



# arrangeit documentation

*Release 0.3.3*

**Ivica Paleka**

Jan 10, 2020

## Contents:

<b>1</b>	<b>Installation</b>	<b>2</b>
1.1	GNU/Linux . . . . .	2
1.2	MS Windows . . . . .	3
<b>2</b>	<b>User guide</b>	<b>4</b>
2.1	Basic principles . . . . .	4
2.2	User interface . . . . .	4
2.3	Main operations . . . . .	5
2.4	Other operations . . . . .	6
2.5	Keyboard shortcuts . . . . .	6
2.6	Options . . . . .	6
<b>3</b>	<b>Development</b>	<b>6</b>
3.1	System requirements . . . . .	7
3.2	Python requirements . . . . .	8
3.3	Additional tools . . . . .	8
<b>4</b>	<b>Functional testing</b>	<b>10</b>
4.1	System requirements . . . . .	10
4.2	Memory and disk space requirements . . . . .	11
4.3	Running tests . . . . .	11
4.4	Maintenance and cleaning . . . . .	12
<b>5</b>	<b>Legal notice</b>	<b>12</b>
5.1	Copyright . . . . .	12

5.2 GNU General Public License . . . . .	12
<b>6 Source documentation</b>	<b>21</b>
6.1 <code>arrangeit</code> – Application package . . . . .	21
6.2 <code>tests</code> – Unit and integration tests . . . . .	63
<b>7 Indices and tables</b>	<b>112</b>
<b>Python Module Index</b>	<b>113</b>
<b>Index</b>	<b>114</b>

---

**arrangeit** is a cross-platform desktop utility that helps you placing your desktop’s open windows. It is a utility mostly based on the mouse movements, with some keyboard shortcuts as helpers.

Version 0.91 was the last published version of a [Win95/98 desktop utility](#) named ArrangeIt. Twenty years later, new software with the name **arrangeit** is born. It has been developed from the scratch in Python and the initial public release v0.3alpha is now [available to download](#) from the Github. Please bear in mind that this version is alpha software and it’s not suitable for production.

It should be possible to run **arrangeit v0.3alpha** under X Windows in GNU/Linux (Wayland is not supported. Yet...) and under MS Windows.

Mac OS X support is expected in the next **v0.4beta** release.

## 1 Installation

You may download **arrangeit v0.3alpha** either in the form of a binary distribution package (pick your platform and Python version from the project’s [releases](#) page on Github) or as a source package distribution.

Minimum requirements for Python is version 3.5.

### 1.1 GNU/Linux

#### Binary distribution

For now, only Debian/Ubuntu binary releases for Python 3.5, 3.6 and 3.7 can be downloaded from the [releases](#) page.

Download binaries of your choice and place them in a temporary directory. Install **arrangeit** by typing the following commands in terminal:

```
$ sudo apt-get install python3-pil.imagetk python3-xlib
$ cd tmp_directory
$ sudo dpkg -i python3-pynput_1.4.2_all.deb
$ sudo dpkg -i arrangeit_0.3alpha_all.deb
```

Run the executable with:

```
$ arrangeit
```

## Uninstallation

If you want, you may uninstall the software with:

```
$ sudo apt-get purge arrangeit
$ sudo apt-get autoremove --purge
```

Also, if you've saved some data or changed some settings from the options dialog, then you may delete every trace of that by removing the data directory:

```
$ rm -rf ~/.local/share/arrangeit
```

## Source distribution

Use the following commands in Ubuntu to prepare and run **arrangeit**:

```
# install requirements
$ sudo apt-get install python3-dev git gcc \
    libgirepository1.0-dev libcairo2-dev pkg-config gir1.2-gtk-3.0

# change current directory to the one where you keep your projects
$ cd ~/projects

# clone arrangeit repository (or you may download it as a compressed directory)
$ git clone https://github.com/ipaleka/arrangeit.git

# create directory for the virtual environments if you don't have it already
$ mkdir venvs
# create virtual environment for arrangeit
$ python3 -m venv venvs/arrangeit
# activate newly created virtual environment
$ source venvs/arrangeit

# install Python dependencies
(arrangeit) $ cd arrangeit
(arrangeit) $ python -m pip install -r requirements/linux.txt

# run arrangeit
(arrangeit) $ python -m arrangeit
```

## 1.2 MS Windows

### Binary distribution

Provided binary release downloaded from the [releases](#) page is in the form of a compressed directory.

Extract it in a directory of your choice and run the `arrangeit.exe` executable by clicking it. You may also, of course, create a shortcut to that executable and place it on the desktop or in some other place of choice.

## Uninstallation

No runtime files will be placed outside that directory during a run. If you've changed some settings or saved the data in the options dialog, then your configuration files would be saved in `arrangeit` directory under your user's directory.

If you want to delete every trace of **arrangeit** software, then you should delete that directory (typically `c:\Users\yourusername\arrangeit`) and the directory where you extracted the binary distribution.

## Source distribution

You should install [Python 3](#) first in order to run **arrangeit** from the source in MS Windows.

Then you should either download [source archive](#) and extract it in a directory of your choice or you may clone **arrangeit** repository (detailed instructions are in the [development](#) page).

You should take the following steps in order to prepare and run **arrangeit** from source distribution in MS Windows:

```
:: change current directory to the one where you keep your projects  
cd projects  
  
:: create directory for the virtual environments if you don't have it already  
mkdir venvs  
:: create virtual environment for arrangeit  
python -m venv venvs\arrangeit  
:: activate newly created virtual environment with  
venvs\arrangeit\Scripts\activate.bat  
  
:: enter the extracted source distribution directory  
(arrangeit) cd arrangeit  
  
:: install Python dependencies  
(arrangeit) python -m pip install -r requirements/windows.txt  
  
:: run arrangeit  
(arrangeit) python -m arrangeit
```

## 2 User guide

### 2.1 Basic principles

**arrangeit** collects all the available open windows and puts them in a queue for operation. Ending operation on the current window is followed by starting operation on the next window in the queue. **arrangeit v0.3alpha** ends its execution after all the windows are exhausted and it should be started again for additional operations.

### 2.2 User interface

A box with the title of currently operating window occupies the top-left position of the main window. That window's application name and icon are placed at the top-right.

The other windows in the queue are placed below the title box, in the order in which **arrangeit** will operate on them.

Workspaces boxes are placed on the right side of the listed windows, with the emphasized color for the current window workspace.

The status bar for displaying software messages is placed at the bottom of the main window.

Quit button and the button bringing options dialog (some basic program settings may be changed from there) are at the bottom-right of the main window.

## 2.3 Main operations

The default state of the program is its operational state. It means that moving the mouse cursor automatically moves the main window. That is the positioning phase of the software in which you're choosing the future position on the screen for the current window.

You switch to resizing phase by pressing the left button of the mouse. There will be no resizing phase for a fixed size window, so a left click in such a case will switch to the positioning phase of the next window in the queue.

Click the middle mouse button or press **Shift** key on your keyboard to release the mouse. That will stop the positioning/resizing phase and the other functionalities of the software will be allowed instead.

### Positioning phase

As you move your mouse you also move the arranged main window. You may set the future position of the window by pressing the left mouse button or by **Enter** key on your keyboard. The starting corner of the resizing phase can be picked either by pressing **Ctrl** key or it can be set automatically by the snapping process.

The main window will snap next to the other windows if it is moved close enough to them. From the options dialog you may choose a snap distance or you may completely turn off the snap functionality.

Press the right mouse button or **Space** on keyboard to skip the current window and start to operate on the next window in the queue.

You may reposition listed windows starting from the desired window by releasing the mouse and clicking its title in the listed window collection. If you don't want to release the mouse for the action, you may activate a window by its ordinal number from the corresponding **F** key (**F1** for the first window in the queue, **F2** for the second, etc.).

To move current window to another workspace/desktop, click that workspace in the workspace list after releasing the mouse. Press a related number on your keyboard to switch to a workspace without releasing the mouse.

When you confirm the position by pressing the left mouse button (or by pressing the **Enter** key on your keyboard), the resizing phase starts for non-fixed size window.

### Resizing phase

Your cursor is positioned at the opposite corner of starting point set in positioning phase. You are setting the ending corner of the window in resizing phase - just press the left mouse button when you are ready. You'll be switched to the next window in the queue afterward, while the previous window will be positioned and resized based on the actions you did.

Target window background image is shown as the main window background during the resizing phase. From the options dialog you may set should that image be in grayscale. you may set the size for the *blur* filter, or you may completely turn off the background image and set the background to be made from the main background color instead.



If you release mouse (mouse middle button click or **Shift** on keyboard) then the resizing phase will be canceled and recapturing the mouse starts from the positioning phase.



## 2.4 Other operations

You access other operations either by keyboard shortcuts or by releasing the mouse cursor and use it to select some other functionality beside positioning/resizing.

### Changing window properties

If the current window is a fixed size window, then there would be no resizing phase after left click - it will be immediately placed and the positioning phase for the next window in the queue will start afterward.

You may change that default behavior by clicking  in the window title box. It works the other way around too: if you click  for resizable window, then it will be just placed after positioning phase like it was a fixed size window.

Similar behavior is applicable for restored/minimized windows too: currently minimized window will be restored after resizing phase if you click , and currently restored window will be minimized if you click .

---

**Note:** In GNU/Linux if a window is pinned (visible in all workspaces) then if you change a workspace during the window operation, you will unpin that window and so make it visible in the selected workspace only.

---

## 2.5 Keyboard shortcuts

Key	Mouse counterpart	Action
Enter	left-click	confirm position
Esc	Quit button	quit program
Space	right-click	skip window
Ctrl	<i>by snapping</i>	cycle corner
Shift	middle-click	release mouse
R	resizable icon	turn on/off resizing phase
M	minimize icon	make window minimized/restored
1-9	click workspace	change workspace
F1-F12	click listed window	restart from selected window

## 2.6 Options

You may change some program settings from the options dialog started after you release the mouse and click the *Options* button at the bottom-right of the main window.

A setting value would be changed and saved immediately after selecting, but some settings changing require program restart in order to take effect.

## 3 Development

This section is about the requirements necessary to develop **arrangeit** software.

## 3.1 System requirements

### GNU/Linux

#### Ubuntu

To start **arrangeit** development on Ubuntu, you should install some system packages by issuing the following command:

```
$ sudo apt-get install python3-dev git gcc pkg-config libcairo2-dev \  
libgirepository1.0-dev gir1.2-gtk-3.0 gir1.2-wnck-3.0
```

If you are planning to build latexpdf documentation then you should install some additional packages with:

```
$ sudo apt-get install texlive texlive-latex-extra latexmk
```

#### elementaryOS 5.0 (juno)

```
$ sudo apt-get install python3-dev python3-venv git python3-tk \  
pkg-config libgirepository1.0-dev
```

#### Debian Stretch

```
$ su  
# apt-get install python3-dev python3-venv python3-tk git pkg-config \  
libcairo2-dev libgirepository1.0-dev gir1.2-gtk-3.0 gir1.2-wnck-3.0
```

#### Debian Buster

```
$ su  
# apt-get install python3-dev python3-venv python3-tk git gcc \  
pkg-config libcairo2-dev libgirepository1.0-dev
```

#### Manjaro 18.04 Xfce

```
$ sudo pacman -S gobject-introspection tk
```

### MS Windows

Official [Python 3 installer](#) and [git for Windows](#) probably represent the easiest way to start development on MS Windows.

### Mac OS X

Download the [official installer](#) and install Python 3 by executing .pkg file. Finally, run provided post-install script **Install Certificates.command**.

## 3.2 Python requirements

You should develop **arrangeit** in a dedicated virtual environment. If you don't have any other preferred way, then probably the easiest way to create a virtual environment would be **venv** integrated with Python 3.5+.

For example, if you place your projects in `projects` directory and path to `arrangeit` root directory is `/home/yourusername/projects/arrangeit` (or `c:\Users\yourusername\projects\arrangeit` on MS Windows), then you may create a directory inside `projects` directory to hold your virtual environments.

```
$ cd ~/projects
$ mkdir venvs
$ cd venvs
```

Create a new virtual environment with:

```
$ python3 -m venv arrangeit
```

The virtual environment is activated on GNU/Linux from `venvs` directory with:

```
$ source arrangeit/bin/activate
```

Or in MS Windows with:

Install the base requirements by issuing the following from the project's root directory:

```
(arrangeit) $ python -m pip install -U -r requirements/linux.txt
```

And all the necessary Python dependency packages for **arrangeit** development with:

```
(arrangeit) $ python -m pip install -U -r requirements/base_development.txt
```

## 3.3 Additional tools

### pygettext

**pygettext** is a Python wrapper for `xgettext` and it ships with Python. To prepare translation template, run the following command on Ubuntu from the project's root directory:

```
find ./arrangeit -iname "*.py" | xargs pygettext3 --verbose --extract-all \
  --default-domain=arrangeit --output-dir=./arrangeit/locale
```

Create a `language` directory inside `locale` directory, and inside that language directory create another directory with the name `LC_MESSAGES`. Finally, copy `arrangeit` translation template into that directory and rename it to `arrangeit.po`.

Here's how it was done from the project's root directory for the Croatian language:

```
$ mkdir -p arrangeit/locale/hr_HR/LC_MESSAGES
$ cp arrangeit/locale/arrangeit.pot arrangeit/locale/hr_HR/LC_MESSAGES/arrangeit.po
```

If an existing translation needs to be updated after `arrangeit.pot` has been changed, then you should update the differences with **msgmerge**:

```
$ msgmerge --update arrangeit/locale/hr_HR/LC_MESSAGES/arrangeit.po arrangeit/locale/
↪arrangeit.pot
```

After the translation is finished, compile the language file with **msgfmt**:



```
$ cd arrangeit/locale/hr_HR/LC_MESSAGES
$ msgfmt -o arrangeit.mo arrangeit.po
```

If you use **Poedit** for translation, then instead of the last command you may create compiled file by clicking the Save button in Poedit.

## black

Any code should be formatted by **black** before commit.

It should have been installed together with other development requirements (`python -m pip install -r requirements/base_development.txt`) or you may install it separately with:

```
$ python3 -m pip install black
```

Run it from the root directory by:

```
$ black arrangeit
```

## pyflakes

Install **pyflakes** linter with:

```
$ python3 -m pip install pyflakes
```

Run it from the project's root directory by:

```
$ python3 -m pyflakes arrangeit
```

## py2deb

**py2deb** is used to build GNU/Linux installation package.

Run the following command to install py2deb dependencies on Debian/Ubuntu:

```
$ sudo apt-get install dpkg-dev fakeroot lintian python3-pip
```

py2deb's dependency `pip-accel` needs `pip` version to be 7.x, so the following commands should probably be run inside a Python 3 virtual environment created for the purpose (add `--user` argument to `pip3` if you want to install py2deb system-wide):

```
$ pip3 install py2deb $ pip3 install pip-accel # it will downgrade pip to version <8.0
```

And then run the following command inside the project's root directory to create Debian installation package in `./dist/` directory:

```
$ mkdir dist
$ py2deb -r ./dist/ --no-name-prefix=arrangeit -y \
  --use-system-package=Pillow,python3-pil \
  --use-system-package=python-xlib,python3-xlib \
  --use-system-package=six,python3-six \
  .
```

## PyInstaller

PyInstaller is used to build MS Windows installation package.

*starter.py* script is created in the project's root directory for the purpose of PyInstaller's dependencies collecting. The specification file `pyinstaller.spec` in the same directory is used to produce MS Windows executable by the following call:

```
(arrangeit) $ python -OO -m PyInstaller pyinstaller.spec
```

## SonarQube

SonarQube is an open-source platform for inspection of code quality for detecting bugs, code smells, and security vulnerabilities.

### Starting server

```
$ ~/opt/repos/sonarqube-7.7/bin/linux-x86-64/sonar.sh console
```

### Starting scanner

You should add scanner executable to your PATH. For example, by adding the following line to your `~/ .bashrc`:

```
export PATH=$PATH:~/opt/repos/sonar-scanner/bin
```

To start scanning, run the scanner from the root directory of the project with:

```
$ sonar-scanner
```

For additional information read the scanner [documentation](#).

## Administration

Prepare coverage's XML report by running the following in the project's root directory:

```
(arrangeit) $ python -m pytest -v --cov-report xml:tests/unit/coverage-linux.xml --  
↳ cov=arrangeit
```

## Overview

Open your browser and point it to <http://localhost:9000>. Login as **admin/admin**.

# 4 Functional testing

## 4.1 System requirements

## VirtualBox

**arrangeit** functional testing is done inside a VirtualBox virtual machine created with Vagrant. In Ubuntu, you may install VirtualBox by issuing the following command:

```
$ sudo apt-get install virtualbox virtualbox-guest-utils \
    virtualbox-guest-x11 virtualbox-guest-dkms
```

## Vagrant

Vagrant may be downloaded from:

<https://www.vagrantup.com/downloads.html>

In Ubuntu, install downloaded package with:

```
$ sudo dpkg -i vagrant_2.2.6_x86_64.deb
```

## Ansible

You may install Ansible in ubuntu

```
$ sudo apt-get install ansible
```

Another way is installation by *pip* for the current user:

```
$ pip install ansible --upgrade --user
```

## 4.2 Memory and disk space requirements

2GB of RAM is assigned to a virtual machine in the *arrangeit Vagrantfile* located in *tests/vm* subdirectory.

A virtual machine will occupy approximately 10GB of disk space upon finished installation, together with the size of related Vagrant box/image.

So in the case of three virtual machines you should have available at least 6GB of RAM and 30GB of disk space if you want to test them all at once. For testing one virtual machine at a time you'll need 2GB of RAM and 10GB of disk space.

## 4.3 Running tests

Robot Framework functional tests for *arrangeit* will run automatically for every Vagrant virtual machine if you invoke the following command from the *tests/vm* directory:

```
$ vagrant up
```

That command will - in serial for all defined Vagrant machines - download the Vagrant box if it isn't already downloaded, install the OS in an idempotent way and finally run the Robot Framework functional tests for *arrangeit*.

Run the same command with added virtual machine name if you want to run tests for a single virtual machine:

```
$ vagrant up xfcevm
```

If the provision phase has failed or you've updated some provisioning ansible task, then you may re-initiate provisioning with:

```
$ vagrant up --provision xfcevm
```

## 4.4 Maintenance and cleaning

To update downloaded Vagrant boxes to the latest available versions, you should invoke the following command:

```
$ vagrant box update
```

You may save extra space by removing the obsolete boxes with:

```
$ vagrant box prune
```

## 5 Legal notice

### 5.1 Copyright

**arrangeit** - cross-platform desktop utility for easy windows management

Copyright © 1999-2019 Ivica Paleka

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 of the License, 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.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

### 5.2 GNU General Public License

*Version 3, 29 June 2007 Copyright © 2007 Free Software Foundation, Inc <<http://fsf.org>>*

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that

you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: **(1)** assert copyright on the software, and **(2)** offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## **TERMS AND CONDITIONS**

### **0. Definitions**

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that **(1)** displays an appropriate copyright notice, and **(2)** tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under

this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

## 1. Source Code

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that **(a)** is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and **(b)** serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## 3. Protecting Users’ Legal Rights From Anti-Circumvention Law

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

#### 4. Conveying Verbatim Copies

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- **a)** The work must carry prominent notices stating that you modified it, and giving a relevant date.
- **b)** The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- **c)** You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- **d)** If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

#### 6. Conveying Non-Source Forms

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- **a)** Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- **b)** Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either **(1)** a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically

performing this conveying of source, or **(2)** access to copy the Corresponding Source from a network server at no charge.

- **c)** Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- **d)** Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- **e)** Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either **(1)** a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or **(2)** anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they



were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- **a)** Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- **b)** Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- **c)** Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- **d)** Limiting the use for publicity purposes of names of licensors or authors of the material; or
- **e)** Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- **f)** Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## 8. Termination

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated **(a)** provisionally, unless and until the copyright holder explicitly and finally terminates your license, and **(b)** permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

## 9. Acceptance Not Required for Having Copies

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

## 10. Automatic Licensing of Downstream Recipients

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

## 11. Patents

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license **(a)** in connection with copies of the covered work conveyed by you (or copies made from those copies), or **(b)** primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

## **12. No Surrender of Others’ Freedom**

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

## **13. Use with the GNU Affero General Public License**

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

## **14. Revised Versions of this License**

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

## 15. Disclaimer of Warranty

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

## 16. Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## 17. Interpretation of Sections 15 and 16

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

*END OF TERMS AND CONDITIONS*

## How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program’s name and a brief idea of what it does.> Copyright (C) <year> <name of author>

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 of the License, 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.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WAR-
RANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under
certain conditions; type 'show c' for details.
```

The hypothetical commands *show w* and *show c* should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

## 6 Source documentation

Contents:

### 6.1 `arrangeit` – Application package

#### `arrangeit.__main__` – Main module of the `arrangeit` application

```
arrangeit.__main__.main()
```

Retrieves, instantiates and runs platform specific app.

Configures simple logger too.

#### `arrangeit.base` – Base classes holding common code for all the platforms

```
class arrangeit.base.BaseApp
```

Bases: `object`

Base App class holding common code for all the platforms.

##### Variables

- `BaseApp.controller` – object that connects data and presentation
- `BaseApp.collector` – object responsible for collecting windows data

```
activate_root(*args)
```

Method must be overridden.

```
change_setting(name="", value=None)
```

Changes provided setting name to provided value

and saves it to user settings file.

If name startswith `_` it means we want to change theme part, so it calls and returns `change_settings_color_group()`.

##### Parameters

- **name** (*str*) – setting name to save
- **value** (*int/float/str*) – setting value to save

**change\_settings\_color\_group** (*group=""*, *value=None*)

Changes values for all settings ending with provided *group*

and saves them to user settings file.

**Parameters**

- **group** (*str*) – settings group to save
- **value** (*int/float/str*) – setting value to save

**collector = None**

**controller = None**

**create\_snapping\_sources** (*for\_model*)

Returns collection of snapping rectangles grouped by workspace.

Snapping rectangle is created around window connected edge points pair with height (or width) of 2\*SNAP\_PIXELS and width (or height) of related window side. Snapping rects for all available monitors are created for each workspace.

**Parameters for\_model** (*WindowModel*) – current model

**Returns** dict (int: list of four-tuples)

**grab\_window\_screen** (*model*, *root\_wid=None*)

Method must be overridden.

**move** (*\*args*)

Method must be overridden.

**move\_and\_resize** (*\*args*)

Method must be overridden.

**move\_to\_workspace** (*\*args*)

Method must be overridden.

**rerun\_from\_window** (*wid*, *remove\_before*)

Restart positioning routine from the window with provided *wid*

without already positioned/skipped windows.

**Parameters wid** (*int*) – windows identifier

**run** ()

Collects data, prepare them for view and finally shows view application.

**run\_task** (*task*, *\*args*)

Runs provided task with provided args

**Parameters task** (*str*) – task name

**save\_default** ()

Saves collection to default filename in user's directory.

Creates application's user data directory if it not exists.

**screenshot\_cleanup** ()

Override if platform needs cleanup after screenshot is taken.

**setup\_collector** ()

Returns platform specific Collector class.

**setup\_controller ()**  
Returns platform specific Controller class.

**class** `arrangeit.base.BaseCollector`

Bases: `object`

Base Collector class holding common code for all the platforms.

**Variables** `collection` – collection of `WindowModel` instances

**add\_window (win)**  
Method must be overridden.

**check\_window (win)**  
Method must be overridden.

**collection = None**

**get\_available\_workspaces ()**  
Method must be overridden.

**get\_monitors\_rects ()**  
Method must be overridden.

**get\_smallest\_monitor\_size ()**  
Returns size of the smallest monitor.

**Returns** tuple (w,h)

**get\_windows ()**  
Method must be overridden.

**get\_workspace\_number (workspace)**  
Method must be overridden.

**get\_workspace\_number\_for\_window (win)**  
Method must be overridden.

**is\_applicable (window\_type)**  
Method must be overridden.

**is\_resizable (window\_type)**  
Method must be overridden.

**is\_restored (window\_type)**  
Method must be overridden.

**is\_valid\_state (window\_type, window\_state)**  
Method must be overridden.

**run ()**  
Populates `collection` with `WindowModel` instances

created from the windows list provided by `get_windows ()` after they are checked for compliance with `check_window ()` by calling `add_window ()`.

**Variables** `win` – current window instance/handle in the loop

**class** `arrangeit.base.BaseController (app)`

Bases: `object`

Base Controller class holding common code for all the platforms.

**Variables**

- `BaseController.app` – platform specific parent app

- *model* – model holding window data
- *BaseController.generator* – generator for retrieving model instances from collection
- *BaseController.view* – Tkinter application showing main window
- *BaseController.mouse* – class responsible for mouse events and queue
- *state* – controller’s state (LOCATE+0..3, RESIZE+0..3 or OTHER)
- *default\_size* – default root window size (width, height)
- *screenshot\_widget* – widget holding background image
- *screenshot* – screenshot image of the window model
- *BaseController.screenshot\_when\_exposed* – should wait for window exposure
- *snapping\_targets* – dictionary of snapping rectangles grouped by workspace number
- *BaseController.timer* – id of active timer

**app = None**

**apply\_snapping** (*new\_x, new\_y, sources, intersections*)

Moves cursor and sets new state and corner if snapping occurred on new side.

State and corner can change only for positioning phase, so for resizing phase this method just calls and returns *BaseMouse.move\_cursor()*.

**Parameters**

- **new\_x** (*int*) – new cursor position on x-axis
- **new\_y** (*int*) – new cursor position on y-axis
- **sources** (tuple of *Rectangle*) – four-tuple of root window snapping rectangles
- **intersections** (*tuple*) – one or two pairs of snapping rectangles that intersect

**Variables new\_state** – positioning state

**change\_position** (*x, y*)

Changes root window position to provided x and y

if snapping criteria is not satisfied.

**Parameters**

- **x** (*int*) – absolute horizontal axis mouse position in pixels
- **y** (*int*) – absolute vertical axis mouse position in pixels

**change\_setting** (*name, value*)

Calls task for changing provided settings name to provided value.

**Parameters**

- **name** (*str/int/float*) – setting name
- **value** – value to change the setting to

**change\_size** (*x, y*)

Changes root window size in regard to provided current x and y

related to model’s changed x and y if calculated size won’t be smaller than minimum and if snapping criteria isn’t satisfied.



### Parameters

- **x** (*int*) – absolute horizontal axis mouse position in pixels
- **y** (*int*) – absolute vertical axis mouse position in pixels

### Variables

- **position** – eventual position of minimum sized root
- **width** – root window calculated width
- **height** – root window calculated height
- **left** – root window calculated position on x-axis
- **top** – root window calculated position on y-axis

### **check\_current\_size** (*x, y*)

Returns True if current size in resizing phase is greater than minimum size defined in settings.

### Parameters

- **x** (*int*) – absolute horizontal axis mouse position in pixels
- **y** (*int*) – absolute vertical axis mouse position in pixels

**Returns** tuple position (int, int) or False

### **check\_mouse** ()

Runs method that corresponds to retrieved item from mouse queue.

There are only two possibilities for item type: Boolean (scroll direction) or tuple (mouse position).

Method calls itself in regular interval defined in settings.

### **check\_snapping** (*x, y*)

Snaps root window and returns True if root window intersects

with any collection window according to snapping rects in current workspace or returns False if no snapping has occurred.

Calls *apply\_snapping()* to change state and corner if snapping occurs on different corner than current state/corner.

### Parameters

- **x** (*int*) – absolute horizontal axis mouse position in pixels
- **y** (*int*) – absolute vertical axis mouse position in pixels

### Variables

- **sources** – four-tuple of root window snapping rectangles
- **intersections** – one or two pairs of snapping rectangles that intersect
- **offset** – offset for axes

**Returns** Boolean

### **check\_snapping\_state** (*sources, intersections*)

Returns new state changed by snapping or None if state should not be changed.

### Parameters

- **sources** (tuple of Rectangle) – four-tuple of root window snapping rectangles

- **intersections** (*tuple*) – one or two pairs of snapping rectangles that intersect

**Variables**

- **new\_state** – positioning state
- **index0** – position of root's first intersected snapping rectangle in sources
- **index1** – position of root's second intersected snapping rectangle in sources

**cycle\_corners** (*counter=False*)

Cycle through corners in positioning phase by changing state.

**default\_size** = None

**display\_message** (*message, permanent=False*)

Displays informational message in view's status bar.

**Variables** *message* – message to display

**generator** = None

**get\_root\_rect** (*x, y*)

Returns current root position and size calculated from provided x, y.

**Parameters**

- **x** (*int*) – current horizontal axis mouse position in pixels
- **y** (*int*) – current vertical axis mouse position in pixels

**Returns** (int, int, int, int)

**listed\_window\_activated** (*wid*)

Calls task that restarts positioning routine from provided window id not including windows prior to current model.

**Parameters** *wid* (*int*) – windows identifier

**listed\_window\_activated\_by\_digit** (*number*)

Activates listed window by its ordinal in list presented by provided number.

**Parameters** *number* (*int*) – number of 1 to 16 representing ordinal in list

**Variables** *windows* – available workspaces in view

**mainloop** ()

Tkinter main loop.

**model** = None

**mouse** = None

**mouse\_move** (*x, y*)

Moves root Tkinter window to provided mouse coordinates.

**Parameters**

- **x** (*int*) – absolute horizontal axis mouse position in pixels
- **y** (*int*) – absolute vertical axis mouse position in pixels

**mouse\_scroll** (*counter=False*)

Cycles through window corners in both directions.

**Parameters** *counter* (*Boolean*) – is scroll in counter direction

**move\_to\_corner** ()

Configures mouse pointer and moves cursor to calculated corner position.

**Variables**

- **x** – absolute horizontal axis mouse position in pixels
- **y** – absolute vertical axis mouse position in pixels

**next** (*first\_time=False, from\_workspace=None*)

Sets controller `model` attribute from the value yielded from `generator` and populates view widgets with new model data.

Sets program to be in positioning phase by setting LOCATE state. Also changes and moves cursor and root window to model's window position. Grabs and sets screenshot image of the model's window. If there are no values left in collection then saves and exits app. Switches workspace if it's changed.

**Parameters**

- **first\_time** (*Boolean*) – is method called for the very first time
- **from\_workspace** (*int*) – workspace passed if listed window activated

**Variables** `old_workspace` – old model's workspace number

**Returns** Boolean

**on\_continue** (*event*)

Restarts positioning routine.

**on\_focus** (*event*)

Calls task top activate root if Tkinter has lost focus.

**on\_key\_pressed** (*event*)

Calls method related to pressed key.

**Parameters** `event` (*Tkinter event*) – caught event

**on\_mouse\_left\_down** (*event*)

Calls `update` with current cursor position

**Parameters** `event` (*Tkinter event*) – caught event

**on\_mouse\_middle\_down** (*event*)

Switches to third state.

**Parameters** `event` (*Tkinter event*) – caught event

**on\_mouse\_right\_down** (*event*)

Skips the current model.

**Parameters** `event` (*Tkinter event*) – caught event

**on\_resizable\_change** (*event*)

Switches model resizable attribute.

**on\_restored\_change** (*event*)

Switches model restored attribute.

**place\_on\_opposite\_corner** ()

Changes and moves cursor to model windows corner opposite to positioning phase and so triggers master resizing.

**Variables**

- **left** – x-axis part of the cursor position
- **top** – y-axis part of the cursor position

**place\_on\_top\_left** ()

Moves cursor to model's top left position and setups that corner widget and cursor.

**prepare\_view** ()

Populates view's workspaces and windows list widgets.

Very first window is our main window so we skip it in listing.

**recapture\_mouse** ()

Starts mouse listener and positioning/resizing routine.

**release\_mouse** ()

Stops positioning/resizing routine and releases mouse.

**remove\_listed\_window** (*wid*)

Destroys window widget from windows list and refreshes the list afterward.

**Parameters** *wid* (*int*) – id of window that will be destroyed

**resizing\_state\_counterpart** ()

Returns resizing counterpart to current positioning state.

**run** (*generator*)

Prepares view, syncs data, starts mouse listener and enters main loop.

Calls *prepare\_view()* to create workspaces and windows list widgets. Sets generator attribute to provided generator and sets window data by calling *next()* for the first time. Calls view application startup routine to show root and calculate visible parameters. Also brings global focus to root window.

**save** ()

Runs task for saving windows collection data to default file.

**screenshot = None**

**screenshot\_when\_exposed = False**

**screenshot\_widget = None**

**set\_default\_geometry** (*root*)

Sets provided root window width and height

calculated from available width and height for screen as quarter of the smaller element. Returned width and height have 16:9 aspect ratio.

**Parameters** *root* (*tkinter.Tk* instance) – root tkinter window

**Variables**

- *width* – root width in pixels
- *height* – root height in pixels

**set\_minimum\_size** (*x*, *y*)

Sets root window size to minimum size defined in settings

and places root's top left position to (*x*, *y*).

**Parameters**

- *x* (*int*) – absolute horizontal axis mouse position in pixels

- **y** (*int*) – absolute vertical axis mouse position in pixels

**set\_screenshot ()**

Creates and places screenshot of model window as background image.

If we can't include window decoration in image then offset is returned and we place image shifted by offset amount of pixels to related axis.

**Variables** **offset** – offset (*x, y*)

**set\_timer ()**

Cancels previous timer if it exists and creates a new one.

**setup ()**

Initializes Tkinter ViewApplication with root window and self as arguments.

Creates and place screenshot widget below view frame, used to hold window image. Sets view attribute to newly created Tkinter application. Temporary hides root window. Tkinter root window from now may be accessed by `self.view.master` attribute.

**setup\_corner ()**

Configures mouse pointer and background to current corner.

**setup\_root\_window (root)**

Sets provided root window appearance common for all platforms.

**Parameters** **root** (`tkinter.Tk` instance) – root tkinter window

**shutdown ()**

Stops mouse listener, destroys Tkinter root window and exits.

**skip\_current\_window ()**

Calls `next ()` and then destroys that new window from the windows list.

**snapping\_targets = None**

**state = None**

**switch\_resizable ()**

Changes current model resizable Boolean value and updates view.

**switch\_restored ()**

Changes current model restored Boolean value and updates view.

**switch\_workspace ()**

Activates workspace and moves root window onto it.

**timer = None**

**update (x, y)**

Calls corresponding state related update method.

**Parameters**

- **x** (*int*) – current horizontal axis mouse position in pixels
- **y** (*int*) – current vertical axis mouse position in pixels

**update\_positioning (x, y)**

Updates model with provided cursor position in LOCATE state and takes action in regard to model type.

**Parameters**

- **x** (*int*) – current horizontal axis mouse position in pixels

- **y** (*int*) – current vertical axis mouse position in pixels

**update\_resizing** (*x, y*)

Updates model related to provided cursor position and current root size and calls move and resize task if window has changed.

Switches to next model anyway.

**Parameters**

- **x** (*int*) – current horizontal axis mouse position in pixels
- **y** (*int*) – current vertical axis mouse position in pixels

**Variables** **params** – rect attributes we’re going to change

**view** = None

**workspace\_activated** (*number*)

Activates workspace with number equal to provided number.

**Parameters** **number** (*int*) – our custom workspace number (screen\*1000 + workspace)

**workspace\_activated\_by\_digit** (*number*)

Activates workspace with humanized number equal to provided number.

**Parameters** **number** (*int*) – number of 1 to 9 representing workspace

**Variables** **workspaces** – available workspaces in view

**class** `arrangeit.base.BaseMouse`

Bases: `object`

Class responsible for listening and controlling system-wide mouse events.

**Variables**

- **queue** – mouse events queue
- **listener** – class as separate thread listening for mouse events
- **control** – class for retrieving and setting cursor position

**control** = None

**cursor\_position** ()

Returns current cursor position.

**Returns** (*int, int*)

**get\_item** ()

Gets next item in queue and returns it.

**Returns** (*x,y*) or `bool` or `None`

**listener** = None

**move\_cursor** (*x, y*)

Moves cursor position to a point defined by provided x and y.

**on\_move** (*x, y*)

Puts provided x and y in queue as position tuple.

NOTE: `int(x)` and `int(y)` are needed for Darwin - making a specific platform mouse module just for that is avoided.

**Parameters**

- **x** (*int*) – absolute horizontal axis mouse position in pixels
- **y** (*int*) – absolute vertical axis mouse position in pixels

**on\_scroll** (*x, y, dx, dy*)

Puts scroll direction as Boolean value in queue.

We are interested only in *dy* that holds either +1 or -1 value, so we converted that to Boolean value.

#### Parameters

- **x** (*int*) – absolute horizontal axis mouse position in pixels
- **y** (*int*) – absolute vertical axis mouse position in pixels
- **dx** (*int*) – scroll vector on x axis
- **dy** (*int*) – scroll vector on y axis

**queue = None**

**start** ()

Initializes and starts listener for move and scroll events.

**stop** ()

Stops listener by raising an exception.

### arrangeit.data – Module with classes holding visible windows data

**class** arrangeit.data.**WindowModel** (\*\**kwargs*)

Bases: object

Class holding window data.

#### Variables

- **WindowModel.wid** – window id (xid, hwnd, ...)
- **rect** – window rectangle (x, y, width, height)
- **WindowModel.resizable** – is window resizable or not
- **WindowModel.restored** – is window restored or minimized
- **WindowModel.title** – window title/caption
- **WindowModel.name** – window's application name
- **WindowModel.icon** – window's application icon
- **workspace** – virtual workspace the window is on in format 1000 \* screen + number
- **changed** – changed window rectangle (x, y, width, height)
- **changed\_ws** – changed window workspace

**changed = ()**

**changed\_h**

**changed\_w**

**changed\_ws = None**

**changed\_x**

**changed\_y**

**clear\_changed()**  
Resets changing related attributes to initial empty values.

**h**

**icon = None**

**is\_changed**  
Checks if model rect has been changed.

**Returns** Boolean

**is\_ws\_changed**  
Checks if workspace has been changed.

**Returns** Boolean

**name = None**

**rect = ()**

**resizable = None**

**restored = None**

**set\_changed(\*\*kwargs)**  
Creates changed attribute from provided arguments.

Accepts “rect” argument, individual rect element(s) as defined by Settings.WINDOW\_MODEL\_RECT\_ELEMENTS or “ws” argument. If some rect part isn’t provided then changed, respectively rect is used for valid changes or rect elements.

Resets to () if any of provided rect arguments is invalid in regard to Settings.WINDOW\_MODEL\_TYPES for “rect”. changed\_ws is reset to None in such a case.

NOTE this method needs refactoring

#### **Variables**

- **index** – argument’s index in rect tuple
- **changed** – temporary collection holding calculated values
- **new\_value** – new value for rect element

**setup(\*\*kwargs)**  
Sets model data from provided kwargs  
or sets the value to None/() if attribute isn’t provided.

**title = None**

**w**

**wid = None**

**workspace = None**

**ws**  
Shorter alias for workspace attribute.

**x**

**y**

**class** arrangeit.data.WindowsCollection  
Bases: object



Class holding visible windows collection.

**add** (*instance*)

Adds given instance to `_members` list.

Raises `ValueError` if given *instance* isn't a `WindowModel` instance.

**Parameters** *instance* (`WindowModel instance`) – window data

**clear** ()

Empties the `_members` list.

**export** ()

Prepares for saving useful data from collection.

**generator** ()

Yields the next member from `_members`.

Raises `ValueError` if given *instance* isn't a `WindowModel` instance. :returns: `WindowModel` instance

**get\_model\_by\_wid** (*wid*)

Returns window model having provided *wid* from collection.

**Parameters** *wid* (*int*) – window id (xid, hwnd, ...)

**Returns** `WindowModel` instance

**get\_windows\_list** ()

Prepares and returns list of windows ids, titles and icons.

**Returns** [(int, str, PIL. Image. Image)]

**repopulate\_for\_wid** (*wid*, *remove\_before*)

Repopulates collection starting from the window with identifier *wid*

without including models placed before provided *remove\_before*.

**Parameters**

- **wid** (*int*) – window id (xid, hwnd, ...)
- **remove\_before** – window id (xid, hwnd, ...)

**Variables**

- **start\_index** – index of model that is going to become the first
- **remove\_index** – index of first model that is not going to be removed

**size**

Returns the size of `_members` list.

**sort** ()

Sorts collection for presentation queue.

First model stays first and the others are sorted by their workspace first and then on current position. <starts from workspace number 0 when all the windows from greater workspaces numbers are exhausted.

**Variables**

- **others** – sorted list without first element
- **index** – index of first element having greater or equal workspace like first

## arrangeit.view – Module with classes and functions holding visual presentation data

**class** `arrangeit.view.CornerWidget` (*master=None, shift=0, background='red'*)

Bases: `object`

Widget holding three frames for emphasizing current corner.

### Variables

- `CornerWidget.master` – parent widget
- `shift` – cursor shift from corner in pixels
- `CornerWidget.background` – widget background color
- `length` – axes frame length (long side) in pixels
- `width` – axes frame width (small side) in pixels
- `box_size` – box frame width/height in pixels

**anchor** (*corner=0*)

Returns anchor for provided corner.

**Returns** `str`

**background** = `'red'`

**box\_size** = `8`

**get\_place\_parameters** (*corner, size\_property*)

Returns parameters for place method for given corner and size method.

**Returns** `dict`

**hide\_corner** ()

Places widget frames in provided corner.

**length** = `20`

**master** = `None`

**max\_box**

Returns box placement on axis related to shift.

**Returns** `int`

**max\_xy**

Returns placement on axis related to shift.

**Returns** `int`

**set\_corner** (*corner=0*)

Places widget frames in provided corner.

**setup\_widgets** ()

Creates all three frames and places them in default corner.

**shift** = `0`

**width** = `4`

**class** `arrangeit.view.ListedWindow` (*master=None, wid=0, title="", icon=<PIL.Image.Image image mode=RGB size=32x32>*)

Bases: `tkinter.Frame`

Tkinter frame holding window title and smaller icon.

## Variables

- *ListedWindow.master* – master widget
- *ListedWindow.wid* – window id
- *ListedWindow.title* – window title
- *ListedWindow.icon* – window’s application icon

**get\_icon\_image** (*icon*)

Returns provided icon resized and converted to format suitable for Tkinter.

**Parameters** *icon* (PIL.Image.Image) – window’s application icon

**Returns** PIL.ImageTk.PhotoImage

**icon** = <PIL.Image.Image image mode=RGB size=32x32>

**master** = None

**on\_widget\_enter** (*event*)

Highlights widget by changing foreground color.

**on\_widget\_leave** (*event*)

Resets widget foreground color.

**setup\_bindings** ()

Binds relevant events to related callback.

**setup\_widgets** ()

Creates and places all the frame’s variables and widgets.

**title** = ''

**wid** = 0

**class** arrangeit.view.**PropertyIcon** (*master=None, background='white', callback=None*)

Bases: tkinter.Label

Tkinter label holding on/off image for a property.

## Variables

- *PropertyIcon.master* – master widget
- *PropertyIcon.images* – collection of two possible images
- *PropertyIcon.colored* – collection of two highlighted images
- *PropertyIcon.background* – main background color
- *PropertyIcon.value* – current widget value (0 or 1)
- *PropertyIcon.on\_name* – image name when the property is on
- *PropertyIcon.off\_name* – image name when the property is off
- *PropertyIcon.callback* – method to call on triggered event

**background** = 'white'

**callback** = None

**colored** = {0: None, 1: None}

**images** = {0: None, 1: None}

**master** = None

