WMTile

WMTile 1.0.0 User Manual

a window tiler for XFCE desktop environment

2024-12-04

`https://pypi.org/project/wmtile`

Carlo Alessandro Verre

`carlo.alessandro.verre@gmail.com`

Contents

1. Forewords

1.1. Introduction

WMTile is a pure-Python Linux-only free and open-source utility to be used with XFCE desktop environment and X11 graphic server protocol to reshape in seven ways the visible windows in current workspace, while minimized windows remain hidden. But if there are no visible windows, WMTile makes visible and reshapes all minimized windows.

WMTile can also create seven panel launchers for usage by mouse, and seven keyboard shortcuts for usage by keyboard.

1.2. Installation

Before installing WMTile, for instance on a Debian-derived Linux type:

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt install wmctrl xdotool x11-utils pipx
```

On other platforms you will use the specific installer instead.

Then install WMTile by typing (without `sudo`):

```
$ pipx install wmtile
$ pipx ensurepath
```

Now you can close the terminal, open another one, and run WMTile. For example, to tile all visible windows in current workspace, type:

```
$ wmtile -t
```

Later you can type:

```
$ pipx upgrade yawp
```

in order to upgrade YAWP to a later version.

2. Usage

2.1. CLI Usage

```
$ wmtile [-h] [-H] [-Y ...] [-V] [-L] [-S] [-m] [-t] [-p] [-l] [-s] [-b] [-c]
```

ARGUMENT	MEANING
-h, --help	show a help message and exit
-H, --user-manual	browse the User Manual in PDF format and exit
-Y, --pdf-browser	PDF browser used by -H, default: 'xdg-open'
-V, --version	show program's version number and exit
-L, --launchers	create 7 panel launchers
-S, --shortcuts	create 7 keyboard shortcuts
-m, --minimize	minimize visible windows
-t, --tile	reshape visible windows as tiles
-p, --portrait	reshape visible windows as portraits
-l, --landscape	reshape visible windows as landscapes
-s, --stack	reshape visible windows as a stack
-b, --maximize	maximize visible windows
-c, --close	gracefully close visible windows

You must give one and only one argument, only `-H` and `-Y` can go together. The `-Y` default value should select the system-default PDF browser, else choose your preferred one, as atril evince okular or xpdf.

2.2. Parameters

WMTile behavior is controlled by four parameters, each value is an unsigned integer indicating a number of pixels.

PARAMETER	DEFAULT
bottom_space	24
right_space	12
top_stack	20
left_stack	20

`bottom_space` and `right_space` define an additional reserved space below and to the right of each window. We need this to prevent window overlapping, because not all windows behave exactly the same way.

`top_stack` and `left_stack` only affect the Stack function ('`wmtile -s`', see 3.5.) and define howmany pixels each window goes respectively down and right, compared to the previous one. `top_stack` should be large enough to read the titles of all windows.

Default values can be altered by the content of the configuration file '`~/.config/wmtile/wmtile.cfg`' which, if exists, should contain lines with syntax:

```
[ parameter '=' integer ] [ '#' comment ]
```

For example:

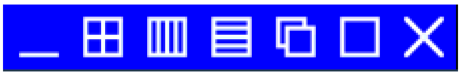
```
$ cat ~/.config/wmtile/wmtile.cfg
# WMTile configuration parameters
left_stack = 10 # 10 is more than enough
```

2.3. Mouse Usage

You can call WMTile directly from terminal, but is more convenient to use it either by mouse or by keyboard.

To use WMTile by mouse you'll create seven panel launchers by typing:

```
$ wmtile -L
Creating 7 panel launchers...
    panel launcher 'wmtile -m' not found, created
    panel launcher 'wmtile -t' not found, created
    panel launcher 'wmtile -p' not found, created
    panel launcher 'wmtile -l' not found, created
    panel launcher 'wmtile -s' not found, created
    panel launcher 'wmtile -b' not found, created
    panel launcher 'wmtile -c' not found, created
```

If you have one panel only, the seven launchers  will be appended immediately at end of your panel, while if you have more than one, you will be asked seven times which panel you want to add each launcher to.

If you run '`wmtile -L`' again, the already existing panel launchers will not be duplicated:

