheckbox (
    self,
    tk.IntVar skip_spaces_checkbox_value,
    str selected_file )
```

updates the checkbox in the gui to show if skip\_spaces is set or not

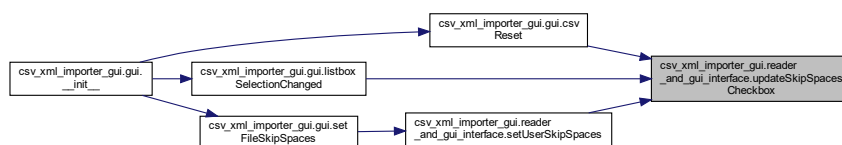
Parameters:

skip\_spaces\_checkbox\_value: the boolean value of the skip\_spaces\_checkbox  
 selected\_file: the file, which has been selected in the main listbox

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.28 updateXMLUserHeader()

```
def csv_xml_importer_gui.reader_and_gui_interface.updateXMLUserHeader (
    self,
    tk.IntVar xml_header_var,
    tk.Listbox listbox )
```

updates the checkbox to show, if the xml file has a header or not

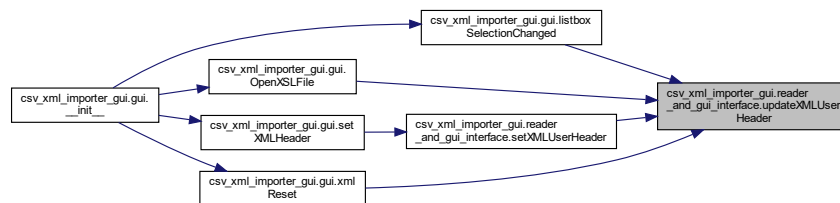
Parameters:

xml\_header\_var: the variable, which holds the selected value of the checkbox for the header  
listbox: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.29 updateXSLFileTextbox()

```
def csv_xml_importer_gui.reader_and_gui_interface.updateXSLFileTextbox (
    self,
    tk.Entry xsl_textbox,
    str selected_file )
```

updates the xsl textbox to write the absolute or relative filepath of the xsl Stylesheet in it

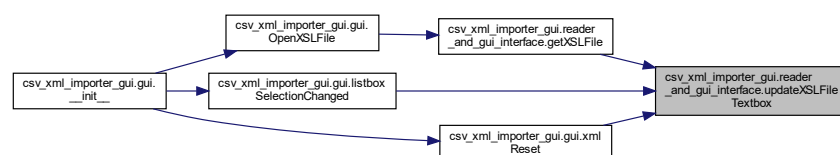
Parameters:

xsl\_textbox: the textbox for the filepath  
selected\_file: the file, which has been selected in the main listbox

Returns:

nothing or void

Here is the caller graph for this function:



### 3.4.3.30 xmlResetFunctionality()

```
def csv_xml_importer_gui.reader_and_gui_interface.xmlResetFunctionality (
    self,
    tk.Listbox listbox )
```

resets the selected xml file to its default values from the xsl Stylesheet

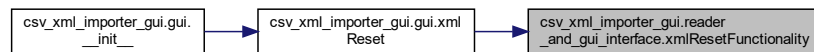
Parameters:

*listbox*: the main listbox at the top of the gui window, to show which files have been selected and in which they have been selected

Returns:

*selected\_file*: the selected file from the main listbox

Here is the caller graph for this function:



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

- C:/Users/Philip/Desktop/Studienarbeit\_Python\_Philip\_Bartmann\_62442491/src/csv\_xml\_importer\_gui.py



# Index

- `__init__`
  - `csv_xml_importer_gui.gui`, 11
  - `csv_xml_importer_gui.reader_and_gui_interface`, 44
  - `reader.reader`, 32
  - `reader_support_file.dataframeAndHeaderHandler`, 5
- `addToFilesDict`
  - `reader.reader`, 32
- `addXMLParameter`
  - `reader.reader`, 33
- `changeXMLParameter`
  - `csv_xml_importer_gui.reader_and_gui_interface`, 44
- `chooseXMLParameter`
  - `csv_xml_importer_gui.reader_and_gui_interface`, 45
- `ClearAllFilesFunctionality`
  - `reader.reader`, 34
- `ClearAllFilesGUI`
  - `csv_xml_importer_gui.gui`, 12
- `ClearAllFilesInterface`
  - `csv_xml_importer_gui.reader_and_gui_interface`, 45
- `csv_xml_importer_gui.gui`, 9
  - `__init__`, 11
  - `ClearAllFilesGUI`, 12
  - `csvReset`, 13
  - `ExportAs`, 14
  - `finalCSVExport`, 15
  - `finalImporter`, 16
  - `finalXMLExport`, 16
  - `ImportAs`, 17
  - `listboxSelectionChanged`, 18
  - `OpenFileGUI`, 19
  - `OpenXSLFile`, 20
  - `RemoveFileGUI`, 21
  - `setFileDelimiter`, 22
  - `setFileEncoding`, 22
  - `setFileHeader`, 23
  - `setFileLineTerminator`, 24
  - `setFileQuotechar`, 25