```

off_name = None
on_name = None
on_widget_enter (event)
    Highlights widget by changing image and its foreground.
on_widget_leave (event)
    Resets widget image and its foreground color.
set_value (value)
    Sets label image in relation to provided value.

    Parameters value (Boolean) – is property on or not
setup_bindings ()
    Binds relevant events to related callback.
setup_widgets ()
    Configures widgets images and sets current image.
value = 1

```

```

class arrangeit.view.Resizable (master=None, background='white')
    Bases: arrangeit.view.PropertyIcon
    Widget holding resizable/non-resizable image.

```

#### Variables

- **Resizable.images** – collection of two possible images
- **Resizable.colored** – collection of two highlighted images
- **Resizable.on\_name** – image name when the property is on
- **Resizable.off\_name** – image name when the property is off

```

colored = {0: None, 1: None}
images = {0: None, 1: None}
off_name = 'move.png'
on_name = 'resize.png'

```

```

class arrangeit.view.Restored (master=None, background='white')
    Bases: arrangeit.view.PropertyIcon
    Widget holding restored/minimized image.

```

#### Variables

- **Restored.images** – collection of two possible images
- **Restored.colored** – collection of two highlighted images
- **Restored.on\_name** – image name when the property is on
- **Restored.off\_name** – image name when the property is off

```

colored = {0: None, 1: None}
images = {0: None, 1: None}
off_name = 'minimize.png'
on_name = 'restore.png'

```

```

class arrangeit.view.Statusbar (master=None)
    Bases: tkinter.Frame

    Tkinter frame showing app messages at the bottom of root window.

    Variables Statusbar.master – master widget

master = None

setup_widgets ()
    Creates and places all the widget's variables and widgets.

```

```

class arrangeit.view.Toolbar (master=None)
    Bases: tkinter.Frame

    Tkinter frame holding options and quit button.

    Variables Toolbar.master – master widget

master = None

on_options_click ()
    Creates and shows options dialog and hides root window.

setup_widgets ()
    Creates and places all the frame's variables and widgets.

```

```

class arrangeit.view.ViewApplication (master=None, controller=None)
    Bases: tkinter.Frame

    Tkinter frame showing current window from the data provided through controller.

    Variables
        • ViewApplication.master – master Tkinter window
        • ViewApplication.controller – controller object providing windows data

controller = None

get_root_wid ()
    Returns windows identifier of master/root window.

    Returns int

hide_root ()
    Hides master/root window.

master = None

reset_bindings ()
    Unbinds all relevant events and binds those for positioning routine.

setup_bindings ()
    Binds relevant events to related controller callbacks.

    bind_all method is used if possible so events can be catch in label widget. It first unbinds Button-1
    events (in case they were bound in reset_bindings ())

    NOTE master is None check exists solely because unit tests.

setup_corner ()
    Creates and places corner widget in the default corner 0.

setup_icon ()
    Sets and places icon label widget.

```

**setup\_name ()**  
Sets and places application name label widget.

**setup\_resizable ()**  
Sets and places resizable label widget.

**setup\_restored ()**  
Sets and places restored label widget.

**setup\_statusbar ()**  
Creates and places `statusbar` widget and sets corresponding variable.

**setup\_title ()**  
Sets and places title label widget.

**setup\_toolbar ()**  
Creates and places `toolbar` widget and sets corresponding variable.

**setup\_widgets ()**  
Calls all the frame's widgets creation and placement methods.

**setup\_windows ()**  
Creates and places `windows` widget and sets corresponding variable.

**setup\_workspaces ()**  
Creates and places `workspaces` widget and sets corresponding variable.

**show\_root ()**  
Shows master/root window.

**startup ()**  
Shows master and then calculates and sets now visible parameters.  
Calls `focus_set ()` so frame can trigger keyboard events.

**update\_widgets (model)**  
Updates widgets with the data from provided `WindowModel` instance.  
Tkinter needs a reference to image so we create `icon_image` reference.

**Parameters model** (`WindowModel` instance) – window data

**class** `arrangeit.view.WindowsList` (*master=None*)

Bases: `tkinter.Frame`

Tkinter frame holding titles and small icons of the windows in queue.

**Variables** `WindowsList.master` – master widget

**add\_windows (windows)**

Creates children widgets from provided windows list.

**Parameters windows** (`[(int, str, PIL.Image.Image)]`) – list of windows tuples (number, title, icon)

**clear\_list ()**

Destroys all children widgets.

**master = None**

**on\_window\_label\_button\_down (event)**

Activates window by wid carried with provided event.

**Parameters event** (`Tkinter event`) – caught event

**place\_children** ()

Place children widgets in order.

Used after the top widget is destroyed.

**place\_widget\_on\_position** (*widget*, *position*)

Configures placement and place provided widget at provided vertical position.

#### Parameters

- **widget** (*ListedWindow*) – Tkinter Frame widget
- **position** (*int*) – vertical position in master starting from top

**class** `arrangeit.view.Workspace` (*master=None*, *number=0*, *name=""*)

Bases: `tkinter.Frame`

Tkinter frame holding individual workspace widget.

#### Variables

- **`Workspace.master`** – master widget
- **`Workspace.number`** – workspace number
- **`Workspace.name`** – workspace name

**get\_humanized\_number** (*number*)

Returns workspace number without screen part and increased by 1 as systems count workspaces from 0, but users expect to be from 1.

**Parameters** **number** (*int*) – workspace number

**master** = `None`

**name** = ''

**number** = 0

**on\_widget\_enter** (*event*)

Highlights widget by changing foreground color.

**on\_widget\_leave** (*event*)

Resets widget foreground color.

**setup\_bindings** ()

Binds relevant events to related callback.

**setup\_widgets** ()

Creates and places all the frame's variables and widgets.

As systems counts workspace from 0, we increase number by 1.

**class** `arrangeit.view.WorkspacesCollection` (*master=None*)

Bases: `tkinter.Frame`

Tkinter frame holding all the available workspaces widgets.

#### Variables

- **`WorkspacesCollection.master`** – master widget
- **`active`** – currently active workspace number
- **`capacity`** – number of children workspaces

**active** = 0

**add\_workspaces** (*workspaces*)

Creates children workspaces widgets from provided list of workspaces

Creates no widget for configuration without multiple workspaces. Widgets are stacked related to their numbers from top right two positions towards bottom and then too the left, Actual workspaces are placed from left to right, then down the same orientation. Still, as a design decision, we use the same size for every configuration having less than 5 workspaces.

**Parameters** *workspaces* (*[(int, str)]*) – list of workspaces two-tuples (number, name)

**Variables** *relwidth* – workspace widget width

**master** = None

**on\_workspace\_label\_button\_down** (*event*)

Activates workspace by number carried with provided event.

**Parameters** *event* (*Tkinter event*) – caught event

**select\_active** (*number*)

Emphasizes active workspace and deemphasizes all others.

Foreground text coloured by setting SELECTED\_COLOR is used to emphasize selection, together with SELECT\_CURSOR setting.

**Parameters** *number* (*int*) – number of workspace to select

**Variables**

- *workspace* – child widget
- *color* – Tkinter color name
- *cursor* – Tkinter cursor name

`arrangeit.view.get_screenshot_widget` (*root*)

Returns Label widget that will hold screenshot image in background.

**Parameters** *root* (*tk.Tk*) – application main window

**Returns** *tk.Label*

`arrangeit.view.get_tkinter_root` ()

Initializes and returns Tkinter root window.

**Returns** *tk.Tk* window instance

## `arrangeit.options` – Module with classes and functions for options dialog

**class** `arrangeit.options.AboutDialog` (*master=None*)

Bases: `tkinter.Toplevel`

Tkinter “About” dialog window.

**Variables**

- *AboutDialog.master* – master widget
- *AboutDialog.logo* – arrangeit logo

**logo** = None

**master** = None

**on\_help\_click** ()

Opens documentation page in user’s default web browser.



**on\_releases\_click()**

Opens download releases page in user's default web browser.

**setup\_widgets()**

Creates and packs all the dialog's widgets.

```
class arrangeit.options.CheckOption (master=None, name="", change_callback=None, initial=False, label="")
```

Bases: `tkinter.Checkbutton`

Tkinter widget for showing and changing Boolean values.

#### Variables

- *CheckOption.master* – master widget
- *CheckOption.name* – setting name to change
- *CheckOption.var* – variable holding the check button value

**master = None**

**name = ''**

**on\_update\_value (\*args)**

**var = None**

```
class arrangeit.options.ColorOption (master=None, name="", change_callback=None, initial="", label="", choices=('white', 'black', 'gray', 'slategray', 'gray25', 'gray75', 'lightblue', 'blue', 'royalblue', 'cyan', 'orange', 'salmon', 'indianred', 'red', 'orchid', 'pink', 'green', 'olivedrab', 'wheat', 'khaki', 'tan', 'lightgoldenrodyellow', 'yellow'))
```

Bases: `tkinter.OptionMenu`

Tkinter widget for showing and changing Boolean values.

#### Variables

- *ColorOption.master* – master widget
- *ColorOption.name* – setting name to change
- *ColorOption.var* – variable holding the choice value

**label = None**

**master = None**

**name = ''**

**on\_update\_value (\*args)**

**var = None**

```
class arrangeit.options.FloatScaleOption (master, **kwargs)
```

Bases: `arrangeit.options.ScaleOption`

Tkinter widget for showing and changing float range settings values.

**on\_update\_value (value)**

```
class arrangeit.options.OptionsDialog (master=None)
```

Bases: `tkinter.Toplevel`

Tkinter dialog window for manipulating of user settings data.

## Variables

- `OptionsDialog.master` – master widget
- `message` – variable holding message log
- `OptionsDialog.timer` – id of active timer

**change\_setting** (*name*=", *value*=None)

Calls sontrroller's change setting method and updates message log.

Also cancels previous timer if it exists and create a new one.

### Parameters

- **name** (*str/int/float*) – setting name
- **value** – value for given name setting

**create\_frame** (*master*)

Creates and returns frame that will holds pair of widgets.

**Parameters** **master** (*Tkinter widget*) – parent widget

**create\_separator** (*master, vertical=False*)

Creates and returns default horizontal separator of vertical if argument set.

### Parameters

- **master** (*Tkinter widget*) – parent widget
- **vertical** (*Boolean*) – is separator oriented vertical instead default horizontal

**create\_widget** (*master, name, \*\*kwargs*)

Creates and returns presentation widget for setting with provided name.

### Parameters

- **master** (`ttk.LabelFrame`) – parent widget
- **name** (*str*) – setting name

**Returns** Tkinter widget instance

**master** = None

**message** = None

**on\_destroy\_options** (*event*)

Brings back root window and destroys options dialog.

**on\_save\_default** ()

Saves windows collection data to default file.

**on\_show\_about** ()

Creates and shows about dialog on top of this dialog.

**set\_timer** ()

Cancels previous timer if it exists and creates a new one.

**setup\_bindings** ()

Binds relevant events to related callback.

**setup\_files\_section** ()

Creates and packs widgets for section dealing with files.

**Returns** `ttk.LabelFrame`

**setup\_section** (*name*, *denominator=6*)  
Creates and packs widgets for section with provided name.

**Returns** `ttk.LabelFrame`

**setup\_widgets** ()  
Creates and packs all the options' widgets.

**timer = None**

**widget\_class\_from\_name** (*name*)  
Returns related widget class from provided setting name.

**Parameters** *name* (*str*) – setting name

**Returns** custom Tkinter widget instance

**class** `arrangeit.options.ScaleOption` (*master=None*, *name=""*, *change\_callback=None*, *initial=0*, *label=""*, *configs={}*)

Bases: `tkinter.Scale`

Tkinter widget for showing and changing range settings values.

#### Variables

- `ScaleOption.master` – master widget
- `ScaleOption.name` – setting name to change

**master = None**

**name = ''**

**on\_update\_value** (*value*)

**class** `arrangeit.options.ThemeOption` (*\*args*, *\*\*kwargs*)  
Bases: `arrangeit.options.ColorOption`

### `arrangeit.utils` – Module holding various utility functions

**class** `arrangeit.utils.Rectangle` (*x0*, *y0*, *x1*, *y1*)  
Bases: `tuple`

**x0**  
Alias for field number 0

**x1**  
Alias for field number 2

**y0**  
Alias for field number 1

**y1**  
Alias for field number 3

`arrangeit.utils.check_intersections` (*sources*, *targets*)  
Returns first pairs that intersects from sources and targets list of Rectangles.

Sources is either four-tuple representing whole window or two-tuple representing specific corner of the window (from first top-left clockwise to forth bottom-left).

We are interested in intersection of odd or even pairs of sources and targets. It means that `sources[0]` or `sources[2]` should intersect with `targets[n][0]` or `targets[n][2]`, respectively `sources[1]` or `sources[3]` should intersect with `targets[n][1]` or `targets[n][3]`.

So we create iterator that first cycle through all even elements pairs and then through all odd elements pairs. Stops iteration when first intersected pair is found. Returns either single pair (even or odd) or tuple of both.

#### Parameters

- **sources** (tuple of *Rectangle*) – two-tuple or four-tuple of root window snapping rectangles
- **targets** (list of *Rectangle*) – collection of other windows snapping rectangles
- **even** ((*Rectangle*, *Rectangle*)) – horizontal intersection pair or False
- **odd** ((*Rectangle*, *Rectangle*)) – vertical intersection pair or False

**Returns** *Rectangle* or (*Rectangle*,:class:*Rectangle*) or False

`arrangeit.utils.get_class(name, platform)`

Helper method for retrieving platform specific class instance

for given name and platform.

If provided platform is None then we use `platform_path()`.

If class can't be imported that means host system isn't implemented (yet...) and so we `sys.exit` with a message.

#### Parameters

- **name** (*string*) – function name suffix
- **platform** (*string* or *None*) – platform name

**Returns** class instance from the platform specific package

`arrangeit.utils.get_component_class(name, platform=None)`

Helper method for retrieving platform specific App class.

**Parameters** **platform** (*string*) – platform name

**Returns** class with provided name from the platform specific package

`arrangeit.utils.get_cursor_name(corner, with_arrow=False)`

Returns cursor name for provided corner.

#### Parameters

- **corner** (*int*) – corner number
- **with\_arrow** (*Boolean*) – indicating should cursor contain an arrow

**Returns** str

`arrangeit.utils.get_prepared_screenshot(image, blur_size=2, grayscale=False)`

Filters provided image and converts it to format suitable for Tkinter.

SCREENSHOT\_BLUR\_PIXELS defines blur depth in pixels.

#### Parameters

- **image** (*PIL.Image.Image*) – raw screenshot image
- **blur\_size** (*int*) – how many pixels in all directions will be blurred
- **grayscale** (*Boolean*) – should image be converted to grayscale

**Returns** *PIL.ImageTk.PhotoImage*

`arrangeit.utils.get_resized_image(filename, size)`

Opens and resizes image with provided filename to provided size.

### Parameters

- **filename** (*str*) – resource file name
- **size** (*tuple*) – width and height to resize image to

**Returns** `PIL.Image`

`arrangeit.utils.get_resource_path(filename)`

Returns full path to resource with provided filename.

**Parameters** **filename** (*str*) – resource file name

**Returns** `str`

`arrangeit.utils.get_snapping_sources_for_rect(rect, snap, corner=None)`

Returns snapping rectangles formatted as `Rectangle(x0,y0,x0,y0)` from provided `rect`.

Snapping rectangle is created around window connected edge points pair with height (or width) of `2*SNAP_PIXELS` and width (or height) of related window side.

All four rectangles are returned for default corner of `None`. If `corner` is provided then it returns two adjacent rectangles for related provided corner (horizontal first, vertical second) where ordinal 0 is top-left corner, with clockwise ordering to bottom-left corner which is ordinal 3.

### Parameters

- **rect** (*(int, int, int, int)*) – window defined by (x, y, width, height)
- **snap** (*int*) – snapping distance in pixels

**Returns** two or four-tuple of `Rectangle`

`arrangeit.utils.get_value_if_valid_type(value, typ)`

Returns provided value if it's of provided type

or returns `None` if it's not. If `value` is `None` then `None` is returned. If provided `value` and `typ` are collections then each element is checked.

### Parameters

- **value** (*Python type*) – value to check for type
- **typ** (*Python type*) – type to check on `value`

**Returns** `value` or `None`

`arrangeit.utils.increased_by_fraction(value, fraction)`

Helper method for increasing provided value by provided fraction.

### Parameters

- **value** (*int*) – value to increase
- **fraction** (*float*) – fraction of a whole to increase value by

**Returns** `int`

`arrangeit.utils.offset_for_intersections(rectangles, snap)`

Checks if single or both axes intersect and returns related offset(s).

### Parameters

- **rectangles** (`Rectangle` or `(Rectangle, Rectangle)`) – one or two intersecting pair of rectangles
- **snap** (*int*) – snapping value in pixels

**Returns** tuple (x,y)

`arrangeit.utils.open_image(filename, background='white', colorized=False, foreground='red')`  
Returns Pillow image instance from provided name and colorizes it if set.

Provided `black` and `white` are used for colorize filter.

#### Parameters

- **filename** (*str*) – image filename
- **background** (*str*) – image background color
- **colorized** (*Boolean*) – should return image be highlighted
- **foreground** (*str*) – image foreground color

**Returns** `PIL.Image`

`arrangeit.utils.platform_path()`  
Returns lowercased string holding platform name.

`arrangeit.utils.platform_user_data_path()`  
Retrieves platform specific user data directory path.

`arrangeit.utils.quarter_by_smaller(width, height, size=3)`  
Helper method for retrieving one-fourth (default) for given width and height  
with aspect ratio of 16:9.

Starting point for calculation is the smaller value - the presumption is that monitors could be stacked in left-to-right **or** top-to-bottom manner.

#### Parameters

- **width** (*int*) – total desktop area width
- **height** (*int*) – total desktop area height
- **size** (*int*) – window size from 1 to 4

**Variables** `denominator` – window size denominator from 6 to 3

**Returns** (int, int)

`arrangeit.utils.set_icon(widget)`  
Sets application icon to provided widget window.

**Parameters** `widget` (`tk.Toplevel` or `tk.Tk`) – Tkinter toplevel widget

## `arrangeit.settings` – Module holding program's constants and settings

**class** `arrangeit.settings.Settings`

Bases: `object`

Class holding all the program's constants and settings.

`BLANK_ICON = <PIL.Image.Image image mode=RGB size=32x32>`

`CORNER_RECT_INDEXES = [(0, 3), (0, 1), (2, 1), (2, 3)]`

`HELP_PAGE_URL = 'https://arrangeit.readthedocs.io/en/latest/userguide.html'`

`ICON_SIZE = 32`

`LOCATE = 0`

```

OTHER = 100
RELEASES_PAGE_URL = 'https://github.com/ipaleka/arrangeit/releases'
RESIZE = 10
WINDOW_MODEL_RECT_ELEMENTS = ('x', 'y', 'w', 'h')
WINDOW_MODEL_TYPES = {'icon': <class 'PIL.Image.Image'>, 'name': <class 'str'>, 'rec

classmethod color_group(group)
    Returns all the Settings members which names end with provided name.

    Parameters group (str) – setting name

    Returns list

classmethod is_setting(name, value)
    Returns True if provided name with value is valid setting.

    Parameters

    • name (str) – setting name

    • value (str/int/float) – value to check type for

    Returns Boolean

classmethod setting_type(name)
    Returns type of setting with provided name.

    Parameters name (str) – setting name

    Returns type

user_settings = {}

class arrangeit.settings.SettingsMetaclass
    Bases: type

    Meta class needed to access Settings class attributes by names.

arrangeit.settings.read_user_settings()
    Reads and returns user settings data from user home directory.

    Returns dict

arrangeit.settings.validate_user_settings()
    Reads, validates and returns dictionary of user settings.

    Returns dict {name: value}

arrangeit.darwin – Subpackage holding code specific to Mac OS platform

arrangeit.darwin.app – Module providing and running main app loop (Mac OS platform specific
code)

class arrangeit.darwin.app.App
    Bases: arrangeit.base.BaseApp

    Main app class with Mac OS specific code.