```
$ wmtile -L
Creating 7 panel launchers...
    panel launcher 'wmtile -m' already exists
    panel launcher 'wmtile -t' already exists
    panel launcher 'wmtile -p' already exists
    panel launcher 'wmtile -l' already exists
    panel launcher 'wmtile -s' already exists
    panel launcher 'wmtile -b' already exists
    panel launcher 'wmtile -c' already exists
```

2.4. Keyboard Usage

To use WMTile by keyboard you'll create seven keyboard shortcuts by typing:

```
$ wmtile -S
Creating 7 keyboard shortcuts...
    keyboard shortcut Alt+Shift+M --> 'wmtile -m' created
    keyboard shortcut Alt+Shift+T --> 'wmtile -t' created
    keyboard shortcut Alt+Shift+P --> 'wmtile -p' created
    keyboard shortcut Alt+Shift+L --> 'wmtile -l' created
    keyboard shortcut Alt+Shift+S --> 'wmtile -s' created
    keyboard shortcut Alt+Shift+B --> 'wmtile -b' created
    keyboard shortcut Alt+Shift+C --> 'wmtile -c' created
Please reboot in order to make the keyboard shortcuts effective.
```

If you run '`wmtile -S`' again, the new seven shortcuts silently overlap the previous ones, without duplication.

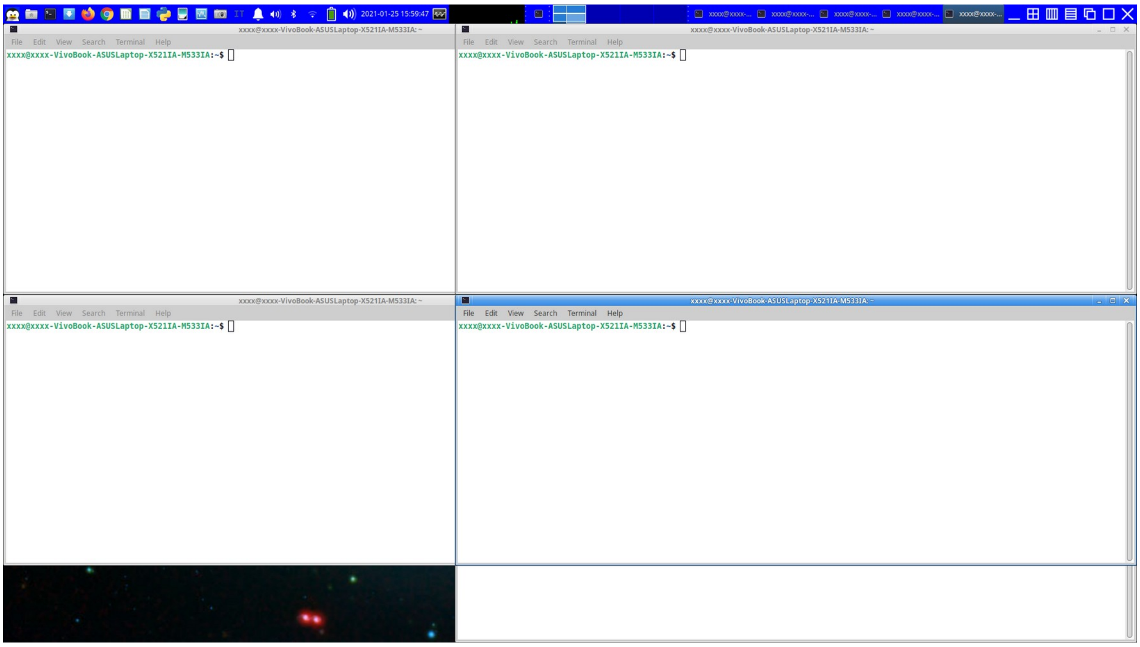
3. Functions

As said before, WMTile reshapes in seven ways the visible windows in current workspace, while minimized windows remain hidden. But if there are no visible windows, WMTile makes visible and reshapes all minimized windows.

For example, imagine you have five windows in the current desktop, three visible and two minimized:

- if you click the Tile ⊞ launcher, the three visible windows are tiled and the two minimized windows remain hidden...
- ...while if you click in sequence the Minimize ▁ launcher followed by the Tile ⊞ launcher, all five windows become visible and tiled.

Let's create five terminal windows in an empty workspace. They will appear randomly overlapping.


2021-01-25 15:59:47
xxxx@xxxx-...
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA: ~
File Edit View Search Terminal Help

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:~$
```

