

# wmtille User Guide

Version 0.9.1

<https://pypi.org/project/wmtille>

Carlo Alessandro Verre

[carlo.alessandro.verre@gmail.com](mailto:carlo.alessandro.verre@gmail.com)

january 25, 2021

## Contents

- [1. Before Usage](#)
  - [1.1. Introduction](#)
  - [1.2. Installation](#)
- [2. Usage](#)
  - [2.1. CLI Usage](#)
  - [2.2. Parameters](#)
  - [2.3. Mouse Usage](#)
  - [2.4. Keyboard Usage](#)
- [3. Functions](#)
  - [3.1. Minimize](#)
  - [3.2. Tiles](#)
  - [3.3. Portraits](#)
  - [3.4. Landscapes](#)
  - [3.5. Stack](#)
  - [3.6. Big](#)
  - [3.7. Close](#)
- [4. Appendices](#)
  - [4.1. Known Bugs](#)
  - [4.2. Acronyms](#)
  - [4.3. Credits](#)
  - [4.4. Changelog](#)

# 1. Before Usage

## 1.1. Introduction

`wmtile` is a small CLI utility written in Python 3, aimed to reshape in various manners the windows in current workspace.

Using `wmctrl` and `xdotool`, `wmtile` is compatible with the EWMH/NetWM specification, so it can work with many Window Mangers as XFCE, Enlightenment, Icewm, Kwin or Sawfish.

Under XFCE `wmtile` can install automatically the panel launchers needed for mouse usage, and the keyboard shortcuts needed for keyboard usage.

## 1.2. Installation

If you lack `pip3` command you must install it, for instance on a Debian-derived Linux type:

```
$ sudo apt-get -y install python3-pip
```

`wmtile` requires also `wmctrl` and `xdotool`, for instance on a Debian-derived Linux type:

```
$ sudo apt-get -y install wmctrl xdotool
```

Now you can install `wmtile` typing (without `sudo`):

```
$ pip3 install wmtile
```

## 2. Usage

### 2.1. CLI Usage

usage: wmtile [-h] [-H] [-V] [-i] [-k] [-m] [-t] [-p] [-l] [-s] [-b] [-c]

Always give one and only one argument.

optional arguments:

-h, --help	show this help message
-H, --user-guide	open User Guide in PDF format
-V, --version	show program's version number
-i, --launchers	install 7 panel launchers (XFCE only)
-k, --shortcuts	install 7 keyboard shortcuts (XFCE only)
-m, --minimize	Minimize all windows in current desktop
-t, --tiles	reshape as Tiles all windows in current desktop
-p, --portraits	reshape as Portraits all windows in current desktop
-l, --landscapes	reshape as Landscapes all windows in current desktop
-s, --stack	reshape as a Stack all windows in current desktop
-b, --big	reshape as Big (maximize) all windows in current desktop
-c, --close	gracefully Close all windows in current desktop

### 2.2. Parameters

wmtile behavior is tuned by eight parameters:

PARAMETER	DEFAULT
top_margin	32
bottom_margin	0
left_margin	0
right_margin	0
bottom_space	36
right_space	12
top_stack	20
left_stack	20

Any value is an unsigned integer, denoting a number of pixels.

`top_margin`, `bottom_margin`, `left_margin` and `right_margin` give the reserved space on the screen where wmtile cannot put the windows. Default for `top_margin` is 32 because we imagine to have a single horizontal panel 32 pixels high at the top of the screen.

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

`top_stack` and `left_stack` only affect the stack function (`wmtile -s`) and says howmany pixels any window goes down and right compared to the previous one. In particular `top_stack` must be sufficient to read the titles of all windows.

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

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

or "empty" lines with syntax:

```
[ '#' comment ]
```

for example:

```
# wmtile configuration parameters
left_stack = 10 # 10 is more than enough
top_margin = 36 # horizontal top panel, 36 pixels high
```

## 2.3. Mouse Usage

You can use `wmtile` directly from terminal, but is more convenient to use it either by mouse or by keyboard, depending on your tastes. For mouse usage you should create by hand seven panel launchers for the `wmtile` seven functions (minimize, tiles, portraits, landscapes, stack, big and close), but under XFCE you can invoke the automatic `wmtile` installer for panel launchers:

```
$ wmtile -i
installing 7 wmtile panel launchers
  launcher --> wmtile -m (Minimize all windows in current workspace)
  launcher --> wmtile -t (reshape as Tiles all windows in current workspace)
  launcher --> wmtile -p (reshape as Portraits all windows in current
workspace)
  launcher --> wmtile -l (reshape as Landscapes all windows in current
workspace)
  launcher --> wmtile -s (reshape as a Stack all windows in current workspace)
  launcher --> wmtile -b (reshape as Big (maximize) all windows in current
workspace)
  launcher --> wmtile -c (gracefully Close all windows in current workspace)
```



which will create the seven launchers:

If you have one panel only, the seven launchers will be added immediately at and of your panel.

