
tkfilebrowser Documentation

Release 2.2.5

Juliette Monsel

Aug 01, 2019

Contents:

1	Installation	3
2	Example	5
3	Documentation	9
4	Changelog	13
5	Index	17
	Index	19

tkfilebrowser is an alternative to `tkinter.filedialog` that allows the user to select files or directories. The GUI is written with `tkinter` but the look is closer to `GTK` and the application uses `GTK` bookmarks (the one displayed in `nautilus` or `thunar` for instance). This filebrowser supports new directory creation and filtype filtering.

This module contains a general *FileBrowser* class which implements the filebrowser and the following functions, similar to the one in `filedialog`:

- *askopenfilename()* that allow the selection of a single file
- *askopenfilenames()* that allow the selection of multiple files
- *askopendirname()* that allow the selection a single folder
- *askopendirnames()* that allow the selection of multiple folders
- *asksaveasfilename()* that returns a single filename and give a warning if the file already exists

Project page: <https://github.com/j4321/tkFileBrowser>

1.1 Requirements

- Linux or Windows
- Python 2.7 or 3.x

And the python packages:

- tkinter (included in the python distribution for Windows)
- psutil
- babel
- pywin32 (Windows only)
- pillow (only if `tkinter.TkVersion < 8.6`)

1.2 Install

- Ubuntu: use the PPA [ppa:j-4321-i/ppa](#)

```
$ sudo add-apt-repository ppa:j-4321-i/ppa
$ sudo apt-get update
$ sudo apt-get install python(3)-tkfilebrowser
```

- Archlinux:
the package is available on [AUR](#)
- With pip:

```
$ pip install tkfilebrowser
```


CHAPTER 2

Example

```
try:
    import tkinter as tk
    import tkinter.ttk as ttk
    from tkinter import filedialog
except ImportError:
    import Tkinter as tk
    import ttk
    import tkinterFileDialog as filedialog
from tkfilebrowser import askopendirname, askopenfilenames, asksaveasfilename

root = tk.Tk()

style = ttk.Style(root)
style.theme_use("clam")

def c_open_file_old():
    rep = filedialog.askopenfilenames(parent=root,
                                      initialdir='/',
                                      initialfile='tmp',
                                      filetypes=[("PNG", "*.png"),
                                                  ("JPEG", "*.jpg"),
                                                  ("All files", "*")])

    print(rep)

def c_open_dir_old():
    rep = filedialog.askdirectory(parent=root, initialdir='/tmp')
    print(rep)

def c_save_old():
    rep = filedialog.asksaveasfilename(parent=root,
```

(continues on next page)

(continued from previous page)

```

        defaultextension=".png",
        initialdir='/tmp',
        initialfile='image.png',
        filetypes=[("PNG", "*.png"),
                    ("JPEG", "*.jpg"),
                    ("All files", "*")]])

    print(rep)

def c_open_file():
    rep = askopenfilenames(parent=root,
                           initialdir='/',
                           initialfile='tmp',
                           filetypes=[("Pictures", "*.png|*.jpg|*.JPG"),
                                       ("All files", "*")]])

    print(rep)

def c_open_dir():
    rep = askopendirname(parent=root,
                         initialdir='/',
                         initialfile='tmp')

    print(rep)

def c_save():
    rep = asksaveasfilename(parent=root,
                           defaulttext=".png",
                           initialdir='/tmp',
                           initialfile='image.png',
                           filetypes=[("Pictures", "*.png|*.jpg|*.JPG"),
                                       ("All files", "*")]])

    print(rep)

ttk.Label(root, text='Default dialogs').grid(row=0, column=0,
                                             padx=4, pady=4,
                                             sticky='ew')
ttk.Label(root, text='tkfilebrowser dialogs').grid(row=0, column=1,
                                                    padx=4, pady=4,
                                                    sticky='ew')
ttk.Button(root, text="Open files", command=c_open_file_old).grid(row=1, column=0,
                                                                  padx=4, pady=4,
                                                                  sticky='ew')
ttk.Button(root, text="Open folder", command=c_open_dir_old).grid(row=2, column=0,
                                                                  padx=4, pady=4,
                                                                  sticky='ew')
ttk.Button(root, text="Save file", command=c_save_old).grid(row=3, column=0,
                                                            padx=4, pady=4,
                                                            sticky='ew')
ttk.Button(root, text="Open files", command=c_open_file).grid(row=1, column=1,
                                                              padx=4, pady=4,
                                                              sticky='ew')
ttk.Button(root, text="Open folder", command=c_open_dir).grid(row=2, column=1, padx=4,
                                                             padx=4, pady=4,
                                                             sticky='ew')
ttk.Button(root, text="Save file", command=c_save).grid(row=3, column=1,,