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:~$
```

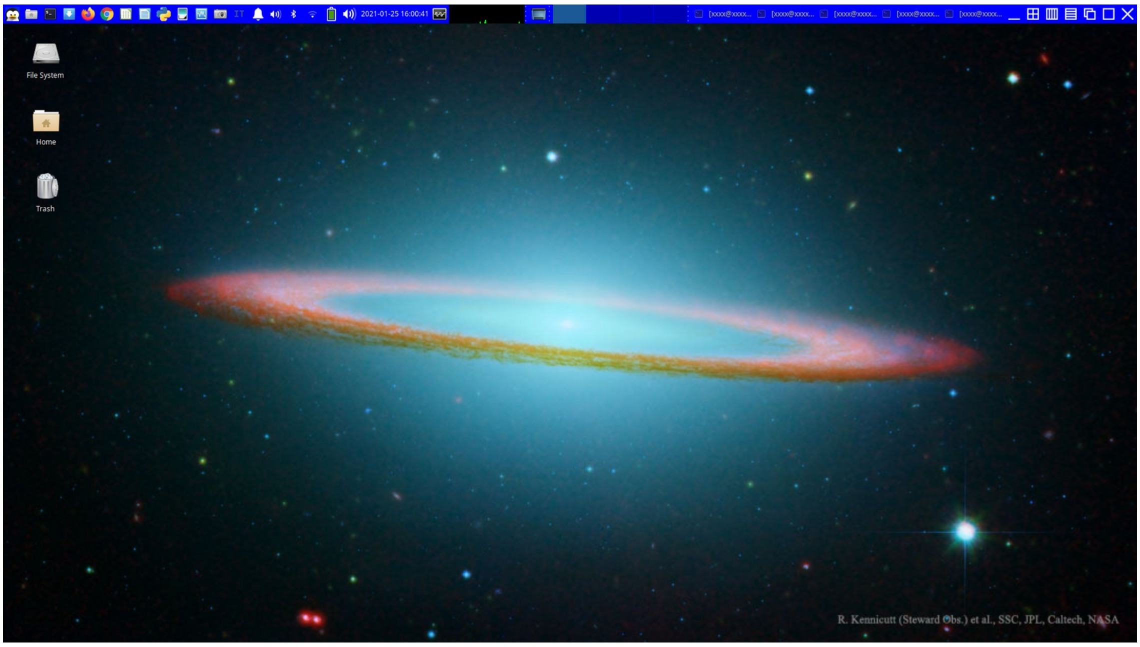
```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:~$
```

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:~$
```

Let's now see how we can reshape these windows with the mouse or the keyboard.