If otherwise you have more than one, you will be asked seven times (sorry...) which panel you want to add each launcher to. Panels are numbered as 0, 1...

**Beware:** if you issue '`wmtile -i`' a second time, you create other seven launchers in your panel, and you will have to delete them one by one.

## 2.4. Keyboard Usage

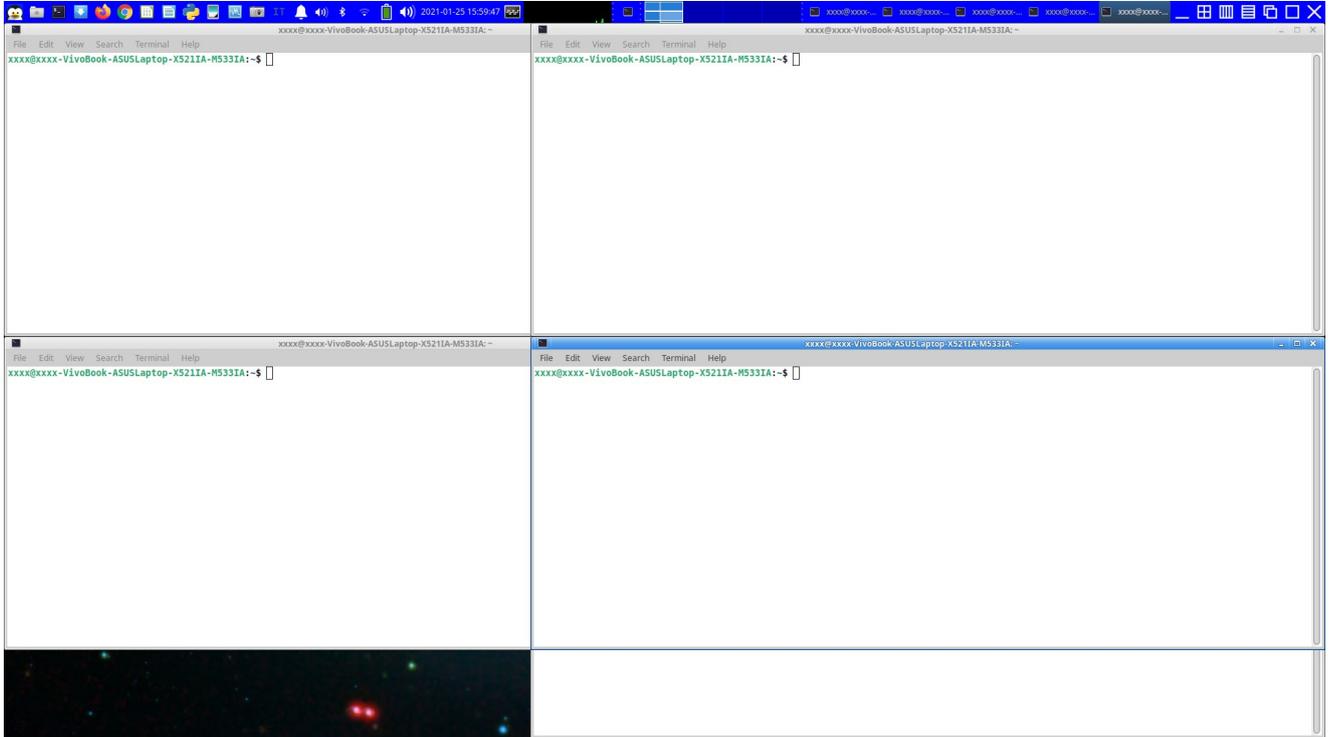
In order to use `wmtile` by keyboard, you should define by hand seven keyboard shortcuts for the `wmtile` seven functions (minimize, tiles, portraits, landscapes, stack, big and close), but under XFCE you can invoke the automatic `wmtile` installer for keyboard shortcuts:

```
$ wmtile -k
installing 7 wmtile keyboard shortcuts
  Alt+Shift+M --> wmtile -m (Minimize all windows in current workspace)
  Alt+Shift+T --> wmtile -t (reshape as Tiles all windows in current workspace)
  Alt+Shift+P --> wmtile -p (reshape as Portraits all windows in current
workspace)
  Alt+Shift+L --> wmtile -l (reshape as Landscapes all windows in current
workspace)
  Alt+Shift+S --> wmtile -s (reshape as a Stack all windows in current
workspace)
  Alt+Shift+B --> wmtile -b (reshape as Big (maximize) all windows in current
workspace)
  Alt+Shift+C --> wmtile -c (gracefully Close all windows in current workspace)
please reboot in order to make wmtile keyboard shortcuts effective
```

If you issue `'wmtilde -k'` a second time, don't worry, the new seven shortcuts overlap the previous ones, without duplication.