```

(continues on next page)

(continued from previous page)

```
padx=4, pady=4,  
sticky='ew')  
  
root.mainloop()
```


3.1 askopendirname

`tkfilebrowser.askopendirname` (*parent=None, title='Open', **kwargs*)

Return '' or the absolute path of the chosen directory.

Arguments:

parent [Tk or Toplevel instance] parent window

title [str] the title of the filebrowser window

initialdir [str] directory whose content is initially displayed

initialfile [str] initially selected item (just the name, not the full path)

filetypes [list [("name", "*.ext1|*.ext2|.."), ...]] only the files of given filetype will be displayed, e.g. to allow the user to switch between displaying only PNG or JPG pictures or displaying all files: `filetypes=[("Pictures", "*.png|*.PNG|*.jpg|*.JPG"), ("All files", "*")]`

okbuttontext [str] text displayed on the validate button, default is “Open”.

cancelbuttontext [str] text displayed on the button that cancels the selection, default is “Cancel”.

foldercreation [bool] enable the user to create new folders if True (default)

3.2 askopendirnames

`tkfilebrowser.askopendirnames` (*parent=None, title='Open', **kwargs*)

Return () or the tuple of the absolute paths of the chosen directories

Arguments:

parent [Tk or Toplevel instance] parent window

title [str] the title of the filebrowser window

initialdir [str] directory whose content is initially displayed

initialfile [str] initially selected item (just the name, not the full path)

filetypes [list [("name", "*.ext1|*.ext2|.."), ...]] only the files of given filetype will be displayed, e.g. to allow the user to switch between displaying only PNG or JPG pictures or displaying all files: `filetypes=[("Pictures", "*.png|*.PNG|*.jpg|*.JPG"), ("All files", "*")]`

okbuttontext [str] text displayed on the validate button, default is “Open”.

cancelbuttontext [str] text displayed on the button that cancels the selection, default is “Cancel”.

foldercreation [bool] enable the user to create new folders if True (default)

3.3 askopenfilename

`tkfilebrowser.askopenfilename` (*parent=None, title='Open', **kwargs*)

Return '' or the absolute path of the chosen file

Arguments:

parent [Tk or Toplevel instance] parent window

title [str] the title of the filebrowser window

initialdir [str] directory whose content is initially displayed

initialfile [str] initially selected item (just the name, not the full path)

filetypes [list [("name", "*.ext1|*.ext2|.."), ...]] only the files of given filetype will be displayed, e.g. to allow the user to switch between displaying only PNG or JPG pictures or displaying all files: `filetypes=[("Pictures", "*.png|*.PNG|*.jpg|*.JPG"), ("All files", "*")]`

okbuttontext [str] text displayed on the validate button, default is “Open”.

cancelbuttontext [str] text displayed on the button that cancels the selection, default is “Cancel”.

foldercreation [bool] enable the user to create new folders if True (default)

3.4 askopenfilenames

`tkfilebrowser.askopenfilenames` (*parent=None, title='Open', **kwargs*)