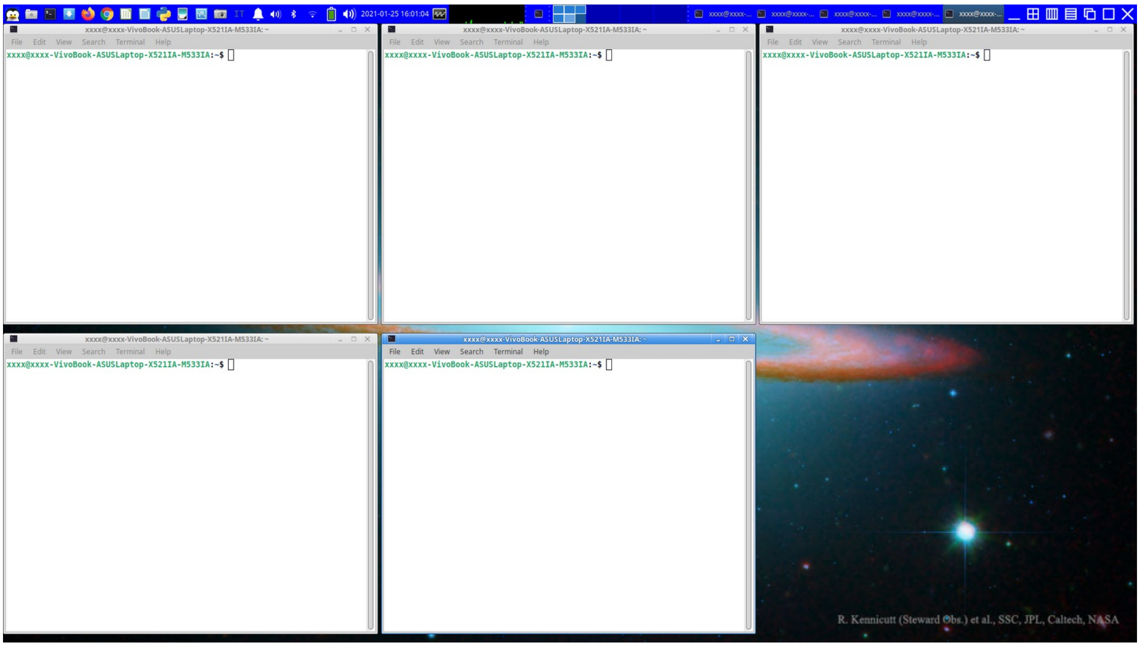
3.1. Minimize

If you left-click the Minimize panel launcher or press Alt+Shift+M, all visible windows in current workspace are minimized and disappear.



3.2. Tile

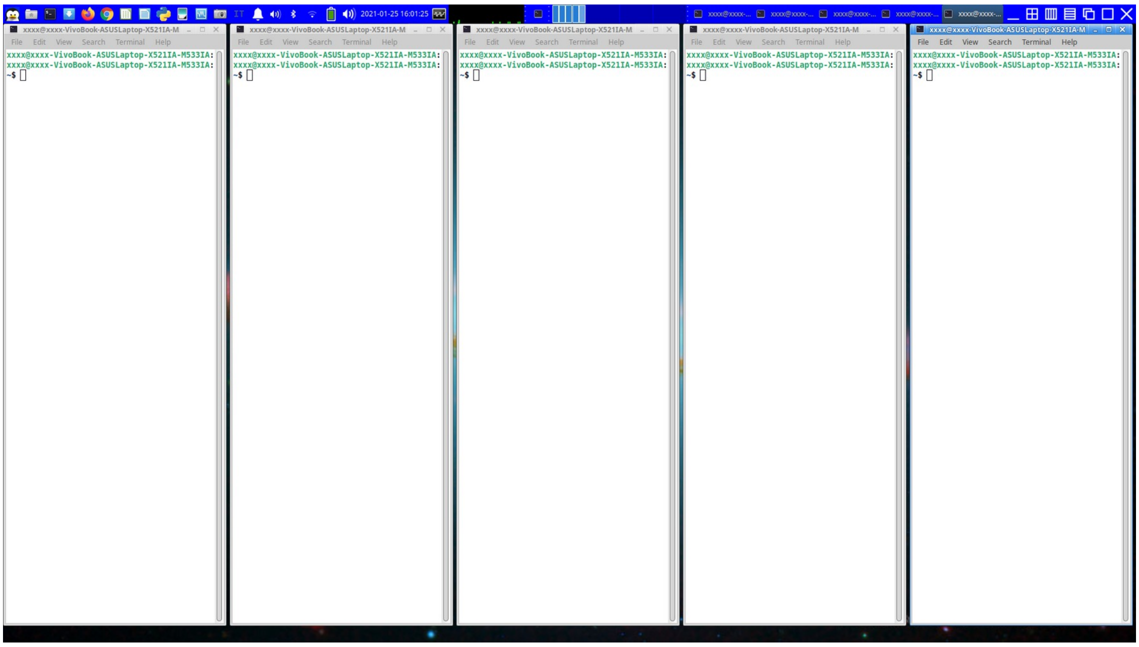
If you left-click the Tile panel launcher or press Alt+Shift+T, all visible windows (or all hidden windows if no windows are visible) in current workspace are arranged in a grid, n rows × n columns, or n rows × (n + 1) columns.