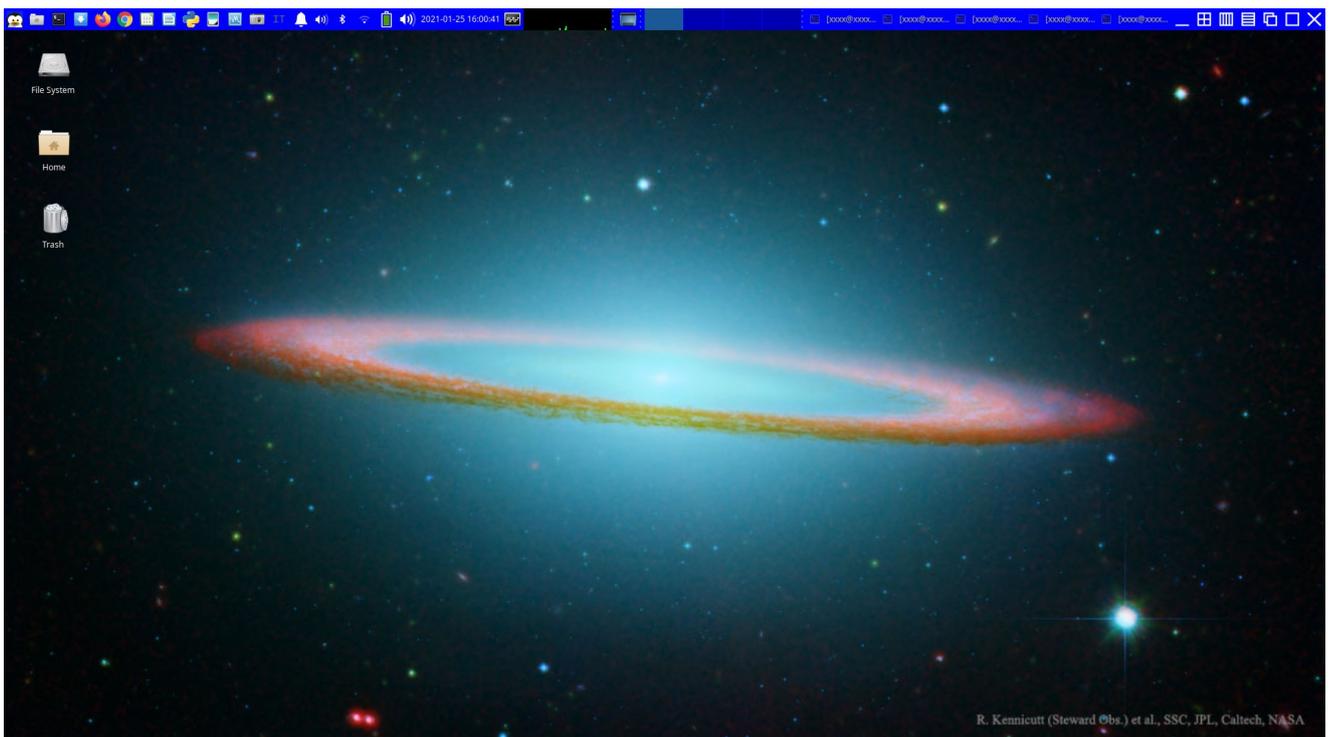
### 3. Functions

Now we'll illustrate by an example the seven `wmtille` functions. If you have more than one workspace, functions operate on all windows in the current workspace only. Let's create, say, five terminal windows, they will all appear as randomly overlapping.