    activate_root(wid)
        TODO implement

```

**Parameters** `wid (int)` – windows id

**grab\_window\_screen** (*model, root\_wid=None*)

Grabs and returns screenshot of the window from provided model.

TODO implement

**Parameters**

- **model** (`WindowModel`) – model of the window we want screenshot from
- **root\_wid** (`int`) – root window identifier

**Returns** (`PIL.ImageTk.PhotoImage, (int, int)`)

**move** (*wid*)

TODO implement

**Parameters** `wid (int)` – windows id

**move\_and\_resize** (*wid*)

Moves and resizes window identified by provided identifier `wid`.

TODO implement

**Parameters** `wid (int)` – windows id

**Returns** `Boolean`

**move\_to\_workspace** (*wid, number*)

TODO implement

**Parameters**

- **wid** (`int`) – root id got from Tkinter
- **number** (`int`) – our custom workspace number

**arrangeit.darwin.collector** – Module responsible for collecting windows (Mac OS platform specific code)

**class** `arrangeit.darwin.collector.Collector`

Bases: `arrangeit.base.BaseCollector`

Collecting windows class with Mac OS specific code.

**add\_window** (*win*)

Creates `WindowModel` instance from provided `win` and adds it to collection.

**Parameters** `win (dict)` – window object

**check\_window** (*win*)

Checks does window qualify to be collected

by checking window type applicability with `is_applicable()` and its current state validity with `is_valid_state()`.

**Parameters** `win (dict)` – window object

**Returns** `Boolean`

**get\_application\_name** (*win*)

Returns application/owner name for the provided `win`.

**Parameters** `win (dict)` – window object



**Returns** str

**get\_available\_workspaces** ()

TODO implement

**Returns** list

**get\_monitors\_rects** ()

Returns list of available monitors position and size rectangles.

**Returns** list [(x,y,w,h)]

**get\_windows** ()

Returns list of all windows as dictionary objects

**Returns** list

**get\_workspace\_number\_for\_window** (*win*)

TODO implement

**Parameters** *win* (*dict*) – window object

**Returns** str

**is\_applicable** (*win*)

Checks if provided win represents window that should be collected.

TODO implement

**Parameters** *win* (*dict*) – window object

**Returns** Boolean

**is\_resizable** (*win*)

TODO implement

**Parameters** *win* (*dict*) – window object

**Returns** Boolean

**is\_restored** (*win*)

TODO implement

**Parameters** *win* (*dict*) – window object

**Returns** Boolean

**is\_valid\_state** (*win*)

Checks if provided win is window with valid state for collecting.

TODO implement

**Parameters** *win* (*dict*) – window object

**Returns** Boolean

**arrangeit.darwin.controller** – Module responsible for connecting data and view (Mac OS platform specific code)

**class** `arrangeit.darwin.controller.Controller` (*app*)

Bases: `arrangeit.base.BaseController`

Controller class with Mac OS specific code.

## arrangeit.darwin.utils – Mac OS specific utility functions

`arrangeit.darwin.utils.user_data_path()`  
Returns Mac OS X specific path for saving user's data.

## arrangeit.linux – Subpackage holding code specific to GNU/Linux platform

### arrangeit.linux.app – Module providing and running main app loop (GNU/Linux platform specific code)

**class** `arrangeit.linux.app.App`

Bases: `arrangeit.base.BaseApp`

Main app class with GNU/Linux specific code.

**activate\_root** (*wid*)

Activates/focuses root window identified by provided *wid*.

**Parameters** *wid* (*int*) – windows id

**grab\_window\_screen** (*model*, *root\_wid=None*)

Grabs and returns screenshot of the window from provided *model*.

We can't include window decoration in image so offset in pixels for both axes is returned.

#### Parameters

- **model** (`WindowModel`) – model of the window we want screenshot from
- **root\_wid** (*int*) – root window identifier - not needed for GNU/Linux

#### Variables

- **window** – model window instance
- **pixbuf** – X11 pixbuf image
- **width** – window width in pixels without window manager decoration
- **height** – window height in pixels without window manager decoration

**Returns** (`PIL.ImageTk.PhotoImage`, (*int*, *int*))

**move** (*wid*)

Just calls `move_and_resize()` as the same method moves and resizes

in `Wnck.Window` class under GNU/Linux.

**Parameters** *wid* (*int*) – windows id

**Returns** Boolean

**move\_and\_resize** (*wid*)

Moves and resizes window identified by provided *wid*.

Gravity stays the same (`Wnck.WindowGravity.CURRENT`) and the other arguments are calculated/retrieved from *model* where `changed` attribute holds needed data.

If returned `mask` is `False` then we don't need to do anything more.

**Parameters** *wid* (*int*) – windows id

#### Variables

- **model** – window data

- **mask** – combination of bits holding information what is changed
- **win** – window instance

**Returns** Boolean

**move\_to\_workspace** (*wid, number*)

Moves root window to provided custom workspace number.

Calls `_move_window_to_workspace()` with `wid` increased by 1.

**Parameters**

- **wid** (*int*) – root id got from Tkinter
- **number** (*int*) – our custom workspace number

**arrangeit.linux.collector** – Module responsible for collecting windows (GNU/Linux platform specific code)

**class** `arrangeit.linux.collector.Collector`

Bases: `arrangeit.base.BaseCollector`

Collecting windows class with GNU/Linux specific code.

**add\_window** (*win*)

Creates `WindowModel` instance from provided `win` and adds it to collection.

**Parameters** `win` (`Wnck.Window` object) – window to create `WindowModel` from it

**check\_window** (*win*)

Checks does window qualify to be collected

by checking window type applicability with `is_applicable()` and its state validity for the type with `is_valid_state()`.

**Parameters** `win` (`Wnck.Window` object) – window instance to check

**Variables**

- **window\_type** – window type
- **window\_state** – window state

**Returns** Boolean

**get\_available\_workspaces** ()

Returns custom list of workspaces available on default screen.

Returned list contains two-tuples of calculated workspace number and corresponding name.

**Variables** `workspaces` – workspaces collection

**Returns** [(int, str)]

**get\_image\_from\_pixbuf** (*pixbuf*)

Returns PIL image converted from provided `pixbuf`.

<https://gist.github.com/mozbugbox/10cd35b2872628246140>

**Returns** `PIL`. Image instance

**get\_monitors\_rects** ()

Returns list of available monitors position and size rectangles.

**Variables**

- **display** – default display
- **area** – monitor working area rect

**Returns** list [(x,y,w,h)]

**get\_window\_by\_wid** (*wid*)

Returns window instance having provided wid.

**Parameters** **wid** (*int*) – window id

**Returns** Wnck.Window object

**get\_window\_move\_resize\_mask** (*model*)

Returns flag indicating what is changed when we move/resize window.

Calls recursive method traversing all rect parts. Returned flag is combination of the X, Y, WIDTH and HEIGHT bits.

**Parameters** **model** (WindowModel instance) – model holding window data

**Returns** flag

**get\_windows** ()

Returns windows list from the Wnck.Screen object.

**Variables** **screen** – provides all the windows instances

**Returns** list of Wnck.Window instances

**get\_wnck\_workspace\_for\_custom\_number** (*number*)

Returns Wnck.Workspace instance from provided custom number.

**Variables** *number* – our custom workspace number

**get\_workspace\_number** (*workspace*)

Returns integer containing screen and workspace numbers of the workspace.

In returned integer screen number represents thousands part, and workspace number represents remainder of division by 1000.

**Parameters** **workspace** (Wnck.workspace) – workspace instance

**Returns** int

**get\_workspace\_number\_for\_window** (*win*)

Returns workspace number for the provided window.

**Parameters** **win** (Wnck.Window) – window instance

**Returns** int

**is\_applicable** (*window\_type*)

Checks if provided window\_type qualifies window for collecting.

**Parameters** **window\_type** (Wnck.WindowType int flag) – type of window

**Returns** Boolean

**is\_resizable** (*window\_type*)

Checks if provided window\_type implies that window is resizable.

**Parameters** **window\_type** (Wnck.WindowType int flag) – type of window

**Returns** Boolean

**is\_restored** (*win*)

Checks if provided win is not minimized.

**Parameters** `win` (`Wnck.Window` object) – window instance to check

**Returns** Boolean

**`is_valid_state`** (`window_type`, `window_state`)

Checks if `window_state` for `window_type` qualifies window to collect.

**Parameters**

- **`window_type`** (`Wnck.WindowType` *int flag*) – type of window
- **`window_state`** (`Wnck.WindowState` *int flag*) – current state of window

**Returns** Boolean

**`arrangeit.linux.controller` – Module responsible for connecting data and view (GNU/Linux platform specific code)**

**class** `arrangeit.linux.controller.Controller` (*app*)

Bases: `arrangeit.base.BaseController`

Controller class with GNU/Linux specific code.

**`setup_root_window`** (*root*)

Sets provided root appearance attributes specific for GNU/Linux.

**`arrangeit.linux.utils` – GNU/Linux specific utility functions**

`arrangeit.linux.utils.user_data_path` ()

Returns GNU/Linux platform specific path for saving user's data.

It first try with `.local/share` in user home directory, and if there's no such directory returns `.arrangeit` directory in user home directory.

**Returns** str path

**`arrangeit.windows` – Subpackage holding code specific to MS Windows platform**

**`arrangeit.windows.app` – Module providing and running main app loop (MS Windows platform specific code)**

**class** `arrangeit.windows.app.App`

Bases: `arrangeit.base.BaseApp`

Main app class with MS Windows specific code.

**`activate_root`** (*hwnd*)

Activates/focuses root window identified by provided `hwnd`.

**`grab_window_screen`** (*model*, *root\_wid=None*)

Setups and returns screenshot of the window from provided `model`.

If DWM composition settings allows then surface of `model` window is taken from root window after thumbnails are created in it.

TODO check why this (-1, -1) fits

**Parameters**

- **model** (*WindowModel*) – model of the window we want screenshot from
- **root\_wid** (*int*) – root window identifier

**Returns** (*PIL. ImageTk. PhotoImage*, (*int*, *int*))

**move** (*hwnd*)

Just calls *move\_and\_resize()* as the same method moves and resizes under MS Windows.

**Parameters** **hwnd** (*int*) – windows id

**move\_and\_resize** (*hwnd*)

Moves and resizes window identified by provided hwnd.

**Parameters** **hwnd** (*int*) – root id got from Tkinter

**Variables** *model* – collected window data

**Returns** Boolean

**move\_other\_to\_workspace** (*hwnd*, *number*)

Moves other process' window to provided workspace number.

**Parameters**

- **hwnd** (*int*) – identifier of the window to move
- **number** (*int*) – workspace number

**move\_to\_workspace** (*hwnd*, *number*)

Moves root window to provided workspace number.

**Parameters**

- **hwnd** (*int*) – root id got from Tkinter
- **number** (*int*) – workspace number

**screenshot\_cleanup** (*\*args*)

Unregisters DWM thumbnails kept in instance's *thumbnails* attribute.

**Variables** **thumbnail** – DWM thumbnail identifier

**thumbnails** = ()

**arrangeit.windows.api** – Module providing helper class and functions for API calls (MS Windows platform specific code)

**class** *arrangeit.windows.api.Api*

Bases: *object*

Helper class for calls to Windows API.

**Variables**

- **packages** – cached collection of packages distincted by windows handles
- **helpers** – object holding helper methods for Windows API functions
- **Api.vdi** – object holding methods of virtual desktop interface

**enum\_windows** (*hwnd=None, enum\_children=False*)

Helper function to enumerate either desktop windows or children windows for window identified by provided hwnd.

**Parameters**

- **hwnd** (*int*) – window id
- **enum\_children** (*Boolean*) – should children windows be enumerated

**Returns** list

**executable\_name\_for\_hwnd** (*hwnd*)

Returns name of the executable associated with provided window identifier.

**Parameters** **hwnd** (*int*) – window handle

**Variables**

- **pid** – process identifier
- **hprocess** – process handle
- **path\_buffer** – buffer holding executable path
- **ret\_val** – function returned value indicating success for value > 0

**Returns** str

**extended\_frame\_rect** (*hwnd*)

Helper function to return DWM frame rect for window with provided hwnd.

**Parameters** **hwnd** (*int*) – window id

**Variables**

- **winrect** – area of window extended bounds
- **ret\_val** – function returned value indicating success for value > 0

**Returns** int

**get\_ancestor\_by\_type** (*hwnd, ancestor\_type*)

Helper function to return hwnd of ancestor window of window with given hwnd.

**Parameters**

- **hwnd** (*int*) – window id
- **ancestor\_type** (*int*) – window ancestor type

**Returns** int

**get\_desktop\_ordinal\_for\_window** (*hwnd*)

Returns corresponding desktop ordinal of the window with provided hwnd.

**Parameters** **hwnd** (*int*) – window id

**Returns** int

**get\_desktops** ()

Returns list of virtual desktops.

Returned list contains two-tuples of desktop numbers in order and their corresponding names. A name is formatted from “Desktop ” translation followed by ordinal increased by 1.

**Returns** [(int, str)]

**get\_last\_active\_popup** (*hwnd*)

Helper function to return hwnd of last popup of window with provided hwnd.

**Parameters** *hwnd* (*int*) – window id

**Returns** int

**get\_package** (*hwnd*)

Returns *Package* holding needed package data from provided window id.

**Parameters** *hwnd* (*int*) – window id

**Variables**

- **full\_name** – buffer holding package full name
- **package\_info\_reference** – reference to package info structure pointer
- **package\_info\_buffer** – buffer holding reference to package info structure
- **package\_info** – structure holding package data

**Returns** *Package*

**helpers = None**

**is\_cloaked** (*hwnd*)

Returns True if window with provided hwnd is cloaked/hidden.

False is returned for Windows 7 and earlier versions (helper method returns error value).

If DWM value confirms cloaked state, then the value from a call to `is_window_in_current_desktop` is returned as all the windows from the other desktops are presented as *cloaked*, so it is implied they are really cloaked. NOTE this behaviour needs additional testing

**Parameters** *hwnd* (*int*) – window id

**Variables**

- **cloaked** – flag holding non-zero value if window is cloaked
- **ret\_val** – function returned value indicating error/success status

**Returns** Boolean

**is\_dwm\_composition\_enabled** ()

Helper function returning True if DWM composition is enabled in system.

**Variables** **enabled** – composition enabled or not value

**Returns** Boolean

**move\_other\_window\_to\_desktop** (*hwnd*, *number*)

Moves other window with provided hwnd to desktop with provided ordinal.

**Parameters**

- **hwnd** (*int*) – window id
- **number** (*int*) – desktop ordinal

**Returns** int

**move\_own\_window\_to\_desktop** (*hwnd*, *number*)

Moves own/root window with provided hwnd to desktop with provided ordinal.

**Parameters**



- **hwnd** (*int*) – window id
- **number** (*int*) – desktop ordinal

**Returns** *int*

**packages** = {}

**setup\_thumbnail** (*from\_hwnd, root\_hwnd, rectangle*)

Create, updates and returns handle of thumbnail of provided source window created in root window.

**Parameters**

- **from\_hwnd** (*int*) – identifier of window to make thumbnail of
- **root\_hwnd** (*int*) – identifier of root window to make thumbnail in
- **rectangle** (*arrangeit.utils.Rectangle*) – area occupied by thumbnail

**Variables**

- **thumbnail\_id** – id of created thumbnail
- **ret\_val** – function returned value indicating error/success status

**Returns** *ctypes.wintypes.HANDLE*

**title\_info\_state** (*hwnd, state*)

Helper function to return title bar info state for window with provided hwnd.

**Parameters**

- **hwnd** (*int*) – window id
- **state** (*int*) – title bar info state type

**Variables**

- **title\_info** – title bar information structure
- **success** – value indicating is call successful

**Returns** *int*

**unregister\_thumbnail** (*thumbnail\_id*)

Unregisters thumbnail with provided identifier.

**Parameters** **thumbnail\_id** (*ctypes.wintypes.HANDLE*) – identifier of thumbnail to unregister

**Variables** **ret\_val** – function returned value indicating error/success status

**Returns** *ctypes.wintypes.HANDLE*

**vdi** = None

**window\_info\_extended\_style** (*hwnd, style*)

Helper function to return extended window style for window with given hwnd.

**Parameters**

- **hwnd** (*int*) – window id
- **style** (*int*) – extended window style type

**Variables**

- **window\_info** – window information structure

- **success** – value indicating is call successful

**Returns** int

**class** `arrangeit.windows.api.DummyVirtualDesktops`

Bases: `object`

Helper class for systems that don't support virtual desktops.

**get\_desktops** (*refresh=False*)

Returns list with single two-tuple of 0 and empty string.

**get\_window\_desktop** (*hwnd, refresh=False*)

Returns two-tuple of 0 and empty string.

**is\_window\_in\_current\_desktop** (*hwnd*)

Just returns True.

**move\_other\_window\_to\_desktop** (*hwnd, desktop\_ordinal*)

Just returns None.

**move\_own\_window\_to\_desktop** (*hwnd, desktop\_ordinal*)

Just returns None.

**class** `arrangeit.windows.api.Helpers`

Bases: `object`

Helper class for calls to WinDLL API.

**class** `arrangeit.windows.api.Package` (*path=""*)

Bases: `object`

Helper class for calls to Windows API.

**Variables**

- **path** – filesystem path to package directory
- **app\_name** – name of package's first application
- **Package.icon** – application icon

**app\_name** = ''

**icon** = <PIL.Image.Image image mode=RGB size=32x32>

**path** = ''

**setup\_package** ()

Retrieves and sets package data.

TODO add call to this method after window is exposed if it was minimized

**Variables** **root** – root element of XML document

`arrangeit.windows.api.platform_supports_packages` ()

Returns Boolean indicating if Windows version supports packages.

**Variables** **version** – platform version data

**Returns** Boolean

`arrangeit.windows.api.platform_supports_virtual_desktops` ()

Returns Boolean indicating if Windows version supports virtual desktops.

**Variables** **version** – platform version data

**Returns** Boolean

**arrangeit.windows.collector** – Module responsible for collecting windows (MS Windows platform specific code)

**class** `arrangeit.windows.collector.Collector`

Bases: `arrangeit.base.BaseCollector`

Collecting windows class with MS Windows specific code.

**add\_window** (*hwnd*)

Creates WindowModel instance from provided hwnd and adds it to collection.

**Parameters** *hwnd* (*int*) – window id

**check\_window** (*hwnd*)

Checks does window qualify to be collected

by checking window type applicability with `is_applicable()` and its current state validity with `is_valid_state()`.

**Parameters** *hwnd* (*int*) – window id

**Returns** Boolean

**get\_application\_name** (*hwnd*)

Returns application name for the window represented by provided handle.

For Windows versions greater than 8.1 it uses package `app_name` if there's cached package for provided `hwnd`.

Otherwise it tries to extract the name from executable path.

If previous methods haven't succeed it returns window's class name.

**Parameters**

- *hwnd* (*int*) – window id
- *app\_name* (*str*) – executable name without extension

**Returns** *str*

**get\_available\_workspaces** ()

Returns custom list of workspaces available on default screen.

Returned list contains two-tuples of workspace number in order and corresponding name.

**Returns** [(*int*, *str*)]

**get\_monitors\_rects** ()

Returns list of available monitors position and size rectangles.

**Returns** list [(*x*,*y*,*w*,*h*)]

**get\_windows** ()

Creates and returns list of all the windows handles

**Returns** list of integers

**get\_workspace\_number\_for\_window** (*hwnd*)

Returns workspace number for the window with provided `hwnd`.

**Parameters** *hwnd* (*int*) – window id

**Returns** *int*

**is\_applicable** (*hwnd*)

Checks if provided `hwnd` represents window type that should be collected.

**Parameters** `hwnd` (*int*) – window id

**Returns** Boolean

**is\_resizable** (*hwnd*)

Checks if provided `hwnd` represents window that can be resized.

**Parameters** `hwnd` (*int*) – window id

**Returns** Boolean

**is\_restored** (*hwnd*)

Checks if provided `hwnd` represents window that is not minimized.

**Parameters** `hwnd` (*int*) – window id

**Returns** Boolean

**is\_valid\_state** (*hwnd*)

Checks if provided `hwnd` represents window with valid state for collecting.

Checking just `_is_activable()` for now.

**Parameters** `hwnd` (*int*) – window id

**Returns** Boolean

**arrangeit.windows.controller** – Module responsible for connecting data and view (MS Windows platform specific code)

**class** `arrangeit.windows.controller.Controller` (*app*)

Bases: `arrangeit.base.BaseController`

Controller class with MS Windows specific code.

**screenshot\_when\_exposed** = `True`

**setup\_root\_window** (*root*)

Sets provided root appearance attributes specific for MS Windows.

**arrangeit.windows.vdi** – Virtual desktop interfaces for MS Windows

Code from this module is based on the blog <<http://www.cyberforum.ru/blogs/105416/blog3671.html>>. The Python implementation is based on the work by @kdschlosser <[https://github.com/DanEdens/Virtual\\_Desktops\\_Plugin/blob/master/Virtualdesktops/\\_\\_int\\_\\_.py](https://github.com/DanEdens/Virtual_Desktops_Plugin/blob/master/Virtualdesktops/__int__.py)> (<<http://www.eventghost.net>>, <<http://www.eventghost.net/forum/viewtopic.php?p=53308#p53308>>)

**class** `arrangeit.windows.vdi.IApplicationView` (*\*args*, *\*\*kwargs*)

Bases: `arrangeit.windows.vdi.IInspectable`

Interface that provides view for the top-level application information.

**class** `arrangeit.windows.vdi.IApplicationViewCollection` (*\*args*, *\*\*kwargs*)

Bases: `sphinx.ext.autodoc.importer._MockObject`

Interface to collection of application views for specified data group.

**class** `arrangeit.windows.vdi.IApplicationViewConsolidatedEventArgs` (*\*args*, *\*\*kwargs*)

Bases: `arrangeit.windows.vdi.IInspectable`

Interface providing the results of application view consolidation operations.

**class** `arrangeit.windows.vdi.IInspectable` (\*args, \*\*kwargs)  
Bases: `sphinx.ext.autodoc.importer._MockObject`

Interface that provides functionality required for all Windows Runtime classes.

**class** `arrangeit.windows.vdi.IObjectArray` (\*args, \*\*kwargs)  
Bases: `sphinx.ext.autodoc.importer._MockObject`

Interface for accessing collection of objects based on IUnknown interface.

**class** `arrangeit.windows.vdi.IServiceProvider` (\*args, \*\*kwargs)  
Bases: `sphinx.ext.autodoc.importer._MockObject`

Interface that provides custom support to other objects.

**class** `arrangeit.windows.vdi.IVirtualDesktop` (\*args, \*\*kwargs)  
Bases: `sphinx.ext.autodoc.importer._MockObject`

Class defining virtual desktop instance accessible through its pointer.

**class** `arrangeit.windows.vdi.IVirtualDesktopManager` (\*args, \*\*kwargs)  
Bases: `sphinx.ext.autodoc.importer._MockObject`

Interface to publicly documented methods dealing with virtual dektops.

**class** `arrangeit.windows.vdi.IVirtualDesktopManagerInternal` (\*args, \*\*kwargs)  
Bases: `sphinx.ext.autodoc.importer._MockObject`

Interface to methods dealing with virtual dektops documented by community.

**class** `arrangeit.windows.vdi.VirtualDesktopsWin10`  
Bases: `object`

Helper class for calls to Windows 10 virtual desktop interfaces.

IVirtualDesktopManager is an interface publicly documented by Microsoft, whilst IVirtualDesktopManagerInternal is documented by community.

#### Variables

- ***manager*** – interface to publicly available API for virtual desktops
- ***internal\_manager*** – interface to internal API for virtual desktops
- ***view\_collection*** – interface to application views collection
- ***desktops*** – collection of virtual desktops ordinals and uids

**desktops = None**

**get\_desktops** (*refresh=False*)

Returns virtual desktops collection available in the system.

Retrieves and sets instance attribute holding collection if it hasn't been set yet or if True value is provided as *refresh* argument.

**Parameters** **refresh** (*Boolean*) – value indicating if desktop collection should be refreshed

**Returns** list of (int, GUID)

**get\_window\_desktop** (*hwnd, refresh=False*)

Returns virtual desktop where window with provided *hwnd* is placed.

**Parameters**

- **hwnd** (*int*) – window handle
- **refresh** (*Boolean*) – value indicating if desktop collection should be refreshed

**Variables** `desktop_id` – virtual desktop’s uid representation

**Returns** (int, GUID)

**internal\_manager** = None

**is\_window\_in\_current\_desktop** (*hwnd*)

Checks if window with provided *hwnd* is placed in current desktop.

**Parameters** `hwnd` (*int*) – window handle

**Returns** Boolean

**manager** = None

**move\_other\_window\_to\_desktop** (*hwnd, desktop\_ordinal*)

Moves other process’ window with provided *hwnd* to the other desktop identified by *desktop\_ordinal*.

**Parameters**

- `hwnd` (*int*) – window handle
- `desktop_ordinal` (*int*) – virtual desktop ordinal in desktops collection

**Variables**

- `desktop_id` – virtual desktop’s uid representation
- `desktop` – virtual desktop instance
- `app_view` – interface to application view

**Returns** False on success, None on failure

**move\_own\_window\_to\_desktop** (*hwnd, desktop\_ordinal*)

Moves root window with provided *hwnd* to the desktop with provided ordinal.

**Parameters**

- `hwnd` (*int*) – window handle
- `desktop_ordinal` (*int*) – virtual desktop ordinal in desktops collection

**Variables**

- `desktop_id` – virtual desktop’s uid representation
- `desktop` – virtual desktop instance

**Returns** False on success, None on failure

**view\_collection** = None

## **arrangeit.windows.utils – MS Windows specific utility functions**

`arrangeit.windows.utils.extract_name_from_bytes_path` (*path*)

Returns name without directory structure and extension from given path.

**Parameters** `path` (*bytes*) – full path to file

**Returns** str

`arrangeit.windows.utils.user_data_path` ()

Returns MS Windows specific path for saving user’s data.

## 6.2 tests – Unit and integration tests

`tests.unit` – arrangeit unit tests package

`tests.unit.test_setup` – Unit tests for main app setup and configuration

**class** `tests.unit.test_setup.TestFiles`

Bases: `object`

Testing class for program resources files.

**test\_resources\_icon\_file\_exist** (*asset*)

**test\_resources\_misc\_file\_exist** (*asset*)

**class** `tests.unit.test_setup.TestSetup`

Bases: `object`

Testing class for main app initialization and configuration.

**test\_main\_calls\_App\_run** (*mock*)

**test\_main\_calls\_get\_component\_class\_App** (*mock*)

**test\_main\_calls\_logging\_basicConfig** (*mock*)

**test\_main\_initializes\_platform\_specific\_App** (*mock*)

**class** `tests.unit.test_setup.TestStructure`

Bases: `object`

Testing class for platform specific subpackages structure.

**test\_host\_platform\_App\_issubclass\_of\_BaseApp** ()

**test\_host\_platform\_Collector\_issubclass\_of\_BaseCollector** ()

**test\_host\_platform\_Controller\_issubclass\_of\_BaseController** ()

`tests.unit.test_base` – Unit tests for base classes (without `BaseController`)

**class** `tests.unit.test_base.TestBaseApp`

Bases: `object`

Testing class for `BaseApp` class.

**test\_BaseApp\_initialize\_snapping\_sources\_calls\_collector\_get\_monitors\_rects** (*mock*)

**test\_BaseApp\_initialize\_snapping\_sources\_calls\_get\_available\_workspaces** (*mock*)

**test\_BaseApp\_initialize\_snapping\_sources\_calls\_get\_snapping\_sources\_for\_rect** (*mock*)

**test\_BaseApp\_initialize\_snapping\_sources\_functionality** (*mock*)

**test\_BaseApp\_save\_setting\_calls\_json\_load\_once** (*mock*)

**test\_BaseApp\_save\_setting\_calls\_platform\_user\_data\_path** (*mock*)

**test\_BaseApp\_save\_setting\_catches\_exception\_and\_continues** (*mock*)

**test\_BaseApp\_save\_setting\_checks\_if\_directory\_exists** (*mock*)

**test\_BaseApp\_save\_setting\_checks\_if\_file\_exists** (*mock*)

**test\_BaseApp\_save\_setting\_creates\_directory** (*mock*)

test\_BaseApp\_save\_setting\_overwrites\_settings\_file\_values (mock)

test\_BaseApp\_save\_setting\_updates\_settings\_file (mock)

test\_BaseApp\_save\_setting\_writes\_to\_settings\_file (mock)

test\_BaseApp\_activate\_root\_raises\_NotImplementedError (mock)

test\_BaseApp\_change\_setting\_calls\_save\_setting (mock)

test\_BaseApp\_change\_setting\_calls\_is\_setting (mock)

test\_BaseApp\_change\_setting\_calls\_is\_setting\_invalid (mock)

test\_BaseApp\_change\_setting\_changes\_valid\_setting (mock)

test\_BaseApp\_change\_setting\_returns\_change\_settings\_color\_group\_BG (mock)

test\_BaseApp\_change\_setting\_returns\_change\_settings\_color\_group\_FG (mock)

test\_BaseApp\_change\_settings\_color\_group\_calls\_Settings\_color\_group (mock)

test\_BaseApp\_change\_settings\_color\_group\_calls\_Settings\_setattr (mock)

test\_BaseApp\_change\_settings\_color\_group\_calls\_save\_setting (mock)

test\_BaseApp\_create\_snapping\_sources\_calls\_initialize\_snapping\_sources (mock)

test\_BaseApp\_create\_snapping\_sources\_calls\_collection\_generator (mock)

test\_BaseApp\_create\_snapping\_sources\_calls\_utils\_get\_snapping\_sources\_for\_rect (mock)

test\_BaseApp\_create\_snapping\_sources\_excludes\_provided\_model (mock, windows, expected)

test\_BaseApp\_create\_snapping\_sources\_functionality (mock, windows, expected)

test\_BaseApp\_create\_snapping\_sources\_includes\_provided\_model (mock, windows, expected)

test\_BaseApp\_create\_snapping\_sources\_returns\_dict (mock)

test\_BaseApp\_create\_snapping\_sources\_uses\_changed\_values\_if\_available (mock, windows, expected)

test\_BaseApp\_defines\_screenshot\_cleanup ()

test\_BaseApp\_grab\_window\_screen\_raises\_NotImplementedError (mock)

test\_BaseApp\_initialization\_calls\_setup\_collector (mock)

test\_BaseApp\_initialization\_calls\_setup\_controller (mock)

test\_BaseApp\_initialization\_instantiates\_collector (mock)

test\_BaseApp\_initialization\_instantiates\_controller (mock)

test\_BaseApp\_initialization\_instantiates\_controller\_with\_app (mock)

test\_BaseApp\_inits\_attr\_as\_None (attr)

test\_BaseApp\_move\_and\_resize\_raises\_NotImplementedError (mock)

test\_BaseApp\_move\_raises\_NotImplementedError (mock)



```

test_BaseApp_move_to_workspace_raises_NotImplementedError (mockер)
test_BaseApp_rerun_from_window_calls_repopulate_for_wid (mockер)
test_BaseApp_run_calls_WindowsCollection_generator (mockер)
test_BaseApp_run_calls_collector_run (mockер)
test_BaseApp_run_calls_controller_run (mockер)
test_BaseApp_run_calls_controller_run_with_valid_argument (mockер)
test_BaseApp_run_task_calls_related_methods (mockер, task, args)
test_BaseApp_save_default_calls_collection_export (mockер)
test_BaseApp_save_default_calls_json_dump (mockер)
test_BaseApp_save_default_calls_platform_user_data_path (mockер)
test_BaseApp_save_default_checks_if_directory_exists (mockер)
test_BaseApp_save_default_creates_directory (mockер)
test_BaseApp_screenshot_cleanup_returns_None (mockер)
test_BaseApp_setup_collector_calls_get_component_class_Collector (mockер)
test_BaseApp_setup_controller_calls_get_component_class_Controller (mockер)
class tests.unit.test_base.TestBaseCollector
    Bases: object
    Testing class for base Collector class.
    test_BaseCollector_add_window_raises_NotImplementedError ()
    test_BaseCollector_check_window_raises_NotImplementedError ()
    test_BaseCollector_get_available_workspaces_raises_NotImplementedError ()
    test_BaseCollector_get_monitors_rects_raises_NotImplementedError ()
    test_BaseCollector_get_smallest_monitor_size_calls_get_monitors_rects (mockер)
    test_BaseCollector_get_smallest_monitor_size_returns_two_tuple (mockер)
    test_BaseCollector_get_windows_raises_NotImplementedError ()
    test_BaseCollector_get_workspace_number_for_window_raises_NotImplementedError ()
    test_BaseCollector_get_workspace_number_raises_NotImplementedError ()
    test_BaseCollector_initialization_instantiates_WindowsCollection (mockер)
    test_BaseCollector_inits_collection_as_None ()
    test_BaseCollector_is_applicable_raises_NotImplementedError ()
    test_BaseCollector_is_resizable_raises_NotImplementedError ()
    test_BaseCollector_is_restored_raises_NotImplementedError ()
    test_BaseCollector_is_valid_state_raises_NotImplementedError ()
    test_BaseCollector_run_calls_add_window (mockер, elements)
    test_BaseCollector_run_calls_check_window (mockер)
    test_BaseCollector_run_calls_collection_sort (mockер)

```

```

    test_BaseCollector_run_calls_get_windows (mockер)
class tests.unit.test_base.TestBaseMouse
    Bases: object
    Testing class for Mouse class methods.
    test_BaseMouse_cursor_position_calls_Controller_position (mockер)
    test_BaseMouse_cursor_position_returns_position (mockер)
    test_BaseMouse_get_item_calls_queue_get (mockер)
    test_BaseMouse_get_item_returns_None_for_Empty (mockер)
    test_BaseMouse_get_item_returns_item (mockер)
    test_BaseMouse_init_instantiates_Controller (mockер)
    test_BaseMouse_init_instantiates_Queue (mockер)
    test_BaseMouse_init_sets_control_attribute (mockер)
    test_BaseMouse_init_sets_queue_attribute (mockер)
    test_BaseMouse_inits_attr_as_None (attr)
    test_BaseMouse_move_cursor_calls_Controller_position (mockер)
    test_BaseMouse_move_cursor_calls_position_with_provided_x_and_y (mockер)
    test_BaseMouse_on_move_puts_in_queue (mockер)
    test_BaseMouse_on_scroll_puts_in_queue (mockер, dy, expected)
    test_BaseMouse_start_instantiates_Listener (mockер)
    test_BaseMouse_start_sets_listener_attribute (mockер)
    test_BaseMouse_start_starts_listener (mockер)
    test_BaseMouse_stop_returns_False (mockер)
    test_BaseMouse_stop_stops_listener (mockер)
        StopException is raised if MagicMock has got StopException attribute.

```

### tests.unit.test\_basecontroller\_domain\_logic – Unit tests for domain logic part of the BaseController class

```

class tests.unit.test_basecontroller_domain_logic.TestBaseControllerDomainLogic
    Bases: object
    Testing class for base Controller class' domain logic methods.
    test_BaseController_apply_snapping_calls_check_snapping_state (mockер)
    test_BaseController_apply_snapping_calls_move_cursor_for_RESIZE (mockер)
    test_BaseController_apply_snapping_calls_setup_corner (mockер)
    test_BaseController_apply_snapping_changes_move_cursor_call (mockер,
                                                                    new_state,
                                                                    state, added_x,
                                                                    added_y)
    test_BaseController_apply_snapping_changes_state (mockер)

```

test\_BaseController\_apply\_snapping\_not\_calling\_setup\_corner (mocked)

test\_BaseController\_check\_snapping\_calls\_apply\_snapping (mocked)

test\_BaseController\_check\_snapping\_calls\_check\_intersection (mocked)

test\_BaseController\_check\_snapping\_calls\_get\_root\_rect (mocked)

test\_BaseController\_check\_snapping\_calls\_get\_snapping\_sources\_for\_locate (mocked)

test\_BaseController\_check\_snapping\_calls\_get\_snapping\_sources\_for\_resize (mocked)

test\_BaseController\_check\_snapping\_calls\_offset\_for\_intersections (mocked)

test\_BaseController\_check\_snapping\_not\_calling\_apply\_snapping (mocked)

test\_BaseController\_check\_snapping\_snapping\_is\_on\_false (mocked)

test\_BaseController\_check\_snapping\_state\_returns\_None\_for\_both\_axes\_snapping (mocked)

test\_BaseController\_check\_snapping\_state\_returns\_None\_for\_single\_axis\_snapping (mocked)

test\_BaseController\_check\_snapping\_state\_returns\_state\_for\_both\_axes\_snapping (mocked)

test\_BaseController\_check\_snapping\_state\_returns\_state\_for\_single\_axis\_snapping (mocked)

test\_BaseController\_listed\_window\_activated\_calls\_display\_message (mocked)

test\_BaseController\_listed\_window\_activated\_calls\_next (mocked)

test\_BaseController\_listed\_window\_activated\_calls\_recapture\_mouse\_for\_OTHER (mocked)

test\_BaseController\_listed\_window\_activated\_calls\_task\_rerun\_from\_window (mocked)

test\_BaseController\_listed\_window\_activated\_calls\_windows\_clear\_list (mocked)

test\_BaseController\_listed\_window\_activated\_calls\_windowslist\_add\_windows (mocked)

test\_BaseController\_listed\_window\_activated\_initializes\_generator (mocked)

test\_BaseController\_listed\_window\_activated\_not\_calling\_recapture\_not\_OTHER (mocked, state)

test\_BaseController\_listed\_window\_activated\_sets\_generator\_attr (mocked)

test\_BaseController\_next\_calls\_create\_snapping\_sources (mocked)

test\_BaseController\_next\_calls\_get\_root\_rect\_for\_first\_time\_True (mocked)

test\_BaseController\_next\_calls\_place\_on\_top\_left (mocked)

test\_BaseController\_next\_calls\_remove\_listed\_window (mocked)

test\_BaseController\_next\_calls\_root\_geometry\_for\_first\_time\_True (mocked)

test\_BaseController\_next\_calls\_save\_on\_StopIteration (mocked)

test\_BaseController\_next\_calls\_set\_default\_geometry (mocked)

test\_BaseController\_next\_calls\_set\_screenshot (mocked)

test\_BaseController\_next\_calls\_shutdown\_on\_StopIteration (mocked)

test\_BaseController\_next\_calls\_switch\_workspace\_from\_workspace (mocked)

test\_BaseController\_next\_calls\_switch\_workspace\_not\_first\_time (mocked)

test\_BaseController\_next\_calls\_update\_widgets (mocked)

test\_BaseController\_next\_not\_calling\_remove\_listed (mocked)

test\_BaseController\_next\_not\_calling\_root\_geometry\_for\_first\_time\_False (mocked)

test\_BaseController\_next\_not\_calling\_save\_on\_StopIteration (mockery)  
test\_BaseController\_next\_not\_calling\_set\_screenshot (mockery)  
test\_BaseController\_next\_not\_calling\_switch\_workspace\_first\_time (mockery)  
test\_BaseController\_next\_not\_calling\_switch\_workspace\_from\_workspace (mockery)  
test\_BaseController\_next\_not\_calling\_switch\_workspace\_same\_workspace (mockery)  
test\_BaseController\_next\_returns\_False (mockery)  
test\_BaseController\_next\_returns\_True\_on\_StopIteration (mockery)  
test\_BaseController\_next\_runs\_generator (mockery)  
test\_BaseController\_next\_sets\_snapping\_targets\_attribute (mockery)  
test\_BaseController\_next\_sets\_state\_attr\_to\_positioning\_corner\_0 (mockery)  
test\_BaseController\_run\_calls\_activate\_root\_task (mockery)  
test\_BaseController\_run\_calls\_display\_message (mockery)  
test\_BaseController\_run\_calls\_mainloop (mockery)  
test\_BaseController\_run\_calls\_mouse\_start (mockery)  
test\_BaseController\_run\_calls\_next (mockery)  
test\_BaseController\_run\_calls\_prepare\_view (mockery)  
test\_BaseController\_run\_calls\_view\_startup (mockery)  
test\_BaseController\_run\_sets\_generator\_attr\_from\_provided\_attr (mockery)  
test\_BaseController\_update\_calls\_display\_message\_for\_LOCATE (mockery)  
test\_BaseController\_update\_calls\_display\_message\_for\_RESIZE (mockery)  
test\_BaseController\_update\_calls\_update\_positioning\_for\_LOCATE (mockery)  
test\_BaseController\_update\_calls\_update\_resizing\_for\_RESIZE (mockery)  
test\_BaseController\_update\_not\_calling\_update\_methods\_for\_other\_states (mockery,  
state)  
test\_BaseController\_update\_positioning\_calls\_next\_for\_not\_resizable (mockery)  
test\_BaseController\_update\_positioning\_calls\_run\_task\_move\_w\_not\_resizable\_ws (mockery)  
test\_BaseController\_update\_positioning\_calls\_run\_task\_move\_window\_not\_resizable (mockery)  
test\_BaseController\_update\_positioning\_calls\_set\_changed (mockery, state,  
sign\_x, sign\_y)  
test\_BaseController\_update\_positioning\_for\_resizable\_calls\_master\_update (mockery)  
test\_BaseController\_update\_positioning\_for\_resizable\_calls\_place\_on\_opposite (mockery)  
test\_BaseController\_update\_positioning\_for\_resizable\_calls\_set\_screenshot (mockery)  
test\_BaseController\_update\_positioning\_for\_resizable\_not\_calling\_set\_screenshot (mockery)  
test\_BaseController\_update\_positioning\_for\_resizable\_sets\_state (mockery)  
test\_BaseController\_update\_positioning\_not\_calling\_run\_task\_move\_w\_not\_resize (mockery)  
test\_BaseController\_update\_resizing\_calls\_next (mockery)  
test\_BaseController\_update\_resizing\_calls\_run\_task\_move\_and\_resize\_for\_ws (mockery)

```

test_BaseController_update_resizing_calls_run_task_move_and_resize_window (mock)
test_BaseController_update_resizing_corner_0_calls_set_changed (mock)
test_BaseController_update_resizing_corner_1_calls_set_changed (mock)
test_BaseController_update_resizing_corner_2_calls_set_changed (mock)
test_BaseController_update_resizing_corner_3_calls_set_changed (mock)
test_BaseController_update_resizing_skips_run_task_move_and_resize_window (mock)
test_BaseController_workspace_activated_calls_display_message (mock)
test_BaseController_workspace_activated_calls_recapture_mouse_for_OTHER (mock)
test_BaseController_workspace_activated_calls_set_changed (mock)
test_BaseController_workspace_activated_calls_task_move_to_workspace (mock)
test_BaseController_workspace_activated_not_calling_recapture_mouse_not_OTHER (mock,
state)

```

### tests.unit.test\_basecontroller – Unit tests for BaseController class (not including domain logic)