2021-01-25 16:01:04
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA: ~
File Edit View Search Terminal Help
R. Kennicutt (Steward Obs.) et al., SSC, JPL, Caltech, NASA

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:~$
```

3.3. Portrait

If you left-click the Portrait panel launcher or press Alt+Shift+P, all visible windows (or all hidden windows if no windows are visible) in current workspace are horizontally arranged as vertical portraits.


2021-01-25 16:01:25
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M
File Edit View Search Terminal Help

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:
~$
```

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:
~$
```

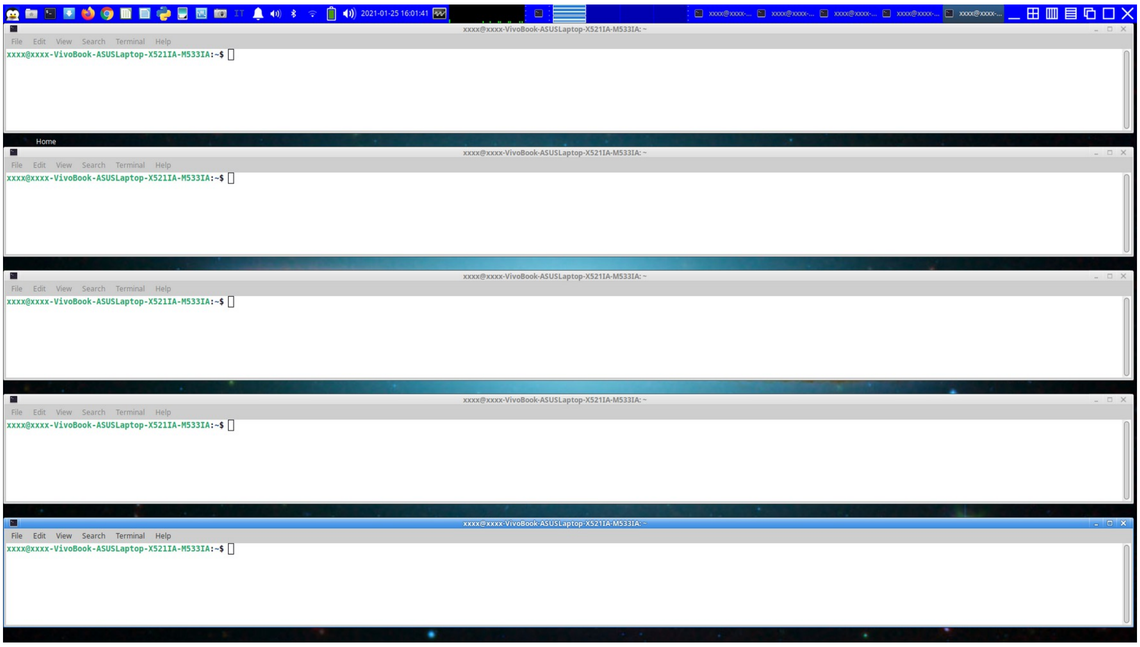
```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:
~$
```

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:
~$
```

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:
~$
```

3.4. Landscape

If you left-click the Landscape panel launcher or press Alt+Shift+L, all visible windows (or all hidden windows if no windows are visible) in current workspace are vertically arranged as horizontal landscapes.


2021-01-25 16:01:41
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA: ~
File Edit View Search Terminal Help
Home

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:~$
```

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:~$
```

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:~$
```

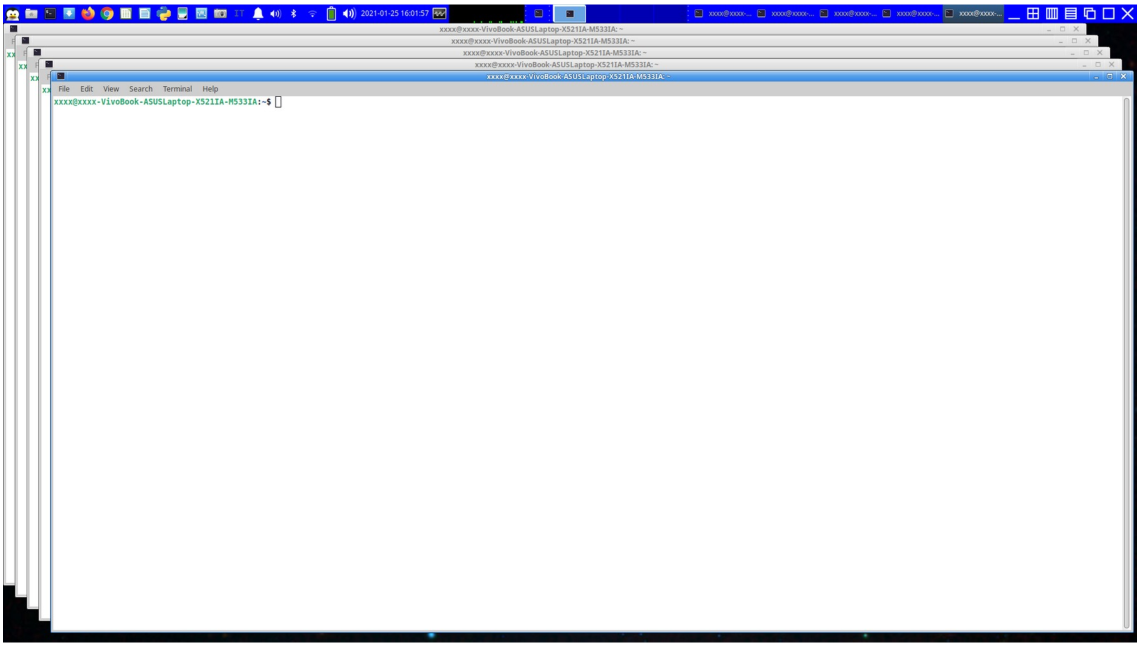
```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:~$
```