#### 3.1. Minimize

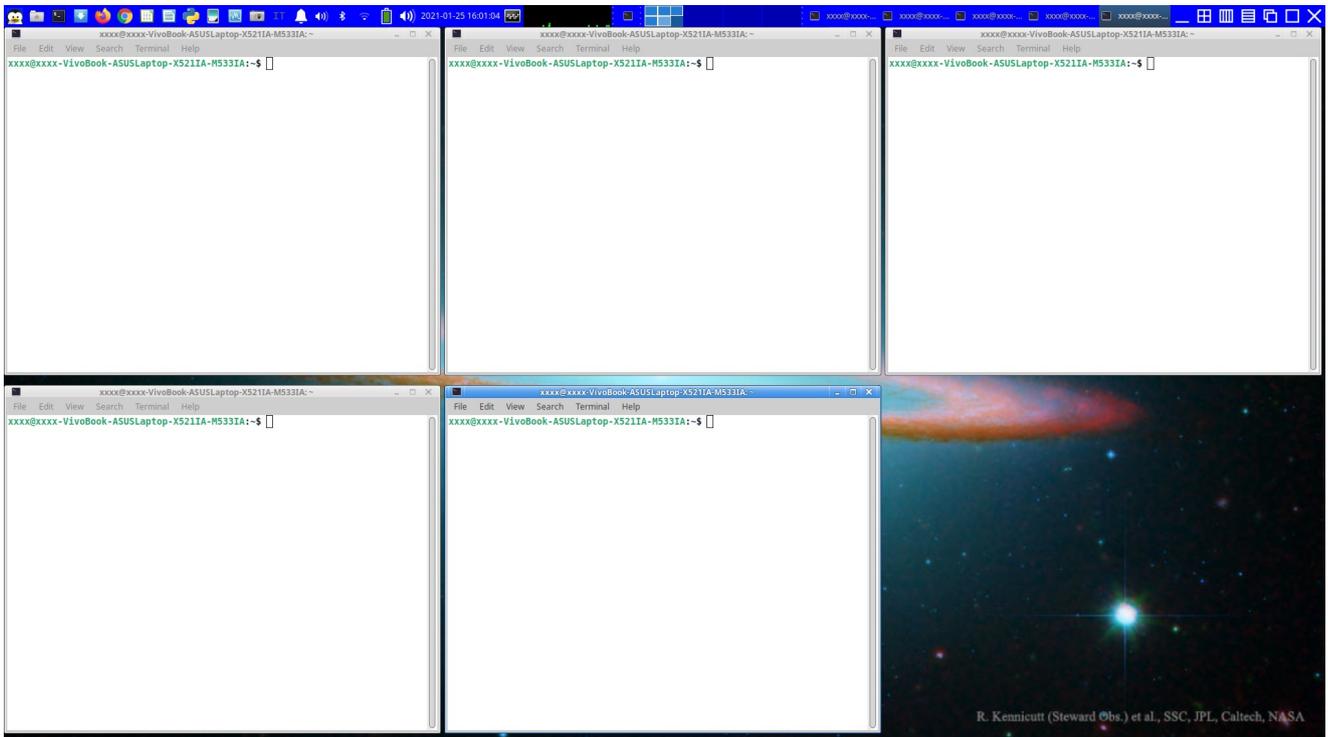
Left-clicking the Minimize panel launcher , or pressing `Alt+Shift+M`, all windows in current workspace are minimized and disappear.