```

class tests.unit.test_basecontroller.TestBaseController

```

```

    Bases: object

```

```

    Testing class for base Controller class.

```

```

test_BaseController_change_position_calls_check_snapping (mock)
test_BaseController_change_position_calls_get_root_rect (mock)
test_BaseController_change_position_calls_master_geometry (mock)
test_BaseController_change_position_not_calling_set_geometry_and_root_rect (mock)
test_BaseController_change_setting_calls_run_task (mock)
test_BaseController_change_size_calls_check_current_size (mock)
test_BaseController_change_size_calls_check_snapping (mock)
test_BaseController_change_size_calls_master_geometry (mock)
test_BaseController_change_size_calls_set_minimum_size_and_returns_for_invalid (mock)
test_BaseController_change_size_not_calling_set_geometry (mock)
test_BaseController_change_size_valid_x_and_y (mock, state, x, y, changed_x,
changed_y, expected)
test_BaseController_change_size_with_min_valid_x_and_y (mock, state,
x, y, changed_x,
changed_y, expected)
test_BaseController_check_current_size_for_invalid_xy_returns_position (mock,
corner,
x,
y,
changed,
expected)

```

**test\_BaseController\_check\_current\_size\_for\_valid\_xy\_returns\_False** (*mock*,  
*corner*,  
*x*, *y*,  
*changed*)

**test\_BaseController\_check\_mouse\_calls\_after\_idle\_with\_mouse\_move** (*mock*)

**test\_BaseController\_check\_mouse\_calls\_after\_idle\_with\_mouse\_scroll** (*mock*)

**test\_BaseController\_check\_mouse\_calls\_after\_with\_itself** (*mock*)

**test\_BaseController\_check\_mouse\_calls\_mouse\_get\_item** (*mock*)

**test\_BaseController\_cycle\_corners\_calls\_display\_message** (*mock*)

**test\_BaseController\_cycle\_corners\_calls\_move\_to\_corner** (*mock*)

**test\_BaseController\_cycle\_corners\_counter\_false\_functionality** (*mock*,  
*state*, *ex-*  
*pected*)

**test\_BaseController\_cycle\_corners\_counter\_true\_functionality** (*mock*, *state*,  
*expected*)

**test\_BaseController\_cycle\_corners\_not\_calling\_move\_to\_corner** (*mock*)

**test\_BaseController\_display\_message\_calls\_set\_timer\_by\_default** (*mock*)

**test\_BaseController\_display\_message\_not\_calling\_set\_timer\_for\_permanent** (*mock*)

**test\_BaseController\_display\_message\_sets\_statusbar\_message** (*mock*)

**test\_BaseController\_get\_root\_rect\_functionality** (*mock*, *state*, *expected*)

**test\_BaseController\_init\_sets\_app\_attribute** (*mock*)

**test\_BaseController\_initialization\_calls\_setup** (*mock*)

**test\_BaseController\_initialization\_instantiates\_Mouse** (*mock*)

**test\_BaseController\_initialization\_instantiates\_WindowModel** (*mock*)

**test\_BaseController\_inits\_attr\_as\_None** (*attr*)

**test\_BaseController\_inits\_screenshot\_when\_exposed\_as\_False** ()

**test\_BaseController\_listed\_window\_activated\_by\_digit\_calls\_l\_window\_activated** (*mock*)

**test\_BaseController\_listed\_window\_activated\_by\_digit\_calls\_wininfo\_children** (*mock*)

**test\_BaseController\_listed\_window\_activated\_by\_digit\_not\_calling\_l\_win\_active** (*mock*)

**test\_BaseController\_mainloop\_calls\_Tkinter\_mainloop** (*mock*)

**test\_BaseController\_mainloop\_calls\_after\_for\_check\_mouse** (*mock*)

**test\_BaseController\_mouse\_move\_calls\_change\_position\_for\_LOCATE** (*mock*)

**test\_BaseController\_mouse\_move\_calls\_change\_size\_for\_RESIZE** (*mock*)

**test\_BaseController\_mouse\_scroll\_calls\_counter\_false\_cycle\_corners** (*mock*)

**test\_BaseController\_move\_to\_corner\_calls\_move\_cursor\_state\_0** (*mock*)

**test\_BaseController\_move\_to\_corner\_calls\_move\_cursor\_state\_1** (*mock*)

**test\_BaseController\_move\_to\_corner\_calls\_move\_cursor\_state\_2** (*mock*)

**test\_BaseController\_move\_to\_corner\_calls\_move\_cursor\_state\_3** (*mock*)

test\_BaseController\_move\_to\_corner\_calls\_setup\_corner (mockery)

test\_BaseController\_on\_continue\_calls\_recapture\_mouse (mockery)

test\_BaseController\_on\_continue\_returns\_break (mockery)

test\_BaseController\_on\_focus\_calls\_focus\_get (mockery)

test\_BaseController\_on\_focus\_calls\_run\_task\_activate\_root (mockery)

test\_BaseController\_on\_focus\_not\_calling\_run\_task (mockery)

test\_BaseController\_on\_focus\_not\_returns\_break (mockery)

test\_BaseController\_on\_focus\_returns\_break (mockery)

test\_BaseController\_on\_key\_pressed\_calls\_cycle\_corners (mockery, key)

test\_BaseController\_on\_key\_pressed\_calls\_release\_mouse (mockery, key)

test\_BaseController\_on\_key\_pressed\_calls\_skip\_current\_window (mockery, key)

test\_BaseController\_on\_key\_pressed\_calls\_switch\_resizable (mockery, key)

test\_BaseController\_on\_key\_pressed\_calls\_switch\_restored (mockery, key)

test\_BaseController\_on\_key\_pressed\_for\_Enter\_calls\_update (mockery, key)

test\_BaseController\_on\_key\_pressed\_for\_Escape\_calls\_shutdown (mockery)

test\_BaseController\_on\_key\_pressed\_for\_digit\_0\_not\_calling\_workspace\_activated\_by\_digi

test\_BaseController\_on\_key\_pressed\_for\_digit\_calls\_workspace\_activated\_by\_digit (mockery, key)

test\_BaseController\_on\_key\_pressed\_for\_func\_keys\_c\_listed\_window\_activated\_by\_d (mockery, key)

test\_BaseController\_on\_key\_pressed\_returns\_break (mockery)

test\_BaseController\_on\_mouse\_left\_down\_calls\_update (mockery)

test\_BaseController\_on\_mouse\_left\_down\_returns\_break (mockery)

test\_BaseController\_on\_mouse\_middle\_down\_calls\_release\_mouse (mockery)

test\_BaseController\_on\_mouse\_middle\_down\_returns\_break (mockery)

test\_BaseController\_on\_mouse\_right\_down\_calls\_skip\_current\_window (mockery)

test\_BaseController\_on\_mouse\_right\_down\_returns\_break (mockery)

test\_BaseController\_on\_mouse\_scroll\_calls\_counter\_true\_cycle\_corners (mockery)

test\_BaseController\_on\_resizable\_calls\_recapture\_mouse (mockery)

test\_BaseController\_on\_resizable\_calls\_switch\_resizable (mockery)

test\_BaseController\_on\_resizable\_returns\_break (mockery)

test\_BaseController\_on\_restored\_change\_calls\_recapture\_mouse (mockery)

test\_BaseController\_on\_restored\_change\_calls\_switch\_restored (mockery)

test\_BaseController\_on\_restored\_change\_returns\_break (mockery)

**test\_BaseController\_place\_on\_opposite\_corner\_calls\_min\_move\_cursor** (*mock*  
*er,*  
*state,*  
*x, y,*  
*w, h,*  
*ex-*  
*pected\_x,*  
*ex-*  
*pected\_y*)

**test\_BaseController\_place\_on\_opposite\_corner\_calls\_move\_cursor** (*mock*  
*er,*  
*state, x, y,*  
*w, h, ex-*  
*pected\_x,*  
*ex-*  
*pected\_y*)

**test\_BaseController\_place\_on\_opposite\_corner\_calls\_setup\_corner** (*mock*  
*er*)

**test\_BaseController\_place\_on\_top\_left\_calls\_move\_cursor** (*mock*  
*er*)

**test\_BaseController\_place\_on\_top\_left\_calls\_setup\_corner** (*mock*  
*er*)

**test\_BaseController\_prepare\_view\_calls\_WindowsList\_add\_windows\_without\_first** (*mock*  
*er*)

**test\_BaseController\_prepare\_view\_calls\_WorkspacesCollection\_add\_workspaces** (*mock*  
*er*)

**test\_BaseController\_recapture\_mouse\_calls\_display\_message** (*mock*  
*er*)

**test\_BaseController\_recapture\_mouse\_calls\_mouse\_start** (*mock*  
*er*)

**test\_BaseController\_recapture\_mouse\_calls\_move\_cursor** (*mock*  
*er*)

**test\_BaseController\_recapture\_mouse\_calls\_set\_default\_geometry** (*mock*  
*er*)

**test\_BaseController\_recapture\_mouse\_calls\_setup\_corner** (*mock*  
*er*)

**test\_BaseController\_recapture\_mouse\_calls\_view\_setup\_bindings** (*mock*  
*er*)

**test\_BaseController\_recapture\_mouse\_changes\_state\_to\_LOCATE** (*mock*  
*er*)

**test\_BaseController\_release\_mouse\_calls\_cursor\_config** (*mock*  
*er*)

**test\_BaseController\_release\_mouse\_calls\_display\_message** (*mock*  
*er*)

**test\_BaseController\_release\_mouse\_calls\_reset\_bindings** (*mock*  
*er*)

**test\_BaseController\_release\_mouse\_calls\_view\_corner\_hide\_corner** (*mock*  
*er*)

**test\_BaseController\_release\_mouse\_changes\_state\_to\_OTHER** (*mock*  
*er*)

**test\_BaseController\_release\_mouse\_stops\_mouse\_listener** (*mock*  
*er*)

**test\_BaseController\_remove\_listed\_window\_calls\_place\_children** (*mock*  
*er*)

**test\_BaseController\_remove\_listed\_window\_calls\_widget\_destroy** (*mock*  
*er*)

**test\_BaseController\_remove\_listed\_window\_not\_calling\_destroy\_for\_wrong\_widget** (*mock*  
*er*)

**test\_BaseController\_resizing\_state\_counterpart** (*mock*  
*er, state, expected*)

**test\_BaseController\_save\_runs\_related\_task** (*mock*  
*er*)

**test\_BaseController\_set\_default\_geometry\_calls\_geometry** (*mock*  
*er*)

**test\_BaseController\_set\_default\_geometry\_calls\_get\_smallest\_monitor\_size** (*mock*  
*er*)

**test\_BaseController\_set\_default\_geometry\_calls\_quarter\_by\_smaller** (*mock*  
*er*)



test\_BaseController\_set\_default\_geometry\_not\_calling\_quarter\_by\_smaller (mockery)

test\_BaseController\_set\_default\_geometry\_not\_changing\_default\_size (mockery)

test\_BaseController\_set\_default\_geometry\_sets\_default\_size (mockery)

test\_BaseController\_set\_minimum\_size\_functionality (mockery)

test\_BaseController\_set\_screenshot\_calls\_grab\_window\_screen (mockery)

test\_BaseController\_set\_screenshot\_calls\_master\_update (mockery)

test\_BaseController\_set\_screenshot\_calls\_run\_task (mockery)

test\_BaseController\_set\_screenshot\_configures\_screenshot\_widget (mockery)

test\_BaseController\_set\_screenshot\_places\_screenshot\_widget (mockery)

test\_BaseController\_set\_screenshot\_returns\_True\_for\_disabled (mockery)

test\_BaseController\_set\_screenshot\_sets\_screenshot\_reference\_variable (mockery)

test\_BaseController\_set\_timer\_calls\_after (mockery)

test\_BaseController\_set\_timer\_calls\_after\_cancel (mockery)

test\_BaseController\_set\_timer\_not\_calling\_after\_cancel (mockery)

test\_BaseController\_set\_timer\_sets\_timer\_attribute (mockery)

test\_BaseController\_setup\_calls\_get\_screenshot\_widget (mockery)

test\_BaseController\_setup\_calls\_get\_tkinter\_root (mockery)

test\_BaseController\_setup\_calls\_setup\_root\_window (mockery)

test\_BaseController\_setup\_corner\_calls\_cursor\_config (mockery, state)

test\_BaseController\_setup\_corner\_calls\_get\_cursor\_name (mockery)

test\_BaseController\_setup\_corner\_calls\_view\_corner\_set\_corner (mockery)

test\_BaseController\_setup\_initializes\_ViewApplication (mockery)

test\_BaseController\_setup\_root\_window\_calls\_config\_background (mockery)

test\_BaseController\_setup\_root\_window\_calls\_wm\_attributes (mockery)

test\_BaseController\_setup\_root\_window\_not\_calling\_alpha (mockery)

test\_BaseController\_shutdown\_calls\_master\_destroy (mockery)

test\_BaseController\_shutdown\_raises\_SystemExit (mockery)

test\_BaseController\_shutdown\_stops\_mouse (mockery)

test\_BaseController\_skip\_current\_window\_calls\_display\_message (mockery)

test\_BaseController\_skip\_current\_window\_calls\_model\_clear\_changed (mockery)

test\_BaseController\_skip\_current\_window\_calls\_next (mockery)

test\_BaseController\_switch\_resizable\_calls\_display\_message (mockery)

test\_BaseController\_switch\_resizable\_calls\_widget\_set\_value (mockery)

test\_BaseController\_switch\_resizable\_functionality (mockery, resizable, expected)

test\_BaseController\_switch\_restored\_calls\_display\_message (mockery)

test\_BaseController\_switch\_restored\_calls\_widget\_set\_value (mockery)

```

test_BaseController_switch_restored_functionality (mocked, restored, expected)
test_BaseController_switch_workspace_calls_display_message (mocked)
test_BaseController_switch_workspace_calls_get_root_wid (mocked)
test_BaseController_switch_workspace_calls_task_move_to_workspace (mocked)
test_BaseController_workspace_activated_by_digit_calls_winfo_children (mocked)
test_BaseController_workspace_activated_by_digit_calls_workspace_activated (mocked)
test_BaseController_workspace_activated_by_digit_not_calling_workspace_active (mocked)

```

## tests.unit.test\_data – Unit tests for window model and collection

```

class tests.unit.test_data.TestWindowModel
    Bases: object

    Testing class for arrangeit.data.WindowModel class.

    test_WindowModel_changed_h_gets_h_from_changed ()
    test_WindowModel_changed_h_gets_h_from_rect ()
    test_WindowModel_changed_w_gets_w_from_changed ()
    test_WindowModel_changed_w_gets_w_from_rect ()
    test_WindowModel_changed_x_gets_x_from_changed ()
    test_WindowModel_changed_x_gets_x_from_rect ()
    test_WindowModel_changed_y_gets_y_from_changed ()
    test_WindowModel_changed_y_gets_y_from_rect ()
    test_WindowModel_clear_changed_sets_changed_to_empty_tuple (mocked)
    test_WindowModel_clear_changed_sets_changed_ws_to_None (mocked)
    test_WindowModel_h_gets_height_from_rect ()
    test_WindowModel_initialization_calls_setup (mocked)
    test_WindowModel_inits_attr_as_None_or_empty_tuple (attr)
    test_WindowModel_inits_changed_as_empty_tuple ()
    test_WindowModel_inits_changed_ws_as_None ()
    test_WindowModel_is_changed_functionality (changed, expected)
    test_WindowModel_is_ws_changed_functionality (changed_ws, ws, expected)
    test_WindowModel_set_changed_creates_empty_tuple_for_invalid (values)
    test_WindowModel_set_changed_creates_empty_tuple_invalid_rect (values)
    test_WindowModel_set_changed_creates_from_rect (values)
    test_WindowModel_set_changed_creates_from_rect_elements_changed (values)
    test_WindowModel_set_changed_creates_from_rect_elements_rect (values)
    test_WindowModel_set_changed_not_changing_same_value (values)
    test_WindowModel_set_changed_sets_changed_ws_and_changed (values)

```

```

test_WindowModel_set_changed_sets_changed_ws_for_provided_ws(ws)
test_WindowModel_set_changed_sets_changed_ws_to_None_for_invalid(ws)
test_WindowModel_setup_calls_get_value_if_valid_type_for_all(mockers, values)
test_WindowModel_setup_set_None_or_empty_for_invalid_type(mockers, values)
test_WindowModel_setup_sets_None_for_values_not_provided(mockers, values)
test_WindowModel_setup_sets_attrs_for_valid_type(mockers, values)
test_WindowModel_setup_sets_attrs_if_provided(mockers, values)
test_WindowModel_w_gets_width_from_rect()
test_WindowModel_ws_is_alias_for_workspace()
test_WindowModel_x_gets_x_from_rect()
test_WindowModel_y_gets_y_from_rect()
class tests.unit.test_data.TestWindowsCollection
    Bases: object
    Testing class for arrangeit.data.WindowsCollection class.
    test_WindowsCollection_add_appends_one_element_to__members()
    test_WindowsCollection_add_raises_for_invalid_argument(arg)
    test_WindowsCollection_clear_empties__members()
    test_WindowsCollection_export(elements)
    test_WindowsCollection_generator_next_yields_value()
    test_WindowsCollection_generator_type()
    test_WindowsCollection_get_model_by_wid_empty_collection()
    test_WindowsCollection_get_model_by_wid_invalid_wid()
    test_WindowsCollection_get_model_by_wid_valid_wid()
    test_WindowsCollection_get_windows_calls_generator(mockers)
    test_WindowsCollection_get_windows_list_returns_list_of_windows()
    test_WindowsCollection_initialization_sets_empty__members()
    test_WindowsCollection_inits____members_as_None()
    test_WindowsCollection_repopulate_for_wid_functionality(elements, wid,
        remove_before,
        expected)
    test_WindowsCollection_size_is_property()
    test_WindowsCollection_size_returns__members_length()
    test_WindowsCollection_sort_functionality(ws_wid, expected)

```

## tests.unit.test\_view – Unit tests for view classes and functions (without ViewApplication)

```
class tests.unit.test_view.TestCornerWidget
```

```
    Bases: object
```

```
    Unit testing class for CornerWidget class.
```

```
    test_view_CornerWidget_anchor_functionality (mock, corner, expected)
```

```
    test_view_CornerWidget_get_place_parameters_functionality (mock, corner, size, expected)
```

```
    test_view_CornerWidget_hide_corner_hides_frames (mock)
```

```
    test_view_CornerWidget_init_calls_setup_widgets (mock)
```

```
    test_view_CornerWidget_init_sets_attributes (mock, attr)
```

```
    test_view_CornerWidget_inits_attributes (attr, value)
```

```
    test_view_CornerWidget_issubclass_of_object ()
```

```
    test_view_CornerWidget_max_box_functionality (mock, shift, expected)
```

```
    test_view_CornerWidget_max_box_is_property ()
```

```
    test_view_CornerWidget_max_xy_functionality (mock, shift, expected)
```

```
    test_view_CornerWidget_max_xy_is_property ()
```

```
    test_view_CornerWidget_set_corner_calls_frame_place (mock)
```

```
    test_view_CornerWidget_set_corner_calls_get_place_parameters_with_max_box (mock)
```

```
    test_view_CornerWidget_set_corner_calls_get_place_parameters_with_max_xy (mock)
```

```
    test_view_CornerWidget_setup_widgets_calls_set_corner (mock)
```

```
    test_view_CornerWidget_setup_widgets_instantiates_frame_and_sets_attribute_for_it (mock, attr, width, height)
```

```
class tests.unit.test_view.TestListedWindow
```

```
    Bases: object
```

```
    Unit testing class for ListedWindow class.
```

```
    test_view_ListedWindow_get_icon_image_calls_ImageTk_PhotoImage (mock)
```

```
    test_view_ListedWindow_init_calls_get_icon_image (mock)
```

```
    test_view_ListedWindow_init_calls_setup_bindings (mock)
```

```
    test_view_ListedWindow_init_calls_setup_widgets (mock)
```

```
    test_view_ListedWindow_init_calls_super_with_master_and_cursor_arg (mock)
```

```
    test_view_ListedWindow_init_sets_attributes (mock, attr)
```

```
    test_view_ListedWindow_inits_attr_as_empty (attr, value)
```

```
    test_view_ListedWindow_issubclass_of_Frame ()
```

```
    test_view_ListedWindow_on_widget_enter_returns_break (mock)
```

```
    test_view_ListedWindow_on_widget_enter_sets_foreground (mock)
```

```
    test_view_ListedWindow_on_widget_leave_returns_break (mock)
```

```

test_view_ListedWindow_on_widget_leave_sets_foreground(mock)
test_view_ListedWindow_setup_bindings_callbacks(mock, event, method)
test_view_ListedWindow_setup_bindings_labels_master_callbacks(mock,
                                                             event,
                                                             method)
test_view_ListedWindow_setup_widgets_calls_config_background(mock)
test_view_ListedWindow_setup_widgets_calls_label_place(mock)
test_view_ListedWindow_setup_widgets_sets_icon_label(mock)
test_view_ListedWindow_setup_widgets_sets_title_label(mock)
class tests.unit.test_view.TestPropertyIcon
    Bases: object
    Unit testing class for PropertyIcon class.
    test_view_PropertyIcon_init_calls_setup_bindings(mock)
    test_view_PropertyIcon_init_calls_setup_widgets(mock)
    test_view_PropertyIcon_init_calls_super_with_master_arg(mock)
    test_view_PropertyIcon_init_sets_attributes(mock, attr)
    test_view_PropertyIcon_inits_attr_as_empty(attr, value)
    test_view_PropertyIcon_issubclass_of_Label()
    test_view_PropertyIcon_on_widget_enter_configures_image(mock)
    test_view_PropertyIcon_on_widget_enter_returns_break(mock)
    test_view_PropertyIcon_on_widget_leave_configures_image(mock)
    test_view_PropertyIcon_on_widget_leave_returns_break(mock)
    test_view_PropertyIcon_set_value_calls_config(mock)
    test_view_PropertyIcon_set_value_sets_value_attribute(mock)
    test_view_PropertyIcon_setup_bindings_callbacks(mock, event, method)
    test_view_PropertyIcon_setup_bindings_labels_master_callbacks(mock,
                                                                event,
                                                                method)
    test_view_PropertyIcon_setup_widgets_configs_label(mock)
    test_view_PropertyIcon_setup_widgets_sets_colorized_icon_image(mock,
                                                                    value,
                                                                    path)
    test_view_PropertyIcon_setup_widgets_sets_icon_image(mock, value, path)
class tests.unit.test_view.TestResizable
    Bases: object
    Unit testing class for Resizable class.
    test_view_Resizable_init_calls_super_with_master_and_background_args(mock)
    test_view_Resizable_inits_attr_as_empty(attr, value)
    test_view_Resizable_inits_image_name(attr, value)

```

```

    test_view_Resizable_issubclass_of_PropertyIcon()
class tests.unit.test_view.TestRestored
    Bases: object
    Unit testing class for Restored class.
    test_view_Restored_init_calls_super_with_master_and_background_args (mock)
    test_view_Restored_inits_attr_as_empty (attr, value)
    test_view_Restored_inits_image_name (attr, value)
    test_view_Restored_issubclass_of_PropertyIcon()
class tests.unit.test_view.TestStatusbar
    Bases: object
    Unit testing class for Statusbar class.
    test_view_Statusbar_init_calls_setup_widgets (mock)
    test_view_Statusbar_init_calls_super_with_master_arg (mock)
    test_view_Statusbar_init_configures_background (mock)
    test_view_Statusbar_init_sets_attributes (mock, attr)
    test_view_Statusbar_inits_attributes (attr, value)
    test_view_Statusbar_issubclass_of_Frame()
    test_view_Statusbar_setup_widgets_calls_label_pack (mock)
    test_view_Statusbar_setup_widgets_sets_message_label (mock)
    test_view_Statusbar_setup_widgets_sets_tk_variable (mock)
class tests.unit.test_view.TestToolbar
    Bases: object
    Unit testing class for Toolbar class.
    test_view_Toolbar_init_calls_setup_widgets (mock)
    test_view_Toolbar_init_calls_super_with_master_arg (mock)
    test_view_Toolbar_init_configures_background (mock)
    test_view_Toolbar_init_sets_attributes (mock, attr)
    test_view_Toolbar_inits_attributes (attr, value)
    test_view_Toolbar_issubclass_of_Frame()
    test_view_Toolbar_on_options_click_hides_root (mock)
    test_view_Toolbar_on_options_click_initializes_Options (mock)
    test_view_Toolbar_on_options_click_sets_topmost_false (mock)
    test_view_Toolbar_on_options_click_sets_topmost_true (mock)
    test_view_Toolbar_setup_widgets_calls_button_place (mock)
    test_view_Toolbar_setup_widgets_sets_options_button (mock)
    test_view_Toolbar_setup_widgets_sets_quit_button (mock)

```

```

class tests.unit.test_view.TestViewFunctions
    Bases: object

    Unit testing class for view module inner functions.

    test_view_get_screenshot_widget_calls_label_place (mock)
    test_view_get_screenshot_widget_initializes_Label (mock)
    test_view_get_screenshot_widget_returns_label_instance (mock)
    test_view_get_tkinter_root_calls_set_icon (mock)
    test_view_get_tkinter_root_initializes_Tk (mock)
    test_view_get_tkinter_root_returns_Tk_instance (mock)
    test_view_get_tkinter_root_sets_title (mock)
    test_view_get_tkinter_root_withdraw_root (mock)

class tests.unit.test_view.TestWindowsList
    Bases: object

    Unit testing class for WindowsList class.

    test_view_WindowsList_add_windows_calls_place_widget_on_position (mock,
                                                                    args)
    test_view_WindowsList_add_windows_initializes_ListedWindow (mock)
    test_view_WindowsList_clear_list_calls_widget_destroy (mock)
    test_view_WindowsList_clear_list_calls_wininfo_children (mock)
    test_view_WindowsList_init_calls_super_with_master_arg (mock)
    test_view_WindowsList_init_configures_background (mock)
    test_view_WindowsList_init_sets_master_attribute (mock)
    test_view_WindowsList_inits_attr_as_None (attr)
    test_view_WindowsList_issubclass_of_Frame ()
    test_view_WindowsList_on_window_label_button_down_calls_listed_window_activated (mock)
    test_view_WindowsList_on_window_label_button_returns_break (mock)
    test_view_WindowsList_place_children_calls_place_widget_on_position (mock)
    test_view_WindowsList_place_widget_on_position_calls_place_on_frame (mock)

class tests.unit.test_view.TestWorkspace
    Bases: object

    Unit testing class for Workspace class.

    test_view_Workspace_get_humanized_number (mock, number)
    test_view_Workspace_init_calls_setup_bindings (mock)
    test_view_Workspace_init_calls_setup_widgets (mock)
    test_view_Workspace_init_calls_super_with_master_arg (mock)
    test_view_Workspace_init_sets_attributes (mock, attr)
    test_view_Workspace_inits_attr_as_empty (attr, value)
    test_view_Workspace_issubclass_of_Frame ()

```

```

test_view_Workspace_on_widget_enter_not_setting_foreground_for_active (mock)
test_view_Workspace_on_widget_enter_returns_break (mock)
test_view_Workspace_on_widget_enter_sets_foreground (mock)
test_view_Workspace_on_widget_leave_not_setting_foreground_for_active (mock)
test_view_Workspace_on_widget_leave_returns_break (mock)
test_view_Workspace_on_widget_leave_sets_foreground (mock)
test_view_Workspace_setup_bindings_callbacks (mock, event, method)
test_view_Workspace_setup_bindings_labels_master_callbacks (mock, event,
method)
test_view_Workspace_setup_widgets_calls_get_humanized_number (mock)
test_view_Workspace_setup_widgets_calls_label_place (mock)
test_view_Workspace_setup_widgets_sets_name_label (mock)
test_view_Workspace_setup_widgets_sets_number_label (mock)
class tests.unit.test_view.TestWorkspacesCollection
Bases: object
Unit testing class for WorkspacesCollection class.
test_view_WorkspacesCollection_add_workspaces_calls_place_on_frame (mock,
args)
test_view_WorkspacesCollection_add_workspaces_initializes_Workspace (mock)
test_view_WorkspacesCollection_add_workspaces_not_calling_place (mock)
test_view_WorkspacesCollection_init_calls_config_background (mock)
test_view_WorkspacesCollection_init_calls_super_with_master_arg (mock)
test_view_WorkspacesCollection_init_sets_master_attribute (mock)
test_view_WorkspacesCollection_inits_attributes (attr, value)
test_view_WorkspacesCollection_issubclass_of_Frame ()
test_view_WorkspacesCollection_on_workspace_label_button_down_calls_workspace_active (m
test_view_WorkspacesCollection_on_workspace_label_button_returns_break (mock)
test_view_WorkspacesCollection_select_active_calls_cursor_config (mock)
test_view_WorkspacesCollection_select_active_calls_label_config (mock)
test_view_WorkspacesCollection_select_active_for_single_workspace (mock)
test_view_WorkspacesCollection_select_active_sets_active_attr (mock)

tests.unit.test_viewapplication – Unit tests for ViewApplication class

class tests.unit.test_viewapplication.TestViewApplication
Bases: object
Unit testing class for ViewApplication class.
test_ViewApplication_get_root_wid_calls_int_and_returns_it (mock)

```



**test\_ViewApplication\_get\_root\_wid\_calls\_master\_frame** (*mock*)  
**test\_ViewApplication\_hide\_root\_calls\_master\_hiding\_up\_method** (*mock*,  
*method*)  
**test\_ViewApplication\_init\_calls\_super\_with\_master\_arg** (*mock*)  
**test\_ViewApplication\_init\_configures\_background** (*mock*)  
**test\_ViewApplication\_init\_sets\_master\_and\_controller\_attributes** (*mock*)  
**test\_ViewApplication\_inits\_calls\_setup\_bindings** (*mock*)  
**test\_ViewApplication\_inits\_calls\_setup\_widgets** (*mock*)  
**test\_ViewApplication\_issubclass\_of\_Frame** ()  
**test\_ViewApplication\_reset\_bindings\_labels\_bind\_callback** (*mock*, *event*,  
*method*)  
**test\_ViewApplication\_reset\_bindings\_unbind\_all** (*mock*, *event*)  
**test\_ViewApplication\_reset\_bindings\_windowslist\_bind\_callback** (*mock*,  
*event*,  
*method*)  
**test\_ViewApplication\_reset\_bindings\_workspaces\_bind\_callback** (*mock*,  
*event*,  
*method*)  
**test\_ViewApplication\_setup\_bindings\_bind\_all\_callbacks** (*mock*, *event*,  
*method*)  
**test\_ViewApplication\_setup\_bindings\_bind\_callbacks** (*mock*, *event*, *method*)  
**test\_ViewApplication\_setup\_bindings\_label\_bind\_callbacks** (*mock*, *event*,  
*method*)  
**test\_ViewApplication\_setup\_bindings\_root\_bind\_callbacks** (*mock*, *event*,  
*method*)  
**test\_ViewApplication\_setup\_bindings\_unbinds\_all\_button\_1** (*mock*)  
**test\_ViewApplication\_setup\_corner\_instantiates\_CornerWidget** (*mock*)  
**test\_ViewApplication\_setup\_corner\_sets\_corner\_attribute** (*mock*)  
**test\_ViewApplication\_setup\_icon\_calls\_label\_place** (*mock*)  
**test\_ViewApplication\_setup\_icon\_sets\_icon\_label** (*mock*)  
**test\_ViewApplication\_setup\_name\_calls\_label\_place** (*mock*)  
**test\_ViewApplication\_setup\_name\_sets\_name\_label** (*mock*)  
**test\_ViewApplication\_setup\_name\_sets\_tk\_variable** (*mock*)  
**test\_ViewApplication\_setup\_resizable\_calls\_label\_place** (*mock*)  
**test\_ViewApplication\_setup\_resizable\_initializes\_Resizable** (*mock*)  
**test\_ViewApplication\_setup\_resizable\_sets\_viewapp\_as\_master** (*mock*)  
**test\_ViewApplication\_setup\_restored\_calls\_label\_place** (*mock*)  
**test\_ViewApplication\_setup\_restored\_initializes\_Restored** (*mock*)  
**test\_ViewApplication\_setup\_restored\_sets\_viewapp\_as\_master** (*mock*)  
**test\_ViewApplication\_setup\_statusbar\_calls\_Statusbar\_place** (*mock*)

test\_ViewApplication\_setup\_statusbar\_initializes\_statusbar (mock)  
test\_ViewApplication\_setup\_statusbar\_sets\_viewapp\_as\_master (mock)  
test\_ViewApplication\_setup\_title\_calls\_label\_place (mock)  
test\_ViewApplication\_setup\_title\_sets\_title\_label (mock)  
test\_ViewApplication\_setup\_title\_sets\_tk\_variable (mock, name, typ)  
test\_ViewApplication\_setup\_toolbar\_calls\_Toolbar\_place (mock)  
test\_ViewApplication\_setup\_toolbar\_initializes\_Toolbar (mock)  
test\_ViewApplication\_setup\_toolbar\_sets\_viewapp\_as\_master (mock)  
test\_ViewApplication\_setup\_widgets\_calls\_setup\_corner (mock)  
test\_ViewApplication\_setup\_widgets\_calls\_setup\_icon (mock)  
test\_ViewApplication\_setup\_widgets\_calls\_setup\_name (mock)  
test\_ViewApplication\_setup\_widgets\_calls\_setup\_resizable (mock)  
test\_ViewApplication\_setup\_widgets\_calls\_setup\_restored (mock)  
test\_ViewApplication\_setup\_widgets\_calls\_setup\_statusbar (mock)  
test\_ViewApplication\_setup\_widgets\_calls\_setup\_title (mock)  
test\_ViewApplication\_setup\_widgets\_calls\_setup\_toolbar (mock)  
test\_ViewApplication\_setup\_widgets\_calls\_setup\_windows (mock)  
test\_ViewApplication\_setup\_widgets\_calls\_setup\_workspaces (mock)  
test\_ViewApplication\_setup\_windows\_calls\_WindowsList\_place (mock)  
test\_ViewApplication\_setup\_windows\_initializes\_WindowsList (mock)  
test\_ViewApplication\_setup\_windows\_sets\_viewapp\_as\_master (mock)  
test\_ViewApplication\_setup\_workspaces\_calls\_WorkspacesCollection\_place (mock)  
test\_ViewApplication\_setup\_workspaces\_initializes\_WorkspacesCollection (mock)  
test\_ViewApplication\_setup\_workspaces\_sets\_viewapp\_as\_master (mock)  
test\_ViewApplication\_show\_root\_calls\_master\_showing\_up\_method (mock, method)  
test\_ViewApplication\_startup\_calls\_configure\_on\_labels (mock)  
test\_ViewApplication\_startup\_calls\_focus\_set\_on\_view\_frame (mock)  
test\_ViewApplication\_startup\_calls\_place\_on\_view\_frame (mock)  
test\_ViewApplication\_startup\_calls\_show\_root (mock)  
test\_ViewApplication\_update\_widgets\_calls\_ImageTk\_PhotoImage (mock)  
test\_ViewApplication\_update\_widgets\_calls\_resizable\_set\_value (mock)  
test\_ViewApplication\_update\_widgets\_calls\_restored\_set\_value (mock)  
test\_ViewApplication\_update\_widgets\_calls\_workspaces\_select\_active (mock)  
test\_ViewApplication\_update\_widgets\_sets\_attr (mock, attr, val, typ)  
test\_ViewApplication\_update\_widgets\_sets\_icon (mock)  
test\_ViewApplication\_update\_widgets\_sets\_icon\_image (mock)

## tests.unit.test\_options – Unit tests for options classes and functions

```
class tests.unit.test_options.TestAboutDialog
```

```
    Bases: object
```

```
    Unit testing class for AboutDialog class.
```

```
    test_AboutDialog_init_calls_geometry_on_master_position (mockler)
```

```
    test_AboutDialog_init_calls_setup_widgets (mockler)
```

```
    test_AboutDialog_init_calls_super_with_master_arg (mockler)
```

```
    test_AboutDialog_init_sets_about_dialog_title (mockler)
```

```
    test_AboutDialog_init_sets_master_attribute (mockler)
```

```
    test_AboutDialog_inits_attributes (attr, value)
```

```
    test_AboutDialog_issubclass_of_Toplevel ()
```

```
    test_AboutDialog_on_help_click_opens_webbrowser (mockler)
```

```
    test_AboutDialog_on_releases_click_opens_webbrowser (mockler)
```

```
    test_AboutDialog_setup_widgets_calls_buttons_pack (mockler)
```

```
    test_AboutDialog_setup_widgets_calls_get_resized_image (mockler)
```

```
    test_AboutDialog_setup_widgets_calls_get_resource_path (mockler)
```

```
    test_AboutDialog_setup_widgets_calls_logo_label_pack (mockler)
```

```
    test_AboutDialog_setup_widgets_calls_notice_label_pack (mockler)
```

```
    test_AboutDialog_setup_widgets_calls_open (mockler)
```

```
    test_AboutDialog_setup_widgets_calls_separator_pack (mockler)
```

```
    test_AboutDialog_setup_widgets_calls_version_pack (mockler)
```

```
    test_AboutDialog_setup_widgets_sets_arrangeit_version_label (mockler)
```

```
    test_AboutDialog_setup_widgets_sets_exit_button (mockler)
```

```
    test_AboutDialog_setup_widgets_sets_gui_version_label (mockler)
```

```
    test_AboutDialog_setup_widgets_sets_help_button (mockler)
```

```
    test_AboutDialog_setup_widgets_sets_logo_label (mockler)
```

```
    test_AboutDialog_setup_widgets_sets_notice_label (mockler)
```

```
    test_AboutDialog_setup_widgets_sets_python_version_label (mockler)
```

```
    test_AboutDialog_setup_widgets_sets_releases_button (mockler)
```

```
    test_AboutDialog_setup_widgets_sets_separator (mockler)
```

```
    test_AboutDialog_setup_widgets_sets_system_version_label (mockler)
```

```
    test_AboutDialog_setup_widgets_sets_tclTk_version_label (mockler)
```

```
    test_OptionsDialog_init_calls_set_icon (mockler)
```

```
class tests.unit.test_options.TestCheckOption
```

```
    Bases: object
```

```
    Unit testing class for CheckOption class.
```

```

test_CheckOption_init_calls_super_with_master_arg (mock)
test_CheckOption_init_configs_attributes (mock)
test_CheckOption_init_deselects_for_initial_value_False (mock)
test_CheckOption_init_selects_for_initial_value_True (mock)
test_CheckOption_init_sets_change_callback_attribute (mock)
test_CheckOption_init_sets_master_attribute (mock)
test_CheckOption_init_sets_name_attribute (mock)
test_CheckOption_init_sets_var_attribute (mock)
test_CheckOption_inits_attributes (attr, value)
test_CheckOption_issubclass_of_Checkbutton ()
test_CheckOption_on_update_value_calls_master_change_setting (mock)
test_CheckOption_on_update_value_returns_break (mock)
class tests.unit.test_options.TestColorOption
    Bases: object
    Unit testing class for ColorOption class.
    test_CheckOption_init_instantiates_and_sets_label_attribute (mock)
    test_CheckOption_init_sets_change_callback_attribute (mock)
    test_ColorOption_init_calls_StringVar_set (mock)
    test_ColorOption_init_calls_super_with_provided_arguments (mock)
    test_ColorOption_init_sets_COLORS_as_initial_choices (mock)
    test_ColorOption_init_sets_master_attribute (mock)
    test_ColorOption_init_sets_name_attribute (mock)
    test_ColorOption_init_sets_var_attribute (mock)
    test_ColorOption_inits_attributes (attr, value)
    test_ColorOption_issubclass_of_OptionMenu ()
    test_ColorOption_on_update_value_calls_change_setting (mock)
    test_ColorOption_on_update_value_returns_break (mock)
class tests.unit.test_options.TestFloatScaleOption
    Bases: object
    Unit testing class for FloatScaleOption class.
    test_FloatScaleOption_init_multiplies_initial_by_100 (mock)
    test_FloatScaleOption_issubclass_of_ScaleOption ()
    test_FloatScaleOption_on_update_value_calls_master_change_setting (mock)
    test_FloatScaleOption_on_update_value_returns_break (mock)
class tests.unit.test_options.TestOptionsDialog
    Bases: object
    Unit testing class for OptionsDialog class.

```

test\_OptionsDialog\_change\_setting\_calls\_controller\_change\_setting (mocked)  
test\_OptionsDialog\_change\_setting\_calls\_set\_timer (mocked)  
test\_OptionsDialog\_change\_setting\_changes\_message\_var (mocked)  
test\_OptionsDialog\_change\_setting\_for\_float\_calls\_controller\_change\_setting (mocked)  
test\_OptionsDialog\_change\_setting\_not\_called\_upon\_startup (mocked)  
test\_OptionsDialog\_create\_frame\_instantiates\_ttk\_Frame (mocked)  
test\_OptionsDialog\_create\_frame\_returns\_frame (mocked)  
test\_OptionsDialog\_create\_separator\_instantiates\_ttk\_Separator (mocked)  
test\_OptionsDialog\_create\_separator\_instantiates\_vertical\_ttk\_Separator (mocked)  
test\_OptionsDialog\_create\_widget\_calls\_widget\_class\_from\_name (mocked)  
test\_OptionsDialog\_create\_widget\_instantiates\_float\_scale\_widget (mocked)  
test\_OptionsDialog\_create\_widget\_instantiates\_widget (mocked)  
test\_OptionsDialog\_create\_widget\_instantiates\_widget\_with\_kwargs (mocked)  
test\_OptionsDialog\_init\_calls\_geometry\_on\_root\_position (mocked)  
test\_OptionsDialog\_init\_calls\_set\_icon (mocked)  
test\_OptionsDialog\_init\_calls\_setup\_bindings (mocked)  
test\_OptionsDialog\_init\_calls\_setup\_widgets (mocked)  
test\_OptionsDialog\_init\_calls\_super\_with\_master\_arg (mocked)  
test\_OptionsDialog\_init\_sets\_master\_attribute (mocked)  
test\_OptionsDialog\_init\_sets\_options\_dialog\_title (mocked)  
test\_OptionsDialog\_inits\_attributes (attr, value)  
test\_OptionsDialog\_issubclass\_of\_Toplevel ()  
test\_OptionsDialog\_on\_destroy\_options\_destroys\_options (mocked)  
test\_OptionsDialog\_on\_destroy\_options\_shows\_root (mocked)  
test\_OptionsDialog\_on\_save\_default\_calls\_controller\_save (mocked)  
test\_OptionsDialog\_on\_save\_default\_calls\_set\_timer (mocked)  
test\_OptionsDialog\_on\_save\_default\_changes\_message\_var (mocked)  
test\_OptionsDialog\_on\_show\_about\_calls\_lift (mocked)  
test\_OptionsDialog\_on\_show\_about\_instantiates\_AboutDialog (mocked)  
test\_OptionsDialog\_set\_timer\_calls\_after (mocked)  
test\_OptionsDialog\_set\_timer\_calls\_after\_cancel\_if\_timer\_exists (mocked)  
test\_OptionsDialog\_set\_timer\_sets\_timer\_attribute (mocked)  
test\_OptionsDialog\_setup\_bindings\_binds\_callback (mocked, event, callback)  
test\_OptionsDialog\_setup\_files\_section\_calls\_button\_pack (mocked)  
test\_OptionsDialog\_setup\_files\_section\_inits\_LabelFrame (mocked)  
test\_OptionsDialog\_setup\_files\_section\_returns\_LabelFrame (mocked)