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:~$
```

3.5. Stack

If you left-click the Stack ⧉ panel launcher or press Alt+Shift+S, all visible windows (or all hidden windows if no windows are visible) in current workspace are reshaped in a stack.

Stack function is controlled by `top_stack` and `left_stack` parameters, which define howmany pixels each window goes respectively down and right, compared to the previous one. `top_stack` should be large enough to read the titles of all windows.

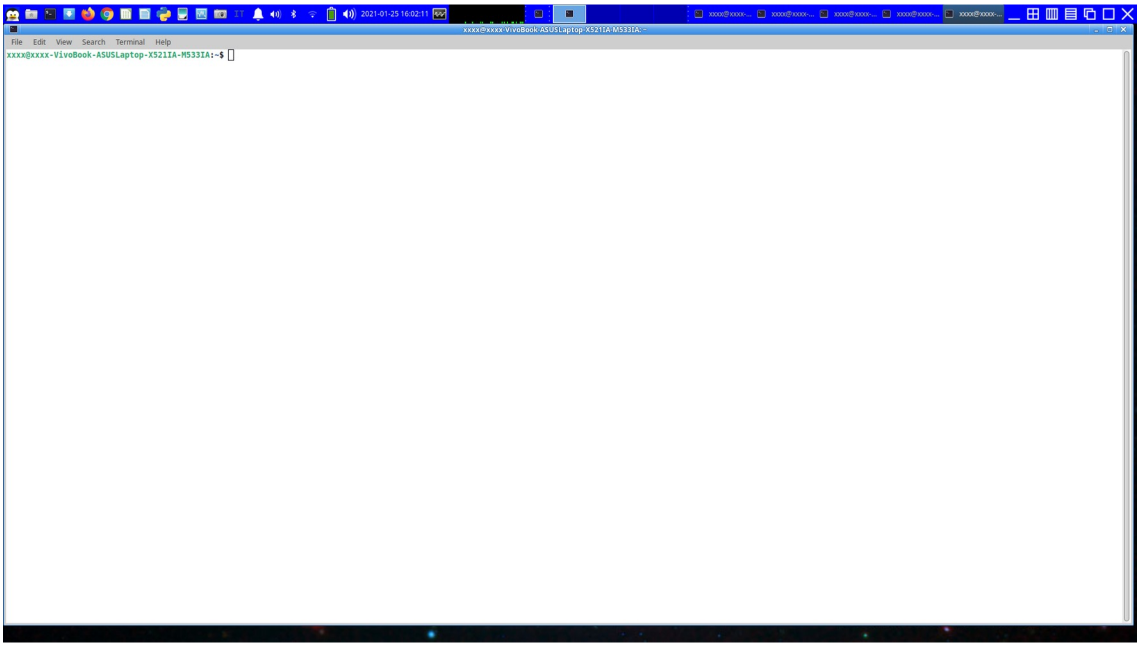

2021-01-25 16:01:57
xxxx@xxxx-...
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA: ~
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA: ~
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA: ~
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA: ~
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA: ~
File Edit View Search Terminal Help

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:~$ 
```

3.6. Maximize

If you left-click the Maximize □ panel launcher or press Alt+Shift+B, all visible windows (or all hidden windows if no windows are visible) in current workspace are maximized, so you will see the most recent one only.

Of course you can access the other windows by Alt+Tab (forward) or Alt+Shift+Tab (backward).