### 3.2. Tiles



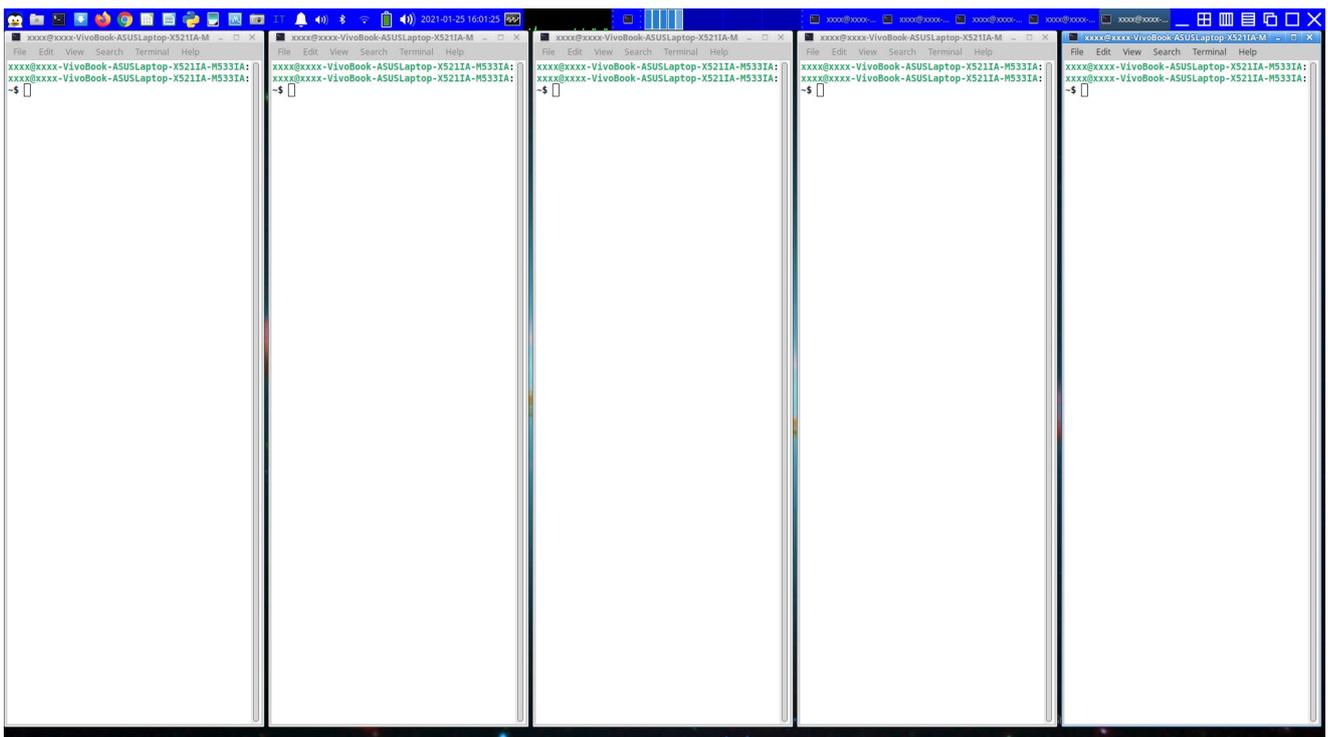
Left-clicking the Tiles panel launcher, or pressing Alt+Shift+T, all windows in current workspace are arranged in a grid,  $n * n$  or  $n * (n + 1)$ .



### 3.3. Portraits



Left-clicking the Portraits panel launcher, or pressing Alt+Shift+P, all windows in current workspace are horizontally arranged in vertical portraits.



### 3.4. Landscapes



Left-clicking the Landscapes panel launcher , or pressing Alt+Shift+L, all windows in current workspace are vertically arranged in horizontal landscapes.



### 3.5. Stack



Left-clicking the Stack panel launcher , or pressing Alt+Shift+S, all windows in current workspace are reshaped in a stack.



### 3.6. Big

Left-clicking the Big panel launcher , or pressing Alt+Shift+B, all windows in current workspace are maximized, so you will see the last window only. For the other windows press Alt-Tab or Alt-Shift-Tab.



### 3.7. Close

Left-clicking the Close panel launcher , or pressing Alt+Shift+C, all windows in current workspace are gracefully closed. Here "gracefully" means that if one of the windows to be closed belongs to an application that is dealing with an open file, then you will be asked whether to save the file.



## 4. Appendices

### 4.1. Known Bugs

For some obscure reason, LibreOffice windows refuse to be reshaped by `wmtile`.

### 4.2. Acronyms

INITIALS	MEANING
CLI	Command Language Interface
DE	Desktop Environment
EWMH	Extended Window Manager Hints
GUI	Graphic User Interface
HTML	HyperText Markup Language
MD	MarkDown
PDF	Page Description Format
PyPI	Python Package Index
WM	Window Manager
XFCE	X Forms Common (or Cool) Environment

### 4.3. Credits

The `wmtile` project has been:

- developed in [Python](#) 3.8.6
- by [Idle](#) 3.8.6
- built and published on PyPI by [flit](#) 3.0.0 on [Linux Xubuntu](#) 20.10
- tested on [Linux Xubuntu](#) 20.10 and [GhostBSD-XFCE](#) 21.01.20

`wmtile` uses two CLI utilities in order to interact with WM:

- [xdotool](#) 1:3:20160805.1-4 for window minimization (which seems impossible by `wmctrl`)
- [wmctrl](#) 1.07-7build1 for any other function

The `wmtile` User Guide file has been:

- written in MD format and exported in HTML format by [ReText](#) 7.1.0
- processed by [toc2md](#) 0.9.2 (see [toc2md](#) User Guide)
- translated from HTML into PDF format by [LibreOffice Writer](#) 7.0.3.1

Thanks to [APOD \(Astronomical Picture of the Day\)](#) for the [Sombrero Galaxy in Infrared](#) image used as desktop background.

### 4.4. Changelog

- Version **0.9.1** - 2021-01-25
  - Changed
    - User Guide has been updated
- Version **0.0.1** - 2020-10-14
  - First version published on Pypi