Return () or the tuple of the absolute paths of the chosen files

Arguments:

parent [Tk or Toplevel instance] parent window

title [str] the title of the filebrowser window

initialdir [str] directory whose content is initially displayed

initialfile [str] initially selected item (just the name, not the full path)

filetypes [list [("name", "*.ext1|*.ext2|.."), ...]] only the files of given filetype will be displayed, e.g. to allow the user to switch between displaying only PNG or JPG pictures or displaying all files: `filetypes=[("Pictures", "*.png|*.PNG|*.jpg|*.JPG"), ("All files", "*")]`

okbuttontext [str] text displayed on the validate button, default is “Open”.

cancelbuttontext [str] text displayed on the button that cancels the selection, default is “Cancel”.

foldercreation [bool] enable the user to create new folders if True (default)

3.5 asksaveasfilename

`tkfilebrowser.asksaveasfilename` (*parent=None, title='Save As', **kwargs*)

Return '' or the chosen absolute path (the file might not exist)

Arguments:

parent [Tk or Toplevel instance] parent window

title [str] the title of the filebrowser window

initialdir [str] directory whose content is initially displayed

initialfile [str] initially selected item (just the name, not the full path)

defaulttext [str (e.g. '.png')] extension added to filename if none is given (default is none)

filetypes [list [("name", "*.ext1|*.ext2|.."), ...]] only the files of given filetype will be displayed, e.g. to allow the user to switch between displaying only PNG or JPG pictures or displaying all files: `filetypes=[("Pictures", "*.png|*.PNG|*.jpg|*.JPG"), ("All files", "*")]`

okbuttontext [str] text displayed on the validate button, default is “Open”.

cancelbuttontext [str] text displayed on the button that cancels the selection, default is “Cancel”.

foldercreation [bool] enable the user to create new folders if True (default)

3.6 FileBrowser

class `tkfilebrowser.FileBrowser` (*parent, initialdir="", initialfile="", mode='openfile', multiple_selection=False, defaulttext="", title='Filebrowser', filetypes=[], okbuttontext=None, cancelbuttontext='Cancel', foldercreation=True, **kw*)

Filebrowser dialog class.

__init__ (*parent, initialdir="", initialfile="", mode='openfile', multiple_selection=False, defaulttext="", title='Filebrowser', filetypes=[], okbuttontext=None, cancelbuttontext='Cancel', foldercreation=True, **kw*)

Create a filebrowser dialog.

Arguments:

parent [Tk or Toplevel instance] parent window

title [str] the title of the filebrowser window

initialdir [str] directory whose content is initially displayed

initialfile [str] initially selected item (just the name, not the full path)

mode [str] kind of dialog: “openfile”, “opendir” or “save”

multiple_selection [bool] whether to allow multiple items selection (open modes only)

defaulttext [str (e.g. '.png')] extension added to filename if none is given (default is none)

filetypes [list [("name", "*.ext1|*.ext2|.."), ...]] only the files of given filetype will be displayed, e.g. to allow the user to switch between displaying only PNG or JPG pictures or displaying all files: `filetypes=[("Pictures", "*.png|*.PNG|*.jpg|*.JPG"), ("All files", "*")]`

okbuttontext [str] text displayed on the validate button, default is “Open”.

cancelbuttontext [str] text displayed on the button that cancels the selection, default is “Cancel”.

foldercreation [bool] enable the user to create new folders if True (default)

create_folder (*event=None*)

Create new folder in current location.

get_result ()

Return selection.

move_item (*item, index*)

Move item to index and update dark/light line alternance.

quit ()

Destroy dialog.

toggle_hidden (*event=None*)

Toggle the visibility of hidden files/folders.

toggle_path_entry (*event*)

Toggle visibility of path entry.

validate (*event=None*)

Validate selection and store it in `self.results` if valid.