```

test_OptionsDialog_setup_files_section_sets_save_default_button (mock)
test_OptionsDialog_setup_section_calls_Frame_pack (mock)
test_OptionsDialog_setup_section_calls_Separator_pack (mock)
test_OptionsDialog_setup_section_calls_create_frame (mock)
test_OptionsDialog_setup_section_calls_create_separator (mock)
test_OptionsDialog_setup_section_calls_create_widget (mock)
test_OptionsDialog_setup_section_calls_widget_label_pack (mock)
test_OptionsDialog_setup_section_calls_widget_pack (mock)
test_OptionsDialog_setup_section_inits_LabelFrame (mock)
test_OptionsDialog_setup_section_returns_section (mock)
test_OptionsDialog_setup_widgets_calls_LabelFrame_pack (mock)
test_OptionsDialog_setup_widgets_calls_about_button_pack (mock)
test_OptionsDialog_setup_widgets_calls_label_pack (mock)
test_OptionsDialog_setup_widgets_calls_quit_button_pack (mock)
test_OptionsDialog_setup_widgets_calls_setup_files_section (mock)
test_OptionsDialog_setup_widgets_calls_setup_section_for_appearance (mock)
test_OptionsDialog_setup_widgets_calls_setup_section_for_colors (mock)
test_OptionsDialog_setup_widgets_sets_about_button (mock)
test_OptionsDialog_setup_widgets_sets_label_for_message (mock)
test_OptionsDialog_setup_widgets_sets_message_var (mock)
test_OptionsDialog_setup_widgets_sets_quit_button (mock)
test_OptionsDialog_widget_class_from_name_calls_setting_type (mock)
test_OptionsDialog_widget_class_from_name_for_typ_None (mock)
test_OptionsDialog_widget_class_from_name_returns_related_class (mock,
                                                                    name,
                                                                    typ)

class tests.unit.test_options.TestOptionsModule
    Bases: object
    Unit testing class for options module and OptionsMetaclass
    test_options_module_CLASSES ()
    test_options_module_COLORS_is_dictionary ()
    test_options_module_MESSAGES_is_dictionary ()
    test_options_module_WIDGETS_has_valid_format_for_all ()
    test_options_module_WIDGETS_is_dictionary ()

class tests.unit.test_options.TestScaleOption
    Bases: object
    Unit testing class for ScaleOption class.
    test_ScaleOption_init_calls_super_with_master_arg (mock)

```

```

test_ScaleOption_init_configs_attributes (mock)
test_ScaleOption_init_configs_command (mock)
test_ScaleOption_init_sets_change_callback_attribute (mock)
test_ScaleOption_init_sets_initial (mock)
test_ScaleOption_init_sets_master_attribute (mock)
test_ScaleOption_init_sets_name_attribute (mock)
test_ScaleOption_inits_attributes (attr, value)
test_ScaleOption_issubclass_of_Scale ()
test_ScaleOption_on_update_value_calls_master_change_setting (mock)
test_ScaleOption_on_update_value_returns_break (mock)
class tests.unit.test_options.TestThemeOption
  Bases: object
  Unit testing class for ThemeOption class.
  test_ThemeOption_init_sets_initial_BG_from_Settings (mock)
  test_ThemeOption_init_sets_initial_FG_from_Settings (mock)
  test_ThemeOption_issubclass_of_ColorOption ()

```

#### tests.unit.test\_utils – Unit tests for helper functions

```

class tests.unit.test_utils.TestUtils
  Bases: object
  Testing class for arrangeit.utils module.
  test_utils_Rectangle_is_namedtuple_class ()
  test_utils_get_snapping_source_by_ordinal_ordinal (rect, expected)
  test_utils_get_snapping_source_by_ordinal_returns_Rectangle ()
  test_utils_check_intersections_calls_intersects_twice_and_returns_two_tuple (mock)
  test_utils_check_intersections_single_calls_intersects_and_returns_False (mock)
  test_utils_check_intersections_single_functionality_for_full_sources (sources,
                                                                    tar-
                                                                    gets,
                                                                    ex-
                                                                    pected)
  test_utils_check_intersections_single_functionality_for_two_sources_corner_0 (sources,
                                                                    tar-
                                                                    gets,
                                                                    ex-
                                                                    pected)
  test_utils_check_intersections_single_functionality_for_two_sources_corner_1 (sources,
                                                                    tar-
                                                                    gets,
                                                                    ex-
                                                                    pected)

```

**test\_utils\_check\_intersections\_single\_functionality\_for\_two\_sources\_corner\_2** (*sources,*  
*tar-*  
*gets,*  
*ex-*  
*pected*)

**test\_utils\_check\_intersections\_single\_functionality\_for\_two\_sources\_corner\_3** (*sources,*  
*tar-*  
*gets,*  
*ex-*  
*pected*)

**test\_utils\_get\_class\_involves\_default\_val\_for\_no\_arg** (*mocked, name*)

**test\_utils\_get\_component\_class\_calls\_get\_class** (*mocked, function*)

**test\_utils\_get\_component\_class\_involves\_provided\_argument** (*mocked, name*)

**test\_utils\_get\_component\_class\_raises\_SystemExit\_for\_invalid\_platform** (*platform,*  
*name*)

**test\_utils\_get\_cursor\_name\_functionality** (*corner, with\_arrow, expected*)

**test\_utils\_get\_prepared\_screenshot\_calls\_filter** (*mocked*)

**test\_utils\_get\_prepared\_screenshot\_calls\_filter\_with\_blur\_size** (*mocked*)

**test\_utils\_get\_prepared\_screenshot\_converts\_to\_grayscale\_if\_set** (*mocked*)

**test\_utils\_get\_prepared\_screenshot\_not\_converting\_to\_grayscale** (*mocked*)

**test\_utils\_get\_prepared\_screenshot\_returns\_ImageTk\_PhotoImage** (*mocked*)

**test\_utils\_get\_resized\_image\_calls\_Image\_open** (*mocked*)

**test\_utils\_get\_resized\_image\_calls\_Image\_resize** (*mocked*)

**test\_utils\_get\_resized\_image\_calls\_and\_returns\_PhotoImage** (*mocked*)

**test\_utils\_get\_resized\_image\_calls\_get\_resource\_path** (*mocked*)

**test\_utils\_get\_resource\_path\_calls\_os\_path\_dirname** (*mocked*)

**test\_utils\_get\_resource\_path\_calls\_os\_path\_join** (*mocked*)

**test\_utils\_get\_resource\_path\_returns\_os\_path\_value** ()

**test\_utils\_get\_snapping\_sources\_for\_rect\_corner\_0** (*rect, expected*)

**test\_utils\_get\_snapping\_sources\_for\_rect\_corner\_1** (*rect, expected*)

**test\_utils\_get\_snapping\_sources\_for\_rect\_corner\_2** (*rect, expected*)

**test\_utils\_get\_snapping\_sources\_for\_rect\_corner\_3** (*rect, expected*)

**test\_utils\_get\_snapping\_sources\_for\_rect\_corner\_None** (*rect, expected*)

**test\_utils\_get\_value\_if\_valid\_type\_for\_collection\_returns\_empty** (*value,*  
*typ*)

**test\_utils\_get\_value\_if\_valid\_type\_for\_collection\_type\_returns\_value** (*value,*  
*typ*)

**test\_utils\_get\_value\_if\_valid\_type\_for\_single\_type\_returns\_None** (*value,*  
*typ*)

**test\_utils\_get\_value\_if\_valid\_type\_for\_single\_type\_returns\_value** (*value,*  
*typ*)



```

test_utils_get_value_if_valid_type_returns_None_for_None_value (value, typ)
test_utils_increased_by_fraction (value, fraction, expected)
test_utils_intersects_functionality (source, target, expected)
test_utils_offset_for_intersecting_pair_corner_0_functionality (pair,
                                                                offset)
test_utils_offset_for_intersecting_pair_corner_1_functionality (pair,
                                                                offset)
test_utils_offset_for_intersecting_pair_corner_2_functionality (pair,
                                                                offset)
test_utils_offset_for_intersecting_pair_corner_3_functionality (pair,
                                                                offset)

test_utils_offset_for_intersecting_pair_returns_False (mockler)
test_utils_offset_for_intersections_calls_offset_once_for_single_pair (mockler)
test_utils_offset_for_intersections_calls_offset_twice_for_two_pairs (mockler)
test_utils_offset_for_intersections_returns_empty_tuple_for_no_rectangles (mockler)
test_utils_offset_for_intersections_returns_opposite_tuple_element (mockler)
test_utils_open_image_calls_ImageOps_colorize (mockler)
test_utils_open_image_calls_Image_convert (mockler)
test_utils_open_image_calls_Image_open (mockler)
test_utils_open_image_calls_different_ImageOps_colorize_if_colorized_set (mockler)
test_utils_open_image_calls_get_resource_path (mockler)
test_utils_open_image_returns_Image (mockler)
test_utils_platform_path_returns_lowercased_system_name (mockler, name)
test_utils_platform_user_data_path_calls_import_module (mockler)
test_utils_platform_user_data_path_calls_user_data_path (mockler)
test_utils_quarter_by_smaller (w, h, size, expected)
test_utils_quarter_by_smaller_out_of_range (size)
test_utils_set_icon_calls_PhotoImage (mockler)
test_utils_set_icon_calls_get_resource_path (mockler)
test_utils_set_icon_calls_tk_call (mockler)

```

## tests.unit.test\_settings – Unit tests for programs settings

```

class tests.unit.test_settings.TestSettings
    Bases: object

    Unit testing class for Settings.

    test_Settings_availability_for_all_constants_in_SETTINGS ()
    test_Settings_color_group_returns_empty_list_for_no_group (mockler)
    test_Settings_color_group_returns_list (mockler)

```

```

test_Settings_color_group_returns_type_for_valid_setting_name (mocked,
                                                                group,
                                                                expected)

test_Settings_initializes_blank_icon ()

test_Settings_initializes_unchangeable_core_program_constant (constant,
                                                                value)

test_Settings_is_setting_returns_False_for_core_setting (mocked)

test_Settings_is_setting_returns_False_for_invalid_setting (mocked)

test_Settings_is_setting_returns_False_for_invalid_value_type (mocked)

test_Settings_is_setting_returns_False_for_value_None (mocked)

test_Settings_is_setting_returns_True_for_valid_setting (mocked)

test_Settings_metaclass_is_SettingsMetaclass ()

test_Settings_setting_type_returns_None_for_invalid (mocked)

test_Settings_setting_type_returns_type_for_valid (mocked)

test_Settings_setting_type_returns_type_for_valid_setting_name (mocked,
                                                                name, typ)

class tests.unit.test_settings.TestSettingsModule
    Bases: object
    Unit testing class for settings module and SettingsMetaclass

test_SettingsMetaclass__getattr__calls_validate_user_settings_just_once (mocked)

test_SettingsMetaclass__getattr__not_changing_core_constant (constant)

test_SettingsMetaclass__getattr__returns_None_for_invalid_name ()

test_SettingsMetaclass__getattr__uses_SETTINGS_for_no_user_setting ()

test_SettingsMetaclass__getattr__uses_user_settings ()

test_SettingsMetaclass_defines__getattr__ ()

test_SettingsMetaclass_is_metaclass ()

test_settings_module_SETTINGS_for_value_type ()

test_settings_module_SETTINGS_has_valid_format_for_all ()

test_settings_module_SETTINGS_is_dictionary ()

test_settings_module_initializes_MESSAGES ()

test_settings_module_initializes_MESSAGES_key (key)

test_settings_module_initializes_SETTINGS ()

test_settings_read_user_settings_calls_json_load (mocked)

test_settings_read_user_settings_calls_platform_user_data_path (mocked)

test_settings_read_user_settings_checks_if_directory_exists (mocked)

test_settings_read_user_settings_returns_dictionary ()

test_settings_read_user_settings_returns_empty_for_exception (mocked)

test_settings_read_user_settings_returns_empty_if_not_exists (mocked)

```

```
test_settings_read_user_settings_returns_read_data_dictionary (mock)
test_settings_validate_user_settings_returns_dictionary ()
test_settings_validate_user_settings_returns_from_read_user (mock)
test_settings_validate_user_settings_returns_only_valid_names_from_read_user (mock)
test_settings_validate_user_settings_returns_only_valid_types_from_read_user (mock)
```

## tests.unit.test\_darwin – Unit tests for Mac OS specific code

```
class tests.unit.test_darwin.TestDarwinApp
```

```
    Bases: object
```

```
    Testing class for arrangeit.darwin.app.App class.
```

```
    test_DarwinApp_activate_root_returns (mock)
```

```
    test_DarwinApp_grab_window_screen_returns (mock)
```

```
    test_DarwinApp_move_and_resize_returns (mock)
```

```
    test_DarwinApp_move_calls_and_returns_move_and_resize (mock)
```

```
    test_DarwinApp_move_to_workspace_returns (mock)
```

```
class tests.unit.test_darwin.TestDarwinCollector
```

```
    Bases: object
```

```
    Testing class for arrangeit.darwin.collector.Collector class.
```

```
    test_DarwinCollector_get_application_icon_calls_Image_open (mock)
```

```
    test_DarwinCollector_get_application_icon_calls_running_apps_ids (mock)
```

```
    test_DarwinCollector_get_application_icon_calls_io_BytesIO (mock)
```

```
    test_DarwinCollector_get_application_icon_returns_Image (mock)
```

```
    test_DarwinCollector_get_window_geometry_calls_valueForKey_element  
                                                                    el-  
                                                                    e-  
                                                                    ment)
```

```
    test_DarwinCollector_get_window_geometry_calls_valueForKey_for_bounds (mock)
```

```
    test_DarwinCollector_get_window_geometry_returns_tuple_of_ints (mock,  
                                                                    element)
```

```
    test_DarwinCollector_get_window_id_calls_valueForKey_ (mock)
```

```
    test_DarwinCollector_get_window_title_calls_valueForKey_ (mock)
```

```
    test_DarwinCollector_running_apps_ids_calls_runningApplications (mock)
```

```
    test_DarwinCollector_running_apps_ids_calls_sharedWorkspace (mock)
```

```
    test_DarwinCollector_running_apps_ids_functionality (mock)
```

```
    test_DarwinCollector_add_window_calls_WindowsCollection_add (mock)
```

```
    test_DarwinCollector_add_window_calls_methods (mock, method)
```

```
    test_DarwinCollector_add_window_inits_WindowModel (mock)
```

```
    test_DarwinCollector_check_window_calls (mock, method)
```

```

test_DarwinCollector_check_window_returns_False_for_not_applicable (mock)
test_DarwinCollector_check_window_returns_False_for_not_valid_state (mock)
test_DarwinCollector_check_window_returns_True (mock)
test_DarwinCollector_get_application_name_calls_valueForKey_ (mock)
test_DarwinCollector_get_available_workspaces_returns (mock)
test_DarwinCollector_get_monitors_rects_calls_NSScreen_screens (mock)
test_DarwinCollector_get_monitors_rects_returns_list_of_rect_parts (mock)
test_DarwinCollector_get_windows_calls_CGWindowListCopyWindowInfo (mock)
test_DarwinCollector_get_windows_returns_list (mock)
test_DarwinCollector_get_workspace_number_for_window_returns (mock)
test_DarwinCollector_is_applicable_calls__running_apps_ids (mock)
test_DarwinCollector_is_applicable_functionality_for_empty_name (mock)
test_DarwinCollector_is_applicable_functionality_for_name_None (mock)
test_DarwinCollector_is_applicable_functionality_for_non_empty_name (mock)
test_DarwinCollector_is_applicable_functionality_for_not_owner (mock)
test_DarwinCollector_is_applicable_functionality_for_owner (mock)
test_DarwinCollector_is_resizable_returns (mock)
test_DarwinCollector_is_restored_returns (mock)
test_DarwinCollector_is_valid_state_returns (mock)

```

```

class tests.unit.test_darwin.TestDarwinUtils

```

```

    Bases: object

```

```

    Testing class for arrangeit.darwin.utils module.

```

```

    test_darwin_utils_user_data_path_calls_NSSearchPathForDirectoriesInDomains (mock)

```

```

    test_darwin_utils_user_data_path_calls_os_path_join (mock)

```

## tests.unit.test\_linux – Unit tests for GNU/Linux specific code

```

class tests.unit.test_linux.TestLinuxApp

```

```

    Bases: object

```

```

    Testing class for arrangeit.linux.app.App class.

```

```

    test_LinuxApp__activate_workspace_calls_get_wnck_workspace_for_custom_number (mock)

```

```

    test_LinuxApp__activate_workspace_calls_workspace_activate (mock)

```

```

    test_LinuxApp__move_window_to_workspace_calls_Wnck_shutdown (mock)

```

```

    test_LinuxApp__move_window_to_workspace_calls__activate_workspace (mock)

```

```

    test_LinuxApp__move_window_to_workspace_calls_get_window_by_wid (mock)

```

```

    test_LinuxApp__move_window_to_workspace_calls_win_activate (mock)

```

```

    test_LinuxApp__move_window_to_workspace_calls_win_move_to_workspace (mock)

```

test\_LinuxApp\_move\_window\_to\_workspace\_returns\_False (mockery)

test\_LinuxApp\_move\_window\_to\_workspace\_returns\_True (mockery)

test\_LinuxApp\_window\_from\_wid\_calls\_get\_default (mockery)

test\_LinuxApp\_window\_from\_wid\_calls\_get\_window\_stack (mockery)

test\_LinuxApp\_window\_from\_wid\_calls\_get\_xid (mockery)

test\_LinuxApp\_window\_from\_wid\_returns\_None (mockery)

test\_LinuxApp\_window\_from\_wid\_returns\_window\_instance (mockery)

test\_LinuxApp\_activate\_root\_calls\_window\_from\_wid (mockery)

test\_LinuxApp\_activate\_root\_calls\_window\_focus (mockery)

test\_LinuxApp\_grab\_window\_screen\_calls\_window\_from\_wid (mockery)

test\_LinuxApp\_grab\_window\_screen\_calls\_get\_prepared\_screenshot (mockery)

test\_LinuxApp\_grab\_window\_screen\_calls\_pixbuf\_get\_from\_window (mockery)

test\_LinuxApp\_grab\_window\_screen\_for\_no\_window\_returns\_empty\_icon (mockery)

test\_LinuxApp\_grab\_window\_screen\_returns\_get\_prepared\_screenshot\_image (mockery)

test\_LinuxApp\_move\_and\_resize\_calls\_WnckWindow\_set\_geometry (mockery)

test\_LinuxApp\_move\_and\_resize\_calls\_move\_window\_to\_workspace (mockery)

test\_LinuxApp\_move\_and\_resize\_calls\_get\_model\_by\_wid (mockery)

test\_LinuxApp\_move\_and\_resize\_calls\_get\_window\_by\_wid (mockery)

test\_LinuxApp\_move\_and\_resize\_calls\_get\_window\_move\_resize\_mask (mockery)

test\_LinuxApp\_move\_and\_resize\_calls\_is\_minimized (mockery)

test\_LinuxApp\_move\_and\_resize\_calls\_minimize (mockery)

test\_LinuxApp\_move\_and\_resize\_calls\_unmaximize (mockery)

test\_LinuxApp\_move\_and\_resize\_calls\_unminimize (mockery)

test\_LinuxApp\_move\_and\_resize\_checks\_maximized (mockery)

test\_LinuxApp\_move\_and\_resize\_not\_calling\_WnckWindow\_set\_geometry (mockery)

test\_LinuxApp\_move\_and\_resize\_not\_calling\_move\_window\_to\_workspace (mockery)

test\_LinuxApp\_move\_and\_resize\_not\_calling\_get\_window\_by\_wid (mockery)

test\_LinuxApp\_move\_and\_resize\_not\_calling\_minimize\_not\_minimized (mockery)

test\_LinuxApp\_move\_and\_resize\_not\_calling\_minimize\_not\_restored (mockery)

test\_LinuxApp\_move\_and\_resize\_not\_calling\_unmaximize (mockery)

test\_LinuxApp\_move\_and\_resize\_not\_calling\_unminimize\_not\_minimized (mockery)

test\_LinuxApp\_move\_and\_resize\_not\_calling\_unminimize\_not\_restored (mockery)

test\_LinuxApp\_move\_and\_resize\_returns\_False (mockery)

test\_LinuxApp\_move\_and\_resize\_returns\_True (mockery)

test\_LinuxApp\_move\_calls\_move\_and\_resize (mockery)

test\_LinuxApp\_move\_to\_workspace\_calls\_move\_window\_to\_workspace (mockery)

```

class tests.unit.test_linux.TestLinuxCollector
    Bases: object

    Testing class for arrangeit.linux.collector.Collector class.

    test_LinuxCollector__check_mask_part_functionality(mock, rect, changed, expected)

    test_LinuxCollector__get_available_wnck_workspaces_calls_Screen_methods(mock, method)

    test_LinuxCollector__get_available_wnck_workspaces_returns_list(mock)

    test_LinuxCollector_add_window_calls_WindowsCollection_add(mock)

    test_LinuxCollector_add_window_calls_Wnck_Window_methods(mock, method)

    test_LinuxCollector_add_window_calls_get_image_from_pixbuf(mock)

    test_LinuxCollector_add_window_calls_get_workspace_number_for_window(mock)

    test_LinuxCollector_add_window_calls_is_resizable(mock)

    test_LinuxCollector_add_window_calls_is_restored(mock)

    test_LinuxCollector_add_window_inits_WindowModel(mock)

    test_LinuxCollector_check_window_calls_W_get_state(mock)

    test_LinuxCollector_check_window_calls_W_get_window_type(mock)

    test_LinuxCollector_check_window_calls_is_applicable(mock)

    test_LinuxCollector_check_window_calls_is_valid_state(mock)

    test_LinuxCollector_check_window_returns_False_for_invalid_state(mock)

    test_LinuxCollector_check_window_returns_False_for_not_is_app(mock)

    test_LinuxCollector_check_window_returns_True_for_both_True(mock)

    test_LinuxCollector_get_available_workspaces_calls_W_workspace_get_name(mock)

    test_LinuxCollector_get_available_workspaces_calls__get_available_wnck(mock)

    test_LinuxCollector_get_available_workspaces_calls_get_workspace_number(mock)

    test_LinuxCollector_get_available_workspaces_functionality(mock)

    test_LinuxCollector_get_available_workspaces_returns_list(mock)

    test_LinuxCollector_get_available_workspaces_returns_one_element(mock)

    test_LinuxCollector_get_image_from_pixbuf_returns_valid_type()

    test_LinuxCollector_get_monitors_rects_calls_GDK_display_get_default(mock)

    test_LinuxCollector_get_monitors_rects_calls_GDK_display_get_monitor(mock)

    test_LinuxCollector_get_monitors_rects_calls_GDK_display_get_n_monitors(mock)

    test_LinuxCollector_get_monitors_rects_calls_GDK_monitor_get_workarea(mock)

    test_LinuxCollector_get_monitors_rects_returns_list_of_rects(mock)

    test_LinuxCollector_get_window_by_wid_calls_Wnck_Window_get(mock)

    test_LinuxCollector_get_window_move_resize_mask_calls__check_mask_part(mock)

    test_LinuxCollector_get_windows_calls_Screen_methods(mock, method)

```

```

test_LinuxCollector_get_wnck_workspace_for_custom_number_calls_get_available (mocked)
test_LinuxCollector_get_wnck_workspace_for_custom_number_calls_get_w_number (mocked)
test_LinuxCollector_get_workspace_number_for_window_calls_W_get_workspace (mocked)
test_LinuxCollector_get_workspace_number_for_window_calls_wn_for_window (mocked)
test_LinuxCollector_get_workspace_number_returns_0 (mocked)
test_LinuxCollector_get_workspace_number_returns_correct_number (mocked,
                                                                    screen,
                                                                    workspace,
                                                                    ex-
                                                                    pected)

test_LinuxCollector_is_applicable (window_type, value)
test_LinuxCollector_is_resizable (window_type, value)
test_LinuxCollector_is_restored (mocked)
test_LinuxCollector_is_valid_state (window_type, window_state, value)
test_LinuxCollector_rect_is_converted_to_tuple (mocked)
test_LinuxCollector_run_functionality (mocked, is_applicable, is_valid_state, value)
test_LinuxCollector_run_super (mocked)

class tests.unit.test_linux.TestLinuxController
    Bases: object

    Testing class for arrangeit.linux.controller.Controller class.

    test_LinuxController_setup_root_window_calls_type_splash (mocked)

class tests.unit.test_linux.TestLinuxUtils
    Bases: object

    Testing class for arrangeit.linux.utils module.

    test_linux_utils_module_user_data_path_for_local_share_not_exists (mocked)
    test_linux_utils_module_user_data_path_checks_local_share_first (mocked)

```

### tests.unit.test\_windows – Unit tests for MS Windows specific code

```

class tests.unit.test_windows.TestWindowsApp
    Bases: object

    Testing class for arrangeit.windowe.app.App class.

    test_WindowsApp_screenshot_with_thumbnails_calls_Rectangle_lower (mocked)
    test_WindowsApp_screenshot_with_thumbnails_calls_Rectangle_right (mocked)
    test_WindowsApp_screenshot_with_thumbnails_calls_setup_thumbnail_lower (mocked)
    test_WindowsApp_screenshot_with_thumbnails_calls_setup_thumbnail_right (mocked)
    test_WindowsApp_screenshot_with_thumbnails_returns_window_area_desktop_sc (mocked)
    test_WindowsApp_screenshot_with_thumbnails_returns_blank_lower (mocked)
    test_WindowsApp_screenshot_with_thumbnails_returns_blank_right (mocked)

```

```

test_WindowsApp_screenshot_with_thumbnails_sets_thumbnails_attribute (mock)
test_WindowsApp_window_area_desktop_screenshot_calls_and_returns_grab (mock)
test_WindowsApp_window_area_desktop_screenshot_calls_extended_frame_rect (mock)
test_WindowsApp_activate_root_calls_SetActiveWindow (mock)
test_WindowsApp_grab_window_screen_calls_screenshot_with_thumbnails (mock)
test_WindowsApp_grab_window_screen_calls_get_prepared_screenshot (mock)
test_WindowsApp_grab_window_screen_calls_is_dwm_composition_enabled (mock)
test_WindowsApp_grab_window_screen_returns_blank (mock)
test_WindowsApp_inits_thumbnails_as_empty_tuple ()
test_WindowsApp_move_and_resize_calls_IsIconic (mock)
test_WindowsApp_move_and_resize_calls_MoveWindow (mock)
test_WindowsApp_move_and_resize_calls_ShowWindow_if_iconic (mock)
test_WindowsApp_move_and_resize_calls_ShowWindow_minimized (mock)
test_WindowsApp_move_and_resize_calls_get_model_by_wid (mock)
test_WindowsApp_move_and_resize_calls_move_other_to_workspace (mock)
test_WindowsApp_move_and_resize_not_calling_MoveWindow (mock)
test_WindowsApp_move_and_resize_not_calling_ShowWindow_if_not_iconic (mock)
test_WindowsApp_move_and_resize_not_calling_ShowWindow_minimized (mock)
test_WindowsApp_move_and_resize_not_calling_move_other_to_workspace (mock)
test_WindowsApp_move_and_resize_returns_False (mock)
test_WindowsApp_move_and_resize_returns_True (mock)
test_WindowsApp_move_calls_move_and_resize (mock)
test_WindowsApp_move_other_to_workspace_calls_api_move_other_window_to_desktop (mock)
test_WindowsApp_move_to_workspace_calls_and_returns_api_move_own_window_to_desktop (mock)
test_WindowsApp_screenshot_cleanup_calls_unregister_thumbnail (mock)
test_WindowsApp_screenshot_cleanup_not_calling_unregister_thumbnail_for_empty (mock)
test_WindowsApp_screenshot_cleanup_sets_thumbnails_attribute_to_empty_tuple (mock)
class tests.unit.test_windows.TestWindowsCollector
  Bases: object
  Testing class for arrangeit.windows.collector.Collector class.
  test_WindowsCollector_get_application_icon_calls_GetClassLong (mock)
  test_WindowsCollector_get_application_icon_calls_SendMessageTimeout (mock)
  test_WindowsCollector_get_application_icon_calls_get_image_from_icon_handle (mock)
  test_WindowsCollector_get_application_icon_calls_get_uwpapp_icon (mock)
  test_WindowsCollector_get_image_from_icon_handle_calls_CreateBitmap (mock)
  test_WindowsCollector_get_image_from_icon_handle_calls_CreateDCFromHandle (mock)

```



test\_WindowsCollector\_get\_image\_from\_icon\_handle\_calls\_GetDC (mock)

test\_WindowsCollector\_get\_image\_from\_icon\_handle\_calls\_Image\_frombuffer (mock)

test\_WindowsCollector\_get\_image\_from\_icon\_handle\_calls\_bitmap\_CreateCompatibleBitmap (mock)

test\_WindowsCollector\_get\_image\_from\_icon\_handle\_calls\_bitmap\_GetBitmapBits (mock)

test\_WindowsCollector\_get\_image\_from\_icon\_handle\_calls\_dc\_CreateCompatibleDC (mock)

test\_WindowsCollector\_get\_image\_from\_icon\_handle\_calls\_dc\_DrawIcon (mock)

test\_WindowsCollector\_get\_image\_from\_icon\_handle\_calls\_dc\_SelectObject (mock)

test\_WindowsCollector\_get\_uwpapp\_icon\_calls\_get\_package (mock)

test\_WindowsCollector\_get\_uwpapp\_icon\_returns\_icon (mock)

test\_WindowsCollector\_get\_uwpapp\_icon\_sets\_api\_packages\_for\_hwnd (mock)

test\_WindowsCollector\_get\_window\_geometry\_calls\_GetWindowPlacement (mock)

test\_WindowsCollector\_get\_window\_geometry\_calls\_Rectangle (mock)

test\_WindowsCollector\_get\_window\_geometry\_calls\_extended\_frame\_rect (mock)

test\_WindowsCollector\_get\_window\_geometry\_calls\_is\_dwm\_composition\_enabled (mock)

test\_WindowsCollector\_get\_window\_geometry\_returns\_tuple\_rect (mock)

test\_WindowsCollector\_get\_window\_title\_calls (mock, method)

test\_WindowsCollector\_get\_window\_title\_functionality (mock, value)

test\_WindowsCollector\_init\_calls\_super (mock)

test\_WindowsCollector\_init\_initializes\_Api\_and\_sets\_it\_as\_attribute (mock)

test\_WindowsCollector\_is\_activable\_calls\_window\_info\_extended\_style (mock)

test\_WindowsCollector\_is\_activable\_return (mock, value, expected)

test\_WindowsCollector\_is\_alt\_tab\_applicable\_calls\_IsWindowVisible (mock)

test\_WindowsCollector\_is\_alt\_tab\_applicable\_calls\_get\_ancestor\_by\_type (mock)

test\_WindowsCollector\_is\_alt\_tab\_applicable\_calls\_get\_last\_active\_popup (mock)

test\_WindowsCollector\_is\_alt\_tab\_applicable\_return\_False (mock)

test\_WindowsCollector\_is\_alt\_tab\_applicable\_return\_True (mock)

test\_WindowsCollector\_is\_cloaked\_calls\_and\_returns\_is\_cloaked (mock)

test\_WindowsCollector\_is\_tool\_window\_calls (mock, method)

test\_WindowsCollector\_is\_tool\_window\_return (mock, value, expected)

test\_WindowsCollector\_is\_tray\_window\_calls\_title\_info\_state (mock)

test\_WindowsCollector\_is\_tray\_window\_return (mock, value, expected)

test\_WindowsCollector\_add\_window\_calls\_WindowsCollection\_add (mock)

test\_WindowsCollector\_add\_window\_calls\_methods (mock, method)

test\_WindowsCollector\_add\_window\_inits\_WindowModel (mock)

test\_WindowsCollector\_check\_window\_calls (mock, method)

test\_WindowsCollector\_check\_window\_functionality (mock, is\_applicable, is\_valid\_state, expected)

```

test_WindowsCollector_get_application_name_calls (mock, method)
test_WindowsCollector_get_application_name_calls_executable_name_for_hwnd (mock)
test_WindowsCollector_get_application_name_existing_package (mock)
test_WindowsCollector_get_application_name_functionality_no_package_no_app_name (mock,
value)
test_WindowsCollector_get_application_name_returns_GetClassName_for_exe_None (mock)
test_WindowsCollector_get_application_name_returns_executable_name_for_hwnd (mock)
test_WindowsCollector_get_available_calls_and_returns_api_get_desktops (mock)
test_WindowsCollector_get_monitors_rects_calls_EnumDisplayMonitors (mock)
test_WindowsCollector_get_monitors_rects_returns_list_of_rect_parts (mock)
test_WindowsCollector_get_windows_calls_api_enum_windows (mock)
test_WindowsCollector_get_workspace_number_for_window_returns_api_get_ordinal (mock)
test_WindowsCollector_is_applicable_calls (mock, method)
test_WindowsCollector_is_applicable_returns_False_for__is_tool_window (mock)
test_WindowsCollector_is_applicable_returns_False_for__is_tray_window (mock)
test_WindowsCollector_is_applicable_returns_False_for_not_IsWindow (mock)
test_WindowsCollector_is_applicable_returns_False_for_not_IsWindowEnabled (mock)
test_WindowsCollector_is_applicable_returns_False_for_not_IsWindowVisible (mock)
test_WindowsCollector_is_applicable_returns_False_for_not__is_alt_tab_applicable (mock)
test_WindowsCollector_is_applicable_returns_True (mock)
test_WindowsCollector_is_resizable_calls (mock, method)
test_WindowsCollector_is_resizable_return (mock, value, expected)
test_WindowsCollector_is_restored_calls (mock, method)
test_WindowsCollector_is_restored_return (mock, value, expected)
test_WindowsCollector_is_valid_state_calls__is_activable (mock)
test_WindowsCollector_is_valid_state_calls__is_cloaked (mock)
test_WindowsCollector_is_valid_state_return_value_for_activable (mock,
method,
value, ex-
pected)
test_WindowsCollector_is_valid_state_return_value_for_cloaked (mock,
method,
value,
expected)
test_WindowsCollector_run_functionality (mock, is_applicable, is_valid_state, value)
class tests.unit.test_windows.TestWindowsController
Bases: object
Testing class for arrangeit.windows.controller.Controller class.
test_WindowsController_inits_screenshot_when_exposed_as_True ()

```

```

    test_WindowsController_setup_root_window_calls_root_overrideredirect (mock)
    test_WindowsController_setup_root_window_calls_super (mock)
class tests.unit.test_windows.TestWindowsUtils
    Bases: object
    Testing class for arrangeit.windows.utils module.
    test_windows_utils_module_extract_name_from_bytes_path_calls_basename (mock)
    test_windows_utils_module_extract_name_from_bytes_path_calls_splitext (mock)
    test_windows_utils_module_extract_name_from_bytes_path_decode (mock)
    test_windows_utils_module_extract_name_from_bytes_path_functionality (path,
                                                                    name)
    test_windows_utils_module_extract_name_from_bytes_path_getdefaultencoding (mock)
    test_windows_utils_module_user_data_path (mock)

```

#### tests.unit.test\_windows\_api – Unit tests for MS Windows api helpers module

```

class tests.unit.test_windows_api.TestDWM_THUMBNAI_PROPERTIES
    Bases: object
    Testing class for arrangeit.windows.api.DWM_THUMBNAI_PROPERTIES class.
    test_windows_api_DWM_THUMBNAI_PROPERTIES_field_and_type (field, typ)
    test_windows_api_DWM_THUMBNAI_PROPERTIES_inits__fields_()
    test_windows_api_DWM_THUMBNAI_PROPERTIES_is_Structure_subclass()
class tests.unit.test_windows_api.TestDummyVirtualDesktops
    Bases: object
    Testing class for arrangeit.windows.api.DummyVirtualDesktops.
    test_api_DummyVirtualDesktops_defines_get_desktops (mock)
    test_api_DummyVirtualDesktops_defines_get_window_desktop (mock)
    test_api_DummyVirtualDesktops_defines_is_window_in_current_desktop (mock)
    test_api_DummyVirtualDesktops_defines_move_other_window_to_desktop (mock)
    test_api_DummyVirtualDesktops_defines_move_own_window_to_desktop (mock)
class tests.unit.test_windows_api.TestPACKAGE_ID
    Bases: object
    Testing class for arrangeit.windows.api.PACKAGE_ID class.
    test_windows_api_PACKAGE_ID_field_and_type (field, typ)
    test_windows_api_PACKAGE_ID_inits__fields_()
    test_windows_api_PACKAGE_ID_is_Structure_subclass()
class tests.unit.test_windows_api.TestPACKAGE_INFO
    Bases: object
    Testing class for arrangeit.windows.api.PACKAGE_INFO class.
    test_windows_api_PACKAGE_INFO_field_and_type (field, typ)

```

```

    test_windows_api_PACKAGE_INFO_inits__fields_()
    test_windows_api_PACKAGE_INFO_is_Structure_subclass()
class tests.unit.test_windows_api.TestPACKAGE_INFO_REFERENCE
    Bases: object
    Testing class for arrangeit.windows.api.PACKAGE_INFO_REFERENCE class.
    test_windows_api_PACKAGE_INFO_REFERENCE_field_and_type (field, typ)
    test_windows_api_PACKAGE_INFO_REFERENCE_inits__fields_()
    test_windows_api_PACKAGE_INFO_REFERENCE_is_Structure_subclass()
class tests.unit.test_windows_api.TestPACKAGE_SUBVERSION
    Bases: object
    Testing class for arrangeit.windows.api.PACKAGE_SUBVERSION class.
    test_windows_api_PACKAGE_SUBVERSION_field_and_type (field, typ)
    test_windows_api_PACKAGE_SUBVERSION_inits__fields_()
    test_windows_api_PACKAGE_SUBVERSION_is_Structure_subclass()
class tests.unit.test_windows_api.TestPACKAGE_VERSION
    Bases: object
    Testing class for arrangeit.windows.api.PACKAGE_VERSION class.
    test_windows_api_PACKAGE_VERSION_field_and_type (field, typ)
    test_windows_api_PACKAGE_VERSION_inits__anonymous_()
    test_windows_api_PACKAGE_VERSION_inits__fields_()
    test_windows_api_PACKAGE_VERSION_is_Structure_subclass()
class tests.unit.test_windows_api.TestPACKAGE_VERSION_U
    Bases: object
    Testing class for arrangeit.windows.api.PACKAGE_VERSION_U class.
    test_windows_api_PACKAGE_VERSION_U_field_and_type (field, typ)
    test_windows_api_PACKAGE_VERSION_U_inits__fields_()
    test_windows_api_PACKAGE_VERSION_U_is_Union_subclass()
class tests.unit.test_windows_api.TestTITLEBARINFO
    Bases: object
    Testing class for arrangeit.windows.api.TITLEBARINFO class.
    test_windows_api_TITLEBARINFO_field_and_type (field, typ)
    test_windows_api_TITLEBARINFO_inits__fields_()
    test_windows_api_TITLEBARINFO_is_Structure_subclass()
class tests.unit.test_windows_api.TestWINDOWINFO
    Bases: object
    Testing class for arrangeit.windows.api.WINDOWINFO class.
    test_windows_api_WINDOWINFO_field_and_type (field, typ)
    test_windows_api_WINDOWINFO_inits__fields_()

```