  - `setFileQuoting`, 25
  - `setFileSkipSpaces`, 26
  - `setXMLHeader`, 27
  - `setXMLParameter`, 28
  - `showXMLParameter`, 28
  - `updatePreview`, 29
  - `xmlReset`, 30
- `csv_xml_importer_gui.reader_and_gui_interface`, 43
  - `__init__`, 44
  - `changeXMLParameter`, 44
  - `chooseXMLParameter`, 45
  - `ClearAllFilesInterface`, 45
  - `csvReset`, 46
  - `finalCSVExportFunctionality`, 46
  - `finalImporterFunctionality`, 47
  - `finalXMLExportFunctionality`, 47
  - `getDataframe`, 48
  - `getXSLFile`, 49
  - `RemoveFilesInterface`, 50
  - `setUserDelimiter`, 50
  - `setUserEncoding`, 51
  - `setUserHeader`, 52
  - `setUserLineTerminator`, 53
  - `setUserQuotechar`, 54
  - `setUserQuoting`, 54
  - `setUserSkipSpaces`, 55
  - `setXMLUserHeader`, 56
  - `ShowFilesInterface`, 57
  - `showXMLParameterFunctionality`, 57
  - `updateDelimiterTextbox`, 58
  - `updateEncodingTextbox`, 58
  - `updateHeaderCheckbox`, 59
  - `updateLineTerminatorTextbox`, 59
  - `updateQuotecharTextbox`, 60
  - `updateQuotingRadioButtons`, 60
  - `updateSkipSpacesCheckbox`, 61
  - `updateXMLUserHeader`, 61
  - `updateXSLFileTextbox`, 62
  - `xmlResetFunctionality`, 62
- `csvReset`
  - `csv_xml_importer_gui.gui`, 13
  - `csv_xml_importer_gui.reader_and_gui_interface`, 46
- `csvSniffer`
  - `reader.reader`, 34
- `ExportAs`
  - `csv_xml_importer_gui.gui`, 14
- `exportAsCSVFile`
  - `reader.reader`, 35
- `exportAsXMLFile`
  - `reader.reader`, 35
- `finalCSVExport`
  - `csv_xml_importer_gui.gui`, 15
- `finalCSVExportFunctionality`

- csv\_xml\_importer\_gui.reader\_and\_gui\_interface, 46
- finalImporter
  - csv\_xml\_importer\_gui.gui, 16
- finalImporterFunctionality
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, 47
- finalXMLExport
  - csv\_xml\_importer\_gui.gui, 16
- finalXMLExportFunctionality
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, 47
- find\_header\_formats
  - reader\_support\_file.dataframeAndHeaderHandler, 6
- getDataframe
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, 48
- getXMLParameters
  - reader.reader, 36
- getXSLFile
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, 49
- giveDataframe
  - reader.reader, 36
- ImportAs
  - csv\_xml\_importer\_gui.gui, 17
- importAsDictionary
  - reader.reader, 36
- importAsListOfLists
  - reader.reader, 37
- importAsNumPyArray
  - reader.reader, 37
- importAsPandasDataframe
  - reader.reader, 37
- ImportFile
  - reader\_support\_file.dataframeAndHeaderHandler, 6
- listboxSelectionChanged
  - csv\_xml\_importer\_gui.gui, 18
- OpenCSVFile
  - reader.reader, 37
- OpenFileGUI
  - csv\_xml\_importer\_gui.gui, 19
- OpenXMLFile
  - reader.reader, 38
- OpenXSLFile
  - csv\_xml\_importer\_gui.gui, 20
- read\_with\_init\_settings
  - reader.reader, 39
- reader.reader, 31
  - \_\_init\_\_, 32
  - addToFilesDict, 32
  - addXMLParameter, 33
  - ClearAllFilesFunctionality, 34
  - csvSniffer, 34
  - exportAsCSVFile, 35
  - exportAsXMLFile, 35
  - getXMLParameters, 36
  - giveDataframe, 36
  - importAsDictionary, 36
  - importAsListOfLists, 37
  - importAsNumPyArray, 37
  - importAsPandasDataframe, 37
  - OpenCSVFile, 37
  - OpenXMLFile, 38
  - read\_with\_init\_settings, 39