4.1 tkfilebrowser 2.3.1

- Fix path bar navigation in Linux

4.2 tkfilebrowser 2.3.0

- Make package compatible with Windows
- Set initial focus on entry in save mode

4.3 tkfilebrowser 2.2.6

- No longer reset path bar when clicking on a path button
- Fix bug caused by broken links

4.4 tkfilebrowser 2.2.5

- Add compatibility with Tk < 8.6.0 (requires PIL.ImageTk)
- Add desktop icon in shortcuts
- Fix handling of spaces in bookmarks
- Fix bug due to spaces in recent file names

4.5 tkfilebrowser 2.2.4

- Fix bug in desktop folder identification

4.6 tkfilebrowser 2.2.3

- Fix `FileNotFoundError` if `initialdir` does not exist
- Add Desktop in shortcuts (if found)
- Improve filetype filtering

4.7 tkfilebrowser 2.2.2

- Fix `ValueError` in `after_cancel` with Python 3.6.5

4.8 tkfilebrowser 2.2.1

- Fix `__main__.py` for python 2

4.9 tkfilebrowser 2.2.0

- Use `babel` instead of `locale` in order not to change the locale globally
- Speed up (a little) folder content display
- Improve example: add comparison with default dialogs
- Add select all on `Ctrl+A` if multiple selection is enabled
- Disable folder creation button if the user does not have write access
- Improve extension management in save mode

4.10 tkfilebrowser 2.1.1

- Fix error if `LOCAL_PATH` does not exists or is not writable

4.11 tkfilebrowser 2.1.0

- Add compatibility with `tkinter.filedialog` keywords `'master'` and `'defaulttextension'`
- Change look of filetype selector
- Fix bugs when navigating without displaying hidden files
- Fix color alternance bug when hiding hidden files
- Fix `setup.py`

- Hide suggestion drop-down when nothing matches anymore

4.12 tkfilebrowser 2.0.0

- Change package name to tkfilebrowser to respect PEP 8
- Display error message when an issue occurs during folder creation
- Cycle only through folders with key browsing in “opendir” mode
- Complete only with folder names in “opendir” mode
- Fix bug: grey/white color alternance not always respected
- Add `__main__.py` with an example
- Add “Recent files” shortcut
- Make the text of the validate and cancel buttons customizable
- Add possibility to disable new folder creation
- Add python 2 support
- Add horizontal scrollbar

4.13 tkFileBrowser 1.1.2

- Add tooltips to display the full path of the shortcut if the mouse stays long enough over it.
- Fix bug: style of browser treeview applied to parent

4.14 tkFileBrowser 1.1.1

- Fix bug: key browsing did not work with capital letters
- Add specific icons for symlinks
- Add handling of symlinks, the real path is returned instead of the link path

4.15 tkFileBrowser 1.1.0

- Fix bug concerning the initialfile argument
- Add column sorting (by name, size, modification date)

4.16 tkFileBrowser 1.0.1

- Set default filebrowser parent to None as for the usual filedialogs and messageboxes.

4.17 tkFileBrowser 1.0.0

- Initial version

CHAPTER 5

Index

Symbols

`__init__()` (*tkfilebrowser.FileBrowser* method), [11](#)

A

`askopendirname()` (*in module tkfilebrowser*), [9](#)

`askopendirnames()` (*in module tkfilebrowser*), [9](#)

`askopenfilename()` (*in module tkfilebrowser*), [10](#)

`askopenfilenames()` (*in module tkfilebrowser*), [10](#)

`asksaveasfilename()` (*in module tkfilebrowser*),
[11](#)

C

`create_folder()` (*tkfilebrowser.FileBrowser*
method), [12](#)

F

`FileBrowser` (*class in tkfilebrowser*), [11](#)

G

`get_result()` (*tkfilebrowser.FileBrowser* method), [12](#)

M

`move_item()` (*tkfilebrowser.FileBrowser* method), [12](#)

Q

`quit()` (*tkfilebrowser.FileBrowser* method), [12](#)

T

`toggle_hidden()` (*tkfilebrowser.FileBrowser*
method), [12](#)

`toggle_path_entry()` (*tkfilebrowser.FileBrowser*
method), [12](#)

V

`validate()` (*tkfilebrowser.FileBrowser* method), [12](#)