```

    test_windows_api_WINDOWINFO_is_Structure_subclass()
class tests.unit.test_windows_api.TestWindowsApiCustomFunctions
    Bases: object
    Testing class for arrangeit.windows.api custom functions.
    test_windows_api_platform_supports_packages_calls_getwindowsversion(mock)
    test_windows_api_platform_supports_packages_for_exception(mock)
    test_windows_api_platform_supports_packages_functionality(mock, major,
                                                                minor, expected)
    test_windows_api_platform_supports_virtual_desktops_calls_getwindowsversion(mock)
    test_windows_api_platform_supports_virtual_desktops_for_exception(mock)
    test_windows_api_platform_supports_virtual_desktops_functionality(mock,
                                                                        major,
                                                                        minor,
                                                                        ex-
                                                                        pected)

class tests.unit.test_windows_api.TestWindowsApiHelpersCommon
    Bases: object
    Testing class for arrangeit.windows.api.Helpers common methods.
    test_windows_api_Helpers__init__calls_setup_base(mock)
    test_windows_api_Helpers__init__calls_setup_common_helpers(mock)
    test_windows_api_Helpers__init__calls_setup_thumbnail_helpers(mock)
    test_windows_api_Helpers__init__calls_setup_win8_helpers(mock)
    test_windows_api_Helpers__init__calls_platform_supports_packages(mock)
    test_windows_api_Helpers__init__not_calling_setup_win8_helpers(mock)
    test_windows_api_Helpers_setup_base_sets_WNDENUMPROC(mock)
    test_windows_api_Helpers_setup_base_sets_WinDLL_dwmapi(mock)
    test_windows_api_Helpers_setup_base_sets_WinDLL_kernel32(mock)
    test_windows_api_Helpers_setup_base_sets_WinDLL_psapi(mock)
    test_windows_api_Helpers_setup_base_sets_WinDLL_user32(mock)
    test_windows_api_Helpers_setup_common_get_windows_thread_process_id(mock)
    test_windows_api_Helpers_setup_common_helpers_close_handle(mock)
    test_windows_api_Helpers_setup_common_helpers_dwm_get_window_attribute(mock)
    test_windows_api_Helpers_setup_common_helpers_enum_child_windows(mock)
    test_windows_api_Helpers_setup_common_helpers_enum_windows(mock)
    test_windows_api_Helpers_setup_common_helpers_get_ancestor(mock)
    test_windows_api_Helpers_setup_common_helpers_get_last_active_popup(mock)
    test_windows_api_Helpers_setup_common_helpers_get_process_image_file_name(mock)
    test_windows_api_Helpers_setup_common_helpers_get_titlebar_info(mock)
    test_windows_api_Helpers_setup_common_helpers_get_window_info(mock)

```

```

test_windows_api_Helpers__setup_common_helpers__open_process (mock)
test_windows_api_Helpers__setup_helper_returns_attr_method (mock)
test_windows_api_Helpers__setup_helper_sets_argtypes (mock)
test_windows_api_Helpers__setup_helper_sets_restype (mock)
test_windows_api_Helpers__setup_thumbnail_help__dwm_update_thumbnail_properties (mock)
test_windows_api_Helpers__setup_thumbnail_helpers__dwm_is_composition_enabled (mock)
test_windows_api_Helpers__setup_thumbnail_helpers__dwm_register_thumbnail (mock)
test_windows_api_Helpers__setup_thumbnail_helpers__dwm_unregister_thumbnail (mock)
class tests.unit.test_windows_api.TestWindowsApiHelpersWin8
  Bases: object
  Testing class for arrangeit.windows.api.Helpers Win8+ functions.
  pytestmark = [Mark(name='skipif', args=(True,)), kwargs={'reason': 'Win 8 and 10 only'}]
  test_windows_api_Helpers__setup_win8_helpers__close_package_info (mock)
  test_windows_api_Helpers__setup_win8_helpers__get_package_full_name (mock)
  test_windows_api_Helpers__setup_win8_helpers__get_package_info (mock)
  test_windows_api_Helpers__setup_win8_helpers__open_package_info_by_full_name (mock)
class tests.unit.test_windows_api.TestWindowsApiPackage
  Bases: object
  Testing class for arrangeit.windows.api.Package.
  test_api_Package__get_first_image_calls_and_returns_resized_Image (mock)
  test_api_Package__get_first_image_calls_open_image_if_not_exists (mock)
  test_api_Package__get_first_image_calls_os_path_exists (mock)
  test_api_Package__get_first_image_calls_os_path_join (mock)
  test_api_Package__get_first_image_calls_product (mock)
  test_api_Package__get_first_image_calls_splitext (mock)
  test_api_Package__get_first_image_catches_exception (mock)
  test_api_Package__get_manifest_root_calls_and_returns_getroot (mock)
  test_api_Package__get_manifest_root_calls_os_path_exists (mock)
  test_api_Package__get_manifest_root_calls_os_path_join (mock)
  test_api_Package__get_manifest_root_calls_parse (mock)
  test_api_Package__get_manifest_root_returns_true_if_not_exists (mock)
  test_api_Package__init__calls_setup_package (mock)
  test_api_Package__init__sets_path_attribute_from_provided (mock)
  test_api_Package__namespace_for_element_calls_re_match (mock)
  test_api_Package__namespace_for_element_returns_empty_string (mock)
  test_api_Package__namespace_for_element_returns_first_group (mock)
  test_api_Package__setup_app_name_calls__namespace_for_element (mock)

```

```

test_api_Package__setup_app_name_calls_iter_on_next (mockер)
test_api_Package__setup_app_name_calls_next (mockер)
test_api_Package__setup_app_name_calls_root_iter (mockер)
test_api_Package__setup_app_name_sets_app_name_attr (mockер)
test_api_Package__setup_icon_appends_once_to_sources_from_Applications (mockер)
test_api_Package__setup_icon_appends_to_sources_from_Properties (mockер)
test_api_Package__setup_icon_calls_namespace_for_element (mockер)
test_api_Package__setup_icon_calls_iter_on_next (mockер)
test_api_Package__setup_icon_calls_next (mockер)
test_api_Package__setup_icon_calls_root_iter (mockер)
test_api_Package__inits_empty_attr (attr)
test_api_Package__inits_empty_icon ()
test_api_Package__setup_package_calls_get_manifest_root (mockер)
test_api_Package__setup_package_calls_setup_app_name (mockер)
test_api_Package__setup_package_calls_setup_icon (mockер)

```

#### tests.unit.test\_windows\_vdi – Unit tests for MS Windows virtual desktop interfaces module

```

class tests.unit.test_windows_vdi.TestWindowsVdiAdjacentDesktop
    Bases: object
    Testing class for arrangeit.windows.vdi.AdjacentDesktop class.
    test_windows_vdi_AdjacentDesktop_field_and_value (field, value)
    test_windows_vdi_AdjacentDesktop_is_INT_subclass ()

class tests.unit.test_windows_vdi.TestWindowsVdiApplicationViewOrientation
    Bases: object
    Testing class for arrangeit.windows.vdi.ApplicationViewOrientation class.
    test_windows_vdi_ApplicationViewOrientation_field_and_value (field, value)
    test_windows_vdi_ApplicationViewOrientation_is_INT_subclass ()

class tests.unit.test_windows_vdi.TestWindowsVdiEventRegistrationToken
    Bases: object
    Testing class for arrangeit.windows.vdi.EventRegistrationToken class.
    test_windows_api_EventRegistrationToken_field_and_type (field, typ)
    test_windows_api_EventRegistrationToken_inits_fields_ ()
    test_windows_vdi_EventRegistrationToken_is_Structure_subclass ()

class tests.unit.test_windows_vdi.TestWindowsVdiHSTRING__
    Bases: object
    Testing class for arrangeit.windows.vdi.HSTRING__ class.
    test_windows_api_HSTRING__field_and_type (field, typ)

```

```

test_windows_api_HSTRING__inits__fields_()
test_windows_vdi_HSTRING__is_Structure_subclass()
class tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView
Bases: object
Testing class for arrangeit.windows.vdi.IApplicationView class.
test_windows_vdi_IApplicationView_field_and_value(field, value)
test_windows_vdi_IApplicationView_is_IInspectable_subclass()
test_windows_vdi_IApplicationView_method_add_Consolidated()
test_windows_vdi_IApplicationView_method_get_AdjacentToLeftDisplayEdge()
test_windows_vdi_IApplicationView_method_get_AdjacentToRightDisplayEdge()
test_windows_vdi_IApplicationView_method_get_Id()
test_windows_vdi_IApplicationView_method_get_IsFullScreen()
test_windows_vdi_IApplicationView_method_get_IsOnLockScreen()
test_windows_vdi_IApplicationView_method_get_IsScreenCaptureEnabled()
test_windows_vdi_IApplicationView_method_get_Orientation()
test_windows_vdi_IApplicationView_method_get_Title()
test_windows_vdi_IApplicationView_method_put_IsScreenCaptureEnabled()
test_windows_vdi_IApplicationView_method_put_Title()
test_windows_vdi_IApplicationView_method_remove_Consolidated()
class tests.unit.test_windows_vdi.TestWindowsVdiIApplicationViewCollection
Bases: object
Testing class for arrangeit.windows.vdi.IApplicationViewCollection class.
test_windows_vdi_IApplicationViewCollection_RegisterForApplicationViewChanges()
test_windows_vdi_IApplicationViewCollection_UnregisterForApplicationViewChanges()
test_windows_vdi_IApplicationViewCollection_field_and_value(field, value)
test_windows_vdi_IApplicationViewCollection_is_IUnknown_subclass()
test_windows_vdi_IApplicationViewCollection_method_GetViewForAppUserModelId()
test_windows_vdi_IApplicationViewCollection_method_GetViewForApplication()
test_windows_vdi_IApplicationViewCollection_method_GetViewForHwnd()
test_windows_vdi_IApplicationViewCollection_method_GetViewInFocus()
test_windows_vdi_IApplicationViewCollection_method_GetViews()
test_windows_vdi_IApplicationViewCollection_method_GetViewsByAppUserModelId()
test_windows_vdi_IApplicationViewCollection_method_GetViewsByZOrder()
test_windows_vdi_IApplicationViewCollection_method_RefreshCollection()
test_windows_vdi_IApplicationViewCollection_method_Unknown1()

```



```

class tests.unit.test_windows_vdi.TestWindowsVdiIApplicationViewConsolidatedEventArgs
    Bases: object
    Testing class for arrangeit.windows.vdi.IApplicationViewConsolidatedEventArgs class.

    test_windows_vdi_IApplicationViewConsolidatedEventArgs_field_and_value (field,
                                                                    value)
    test_windows_vdi_IApplicationViewConsolidatedEventArgs_is_IInspectable_subclass ()
    test_windows_vdi_IApplicationViewConsolidatedEventArgs_m_get_IsUserInitiated ()

class tests.unit.test_windows_vdi.TestWindowsVdiIInspectable
    Bases: object
    Testing class for arrangeit.windows.vdi.IInspectable class.

    test_windows_vdi_IInspectable_field_and_value (field, value)
    test_windows_vdi_IInspectable_is_IUnknown_subclass ()
    test_windows_vdi_IInspectable_method_GetIids ()
    test_windows_vdi_IInspectable_method_GetRuntimeClassName ()
    test_windows_vdi_IInspectable_method_GetTrustLevel ()

class tests.unit.test_windows_vdi.TestWindowsVdiIObjectArray
    Bases: object
    Testing class for arrangeit.windows.vdi.IObjectArray class.

    test_windows_vdi_IObjectArray_field_and_value (field, value)
    test_windows_vdi_IObjectArray_is_IUnknown_subclass ()
    test_windows_vdi_IObjectArray_method_GetAt ()
    test_windows_vdi_IObjectArray_method_GetCount ()

class tests.unit.test_windows_vdi.TestWindowsVdiIServiceProvider
    Bases: object
    Testing class for arrangeit.windows.vdi.IServiceProvider class.

    test_windows_vdi_IServiceProvider_field_and_value (field, value)
    test_windows_vdi_IServiceProvider_is_IUnknown_subclass ()
    test_windows_vdi_IServiceProvider_method_QueryService ()

class tests.unit.test_windows_vdi.TestWindowsVdiIVirtualDesktop
    Bases: object
    Testing class for arrangeit.windows.vdi.IVirtualDesktop class.

    test_windows_vdi_IVirtualDesktop_field_and_value (field, value)
    test_windows_vdi_IVirtualDesktop_is_IUnknown_subclass ()
    test_windows_vdi_IVirtualDesktop_method_GetID ()
    test_windows_vdi_IVirtualDesktop_method_IsViewVisible ()

class tests.unit.test_windows_vdi.TestWindowsVdiIVirtualDesktopManager
    Bases: object
    Testing class for arrangeit.windows.vdi.IVirtualDesktopManager class.

```

```

test_windows_vdi_IVirtualDesktopManager_field_and_value (field, value)
test_windows_vdi_IVirtualDesktopManager_is_IUnknown_subclass ()
test_windows_vdi_IVirtualDesktopManager_method_GetWindowDesktopId ()
test_windows_vdi_IVirtualDesktopManager_method_IsWindowOnCurrentVirtualDesktop ()
test_windows_vdi_IVirtualDesktopManager_method_MoveWindowToDesktop ()
class tests.unit.test_windows_vdi.TestWindowsVdiIVirtualDesktopManagerInternal
  Bases: object
  Testing class for arrangeit.windows.vdi.IVirtualDesktopManagerInternal class.
test_windows_vdi_IVirtualDesktopManagerInternal_field_and_value (field,
                                                                    value)
test_windows_vdi_IVirtualDesktopManagerInternal_is_IUnknown_subclass ()
test_windows_vdi_IVirtualDesktopManagerInternal_method_CanViewMoveDesktops ()
test_windows_vdi_IVirtualDesktopManagerInternal_method_CreateDesktopW ()
test_windows_vdi_IVirtualDesktopManagerInternal_method_FindDesktop ()
test_windows_vdi_IVirtualDesktopManagerInternal_method_GetAdjacentDesktop ()
test_windows_vdi_IVirtualDesktopManagerInternal_method_GetCount ()
test_windows_vdi_IVirtualDesktopManagerInternal_method_GetCurrentDesktop ()
test_windows_vdi_IVirtualDesktopManagerInternal_method_GetDesktops ()
test_windows_vdi_IVirtualDesktopManagerInternal_method_MoveViewToDesktop ()
test_windows_vdi_IVirtualDesktopManagerInternal_method_RemoveDesktop ()
test_windows_vdi_IVirtualDesktopManagerInternal_method_SwitchDesktop ()
class tests.unit.test_windows_vdi.TestWindowsVdiModuleUids
  Bases: object
  Testing class for arrangeit.windows.vdi module level attributes.
test_windows_vdi_module_instantiates_clsid_attribute (attr, uid)
test_windows_vdi_module_instantiates_iid_attribute (attr, uid)
class tests.unit.test_windows_vdi.TestWindowsVdiTrustLevel
  Bases: object
  Testing class for arrangeit.windows.vdi.TrustLevel class.
test_windows_vdi_TrustLevel_field_and_value (field, value)
test_windows_vdi_TrustLevel_is_INT_subclass ()
class tests.unit.test_windows_vdi.TestWindowsVdiVirtualDesktopsWin10
  Bases: object
  Testing class for arrangeit.windows.vdi.VirtualDesktopsWin10 class.
  pytestmark = [Mark(name='skipif', args=(True,), kwargs={'reason': 'Win 10 only'})]
test_windows_vdi_VDWin10_get_desktop_id_from_array_calls_POINTER (mock)
test_windows_vdi_VDWin10_get_desktop_id_from_array_calls_and_returns_GetID (mock)
test_windows_vdi_VDWin10_get_desktop_id_from_array_calls_array_GetAt (mock)

```



```

test_windows_vdi_VDWin10_move_own_window_to_desktop_calls_MoveWindowToDesktop (mock)
test_windows_vdi_VDWin10_move_own_window_to_desktop_calls_SwitchDesktop (mock)
test_windows_vdi_VDWin10_move_own_window_to_desktop_calls_byref (mock)
test_windows_vdi_VDWin10_move_own_window_to_desktop_calls_byref_find (mock)
test_windows_vdi_VDWin10_move_own_window_to_desktop_get_desktop_id_from_ordinal (mock)
test_windows_vdi_VDWin10_move_own_window_to_desktop_returns_False (mock)
test_windows_vdi_VDWin10_move_own_window_to_desktop_returns_None_for_Switch (mock)
test_windows_vdi_VDWin10_move_own_window_to_desktop_returns_None_for_move_not_ok (mock)
test_windows_vdi_VirtualDesktopsWin10__init__calls_setup (mock)
test_windows_vdi_VirtualDesktopsWin10__setup_calls_CoInitialize (mock)
test_windows_vdi_VirtualDesktopsWin10__setup_calls__get_internal_manager_attr (mock)
test_windows_vdi_VirtualDesktopsWin10__setup_calls__get_manager_and_sets_attr (mock)
test_windows_vdi_VirtualDesktopsWin10__setup_calls__get_service_provider (mock)
test_windows_vdi_VirtualDesktopsWin10__setup_calls__get_view_collection_attr (mock)
test_windows_vdi_VirtualDesktopsWin10_inits_attr_as_None (attr)

```

#### tests.unit.test\_windows\_api\_api – Unit tests for MS Windows api helpers module Api class

```

class tests.unit.test_windows_api_api.TestWindowsApiApiPrivate
    Bases: object

```

Testing class for `arrangeit.windows.api.Api` private methods.

```

test_api_Api__rectangle_to_wintypes_rect_calls_and_returns_wintypes_RECT (mock)
test_api_Api__rectangle_to_wintypes_rect_sets_attributtes (mock)
test_api_Api__update_thumbnail_calls_DWM_THUMBNAIL_PROPERTIES (mock)
test_api_Api__update_thumbnail_calls_dwm_update_thumbnail_properties (mock)
test_api_Api__update_thumbnail_calls__rectangle_to_wintypes_rect (mock)
test_api_Api__update_thumbnail_calls_returns_None (mock)
test_api_Api__update_thumbnail_calls_returns_thumbnail_id (mock)
test_api_Api__update_thumbnail_calls_wintypes_BYTE (mock)
test_api_Api__update_thumbnail_calls_wintypes_byref (mock)
test_api_Api__update_thumbnail_sets_dwFlags (mock)
test_api_Api__update_thumbnail_sets_fSourceClientAreaOnly (mock)
test_api_Api__update_thumbnail_sets_fVisible (mock)
test_api_Api__update_thumbnail_sets_opacity (mock)
test_api_Api__update_thumbnail_sets_rcDestination (mock)
test_api_Api__update_thumbnail_sets_rcSource (mock)
test_api_Api__wintypes_rect_to_rectangle_calls_and_returns_Rectangle (mock)

```

```

class tests.unit.test_windows_api_api.TestWindowsApiApiPrivateWin8
    Bases: object
    Testing class for arrangeit.windows.api.Api Win8+ private methods.
    pytestmark = [Mark(name='skipif', args=(True,), kwargs={'reason': 'Win 8 and 10 only'})
    test_Api_package_full_name_from_handle_calls_again_get_package_full_name(mocker)
    test_Api_package_full_name_from_handle_calls_create_unicode_buffer(mocker)
    test_Api_package_full_name_from_handle_calls_first_time_get_package_full_name(mocker)
    test_Api_package_full_name_from_handle_returns_None_for_no_package(mocker)
    test_Api_package_full_name_from_handle_returns_None_for_no_success(mocker)
    test_Api_package_full_name_from_handle_returns_full_name(mocker)
    test_Api_package_full_name_from_hwnd_calls_close_handle(mocker)
    test_Api_package_full_name_from_hwnd_calls_get_windows_thread_process_id(mocker)
    test_Api_package_full_name_from_hwnd_calls_open_process(mocker)
    test_Api_package_full_name_from_hwnd_calls_package_full_name_from_handle(mocker)
    test_Api_package_full_name_from_hwnd_calls_enum_windows(mocker)
    test_Api_package_full_name_from_hwnd_calls_wintypes_DWORD(mocker)
    test_Api_package_full_name_from_hwnd_returns_None(mocker)
    test_Api_package_full_name_from_hwnd_returns_full_name(mocker)
    test_Api_package_info_buffer_from_reference_calls_again_get_package_info(mocker)
    test_Api_package_info_buffer_from_reference_calls_cast(mocker)
    test_Api_package_info_buffer_from_reference_calls_create_string_buffer(mocker)
    test_Api_package_info_buffer_from_reference_calls_first_time_get_package_info(mocker)
    test_Api_package_info_buffer_from_reference_returns_None_for_no_success(mocker)
    test_Api_package_info_buffer_from_reference_returns_None_for_not_insufficient(mocker)
    test_Api_package_info_buffer_from_reference_returns_buffer(mocker)
    test_Api_package_info_ref_from_full_name_calls_open_package_info_by_full_name(mocker)
    test_Api_package_info_reference_from_full_name_calls_PACKAGE_INFO_REFERENCE(mocker)
    test_Api_package_info_reference_from_full_name_calls_pointer(mocker)
    test_Api_package_info_reference_from_full_name_returns_None(mocker)
    test_Api_package_info_reference_from_full_name_returns_package_info_reference(mocker)

class tests.unit.test_windows_api_api.TestWindowsApiApiPublic
    Bases: object
    Testing class for arrangeit.windows.api.Api public methods.
    test_Api_executable_name_for_hwnd_calls_close_handle(mocker)
    test_Api_executable_name_for_hwnd_calls_get_process_image_file_name(mocker)
    test_Api_executable_name_for_hwnd_calls_get_windows_thread_process_id(mocker)
    test_Api_executable_name_for_hwnd_calls_open_process(mocker)

```

test\_Api\_executable\_name\_for\_hwnd\_calls\_and\_returns\_extract\_name\_from\_bytes\_path (mock)  
test\_Api\_executable\_name\_for\_hwnd\_calls\_create\_string\_buffer (mock)  
test\_Api\_executable\_name\_for\_hwnd\_calls\_wintypes\_DWORD (mock)  
test\_Api\_executable\_name\_for\_hwnd\_returns\_None (mock)  
test\_Api\_extended\_frame\_rect\_calls\_dwm\_get\_window\_attribute (mock)  
test\_Api\_extended\_frame\_rect\_calls\_and\_returns\_wintypes\_rect\_to\_rectangle (mock)  
test\_Api\_extended\_frame\_rect\_calls\_ctypes\_byref (mock)  
test\_Api\_extended\_frame\_rect\_calls\_ctypes\_sizeof (mock)  
test\_Api\_extended\_frame\_rect\_calls\_wintypes\_RECT (mock)  
test\_Api\_extended\_frame\_rect\_returns\_None (mock)  
test\_Api\_get\_ancestor\_by\_type\_calls\_and\_returns\_get\_ancestor (mock)  
test\_Api\_get\_desktop\_ordinal\_for\_window\_calls\_dummy\_vdi\_get\_window\_desktop (mock)  
test\_Api\_get\_desktop\_ordinal\_for\_window\_calls\_win10\_vdi\_get\_window\_desktop (mock)  
test\_Api\_get\_desktops\_calls\_dummy\_vdi\_get\_desktops (mock)  
test\_Api\_get\_desktops\_calls\_win10\_vdi\_get\_desktops (mock)  
test\_Api\_get\_desktops\_returns\_list\_of\_two\_tuples (mock)  
test\_Api\_get\_last\_active\_popup\_calls\_and\_returns\_get\_ancestor (mock)  
test\_Api\_is\_cloaked\_calls\_dwm\_get\_window\_attribute (mock)  
test\_Api\_is\_cloaked\_calls\_and\_returns\_vdi\_is\_window\_in\_current\_for\_dwm\_value\_dummy (mock)  
test\_Api\_is\_cloaked\_calls\_and\_returns\_vdi\_is\_window\_in\_current\_for\_dwm\_value\_win10 (mock)  
test\_Api\_is\_cloaked\_calls\_ctypes\_byref (mock)  
test\_Api\_is\_cloaked\_calls\_ctypes\_sizeof (mock)  
test\_Api\_is\_cloaked\_calls\_wintypes\_DWORD (mock)  
test\_Api\_is\_cloaked\_returns\_False\_for\_0\_dwm\_value (mock)  
test\_Api\_is\_cloaked\_returns\_False\_for\_not\_ok (mock)  
test\_Api\_is\_dwm\_composition\_enabled\_calls\_dwm\_is\_composition\_enabled (mock)  
test\_Api\_is\_dwm\_composition\_enabled\_calls\_ctypes\_byref (mock)  
test\_Api\_is\_dwm\_composition\_enabled\_calls\_wintypes\_BOOL (mock)  
test\_Api\_is\_dwm\_composition\_enabled\_returns\_value (mock)  
test\_Api\_move\_other\_window\_to\_desktop\_calls\_dummy\_vdi\_move\_other\_window\_to\_desktop (mock)  
test\_Api\_move\_other\_window\_to\_desktop\_calls\_win10\_vdi\_move\_other\_window\_to (mock)  
test\_Api\_move\_own\_window\_to\_desktop\_calls\_dummy\_vdi\_move\_own\_window\_to\_desktop (mock)  
test\_Api\_move\_own\_window\_to\_desktop\_calls\_win10\_vdi\_move\_own\_window\_to\_desktop (mock)  
test\_Api\_setup\_thumbnail\_calls\_dwm\_register\_thumbnail (mock)  
test\_Api\_setup\_thumbnail\_calls\_and\_returns\_update\_thumbnail (mock)  
test\_Api\_setup\_thumbnail\_calls\_wintypes\_HANDLE (mock)