2021-01-25 16:02:11
xxxx@xxxx-...
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA: ~
File Edit View Search Terminal Help

```
xxxx@xxxx-VivoBook-ASUSLaptop-X521IA-M533IA:~$ 
```

3.7. Close

If you left-click the Close ☒ panel launcher or press Alt+Shift+C, all visible windows (or all hidden windows if no windows are visible) in current workspace are gracefully closed.

Here "gracefully" means that:

- if a window to close belongs to an application containing an open file, you will be asked whether to save the file before closing;
- if a window to close is a terminal containing an active process, you will be asked to confirm the window closing and the consequent process killing.

4. Afterwords

4.1. Versions

Version 1.0.0, 2024-12-04

- Changed: WMTile reshapes only visible windows (but it reshapes all hidden windows if there are no visible windows)
- Changed: creation of panel launchers does not recreates the already existing launchers, avoiding duplications
- Added: the new `-Y` argument allows to choose the PDF browser used to read this User Manual by `-H` argument
- Changed: arguments `-i, --install, -k, --keyboard` and `--big` have been renamed to `-L`, `--launchers`, `-S`, `--shortcuts` and –minimize respectively
- Changed: configuration file '`~/.config/wmtile/parameters.cfg`' has been renamed to '`~/.config/wmtile/wmtile.cfg`'
- Removed: parameters `top_margin`, `bottom_margin`, `left_margin` and `right_margin` are automatically computed and are no more accepted in configuration file
- Unchanged: panel launchers and keyboard shortcuts created by WMTile 0.9.4 are fully compatible with WMTile 1.0.0

Version 0.9.4, 2021-01-29

- Changed: some bug has been fixed

Version 0.9.3, 2021-01-29

- Changed: `README.txt` has been converted into `README.md`

Version 0.9.2, 2021-01-29

- Changed: Help and User Manual have been updated

Version 0.9.1, 2021-01-25

- Changed: User Manual has been updated

Version 0.0.1, 2020-10-14

- First version published on Pypi

4.2. Dimensions

WHAT	LINES	WORDS	CHARS
Python code User Manual	334 380	1430 1929	13848 14276
TOTAL	**714**	**3359**	**28124**

4.3. Bugs

Please report bugs and comments by an e-mail to:

`carlo.alessandro.verre@gmail.com`

4.4. Credits

The WMTile project has been:

- developed in Python 3.11.2 by IDLE 3.11.2 (`https://www.python.org`)
- built and published on PyPI by flit 3.10.1 (`https://pypi.org/project/flit`)
- on Linux Debian 12.7 (`https://www.debian.org`) with XFCE 4.18 (`https://www.xfce.org`)

WMTile uses some CLI utilities in order to interact with XFCE:

- xdotool 1:3:20160805.1-5 for window minimization and maximization
- xwininfo in x11-utils 7.7+5 for window status analysis
- wmctrl 1.07-7+b1 for any other function

This WMTile User Manual has been written in ODT format and exported into PDF format by:

- LibreOffice Writer 7.4.7.2. (`https://www.libreoffice.org`)

Panel launcher icons have been created by:

- GIMP 2.10.34 (`https://www.gimp.org`)

Thanks to Brad Whitehead for the idea of reshaping only the visible windows.

Thanks to APOD (`https://apod.nasa.gov`) for the Sombrero galaxy in infrared image used as desktop background.

4.5. Acronyms

ACRONYM	MEANING
APOD	Astronomical Picture Of the Day
CLI	Command Language Interface
DE	Desktop Environment
FOSS	Free and Open-Source Software
FSF	Free Software Foundation
GIMP	GNU Image Manipulation Program
GNU	GNU is Not Unix
GPL	General Public License
GUI	Graphic User Interface
HTML	HyperText Markup Language
IDLE	Integrated Development and Learning Environment
MD	MarkDown
ODT	Open Document Text
PDF	Page Description Format
PyPI	Python Package Index
WM	Window Manager
XFCE	X Forms Common (or Cool) Environment

4.6. License

This program is free software, you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3, or (at your option) any later version. This program is distributed in the hope that it will be useful, but without any warranty, without even the implied warranty of merchantability or fitness for a particular purpose.

See the GNU General Public License for more details. To receive a copy of the GNU General Public License, see `https://www.gimp.org` or contact Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02111-1301 USA.