  - RemoveFilesFunctionality, 39
  - reset, 40
  - update\_csv\_with\_personal\_settings, 40
  - update\_dataframe, 41
- reader\_support\_file.dataframeAndHeaderHandler, 5
  - \_\_init\_\_, 5
  - find\_header\_formats, 6
  - ImportFile, 6
  - regex\_list\_filler, 7
  - regex\_tester, 7
  - reset, 8
- regex\_list\_filler
  - reader\_support\_file.dataframeAndHeaderHandler, 7
- regex\_tester
  - reader\_support\_file.dataframeAndHeaderHandler, 7
- RemoveFileGUI
  - csv\_xml\_importer\_gui.gui, 21
- RemoveFilesFunctionality
  - reader.reader, 39
- RemoveFilesInterface
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, 50
- reset
  - reader.reader, 40
  - reader\_support\_file.dataframeAndHeaderHandler, 8
- setFileDelimiter
  - csv\_xml\_importer\_gui.gui, 22
- setFileEncoding
  - csv\_xml\_importer\_gui.gui, 22
- setFileHeader
  - csv\_xml\_importer\_gui.gui, 23
- setFileLineTerminator
  - csv\_xml\_importer\_gui.gui, 24
- setFileQuotechar
  - csv\_xml\_importer\_gui.gui, 25
- setFileQuoting
  - csv\_xml\_importer\_gui.gui, 25
- setFileSkipSpaces
  - csv\_xml\_importer\_gui.gui, 26
- setUserDelimiter
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, 50



- setUserEncoding
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [51](#)
- setUserHeader
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [52](#)
- setUserLineTerminator
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [53](#)
- setUserQuotechar
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [54](#)
- setUserQuoting
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [54](#)
- setUserSkipSpaces
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [55](#)
- setXMLHeader
  - csv\_xml\_importer\_gui.gui, [27](#)
- setXMLParameter
  - csv\_xml\_importer\_gui.gui, [28](#)
- setXMLUserHeader
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [56](#)
- ShowFilesInterface
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [57](#)
- showXMLParameter
  - csv\_xml\_importer\_gui.gui, [28](#)
- showXMLParameterFunctionality
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [57](#)
- update\_csv\_with\_personal\_settings
  - reader.reader, [40](#)
- update\_dataframe
  - reader.reader, [41](#)
- updateDelimiterTextbox
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [58](#)
- updateEncodingTextbox
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [58](#)
- updateHeaderCheckbox
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [59](#)
- updateLineTerminatorTextbox
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [59](#)
- updatePreview
  - csv\_xml\_importer\_gui.gui, [29](#)
- updateQuotecharTextbox
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [60](#)
- updateQuotingRadioButtons
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [60](#)
- updateSkipSpacesCheckbox
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [61](#)
- updateXMLUserHeader
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [61](#)
- updateXSLFileTextbox
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [62](#)
- xmlReset
  - csv\_xml\_importer\_gui.gui, [30](#)
- xmlResetFunctionality
  - csv\_xml\_importer\_gui.reader\_and\_gui\_interface, [62](#)