```

test_api_setup_thumbnail_calls_wintypes_byref (mock)
test_api_setup_thumbnail_returns_None (mock)
test_api_title_info_state_calls_TITLEBARINFO (mock)
test_api_title_info_state_calls_get_titlebar_info (mock)
test_api_title_info_state_calls_ctypes_byref (mock)
test_api_title_info_state_calls_ctypes_sizeof (mock)
test_api_title_info_state_returns_None (mock)
test_api_title_info_state_returns_value (mock)
test_api_unregister_thumbnail_calls_dwm_unregister_thumbnail (mock)
test_api_unregister_thumbnail_returns_True_on_error (mock)
test_api_window_info_extended_style_calls_WINDOWINFO (mock)
test_api_window_info_extended_style_calls_get_window_info (mock)
test_api_window_info_extended_style_calls_ctypes_byref (mock)
test_api_window_info_extended_style_returns_None (mock)
test_api_window_info_extended_style_returns_value (mock)
test_api_Api_init_calls_platform_supports_virtual_desktops (mock)
test_api_Api_init_initializes_and_sets_dummy_vdi (mock)
test_api_Api_init_initializes_and_sets_helpers (mock)
test_api_Api_init_initializes_and_sets_vdi (mock)
test_api_Api_enum_windows_calls_WNDENUMPROC (mock)
test_api_Api_enum_windows_calls_enum_child_windows (mock)
test_api_Api_enum_windows_calls_enum_windows (mock)
test_api_Api_enum_windows_nested_append_to_collection (mock)
test_api_Api_enum_windows_returns_non_empty_list (mock)
test_api_Api_inits_attr_as_None (attr)
test_api_Api_inits_empty_attr (attr)
class tests.unit.test_windows_api_api.TestWindowsApiApiPublicWin8
    Bases: object
    Testing class for arrangeit.windows.api.Api Win8 and 10 public methods.
    pytestmark = [Mark(name='skipif', args=(True,), kwargs={'reason': 'Win 8 and 10 only'})
test_api_Api_get_package_calls_PACKAGE_INFO_from_buffer (mock)
test_api_Api_get_package_calls_Package (mock)
test_api_Api_get_package_calls_close_package_info (mock)
test_api_Api_get_package_calls_package_full_name_from_hwnd (mock)
test_api_Api_get_package_calls_package_info_buffer_from_reference (mock)
test_api_Api_get_package_calls_package_info_reference_from_full_name (mock)

```

`test_api_Api_get_package_returns_empty_Package` (*mock*)

**tests.unit.fixtures** – Sample fixtures for unit testing

**tests.unit.mock\_helpers** – Unit testing mock helper functions

tests.unit.mock\_helpers.**controller\_mocked\_app** (*mock*)

tests.unit.mock\_helpers.**controller\_mocked\_for\_next** (*mock*)

tests.unit.mock\_helpers.**controller\_mocked\_for\_run** (*mock*)

tests.unit.mock\_helpers.**controller\_mocked\_key\_press** (*mock*, *key*)

tests.unit.mock\_helpers.**controller\_mocked\_next** (*mock*)

tests.unit.mock\_helpers.**mocked\_for\_about** (*mock*)

tests.unit.mock\_helpers.**mocked\_for\_about\_setup** (*mock*)

tests.unit.mock\_helpers.**mocked\_for\_options** (*mock*)

tests.unit.mock\_helpers.**mocked\_for\_options\_setup** (*mock*, *without\_section=False*, *without\_files=False*)

tests.unit.mock\_helpers.**mocked\_setup** (*mock*)

tests.unit.mock\_helpers.**mocked\_setup\_root** (*mock*)

tests.unit.mock\_helpers.**mocked\_setup\_view** (*mock*)

**tests.unit.nested\_helper** – Helper module for unit testing of inner/nested functions

tests.unit.nested\_helper.**freeVar** (*val*)

tests.unit.nested\_helper.**nested** (*outer*, *innerName*, *\*\*freeVars*)

**tests.functional** – arrangeit functional tests

**tests.functional.arrangeit\_keywords** – arrangeit keywords library for Robot Framework

**tests.vm** – Vagrant’s virtual machines for running tests in various environments

## 7 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)



## Python Module Index

### a

arrangeit, 21  
arrangeit.\_\_main\_\_, 21  
arrangeit.base, 21  
arrangeit.darwin, 47  
arrangeit.darwin.app, 47  
arrangeit.darwin.collector, 48  
arrangeit.darwin.controller, 49  
arrangeit.darwin.utils, 50  
arrangeit.data, 31  
arrangeit.linux, 50  
arrangeit.linux.app, 50  
arrangeit.linux.collector, 51  
arrangeit.linux.controller, 53  
arrangeit.linux.utils, 53  
arrangeit.options, 40  
arrangeit.settings, 46  
arrangeit.utils, 43  
arrangeit.view, 34  
arrangeit.windows, 53  
arrangeit.windows.api, 54  
arrangeit.windows.app, 53  
arrangeit.windows.collector, 59  
arrangeit.windows.controller, 60  
arrangeit.windows.utils, 62  
arrangeit.windows.vdi, 60

tests.unit.test\_windows\_vdi, 103  
tests.vm, 112

### t

tests, 63  
tests.functional, 112  
tests.unit, 63  
tests.unit.fixtures, 112  
tests.unit.mock\_helpers, 112  
tests.unit.nested\_helper, 112  
tests.unit.test\_base, 63  
tests.unit.test\_basecontroller, 69  
tests.unit.test\_basecontroller\_domain\_logic,  
66  
tests.unit.test\_darwin, 91  
tests.unit.test\_data, 74  
tests.unit.test\_linux, 92  
tests.unit.test\_options, 83  
tests.unit.test\_settings, 89  
tests.unit.test\_setup, 63  
tests.unit.test\_utils, 87  
tests.unit.test\_view, 76  
tests.unit.test\_viewapplication, 80  
tests.unit.test\_windows, 95  
tests.unit.test\_windows\_api, 99  
tests.unit.test\_windows\_api\_api, 108

## Index

### A

AboutDialog (class in *arrangeit.options*), 40  
activate\_root () (*arrangeit.base.BaseApp* method), 21  
activate\_root () (*arrangeit.darwin.app.App* method), 47  
activate\_root () (*arrangeit.linux.app.App* method), 50  
activate\_root () (*arrangeit.windows.app.App* method), 53  
active (*arrangeit.view.WorkspacesCollection* attribute), 39  
add () (*arrangeit.data.WindowsCollection* method), 33  
add\_window () (*arrangeit.base.BaseCollector* method), 23  
add\_window () (*arrangeit.darwin.collector.Collector* method), 48  
add\_window () (*arrangeit.linux.collector.Collector* method), 51  
add\_window () (*arrangeit.windows.collector.Collector* method), 59  
add\_windows () (*arrangeit.view.WindowsList* method), 38  
add\_workspaces () (*arrangeit.view.WorkspacesCollection* method), 39  
anchor () (*arrangeit.view.CornerWidget* method), 34  
Api (class in *arrangeit.windows.api*), 54  
app (*arrangeit.base.BaseController* attribute), 24  
App (class in *arrangeit.darwin.app*), 47  
App (class in *arrangeit.linux.app*), 50  
App (class in *arrangeit.windows.app*), 53  
app\_name (*arrangeit.windows.api.Package* attribute), 58  
apply\_snapping () (*arrangeit.base.BaseController* method), 24  
arrangeit (module), 21  
arrangeit.\_\_main\_\_ (module), 21  
arrangeit.base (module), 21  
arrangeit.darwin (module), 47  
arrangeit.darwin.app (module), 47  
arrangeit.darwin.collector (module), 48  
arrangeit.darwin.controller (module), 49  
arrangeit.darwin.utils (module), 50  
arrangeit.data (module), 31  
arrangeit.linux (module), 50  
arrangeit.linux.app (module), 50  
arrangeit.linux.collector (module), 51  
arrangeit.linux.controller (module), 53  
arrangeit.linux.utils (module), 53

arrangeit.options (module), 40  
arrangeit.settings (module), 46  
arrangeit.utils (module), 43  
arrangeit.view (module), 34  
arrangeit.windows (module), 53  
arrangeit.windows.api (module), 54  
arrangeit.windows.app (module), 53  
arrangeit.windows.collector (module), 59  
arrangeit.windows.controller (module), 60  
arrangeit.windows.utils (module), 62  
arrangeit.windows.vdi (module), 60

### B

background (*arrangeit.view.CornerWidget* attribute), 34  
background (*arrangeit.view.PropertyIcon* attribute), 35  
BaseApp (class in *arrangeit.base*), 21  
BaseCollector (class in *arrangeit.base*), 23  
BaseController (class in *arrangeit.base*), 23  
BaseMouse (class in *arrangeit.base*), 30  
BLANK\_ICON (*arrangeit.settings.Settings* attribute), 46  
box\_size (*arrangeit.view.CornerWidget* attribute), 34

### C

callback (*arrangeit.view.PropertyIcon* attribute), 35  
change\_position () (*arrangeit.base.BaseController* method), 24  
change\_setting () (*arrangeit.base.BaseApp* method), 21  
change\_setting () (*arrangeit.base.BaseController* method), 24  
change\_setting () (*arrangeit.options.OptionsDialog* method), 42  
change\_settings\_color\_group () (*arrangeit.base.BaseApp* method), 22  
change\_size () (*arrangeit.base.BaseController* method), 24  
changed (*arrangeit.data.WindowModel* attribute), 31  
changed\_h (*arrangeit.data.WindowModel* attribute), 31  
changed\_w (*arrangeit.data.WindowModel* attribute), 31  
changed\_ws (*arrangeit.data.WindowModel* attribute), 31  
changed\_x (*arrangeit.data.WindowModel* attribute), 31  
changed\_y (*arrangeit.data.WindowModel* attribute), 31

check\_current\_size() (arrangeit.base.BaseController method), 25  
 check\_intersections() (in module arrangeit.utils), 43  
 check\_mouse() (arrangeit.base.BaseController method), 25  
 check\_snapping() (arrangeit.base.BaseController method), 25  
 check\_snapping\_state() (arrangeit.base.BaseController method), 25  
 check\_window() (arrangeit.base.BaseCollector method), 23  
 check\_window() (arrangeit.darwin.collector.Collector method), 48  
 check\_window() (arrangeit.linux.collector.Collector method), 51  
 check\_window() (arrangeit.windows.collector.Collector method), 59  
 CheckOption (class in arrangeit.options), 41  
 clear() (arrangeit.data.WindowsCollection method), 33  
 clear\_changed() (arrangeit.data.WindowModel method), 31  
 clear\_list() (arrangeit.view.WindowsList method), 38  
 collection (arrangeit.base.BaseCollector attribute), 23  
 collector (arrangeit.base.BaseApp attribute), 22  
 Collector (class in arrangeit.darwin.collector), 48  
 Collector (class in arrangeit.linux.collector), 51  
 Collector (class in arrangeit.windows.collector), 59  
 color\_group() (arrangeit.settings.Settings class method), 47  
 colorized (arrangeit.view.PropertyIcon attribute), 35  
 colorized (arrangeit.view.Resizable attribute), 36  
 colorized (arrangeit.view.Restored attribute), 36  
 ColorOption (class in arrangeit.options), 41  
 control (arrangeit.base.BaseMouse attribute), 30  
 controller (arrangeit.base.BaseApp attribute), 22  
 controller (arrangeit.view.ViewApplication attribute), 37  
 Controller (class in arrangeit.darwin.controller), 49  
 Controller (class in arrangeit.linux.controller), 53  
 Controller (class in arrangeit.windows.controller), 60  
 controller\_mocked\_app() (in module tests.unit.mock\_helpers), 112  
 controller\_mocked\_for\_next() (in module tests.unit.mock\_helpers), 112  
 controller\_mocked\_for\_run() (in module tests.unit.mock\_helpers), 112  
 controller\_mocked\_key\_press() (in module tests.unit.mock\_helpers), 112  
 controller\_mocked\_next() (in module tests.unit.mock\_helpers), 112  
 CORNER\_RECT\_INDEXES (arrangeit.settings.Settings attribute), 46  
 CornerWidget (class in arrangeit.view), 34  
 create\_frame() (arrangeit.options.OptionsDialog method), 42  
 create\_separator() (arrangeit.options.OptionsDialog method), 42  
 create\_snapping\_sources() (arrangeit.base.BaseApp method), 22  
 create\_widget() (arrangeit.options.OptionsDialog method), 42  
 cursor\_position() (arrangeit.base.BaseMouse method), 30  
 cycle\_corners() (arrangeit.base.BaseController method), 26

## D

default\_size (arrangeit.base.BaseController attribute), 26  
 desktops (arrangeit.windows.vdi.VirtualDesktopsWin10 attribute), 61  
 display\_message() (arrangeit.base.BaseController method), 26  
 DummyVirtualDesktops (class in arrangeit.windows.api), 58

## E

enum\_windows() (arrangeit.windows.api.Api method), 54  
 executable\_name\_for\_hwnd() (arrangeit.windows.api.Api method), 55  
 export() (arrangeit.data.WindowsCollection method), 33  
 extended\_frame\_rect() (arrangeit.windows.api.Api method), 55  
 extract\_name\_from\_bytes\_path() (in module arrangeit.windows.utils), 62

## F

FloatScaleOption (class in arrangeit.options), 41  
 freeVar() (in module tests.unit.nested\_helper), 112

## G

generator (arrangeit.base.BaseController attribute), 26  
 generator() (arrangeit.data.WindowsCollection method), 33  
 get\_ancestor\_by\_type() (arrangeit.windows.api.Api method), 55

get\_application\_name() (*arrangeit.darwin.collector.Collector method*), 48  
 get\_application\_name() (*arrangeit.windows.collector.Collector method*), 59  
 get\_available\_workspaces() (*arrangeit.base.BaseCollector method*), 23  
 get\_available\_workspaces() (*arrangeit.darwin.collector.Collector method*), 49  
 get\_available\_workspaces() (*arrangeit.linux.collector.Collector method*), 51  
 get\_available\_workspaces() (*arrangeit.windows.collector.Collector method*), 59  
 get\_class() (*in module arrangeit.utils*), 44  
 get\_component\_class() (*in module arrangeit.utils*), 44  
 get\_cursor\_name() (*in module arrangeit.utils*), 44  
 get\_desktop\_ordinal\_for\_window() (*arrangeit.windows.api.Api method*), 55  
 get\_desktops() (*arrangeit.windows.api.Api method*), 55  
 get\_desktops() (*arrangeit.windows.api.DummyVirtualDesktops method*), 58  
 get\_desktops() (*arrangeit.windows.vdi.VirtualDesktopsWin10 method*), 61  
 get\_humanized\_number() (*arrangeit.view.Workspace method*), 39  
 get\_icon\_image() (*arrangeit.view.ListedWindow method*), 35  
 get\_image\_from\_pixbuf() (*arrangeit.linux.collector.Collector method*), 51  
 get\_item() (*arrangeit.base.BaseMouse method*), 30  
 get\_last\_active\_popup() (*arrangeit.windows.api.Api method*), 55  
 get\_model\_by\_wid() (*arrangeit.data.WindowsCollection method*), 33  
 get\_monitors\_rects() (*arrangeit.base.BaseCollector method*), 23  
 get\_monitors\_rects() (*arrangeit.darwin.collector.Collector method*), 49  
 get\_monitors\_rects() (*arrangeit.linux.collector.Collector method*), 51  
 get\_monitors\_rects() (*arrangeit.windows.collector.Collector method*), 59  
 get\_package() (*arrangeit.windows.api.Api method*), 56  
 get\_place\_parameters() (*arrangeit.view.CornerWidget method*), 34  
 get\_prepared\_screenshot() (*in module arrangeit.utils*), 44  
 get\_resized\_image() (*in module arrangeit.utils*), 44  
 get\_resource\_path() (*in module arrangeit.utils*), 45  
 get\_root\_rect() (*arrangeit.base.BaseController method*), 26  
 get\_root\_wid() (*arrangeit.view.ViewApplication method*), 37  
 get\_screenshot\_widget() (*in module arrangeit.view*), 40  
 get\_smallest\_monitor\_size() (*arrangeit.base.BaseCollector method*), 23  
 get\_snapping\_sources\_for\_rect() (*in module arrangeit.utils*), 45  
 get\_tkinter\_root() (*in module arrangeit.view*), 40  
 get\_value\_if\_valid\_type() (*in module arrangeit.utils*), 45  
 get\_window\_by\_wid() (*arrangeit.linux.collector.Collector method*), 52  
 get\_window\_desktop() (*arrangeit.windows.api.DummyVirtualDesktops method*), 58  
 get\_window\_desktop() (*arrangeit.windows.vdi.VirtualDesktopsWin10 method*), 61  
 get\_window\_move\_resize\_mask() (*arrangeit.linux.collector.Collector method*), 52  
 get\_windows() (*arrangeit.base.BaseCollector method*), 23  
 get\_windows() (*arrangeit.darwin.collector.Collector method*), 49  
 get\_windows() (*arrangeit.linux.collector.Collector method*), 52  
 get\_windows() (*arrangeit.windows.collector.Collector method*), 59  
 get\_windows\_list() (*arrangeit.data.WindowsCollection method*), 33  
 get\_wnck\_workspace\_for\_custom\_number() (*arrangeit.linux.collector.Collector method*), 52  
 get\_workspace\_number() (*arrangeit.base.BaseCollector method*), 23

get\_workspace\_number() (*arrangeit.linux.collector.Collector* method), 52  
 get\_workspace\_number\_for\_window() (*arrangeit.base.BaseCollector* method), 23  
 get\_workspace\_number\_for\_window() (*arrangeit.darwin.collector.Collector* method), 49  
 get\_workspace\_number\_for\_window() (*arrangeit.linux.collector.Collector* method), 52  
 get\_workspace\_number\_for\_window() (*arrangeit.windows.collector.Collector* method), 59  
 grab\_window\_screen() (*arrangeit.base.BaseApp* method), 22  
 grab\_window\_screen() (*arrangeit.darwin.app.App* method), 48  
 grab\_window\_screen() (*arrangeit.linux.app.App* method), 50  
 grab\_window\_screen() (*arrangeit.windows.app.App* method), 53

## H

h (*arrangeit.data.WindowModel* attribute), 32  
 HELP\_PAGE\_URL (*arrangeit.settings.Settings* attribute), 46  
 helpers (*arrangeit.windows.api.Api* attribute), 56  
 Helpers (*class in arrangeit.windows.api*), 58  
 hide\_corner() (*arrangeit.view.CornerWidget* method), 34  
 hide\_root() (*arrangeit.view.ViewApplication* method), 37

## I

IApplicationView (*class in arrangeit.windows.vdi*), 60  
 IApplicationViewCollection (*class in arrangeit.windows.vdi*), 60  
 IApplicationViewConsolidatedEventArgs (*class in arrangeit.windows.vdi*), 60  
 icon (*arrangeit.data.WindowModel* attribute), 32  
 icon (*arrangeit.view.ListedWindow* attribute), 35  
 icon (*arrangeit.windows.api.Package* attribute), 58  
 ICON\_SIZE (*arrangeit.settings.Settings* attribute), 46  
 IInspectable (*class in arrangeit.windows.vdi*), 60  
 images (*arrangeit.view.PropertyIcon* attribute), 35  
 images (*arrangeit.view.Resizable* attribute), 36  
 images (*arrangeit.view.Restored* attribute), 36  
 increased\_by\_fraction() (*in module arrangeit.utils*), 45  
 internal\_manager (*arrangeit.windows.vdi.VirtualDesktopsWin10* attribute), 62  
 IObjectArray (*class in arrangeit.windows.vdi*), 61  
 is\_applicable() (*arrangeit.base.BaseCollector* method), 23  
 is\_applicable() (*arrangeit.darwin.collector.Collector* method), 49  
 is\_applicable() (*arrangeit.linux.collector.Collector* method), 52  
 is\_applicable() (*arrangeit.windows.collector.Collector* method), 59  
 is\_changed (*arrangeit.data.WindowModel* attribute), 32  
 is\_cloaked() (*arrangeit.windows.api.Api* method), 56  
 is\_dwm\_composition\_enabled() (*arrangeit.windows.api.Api* method), 56  
 is\_resizable() (*arrangeit.base.BaseCollector* method), 23  
 is\_resizable() (*arrangeit.darwin.collector.Collector* method), 49  
 is\_resizable() (*arrangeit.linux.collector.Collector* method), 52  
 is\_resizable() (*arrangeit.windows.collector.Collector* method), 60  
 is\_restored() (*arrangeit.base.BaseCollector* method), 23  
 is\_restored() (*arrangeit.darwin.collector.Collector* method), 49  
 is\_restored() (*arrangeit.linux.collector.Collector* method), 52  
 is\_restored() (*arrangeit.windows.collector.Collector* method), 60  
 is\_setting() (*arrangeit.settings.Settings* class method), 47  
 is\_valid\_state() (*arrangeit.base.BaseCollector* method), 23  
 is\_valid\_state() (*arrangeit.darwin.collector.Collector* method), 49  
 is\_valid\_state() (*arrangeit.linux.collector.Collector* method), 53  
 is\_valid\_state() (*arrangeit.windows.collector.Collector* method), 60  
 is\_window\_in\_current\_desktop() (*arrangeit.windows.api.DummyVirtualDesktops* method), 58  
 is\_window\_in\_current\_desktop() (*ar-*

*rangeit.windows.vdi.VirtualDesktopsWin10*  
 method), 62  
 is\_ws\_changed (arrangeit.data.WindowModel  
 attribute), 32  
 IServiceProvider (class in arrangeit.windows.vdi),  
 61  
 IVirtualDesktop (class in arrangeit.windows.vdi),  
 61  
 IVirtualDesktopManager (class in ar-  
 rangeit.windows.vdi), 61  
 IVirtualDesktopManagerInternal (class in ar-  
 rangeit.windows.vdi), 61

## L

label (arrangeit.options.ColorOption attribute), 41  
 length (arrangeit.view.CornerWidget attribute), 34  
 listed\_window\_activated() (ar-  
 rangeit.base.BaseController method), 26  
 listed\_window\_activated\_by\_digit ()  
 (arrangeit.base.BaseController method), 26  
 ListedWindow (class in arrangeit.view), 34  
 listener (arrangeit.base.BaseMouse attribute), 30  
 LOCATE (arrangeit.settings.Settings attribute), 46  
 logo (arrangeit.options.AboutDialog attribute), 40

## M

main () (in module arrangeit.\_\_main\_\_), 21  
 mainloop () (arrangeit.base.BaseController method),  
 26  
 manager (arrangeit.windows.vdi.VirtualDesktopsWin10  
 attribute), 62  
 master (arrangeit.options.AboutDialog attribute), 40  
 master (arrangeit.options.CheckOption attribute), 41  
 master (arrangeit.options.ColorOption attribute), 41  
 master (arrangeit.options.OptionsDialog attribute), 42  
 master (arrangeit.options.ScaleOption attribute), 43  
 master (arrangeit.view.CornerWidget attribute), 34  
 master (arrangeit.view.ListedWindow attribute), 35  
 master (arrangeit.view.PropertyIcon attribute), 35  
 master (arrangeit.view.Statusbar attribute), 37  
 master (arrangeit.view.Toolbar attribute), 37  
 master (arrangeit.view.ViewApplication attribute), 37  
 master (arrangeit.view.WindowsList attribute), 38  
 master (arrangeit.view.Workspace attribute), 39  
 master (arrangeit.view.WorkspacesCollection at-  
 tribute), 40  
 max\_box (arrangeit.view.CornerWidget attribute), 34  
 max\_xy (arrangeit.view.CornerWidget attribute), 34  
 message (arrangeit.options.OptionsDialog attribute),  
 42  
 mocked\_for\_about () (in module  
 tests.unit.mock\_helpers), 112  
 mocked\_for\_about\_setup () (in module  
 tests.unit.mock\_helpers), 112  
 mocked\_for\_options () (in module  
 tests.unit.mock\_helpers), 112  
 mocked\_for\_options\_setup () (in module  
 tests.unit.mock\_helpers), 112  
 mocked\_setup () (in module tests.unit.mock\_helpers),  
 112  
 mocked\_setup\_root () (in module  
 tests.unit.mock\_helpers), 112  
 mocked\_setup\_view () (in module  
 tests.unit.mock\_helpers), 112  
 model (arrangeit.base.BaseController attribute), 26  
 mouse (arrangeit.base.BaseController attribute), 26  
 mouse\_move () (arrangeit.base.BaseController  
 method), 26  
 mouse\_scroll () (arrangeit.base.BaseController  
 method), 26  
 move () (arrangeit.base.BaseApp method), 22  
 move () (arrangeit.darwin.app.App method), 48  
 move () (arrangeit.linux.app.App method), 50  
 move () (arrangeit.windows.app.App method), 54  
 move\_and\_resize () (arrangeit.base.BaseApp  
 method), 22  
 move\_and\_resize () (arrangeit.darwin.app.App  
 method), 48  
 move\_and\_resize () (arrangeit.linux.app.App  
 method), 50  
 move\_and\_resize () (arrangeit.windows.app.App  
 method), 54  
 move\_cursor () (arrangeit.base.BaseMouse method),  
 30  
 move\_other\_to\_workspace () (ar-  
 rangeit.windows.app.App method), 54  
 move\_other\_window\_to\_desktop () (ar-  
 rangeit.windows.api.Api method), 56  
 move\_other\_window\_to\_desktop () (ar-  
 rangeit.windows.api.DummyVirtualDesktops  
 method), 58  
 move\_other\_window\_to\_desktop () (ar-  
 rangeit.windows.vdi.VirtualDesktopsWin10  
 method), 62  
 move\_own\_window\_to\_desktop () (ar-  
 rangeit.windows.api.Api method), 56  
 move\_own\_window\_to\_desktop () (ar-  
 rangeit.windows.api.DummyVirtualDesktops  
 method), 58  
 move\_own\_window\_to\_desktop () (ar-  
 rangeit.windows.vdi.VirtualDesktopsWin10  
 method), 62  
 move\_to\_corner () (arrangeit.base.BaseController  
 method), 26  
 move\_to\_workspace () (arrangeit.base.BaseApp  
 method), 22  
 move\_to\_workspace () (arrangeit.darwin.app.App  
 method), 48

*move\_to\_workspace()* (*arrangeit.linux.app.App method*), 51  
*move\_to\_workspace()* (*arrangeit.windows.app.App method*), 54

## N

*name* (*arrangeit.data.WindowModel attribute*), 32  
*name* (*arrangeit.options.CheckOption attribute*), 41  
*name* (*arrangeit.options.ColorOption attribute*), 41  
*name* (*arrangeit.options.ScaleOption attribute*), 43  
*name* (*arrangeit.view.Workspace attribute*), 39  
*nested()* (*in module tests.unit.nested\_helper*), 112  
*next()* (*arrangeit.base.BaseController method*), 27  
*number* (*arrangeit.view.Workspace attribute*), 39

## O

*off\_name* (*arrangeit.view.PropertyIcon attribute*), 35  
*off\_name* (*arrangeit.view.Resizable attribute*), 36  
*off\_name* (*arrangeit.view.Restored attribute*), 36  
*offset\_for\_intersections()* (*in module arrangeit.utils*), 45  
*on\_continue()* (*arrangeit.base.BaseController method*), 27  
*on\_destroy\_options()* (*arrangeit.options.OptionsDialog method*), 42  
*on\_focus()* (*arrangeit.base.BaseController method*), 27  
*on\_help\_click()* (*arrangeit.options.AboutDialog method*), 40  
*on\_key\_pressed()* (*arrangeit.base.BaseController method*), 27  
*on\_mouse\_left\_down()* (*arrangeit.base.BaseController method*), 27  
*on\_mouse\_middle\_down()* (*arrangeit.base.BaseController method*), 27  
*on\_mouse\_right\_down()* (*arrangeit.base.BaseController method*), 27  
*on\_move()* (*arrangeit.base.BaseMouse method*), 30  
*on\_name* (*arrangeit.view.PropertyIcon attribute*), 36  
*on\_name* (*arrangeit.view.Resizable attribute*), 36  
*on\_name* (*arrangeit.view.Restored attribute*), 36  
*on\_options\_click()* (*arrangeit.view.Toolbar method*), 37  
*on\_releases\_click()* (*arrangeit.options.AboutDialog method*), 41  
*on\_resizable\_change()* (*arrangeit.base.BaseController method*), 27  
*on\_restored\_change()* (*arrangeit.base.BaseController method*), 27  
*on\_save\_default()* (*arrangeit.options.OptionsDialog method*), 42  
*on\_scroll()* (*arrangeit.base.BaseMouse method*), 31  
*on\_show\_about()* (*arrangeit.options.OptionsDialog method*), 42  
*on\_update\_value()* (*arrangeit.options.CheckOption method*), 41  
*on\_update\_value()* (*arrangeit.options.ColorOption method*), 41  
*on\_update\_value()* (*arrangeit.options.FloatScaleOption method*), 41  
*on\_update\_value()* (*arrangeit.options.ScaleOption method*), 43  
*on\_widget\_enter()* (*arrangeit.view.ListedWindow method*), 35  
*on\_widget\_enter()* (*arrangeit.view.PropertyIcon method*), 36  
*on\_widget\_enter()* (*arrangeit.view.Workspace method*), 39  
*on\_widget\_leave()* (*arrangeit.view.ListedWindow method*), 35  
*on\_widget\_leave()* (*arrangeit.view.PropertyIcon method*), 36  
*on\_widget\_leave()* (*arrangeit.view.Workspace method*), 39  
*on\_window\_label\_button\_down()* (*arrangeit.view.WindowsList method*), 38  
*on\_workspace\_label\_button\_down()* (*arrangeit.view.WorkspacesCollection method*), 40  
*open\_image()* (*in module arrangeit.utils*), 46  
*OptionsDialog* (*class in arrangeit.options*), 41  
*OTHER* (*arrangeit.settings.Settings attribute*), 46

## P

*Package* (*class in arrangeit.windows.api*), 58  
*packages* (*arrangeit.windows.api.Api attribute*), 57  
*path* (*arrangeit.windows.api.Package attribute*), 58  
*place\_children()* (*arrangeit.view.WindowsList method*), 38  
*place\_on\_opposite\_corner()* (*arrangeit.base.BaseController method*), 27  
*place\_on\_top\_left()* (*arrangeit.base.BaseController method*), 28  
*place\_widget\_on\_position()* (*arrangeit.view.WindowsList method*), 39  
*platform\_path()* (*in module arrangeit.utils*), 46  
*platform\_supports\_packages()* (*in module arrangeit.windows.api*), 58  
*platform\_supports\_virtual\_desktops()* (*in module arrangeit.windows.api*), 58  
*platform\_user\_data\_path()* (*in module arrangeit.utils*), 46  
*prepare\_view()* (*arrangeit.base.BaseController method*), 28  
*PropertyIcon* (*class in arrangeit.view*), 35

pytestmark (*tests.unit.test\_windows\_api.TestWindowsAppHelperWin8\_cleanup()* (*arrangeit.base.BaseApp attribute*), 102  
 pytestmark (*tests.unit.test\_windows\_api.TestWindowsAppPrivateWin8\_cleanup()* (*arrangeit.windows.app.App method*), 54  
 pytestmark (*tests.unit.test\_windows\_api.TestWindowsAppPublicWin8\_on\_exposed* (*arrangeit.base.BaseController attribute*), 28  
 pytestmark (*tests.unit.test\_windows\_vdi.TestWindowsVirtualDesktopWin10\_on\_exposed* (*arrangeit.windows.controller.Controller attribute*), 60

## Q

quarter\_by\_smaller() (*in module arrangeit.utils*), 46  
 queue (*arrangeit.base.BaseMouse attribute*), 31

## R

read\_user\_settings() (*in module arrangeit.settings*), 47  
 recapture\_mouse() (*arrangeit.base.BaseController method*), 28  
 rect (*arrangeit.data.WindowModel attribute*), 32  
 Rectangle (*class in arrangeit.utils*), 43  
 release\_mouse() (*arrangeit.base.BaseController method*), 28  
 RELEASES\_PAGE\_URL (*arrangeit.settings.Settings attribute*), 47  
 remove\_listed\_window() (*arrangeit.base.BaseController method*), 28  
 repopulate\_for\_wid() (*arrangeit.data.WindowsCollection method*), 33  
 rerun\_from\_window() (*arrangeit.base.BaseApp method*), 22  
 reset\_bindings() (*arrangeit.view.ViewApplication method*), 37  
 resizable (*arrangeit.data.WindowModel attribute*), 32  
 Resizable (*class in arrangeit.view*), 36  
 RESIZE (*arrangeit.settings.Settings attribute*), 47  
 resizing\_state\_counterpart() (*arrangeit.base.BaseController method*), 28  
 restored (*arrangeit.data.WindowModel attribute*), 32  
 Restored (*class in arrangeit.view*), 36  
 run() (*arrangeit.base.BaseApp method*), 22  
 run() (*arrangeit.base.BaseCollector method*), 23  
 run() (*arrangeit.base.BaseController method*), 28  
 run\_task() (*arrangeit.base.BaseApp method*), 22

## S

save() (*arrangeit.base.BaseController method*), 28  
 save\_default() (*arrangeit.base.BaseApp method*), 22  
 ScaleOption (*class in arrangeit.options*), 43  
 screenshot (*arrangeit.base.BaseController attribute*), 28

screenshot\_widget (*arrangeit.base.BaseController attribute*), 28  
 select\_active() (*arrangeit.view.WorkspacesCollection method*), 40  
 set\_changed() (*arrangeit.data.WindowModel method*), 32  
 set\_corner() (*arrangeit.view.CornerWidget method*), 34  
 set\_default\_geometry() (*arrangeit.base.BaseController method*), 28  
 set\_icon() (*in module arrangeit.utils*), 46  
 set\_minimum\_size() (*arrangeit.base.BaseController method*), 28  
 set\_screenshot() (*arrangeit.base.BaseController method*), 29  
 set\_timer() (*arrangeit.base.BaseController method*), 29  
 set\_timer() (*arrangeit.options.OptionsDialog method*), 42  
 set\_value() (*arrangeit.view.PropertyIcon method*), 36  
 setting\_type() (*arrangeit.settings.Settings class method*), 47  
 Settings (*class in arrangeit.settings*), 46  
 SettingsMetaclass (*class in arrangeit.settings*), 47  
 setup() (*arrangeit.base.BaseController method*), 29  
 setup() (*arrangeit.data.WindowModel method*), 32  
 setup\_bindings() (*arrangeit.options.OptionsDialog method*), 42  
 setup\_bindings() (*arrangeit.view.ListedWindow method*), 35  
 setup\_bindings() (*arrangeit.view.PropertyIcon method*), 36  
 setup\_bindings() (*arrangeit.view.ViewApplication method*), 37  
 setup\_bindings() (*arrangeit.view.Workspace method*), 39  
 setup\_collector() (*arrangeit.base.BaseApp method*), 22  
 setup\_controller() (*arrangeit.base.BaseApp method*), 22  
 setup\_corner() (*arrangeit.base.BaseController method*), 29



setup\_corner() (arrangeit.view.ViewApplication method), 37  
 setup\_files\_section() (arrangeit.options.OptionsDialog method), 42  
 setup\_icon() (arrangeit.view.ViewApplication method), 37  
 setup\_name() (arrangeit.view.ViewApplication method), 37  
 setup\_package() (arrangeit.windows.api.Package method), 58  
 setup\_resizable() (arrangeit.view.ViewApplication method), 38  
 setup\_restored() (arrangeit.view.ViewApplication method), 38  
 setup\_root\_window() (arrangeit.base.BaseController method), 29  
 setup\_root\_window() (arrangeit.linux.controller.Controller method), 53  
 setup\_root\_window() (arrangeit.windows.controller.Controller method), 60  
 setup\_section() (arrangeit.options.OptionsDialog method), 42  
 setup\_statusbar() (arrangeit.view.ViewApplication method), 38  
 setup\_thumbnail() (arrangeit.windows.api.Api method), 57  
 setup\_title() (arrangeit.view.ViewApplication method), 38  
 setup\_toolbar() (arrangeit.view.ViewApplication method), 38  
 setup\_widgets() (arrangeit.options.AboutDialog method), 41  
 setup\_widgets() (arrangeit.options.OptionsDialog method), 43  
 setup\_widgets() (arrangeit.view.CornerWidget method), 34  
 setup\_widgets() (arrangeit.view.ListedWindow method), 35  
 setup\_widgets() (arrangeit.view.PropertyIcon method), 36  
 setup\_widgets() (arrangeit.view.Statusbar method), 37  
 setup\_widgets() (arrangeit.view.Toolbar method), 37  
 setup\_widgets() (arrangeit.view.ViewApplication method), 38  
 setup\_widgets() (arrangeit.view.Workspace method), 39  
 setup\_windows() (arrangeit.view.ViewApplication method), 38  
 setup\_workspaces() (arrangeit.view.ViewApplication method), 38  
 shift (arrangeit.view.CornerWidget attribute), 34  
 show\_root() (arrangeit.view.ViewApplication method), 38  
 shutdown() (arrangeit.base.BaseController method), 29  
 size (arrangeit.data.WindowsCollection attribute), 33  
 skip\_current\_window() (arrangeit.base.BaseController method), 29  
 snapping\_targets (arrangeit.base.BaseController attribute), 29  
 sort() (arrangeit.data.WindowsCollection method), 33  
 start() (arrangeit.base.BaseMouse method), 31  
 startup() (arrangeit.view.ViewApplication method), 38  
 state (arrangeit.base.BaseController attribute), 29  
 Statusbar (class in arrangeit.view), 36  
 stop() (arrangeit.base.BaseMouse method), 31  
 switch\_resizable() (arrangeit.base.BaseController method), 29  
 switch\_restored() (arrangeit.base.BaseController method), 29  
 switch\_workspace() (arrangeit.base.BaseController method), 29

## T

test\_AboutDialog\_init\_calls\_geometry\_on\_master\_pos() (tests.unit.test\_options.TestAboutDialog method), 83  
 test\_AboutDialog\_init\_calls\_setup\_widgets() (tests.unit.test\_options.TestAboutDialog method), 83  
 test\_AboutDialog\_init\_calls\_super\_with\_master\_arg() (tests.unit.test\_options.TestAboutDialog method), 83  
 test\_AboutDialog\_init\_sets\_about\_dialog\_title() (tests.unit.test\_options.TestAboutDialog method), 83  
 test\_AboutDialog\_init\_sets\_master\_attribute() (tests.unit.test\_options.TestAboutDialog method), 83  
 test\_AboutDialog\_inits\_attributes() (tests.unit.test\_options.TestAboutDialog method), 83  
 test\_AboutDialog\_issubclass\_of\_Toplevel() (tests.unit.test\_options.TestAboutDialog method), 83  
 test\_AboutDialog\_on\_help\_click\_opens\_webbrowser() (tests.unit.test\_options.TestAboutDialog method), 83  
 test\_AboutDialog\_on\_releases\_click\_opens\_webbrowser() (tests.unit.test\_options.TestAboutDialog method), 83





test\_api\_Api\_get\_package\_calls\_\_package\_infb\_Apfferecttableframefor\_hwnd\_calls\_wintypes\_DW  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublicWintypes.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 111 method), 110

test\_api\_Api\_get\_package\_calls\_\_package\_infb\_Apfferecttableframefor\_hwnd\_returns\_None()  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublicWintypes.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 111 method), 110

test\_api\_Api\_get\_package\_calls\_Package()test\_Api\_extended\_frame\_rect\_calls\_\_dwm\_get\_window  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublicWintypes.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 111 method), 110

test\_api\_Api\_get\_package\_calls\_PACKAGE\_INFORMATION\_extended\_frame\_rect\_calls\_and\_returns\_wint  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublicWintypes.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 111 method), 110

test\_api\_Api\_get\_package\_returns\_empty\_Package\_Api\_extended\_frame\_rect\_calls\_ctypes\_byref()  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublicWintypes.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 111 method), 110

test\_api\_Api\_inits\_attr\_as\_None() test\_Api\_extended\_frame\_rect\_calls\_ctypes\_sizeof()  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 111 method), 110

test\_api\_Api\_inits\_empty\_attr() test\_Api\_extended\_frame\_rect\_calls\_wintypes\_RECT()  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 111 method), 110

test\_api\_DummyVirtualDesktops\_defines\_getest\_Api\_extended\_frame\_rect\_returns\_None()  
 (tests.unit.test\_windows\_api.TestDummyVirtualDesktops (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 99 method), 110

test\_api\_DummyVirtualDesktops\_defines\_getest\_Api\_get\_desktops\_by\_type\_calls\_and\_returns\_\_ge  
 (tests.unit.test\_windows\_api.TestDummyVirtualDesktops (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 99 method), 110

test\_api\_DummyVirtualDesktops\_defines\_istest\_Api\_get\_desktops\_for\_window\_calls\_dummy  
 (tests.unit.test\_windows\_api.TestDummyVirtualDesktops (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 99 method), 110

test\_api\_DummyVirtualDesktops\_defines\_movestest\_Api\_get\_desktops\_for\_window\_calls\_wint  
 (tests.unit.test\_windows\_api.TestDummyVirtualDesktops (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 99 method), 110

test\_api\_DummyVirtualDesktops\_defines\_movestest\_Api\_get\_desktops\_for\_window\_calls\_wint  
 (tests.unit.test\_windows\_api.TestDummyVirtualDesktops (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 99 method), 110

test\_Api\_executable\_name\_for\_hwnd\_calls\_test\_Api\_get\_desktops\_calls\_win10\_vdi\_get\_desktops  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 109 method), 110

test\_Api\_executable\_name\_for\_hwnd\_calls\_tget\_Process\_data\_of\_process\_id\_list\_of\_two\_tuples()  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 109 method), 110

test\_Api\_executable\_name\_for\_hwnd\_calls\_tget\_Process\_data\_of\_process\_id\_calls\_and\_returns\_c  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 109 method), 110

test\_Api\_executable\_name\_for\_hwnd\_calls\_topen\_Process\_id\_calls\_\_dwm\_get\_window\_attribute  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 109 method), 110

test\_Api\_executable\_name\_for\_hwnd\_calls\_andstr\_Api\_returns\_exported\_names\_from\_bytestrapvdi()is\_window  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 109 method), 110

test\_Api\_executable\_name\_for\_hwnd\_calls\_tcreatestr\_Api\_returns\_exported\_names\_from\_bytestrapvdi()is\_window  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 110 method), 110

*test\_api\_is\_cloaked\_calls\_ctypes\_byref()* *test\_api\_Package\_\_get\_first\_image\_calls\_splitext()*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_is\_cloaked\_calls\_ctypes\_sizeof()* *test\_api\_Package\_\_get\_first\_image\_catches\_exception*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_is\_cloaked\_calls\_wintypes\_DWORD()* *test\_api\_Package\_\_get\_manifest\_root\_calls\_and\_return*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_is\_cloaked\_returns\_False\_for\_0\_test\_variable\_Package\_\_get\_manifest\_root\_calls\_os\_path\_e*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_is\_cloaked\_returns\_False\_for\_noteak()* *test\_api\_Package\_\_get\_manifest\_root\_calls\_os\_path\_e*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_is\_dwm\_composition\_enabled\_callstestwapisPackage\_\_get\_manifest\_root\_calls\_parse()*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_is\_dwm\_composition\_enabled\_callstesttypes\_Backstage\_\_get\_manifest\_root\_returns\_true\_i*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_is\_dwm\_composition\_enabled\_callstewintypes\_BAB0014\_\_init\_\_calls\_setup\_package()*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_is\_dwm\_composition\_enabled\_returnstvariable\_Package\_\_init\_\_sets\_path\_attribute\_from\_pi*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_move\_other\_window\_to\_desktop\_callstestdummyPackage\_\_move\_other\_window\_to\_desktop\_calls\_re\_ma*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_move\_other\_window\_to\_desktop\_callstewapi10Package\_\_move\_other\_window\_to\_desktop\_\_element\_returns\_emp*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_move\_own\_window\_to\_desktop\_callstestwapisPackage\_\_own\_window\_for\_desktop\_\_returns\_fi*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_move\_own\_window\_to\_desktop\_callstewinapiPackage\_\_own\_window\_for\_desktop\_\_namespace\_*  
*(tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 110* *method), 102*

*test\_api\_Package\_\_get\_first\_image\_calls\_andrepairPackage\_\_get\_size\_of\_image\_\_p\_name\_calls\_iter\_on\_next*  
*(tests.unit.test\_windows\_api.TestWindowsApiPackage* *(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 102* *method), 102*

*test\_api\_Package\_\_get\_first\_image\_calls\_bopen\_imagePackage\_\_get\_size\_of\_image\_\_p\_name\_calls\_next()*  
*(tests.unit.test\_windows\_api.TestWindowsApiPackage* *(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 102* *method), 103*

*test\_api\_Package\_\_get\_first\_image\_calls\_bspapiPackage\_\_setup\_app\_name\_calls\_root\_iter()*  
*(tests.unit.test\_windows\_api.TestWindowsApiPackage* *(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 102* *method), 103*

*test\_api\_Package\_\_get\_first\_image\_calls\_bspapiPackage\_\_setup\_app\_name\_sets\_app\_name\_att*  
*(tests.unit.test\_windows\_api.TestWindowsApiPackage* *(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 102* *method), 103*

*test\_api\_Package\_\_get\_first\_image\_calls\_pstestapi()Package\_\_setup\_icon\_appends\_once\_to\_source*  
*(tests.unit.test\_windows\_api.TestWindowsApiPackage* *(tests.unit.test\_windows\_api.TestWindowsApiPackage*  
*method), 102* *method), 103*

test\_api\_package\_setup\_icon\_appends\_to\_search\_api\_from\_property\_state\_calls\_TITLEBARINFO()  
 (tests.unit.test\_windows\_api.TestWindowsApiPackage (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 103 method), 111

test\_api\_package\_setup\_icon\_calls\_namespace\_for\_element\_info\_state\_returns\_None()  
 (tests.unit.test\_windows\_api.TestWindowsApiPackage (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 103 method), 111

test\_api\_package\_setup\_icon\_calls\_iter\_base\_api\_title\_info\_state\_returns\_value()  
 (tests.unit.test\_windows\_api.TestWindowsApiPackage (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 103 method), 111

test\_api\_package\_setup\_icon\_calls\_next(test\_api\_unregister\_thumbnail\_calls\_\_dwm\_unregister\_thumbnail\_calls\_\_dwm\_unregister\_thumbnail\_returns\_value())  
 (tests.unit.test\_windows\_api.TestWindowsApiPackage (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 103 method), 111

test\_api\_package\_setup\_icon\_calls\_root\_iter(test\_api\_unregister\_thumbnail\_returns\_True\_on\_error())  
 (tests.unit.test\_windows\_api.TestWindowsApiPackage (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 103 method), 111

test\_api\_package\_inits\_empty\_attr() test\_api\_window\_info\_extended\_style\_calls\_\_get\_window\_info\_extended\_style()  
 (tests.unit.test\_windows\_api.TestWindowsApiPackage (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 103 method), 111

test\_api\_package\_inits\_empty\_icon() test\_api\_window\_info\_extended\_style\_calls\_ctypes\_byref()  
 (tests.unit.test\_windows\_api.TestWindowsApiPackage (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 103 method), 111

test\_api\_package\_setup\_package\_calls\_\_get\_main\_api\_window(info\_extended\_style\_calls\_WINDOWINFO\_EXTENDED\_STYLE)  
 (tests.unit.test\_windows\_api.TestWindowsApiPackage (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 103 method), 111

test\_api\_package\_setup\_package\_calls\_\_setup\_main\_api\_window(info\_extended\_style\_returns\_None())  
 (tests.unit.test\_windows\_api.TestWindowsApiPackage (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 103 method), 111

test\_api\_package\_setup\_package\_calls\_\_setup\_api(window\_info\_extended\_style\_returns\_value())  
 (tests.unit.test\_windows\_api.TestWindowsApiPackage (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic  
 method), 103 method), 111

test\_api\_setup\_thumbnail\_calls\_\_dwm\_register\_thumbnail(test\_BaseApp\_\_initialize\_snapping\_sources\_calls\_collect\_snapping\_sources())  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic (tests.unit.test\_base.TestBaseApp method), 63  
 method), 110 test\_BaseApp\_\_initialize\_snapping\_sources\_calls\_get\_snapping\_sources())  
 (tests.unit.test\_base.TestBaseApp method), 63

test\_api\_setup\_thumbnail\_calls\_and\_returns\_updated\_api(test\_BaseApp\_\_initialize\_snapping\_sources\_calls\_collect\_snapping\_sources())  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic (tests.unit.test\_base.TestBaseApp method), 63  
 method), 110 test\_BaseApp\_\_initialize\_snapping\_sources\_calls\_get\_snapping\_sources())  
 (tests.unit.test\_base.TestBaseApp method), 63

test\_api\_setup\_thumbnail\_calls\_wintypes\_byref(test\_BaseApp\_\_initialize\_snapping\_sources\_functionality())  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic (tests.unit.test\_base.TestBaseApp method), 63  
 method), 110 test\_BaseApp\_\_save\_setting\_calls\_json\_load\_once())  
 (tests.unit.test\_base.TestBaseApp method), 63

test\_api\_setup\_thumbnail\_calls\_wintypes\_HANDLE(test\_BaseApp\_\_save\_setting\_calls\_platform\_user\_data())  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic (tests.unit.test\_base.TestBaseApp method), 63  
 method), 110 test\_BaseApp\_\_save\_setting\_catches\_exception\_and\_catches\_exception())  
 (tests.unit.test\_base.TestBaseApp method), 63

test\_api\_setup\_thumbnail\_returns\_None() test\_BaseApp\_\_save\_setting\_checks\_if\_directory\_exists()  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic (tests.unit.test\_base.TestBaseApp method), 63  
 method), 111 test\_BaseApp\_\_save\_setting\_checks\_if\_file\_exists())  
 (tests.unit.test\_base.TestBaseApp method), 63

test\_api\_title\_info\_state\_calls\_\_get\_titlebar\_info(test\_BaseApp\_\_save\_setting\_creates\_directory())  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic (tests.unit.test\_base.TestBaseApp method), 63  
 method), 111 test\_BaseApp\_\_save\_setting\_overwrites\_settings\_file())  
 (tests.unit.test\_base.TestBaseApp method), 63

test\_api\_title\_info\_state\_calls\_ctypes\_byref(test\_BaseApp\_\_save\_setting\_updates\_settings\_file())  
 (tests.unit.test\_windows\_api\_api.TestWindowsApiApiPublic (tests.unit.test\_base.TestBaseApp method), 64  
 method), 111

test\_BaseApp\_save\_setting\_writes\_to\_settings\_file() (tests.unit.test\_base.TestBaseApp method), 64  
 test\_BaseApp\_activate\_root\_raises\_NotImplementedError (tests.unit.test\_base.TestBaseApp method), 64  
 test\_BaseApp\_change\_setting\_calls\_save\_setting (tests.unit.test\_base.TestBaseApp method), 64  
 test\_BaseApp\_change\_setting\_calls\_is\_setting (tests.unit.test\_base.TestBaseApp method), 64  
 test\_BaseApp\_change\_setting\_calls\_is\_setting\_in (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_change\_setting\_changes\_valid\_setting (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_change\_setting\_returns\_change\_setting (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_change\_setting\_returns\_change\_setting (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_change\_settings\_color\_group\_calls (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_change\_settings\_color\_group\_calls (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_change\_settings\_color\_group\_calls (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_change\_settings\_color\_group\_calls (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_create\_snapping\_sources\_calls\_init (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_create\_snapping\_sources\_calls\_collect (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_create\_snapping\_sources\_calls\_util (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_create\_snapping\_sources\_excludes (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_create\_snapping\_sources\_functional (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_create\_snapping\_sources\_includes (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_create\_snapping\_sources\_returns\_dir (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_create\_snapping\_sources\_uses\_change (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_defines\_screenshot\_cleanup (tests.unit.test\_base.TestBaseApp method), 64  
 test\_BaseApp\_grab\_window\_screen\_raises\_NotImplementedError (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_initialization\_calls\_setup\_collect (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_initialization\_calls\_setup\_test\_base (tests.unit.test\_base.TestBaseApp method), 64  
 test\_BaseApp\_initialization\_instantiates\_collec (tests.unit.test\_base.TestBaseApp method), 64  
 test\_BaseApp\_initialization\_instantiates\_contr (tests.unit.test\_base.TestBaseApp method), 64  
 test\_BaseApp\_initialization\_instantiatestest (tests.unit.test\_base.TestBaseApp method), 64  
 test\_BaseApp\_inits\_attr\_as\_None (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_move\_and\_resize\_raises\_NotImplementedError (tests.unit.test\_base.TestBaseApp method), 64  
 test\_BaseApp\_move\_raises\_NotImplementedError (tests.unit.test\_base.TestBaseApp method), 64  
 test\_BaseApp\_move\_to\_workspace\_raises\_NotImplementedError (tests.unit.test\_base.TestBaseApp method), 64  
 test\_BaseApp\_rerun\_from\_window\_calls\_repopulate\_for (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_run\_calls\_collector\_run (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_run\_calls\_controller\_run (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_run\_calls\_controller\_run\_with\_valid\_a (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_run\_calls\_WindowsCollection\_generator (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_run\_task\_calls\_related\_methods (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_save\_default\_calls\_collection\_export (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_save\_default\_calls\_json\_dump (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_save\_default\_calls\_platform\_user\_data (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_save\_default\_checks\_if\_directory\_exists (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_save\_default\_creates\_directory (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_screenshot\_cleanup\_returns\_None (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_setup\_collector\_calls\_get\_component\_c (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_setup\_controller\_calls\_get\_component\_c (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_setup\_controller\_calls\_get\_monitors\_rects\_raises\_NotImplementedError (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_setup\_controller\_calls\_get\_smallest\_monitor\_size\_calls (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_setup\_controller\_calls\_get\_smallest\_monitor\_size\_returns (tests.unit.test\_base.TestBaseApp method), 65  
 test\_BaseApp\_setup\_controller\_calls\_get\_windows\_raises\_NotImplementedError (tests.unit.test\_base.TestBaseApp method), 65

*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller\_domain\_logic.TestBaseControllerL*  
*method), 66*

*test\_BaseCollector\_get\_workspace\_number\_festwBaseControllerRaisesNotImplementedErrorcalls\_check\_sna*  
*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseCollector\_get\_workspace\_number\_raisesBaseControllerRaisesRange(position\_calls\_get\_root*  
*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseCollector\_initialization\_instantiatedBaseControllerChange\_position\_calls\_master\_ge*  
*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseCollector\_inits\_collection\_as\_Nbase()BaseController\_change\_position\_not\_calling\_set*  
*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseCollector\_is\_applicable\_raises\_NotImplementedError(change\_setting\_calls\_run\_task)*  
*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseCollector\_is\_resizable\_raises\_NotImplementedError(change\_size\_calls\_check\_current*  
*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseCollector\_is\_restored\_raises\_NotImplementedError(change\_size\_calls\_check\_snappin*  
*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseCollector\_is\_valid\_state\_raises\_NotImplementedError(change\_size\_calls\_master\_geomet*  
*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseCollector\_run\_calls\_add\_window(test\_BaseController\_change\_size\_calls\_set\_minimum*  
*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseCollector\_run\_calls\_check\_windowwidthtest\_BaseController\_change\_size\_not\_calling\_set\_ge*  
*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseCollector\_run\_calls\_collection\_set(BaseController\_change\_size\_valid\_x\_and\_y)*  
*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseCollector\_run\_calls\_get\_windowswidthtest\_BaseController\_change\_size\_with\_min\_valid\_x\_a*  
*(tests.unit.test\_base.TestBaseCollector*  
*method), 65*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseController\_apply\_snapping\_calls\_test\_BaseControllerCheck\_current\_size\_for\_invalid*  
*(tests.unit.test\_basecontroller\_domain\_logic.TestBaseControllerDomainLogic*  
*method), 66*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 69*

*test\_BaseController\_apply\_snapping\_calls\_test\_BaseControllerCheck\_current\_size\_for\_valid\_xy*  
*(tests.unit.test\_basecontroller\_domain\_logic.TestBaseControllerDomainLogic*  
*method), 66*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 70*

*test\_BaseController\_apply\_snapping\_calls\_test\_BaseControllerCheck\_mouse\_calls\_after\_idle\_w*  
*(tests.unit.test\_basecontroller\_domain\_logic.TestBaseControllerDomainLogic*  
*method), 66*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 70*

*test\_BaseController\_apply\_snapping\_changes\_test\_BaseControllerCheck\_mouse\_calls\_after\_idle\_w*  
*(tests.unit.test\_basecontroller\_domain\_logic.TestBaseControllerDomainLogic*  
*method), 66*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 70*

*test\_BaseController\_apply\_snapping\_changes\_test\_BaseControllerCheck\_mouse\_calls\_after\_with\_it*  
*(tests.unit.test\_basecontroller\_domain\_logic.TestBaseControllerDomainLogic*  
*method), 66*

*(tests.unit.test\_basecontroller.TestBaseController*  
*method), 70*

*test\_BaseController\_apply\_snapping\_not\_called\_in\_BaseControllerCheck\_mouse\_calls\_mouse\_get\_it*



(tests.unit.test\_basecontroller:TestBaseController  
 method), 70

test\_BaseController\_check\_snapping\_calls\_apply\_base\_snapping\_display\_message\_not\_calling\_set  
 (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerDomainLogicBaseController:TestBaseController  
 method), 67

test\_BaseController\_check\_snapping\_calls\_set\_base\_controller\_display\_message\_sets\_status\_bar  
 (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerDomainLogicBaseController:TestBaseController  
 method), 67

test\_BaseController\_check\_snapping\_calls\_get\_base\_controller\_get\_root\_rect\_functionality()  
 (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerDomainLogicBaseController:TestBaseController  
 method), 67

test\_BaseController\_check\_snapping\_calls\_get\_base\_controller\_res\_info\_for\_size\_app\_attribute()  
 (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerDomainLogicBaseController:TestBaseController  
 method), 67

test\_BaseController\_check\_snapping\_calls\_get\_base\_controller\_res\_info\_for\_resize\_app\_calls\_setup()  
 (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerDomainLogicBaseController:TestBaseController  
 method), 67

test\_BaseController\_check\_snapping\_calls\_test\_base\_controller\_sectionalization\_instantiates\_Mo  
 (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerDomainLogicBaseController:TestBaseController  
 method), 67

test\_BaseController\_check\_snapping\_not\_called\_in\_base\_controller\_initialization\_instantiates\_Win  
 (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerDomainLogicBaseController:TestBaseController  
 method), 67

test\_BaseController\_check\_snapping\_snapping\_test\_base\_controller\_inits\_attr\_as\_None()  
 (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerDomainLogicBaseController:TestBaseController  
 method), 67

test\_BaseController\_check\_snapping\_state\_test\_base\_controller\_for\_both\_axes\_snapping\_when\_exposed  
 (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerDomainLogicBaseController:TestBaseController  
 method), 67

test\_BaseController\_check\_snapping\_state\_test\_base\_controller\_for\_single\_axis\_snapping\_activated\_by\_dig  
 (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerDomainLogicBaseController:TestBaseController  
 method), 67

test\_BaseController\_check\_snapping\_state\_test\_base\_controller\_for\_both\_axes\_snapping\_activated\_by\_dig  
 (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerDomainLogicBaseController:TestBaseController  
 method), 67

test\_BaseController\_check\_snapping\_state\_test\_base\_controller\_for\_single\_axis\_snapping\_activated\_by\_dig  
 (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerDomainLogicBaseController:TestBaseController  
 method), 67

test\_BaseController\_cycle\_corners\_calls\_display\_message\_listed\_window\_activated\_calls\_  
 (tests.unit.test\_basecontroller:TestBaseController (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerL  
 method), 70

test\_BaseController\_cycle\_corners\_calls\_move\_base\_controller\_listed\_window\_activated\_calls\_  
 (tests.unit.test\_basecontroller:TestBaseController (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerL  
 method), 70

test\_BaseController\_cycle\_corners\_counters\_base\_controller\_listed\_window\_activated\_calls\_  
 (tests.unit.test\_basecontroller:TestBaseController (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerL  
 method), 70

test\_BaseController\_cycle\_corners\_counters\_base\_controller\_listed\_window\_activated\_calls\_  
 (tests.unit.test\_basecontroller:TestBaseController (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerL  
 method), 70

test\_BaseController\_cycle\_corners\_not\_called\_in\_base\_controller\_listed\_window\_activated\_calls\_  
 (tests.unit.test\_basecontroller:TestBaseController (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerL  
 method), 70

test\_BaseController\_display\_message\_calls\_set\_base\_controller\_listed\_window\_activated\_calls\_  
 (tests.unit.test\_basecontroller:TestBaseController (tests.unit.test\_basecontroller\_domain\_logic:TestBaseControllerL  
 method), 70



*(tests.unit.test\_basecontroller\_domain\_logic.TestBaseControllerDomainLogicTestBaseController*  
*method), 68* *(tests.unit.test\_basecontroller\_domain\_logic.TestBaseControllerDomainLogicTestBaseController*  
*method), 71*  
test\_BaseController\_next\_sets\_state\_attr\_test\_BaseController\_next\_sets\_state\_attr\_key\_pressed\_returns\_break ()  
*(tests.unit.test\_basecontroller\_domain\_logic.TestBaseControllerDomainLogicTestBaseController*  
*method), 68* *(tests.unit.test\_basecontroller\_domain\_logic.TestBaseControllerDomainLogicTestBaseController*  
*method), 71*  
test\_BaseController\_on\_continue\_calls\_recapture\_mouse\_controller\_on\_mouse\_left\_down\_calls\_update  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_continue\_returns\_break\_BaseController\_on\_mouse\_left\_down\_returns\_break  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_focus\_calls\_focus\_get(BaseController\_on\_mouse\_middle\_down\_calls\_release  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_focus\_calls\_run\_task(BaseController\_on\_mouse\_middle\_down\_returns\_break  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_focus\_not\_calling\_test\_BaseController\_on\_mouse\_right\_down\_calls\_skip  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_focus\_not\_returns\_break\_BaseController\_on\_mouse\_right\_down\_returns\_break  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_focus\_returns\_break\_test\_BaseController\_on\_mouse\_scroll\_calls\_counter\_t  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_key\_pressed\_calls\_test\_BaseController\_on\_resizable\_calls\_recapture\_m  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_key\_pressed\_calls\_test\_BaseController\_on\_resizable\_calls\_switch\_resi  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_key\_pressed\_calls\_test\_BaseController\_on\_resizable\_returns\_break ()  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_key\_pressed\_calls\_test\_BaseController\_on\_restored\_change\_calls\_recapt  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_key\_pressed\_calls\_test\_BaseController\_on\_restored\_change\_calls\_switc  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_key\_pressed\_for\_digit\_BaseController\_on\_restored\_change\_by\_digit\_key  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_key\_pressed\_for\_digit\_BaseController\_on\_restored\_change\_by\_digit\_key  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 71*  
test\_BaseController\_on\_key\_pressed\_for\_Enter\_BaseController\_on\_restored\_change\_calls  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 72*  
test\_BaseController\_on\_key\_pressed\_for\_Escape\_BaseController\_on\_restored\_change\_calls  
*(tests.unit.test\_basecontroller.TestBaseController* *(tests.unit.test\_basecontroller.TestBaseController*  
*method), 71* *method), 72*  
test\_BaseController\_on\_key\_pressed\_for\_funct\_BaseController\_on\_restored\_change\_calls\_move\_cu

(tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_place\_on\_top\_left\_calls\_test\_BaseController\_remove\_listed\_window\_not\_calling  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_prepare\_view\_calls\_WindowBaseController\_windows\_without\_fixes\_counterpart ()  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_prepare\_view\_calls\_WorkspaceController\_add\_workspaces\_viate\_root\_task ()  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_recapture\_mouse\_calls\_test\_BaseController\_run\_calls\_display\_message ()  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_recapture\_mouse\_calls\_test\_BaseController\_run\_calls\_mainloop ()  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_recapture\_mouse\_calls\_test\_BaseController\_run\_calls\_mouse\_start ()  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_recapture\_mouse\_calls\_test\_BaseController\_run\_calls\_next ()  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_recapture\_mouse\_calls\_test\_BaseController\_run\_calls\_prepare\_view ()  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_recapture\_mouse\_calls\_test\_BaseController\_run\_calls\_view\_startup ()  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_recapture\_mouse\_changes\_test\_BaseController\_run\_sets\_generator\_attr\_from\_pi  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_release\_mouse\_calls\_test\_BaseController\_save\_runs\_related\_task ()  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_release\_mouse\_calls\_test\_BaseController\_set\_default\_geometry\_calls\_geomet  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_release\_mouse\_calls\_test\_BaseController\_set\_default\_geometry\_calls\_get  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_release\_mouse\_calls\_test\_BaseController\_set\_default\_geometry\_calls\_qua  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_release\_mouse\_changes\_test\_BaseController\_set\_default\_geometry\_not\_callin  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_release\_mouse\_stops\_test\_BaseController\_set\_default\_geometry\_not\_chang  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_remove\_listed\_window\_calls\_test\_BaseController\_set\_default\_geometry\_sets\_defau  
 (tests.unit.test\_basecontroller.TestBaseController  
 method), 72

test\_BaseController\_remove\_listed\_window\_calls\_test\_BaseController\_set\_minimum\_size\_functionality

<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_set_screenshot_calls_grab_window_screen(\$setup_root_window_calls_config)	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_set_screenshot_calls_master_base_controller_setup_root_window_calls_wm_att	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_set_screenshot_calls_test_base_controller_setup_root_window_not_calling_	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_set_screenshot_config_sets_base_controller_width_down_calls_master_destroy(	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_set_screenshot_places_test_base_controller(\$shutdown_raises_SystemExit(	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_set_screenshot_returns_test_base_controller(\$shutdown_stops_mouse(	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_set_screenshot_sets_test_base_controller_resizable(\$t_window_calls_disp	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_set_timer_calls_after_test_base_controller_skip_current_window_calls_mode	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_set_timer_calls_after_test_base_controller_skip_current_window_calls_next	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_set_timer_not_calling_test_base_controller_switch_resizable_calls_display	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_set_timer_sets_timer_test_base_controller_switch_resizable_calls_widget_	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_setup_calls_get_screenshot_widget(\$t_controller_switch_resizable_functionality	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_setup_calls_get_tkint_test_base_controller_switch_restored_calls_display_r	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_setup_calls_setup_root_widget(\$t_controller_switch_restored_calls_widget_se	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_setup_corner_calls_ctest_base_controller_switch_restored_functionality(	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>
test_BaseController_setup_corner_calls_get_test_base_controller_switch_workspace_calls_display	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 74</i>
test_BaseController_setup_corner_calls_view_test_base_controller_switch_workspace_calls_get_root	
<i>(tests.unit.test_basecontroller:TestBaseController method), 73</i>	<i>(tests.unit.test_basecontroller:TestBaseController method), 74</i>
test_BaseController_setup_initializes_ViewAppBaseController(\$t_controller_switch_workspace_calls_task_mov	





*(tests.unit.test\_options.TestColorOption method), 84*
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*

*test\_ColorOption\_on\_update\_value\_calls\_changeDarwinCollector\_\_running\_apps\_ids\_calls\_runni*  
*(tests.unit.test\_options.TestColorOption method), 84*
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*

*test\_ColorOption\_on\_update\_value\_returns\_testDarwinCollector\_\_running\_apps\_ids\_calls\_share*  
*(tests.unit.test\_options.TestColorOption method), 84*
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*

*test\_darwin\_utils\_user\_data\_path\_calls\_NSSearchPathForDirectoriesInHomeApps(ids\_functionalit*  
*(tests.unit.test\_darwin.TestDarwinUtils method), 92*
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*

*test\_darwin\_utils\_user\_data\_path\_calls\_ospathDarwinCollector\_add\_window\_calls\_methods()*  
*(tests.unit.test\_darwin.TestDarwinUtils method), 92*
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*

*test\_DarwinApp\_activate\_root\_returns()* *test\_DarwinCollector\_add\_window\_calls\_WindowsColle*  
*(tests.unit.test\_darwin.TestDarwinApp method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*

*test\_DarwinApp\_grab\_window\_screen\_returns\_test\_DarwinCollector\_add\_window\_inits\_WindowModel()*  
*(tests.unit.test\_darwin.TestDarwinApp method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*

*test\_DarwinApp\_move\_and\_resize\_returns()* *test\_DarwinCollector\_check\_window\_calls()*  
*(tests.unit.test\_darwin.TestDarwinApp method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*

*test\_DarwinApp\_move\_calls\_and\_returns\_moves\_and\_resize\_Collector\_check\_window\_returns\_False\_for*  
*(tests.unit.test\_darwin.TestDarwinApp method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*

*test\_DarwinApp\_move\_to\_workspace\_returns\_test\_DarwinCollector\_check\_window\_returns\_False\_for*  
*(tests.unit.test\_darwin.TestDarwinApp method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 92*

*test\_DarwinCollector\_\_get\_application\_ictestDarwinConnectAppsheds(window\_returns\_True())*  
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 92*

*test\_DarwinCollector\_\_get\_application\_ictestDarwinCgOpen()get\_application\_name\_calls\_val*  
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 92*

*test\_DarwinCollector\_\_get\_application\_ictestDarwinCByTestOf()get\_available\_workspaces\_retu*  
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 92*

*test\_DarwinCollector\_\_get\_application\_ictestDarwinCImage()or\_get\_monitors\_rects\_calls\_NSScr*  
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 92*

*test\_DarwinCollector\_\_get\_window\_geometrytestDarwinCForKeyordememritors\_rects\_returns\_li*  
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 92*

*test\_DarwinCollector\_\_get\_window\_geometrytestDarwinCForKeyordgetbwindows\$ calls\_CGWindowList*  
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 92*

*test\_DarwinCollector\_\_get\_window\_geometrytestDarwinCpleeofointget()windows\_returns\_list()*  
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 92*

*test\_DarwinCollector\_\_get\_window\_id\_calls\_testDarwinCForKeyordector\_get\_workspace\_number\_for\_windo*  
*(tests.unit.test\_darwin.TestDarwinCollector method), 91*
*(tests.unit.test\_darwin.TestDarwinCollector method), 92*

*test\_DarwinCollector\_\_get\_window\_title\_calls\_testDarwinCForKeyordor\_is\_applicable\_calls\_\_running\_a*





test\_LinuxApp\_move\_and\_resize\_calls\_WnckWindow\_geometry() (tests.unit.test\_linux.TestLinuxApp method), 93  
 test\_LinuxApp\_move\_and\_resize\_checks\_maximized (tests.unit.test\_linux.TestLinuxCollector (tests.unit.test\_linux.TestLinuxApp method), 93 method), 94  
 test\_LinuxApp\_move\_and\_resize\_not\_calling\_get\_monitors\_rects\_for\_window\_calls\_is\_applicable (tests.unit.test\_linux.TestLinuxApp method), 93 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxApp\_move\_and\_resize\_not\_calling\_get\_wnck\_window\_wid() (tests.unit.test\_linux.TestLinuxApp method), 93 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxApp\_move\_and\_resize\_not\_calling\_minimize (tests.unit.test\_linux.TestLinuxCollector (tests.unit.test\_linux.TestLinuxApp method), 93 method), 94  
 test\_LinuxApp\_move\_and\_resize\_not\_calling\_restore\_down\_window\_calls\_W\_get\_state (tests.unit.test\_linux.TestLinuxApp method), 93 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxApp\_move\_and\_resize\_not\_calling\_unmaximize (tests.unit.test\_linux.TestLinuxCollector (tests.unit.test\_linux.TestLinuxApp method), 93 method), 94  
 test\_LinuxApp\_move\_and\_resize\_not\_calling\_unminimize (tests.unit.test\_linux.TestLinuxCollector (tests.unit.test\_linux.TestLinuxApp method), 93 method), 94  
 test\_LinuxApp\_move\_and\_resize\_not\_calling\_get\_geometry (tests.unit.test\_linux.TestLinuxApp method), 93 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxApp\_move\_and\_resize\_returns\_False() (tests.unit.test\_linux.TestLinuxCollector (tests.unit.test\_linux.TestLinuxApp method), 93 method), 94  
 test\_LinuxApp\_move\_and\_resize\_returns\_True() (tests.unit.test\_linux.TestLinuxCollector (tests.unit.test\_linux.TestLinuxApp method), 93 method), 94  
 test\_LinuxApp\_move\_calls\_move\_and\_resize() (tests.unit.test\_linux.TestLinuxApp method), 93 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxApp\_move\_to\_workspace\_calls\_move\_window (tests.unit.test\_linux.TestLinuxCollector (tests.unit.test\_linux.TestLinuxApp method), 93 method), 94  
 test\_LinuxCollector\_\_check\_mask\_part\_functionality (tests.unit.test\_linux.TestLinuxCollector method), 94 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxCollector\_\_get\_available\_wnck\_workspaces\_calls\_get\_monitors\_rects\_for\_window (tests.unit.test\_linux.TestLinuxCollector method), 94 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxCollector\_\_get\_available\_wnck\_workspaces\_calls\_get\_monitors\_rects\_for\_window (tests.unit.test\_linux.TestLinuxCollector method), 94 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxCollector\_\_get\_available\_wnck\_workspaces\_calls\_get\_monitors\_rects\_for\_window (tests.unit.test\_linux.TestLinuxCollector method), 94 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxCollector\_add\_window\_calls\_get\_image\_from\_pixbuf (tests.unit.test\_linux.TestLinuxCollector method), 94 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxCollector\_add\_window\_calls\_get\_workspace\_number\_of\_existing\_workspace (tests.unit.test\_linux.TestLinuxCollector method), 94 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxCollector\_add\_window\_calls\_is\_resizeable (tests.unit.test\_linux.TestLinuxCollector method), 94 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxCollector\_add\_window\_calls\_is\_restore\_down (tests.unit.test\_linux.TestLinuxCollector method), 94 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxCollector\_add\_window\_calls\_WindowCollection.add (tests.unit.test\_linux.TestLinuxCollector method), 94 (tests.unit.test\_linux.TestLinuxCollector method), 94  
 test\_LinuxCollector\_add\_window\_calls\_WnckWindowCollection.get\_monitors\_rects\_calls\_GDK\_monitors (tests.unit.test\_linux.TestLinuxCollector method), 94 (tests.unit.test\_linux.TestLinuxCollector method), 94

<i>method</i> ), 94	<i>method</i> ), 95
test_LinuxCollector_get_monitors_rects_calls GDKWindowController_get_workspace_window_calls_type_	(tests.unit.test_linux.TestLinuxController
(tests.unit.test_linux.TestLinuxCollector	method), 95
method), 94	
test_LinuxCollector_get_monitors_rects_returns_main_stack_of_apps()	(tests.unit.test_setup.TestSetup
(tests.unit.test_linux.TestLinuxCollector	method),
method), 94	63
test_LinuxCollector_get_window_by_wid_calls WnckWindow_get(component_class_App()	(tests.unit.test_setup.TestSetup
(tests.unit.test_linux.TestLinuxCollector	method), 63
method), 94	test_main_calls_logging_basicConfig()
test_LinuxCollector_get_window_move_resize_mask(tests.unit.test_setup.TestSetup	method), 63
(tests.unit.test_linux.TestLinuxCollector	test_main_initializes_platform_specific_App()
method), 94	(tests.unit.test_setup.TestSetup
	method), 63
test_LinuxCollector_get_windows_calls_Screen_methods(module_CLASSES())	(tests.unit.test_options.TestOptionsModule
(tests.unit.test_linux.TestLinuxCollector	method), 86
method), 94	
test_LinuxCollector_get_wnck_workspace_fbtest_options_module_COLORS_dictionary()	(tests.unit.test_options.TestOptionsModule
(tests.unit.test_linux.TestLinuxCollector	method), 86
method), 94	
test_LinuxCollector_get_wnck_workspace_fbtest_options_module_MESSAGES_number_dictionary()	(tests.unit.test_options.TestOptionsModule
(tests.unit.test_linux.TestLinuxCollector	method), 86
method), 95	
test_LinuxCollector_get_workspace_number_for_windows_calls WnckWorkspace_valid_format_for_a	(tests.unit.test_options.TestOptionsModule
(tests.unit.test_linux.TestLinuxCollector	method), 86
method), 95	
test_LinuxCollector_get_workspace_number_for_windows_calls WnckWorkspace_dictionary()	(tests.unit.test_options.TestOptionsModule
(tests.unit.test_linux.TestLinuxCollector	method), 86
method), 95	
test_LinuxCollector_get_workspace_number_for_options(Dialog_change_setting_calls_controller	(tests.unit.test_options.TestOptionsDialog
(tests.unit.test_linux.TestLinuxCollector	method), 84
method), 95	
test_LinuxCollector_get_workspace_number_for_options(Dialog_number_of_setting_calls_set_timer()	(tests.unit.test_options.TestOptionsDialog
(tests.unit.test_linux.TestLinuxCollector	method), 85
method), 95	
test_LinuxCollector_is_applicable()	test_OptionsDialog_change_setting_changes_message_v
(tests.unit.test_linux.TestLinuxCollector	(tests.unit.test_options.TestOptionsDialog
method), 95	method), 85
test_LinuxCollector_is_resizable()	test_OptionsDialog_change_setting_for_float_calls_c
(tests.unit.test_linux.TestLinuxCollector	(tests.unit.test_options.TestOptionsDialog
method), 95	method), 85
test_LinuxCollector_is_restored()	test_OptionsDialog_change_setting_not_called_upon_s
(tests.unit.test_linux.TestLinuxCollector	(tests.unit.test_options.TestOptionsDialog
method), 95	method), 85
test_LinuxCollector_is_valid_state()	test_OptionsDialog_create_frame_instantiates_ttk_Fr
(tests.unit.test_linux.TestLinuxCollector	(tests.unit.test_options.TestOptionsDialog
method), 95	method), 85
test_LinuxCollector_rect_is_converted_to_explicit_optionsDialog_create_frame_returns_frame()	(tests.unit.test_options.TestOptionsDialog
(tests.unit.test_linux.TestLinuxCollector	method), 85
method), 95	
test_LinuxCollector_run_functionality()	test_OptionsDialog_create_separator_instantiates_tt
(tests.unit.test_linux.TestLinuxCollector	(tests.unit.test_options.TestOptionsDialog
method), 95	method), 85
test_LinuxCollector_run_super()	test_OptionsDialog_create_separator_instantiates_ve
(tests.unit.test_linux.TestLinuxCollector	(tests.unit.test_options.TestOptionsDialog

<i>method</i> ), 85	<i>method</i> ), 85
test_OptionsDialog_create_widget_calls_widget_options_dialog_game(save_default_changes_message (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85
test_OptionsDialog_create_widget_instantiates_options_dialog_widget_show_about_calls_lift() (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85
test_OptionsDialog_create_widget_instantiates_options_dialog_on_show_about_instantiates_About (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85
test_OptionsDialog_create_widget_instantiates_options_dialog_with_keywords_inmer_calls_after() (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85
test_OptionsDialog_init_calls_geometry_on_options_dialog_set_timer_calls_after_cancel_if (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85
test_OptionsDialog_init_calls_set_icon()test_OptionsDialog_set_timer_sets_timer_attribute() (tests.unit.test_options.TestAboutDialog <i>method</i> ), 83	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85
test_OptionsDialog_init_calls_set_icon()test_OptionsDialog_setup_bindings_binds_callback() (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85
test_OptionsDialog_init_calls_setup_bindings_options_dialog_setup_files_section_calls_button (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85
test_OptionsDialog_init_calls_setup_widget_test()OptionsDialog_setup_files_section_inits_Label (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85
test_OptionsDialog_init_calls_super_with_master_options_dialog_setup_files_section_returns_Label (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85
test_OptionsDialog_init_sets_master_attributeOptionsDialog_setup_files_section_sets_save_de (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85
test_OptionsDialog_init_sets_options_dialog_get_options_dialog_setup_section_calls_create_frame (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 86
test_OptionsDialog_inits_attributes() test_OptionsDialog_setup_section_calls_create_separ (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 86
test_OptionsDialog_issubclass_of_Topleveltest_OptionsDialog_setup_section_calls_create_widg (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 86
test_OptionsDialog_on_destroy_options_destroy_options_dialog_setup_section_calls_Frame_pack() (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 86
test_OptionsDialog_on_destroy_options_show_options_dialog_setup_section_calls_Separator_pa (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 86
test_OptionsDialog_on_save_default_calls_test_options_base_dialog_setup_section_calls_widget_label (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 86
test_OptionsDialog_on_save_default_calls_test_options_dialog_setup_section_calls_widget_pack (tests.unit.test_options.TestOptionsDialog <i>method</i> ), 85	(tests.unit.test_options.TestOptionsDialog <i>method</i> ), 86

*method*), 86  
 test\_OptionsDialog\_setup\_section\_inits\_LabelFrame(*method*), 86  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_setup\_section\_returns\_section(*method*), 87  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_setup\_widgets\_calls\_about\_button\_pack(*method*), 87  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_setup\_widgets\_calls\_label\_pack(*method*), 87  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_setup\_widgets\_calls\_LabelFrame\_pack(*method*), 87  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_setup\_widgets\_calls\_quit\_button\_pack(*method*), 87  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_setup\_widgets\_calls\_setup\_final\_section(*method*), 87  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_setup\_widgets\_calls\_setup\_section\_for\_appearance(*method*), 87  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_setup\_widgets\_calls\_setup\_section\_for\_colors(*method*), 87  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_setup\_widgets\_sets\_about\_button(*method*), 87  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_setup\_widgets\_sets\_label\_for\_about(*method*), 87  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_setup\_widgets\_sets\_message\_var(*method*), 89  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_setup\_widgets\_sets\_quit\_button(*method*), 89  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_widget\_class\_from\_name\_calls\_setting\_type(*method*), 89  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_widget\_class\_from\_name\_for\_type\_None(*method*), 89  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_OptionsDialog\_widget\_class\_from\_name\_returns\_related\_class(*method*), 90  
     (*tests.unit.test\_options.TestOptionsDialog method*), 86  
 test\_resources\_icon\_file\_exist(*method*), 90  
     (*tests.unit.test\_setup.TestFiles method*), 63  
 test\_resources\_misc\_file\_exist(*method*), 90  
     (*tests.unit.test\_setup.TestFiles method*), 63  
 test\_ScaleOption\_init\_calls\_super\_with\_master\_settings\_is\_setting\_returns\_False\_for\_invalid(*method*), 86  
     (*tests.unit.test\_options.TestScaleOption method*), 86  
     test\_ScaleOption\_init\_configs\_attributes(*method*), 86  
         (*tests.unit.test\_options.TestScaleOption method*), 87  
     test\_ScaleOption\_init\_configs\_command(*method*), 87  
         (*tests.unit.test\_options.TestScaleOption method*), 87  
     test\_ScaleOption\_init\_sets\_change\_callback\_attribute(*method*), 87  
         (*tests.unit.test\_options.TestScaleOption method*), 87  
     test\_ScaleOption\_init\_sets\_initial(*method*), 87  
         (*tests.unit.test\_options.TestScaleOption method*), 87  
     test\_ScaleOption\_init\_sets\_master\_attribute(*method*), 87  
         (*tests.unit.test\_options.TestScaleOption method*), 87  
     test\_ScaleOption\_init\_sets\_name\_attribute(*method*), 87  
         (*tests.unit.test\_options.TestScaleOption method*), 87  
     test\_ScaleOption\_inits\_attributes(*method*), 87  
         (*tests.unit.test\_options.TestScaleOption method*), 87  
     test\_ScaleOption\_issubclass\_of\_Scale(*method*), 87  
         (*tests.unit.test\_options.TestScaleOption method*), 87  
     test\_ScaleOption\_on\_update\_value\_calls\_master\_change\_callback(*method*), 87  
         (*tests.unit.test\_options.TestScaleOption method*), 87  
     test\_ScaleOption\_on\_update\_value\_returns\_break(*method*), 87  
         (*tests.unit.test\_options.TestScaleOption method*), 87  
     test\_Settings\_availability\_for\_all\_constants\_in\_SEF(*method*), 87  
         (*tests.unit.test\_settings.TestSettings method*), 87  
     test\_Settings\_color\_group\_returns\_empty\_list\_for\_name(*method*), 87  
         (*tests.unit.test\_settings.TestSettings method*), 87  
     test\_Settings\_color\_group\_returns\_list(*method*), 87  
         (*tests.unit.test\_settings.TestSettings method*), 87  
     test\_Settings\_color\_group\_returns\_type\_for\_valid\_setting(*method*), 87  
         (*tests.unit.test\_settings.TestSettings method*), 87  
     test\_Settings\_initializes\_blank\_icon(*method*), 87  
         (*tests.unit.test\_settings.TestSettings method*), 87  
     test\_Settings\_initializes\_unchangeable\_core\_program(*method*), 87  
         (*tests.unit.test\_settings.TestSettings method*), 87  
     test\_Settings\_is\_setting\_returns\_False\_for\_core\_setting(*method*), 87  
         (*tests.unit.test\_settings.TestSettings method*), 87  
     test\_Settings\_is\_setting\_returns\_False\_for\_invalid\_setting(*method*), 87  
         (*tests.unit.test\_settings.TestSettings method*), 87

```

        (tests.unit.test_settings.TestSettings method),          (tests.unit.test_settings.TestSettings method),
        90                                                       90
test_Settings_is_setting_returns_False_for_invalid_setting_type(type_returns_type_for_valid(
        (tests.unit.test_settings.TestSettings method),          (tests.unit.test_settings.TestSettings method),
        90                                                       90
test_Settings_is_setting_returns_False_for_valid_setting($setting_type_returns_type_for_valid(
        (tests.unit.test_settings.TestSettings method),          (tests.unit.test_settings.TestSettings method),
        90                                                       90
test_Settings_is_setting_returns_True_for_valid_setting(validate_user_settings_returns_dictio
        (tests.unit.test_settings.TestSettings method),          (tests.unit.test_settings.TestSettingsModule
        90                                                       method), 91
test_Settings_metaclass_is_SettingsMetaclass($settings_validate_user_settings_returns_from_
        (tests.unit.test_settings.TestSettings method),          (tests.unit.test_settings.TestSettingsModule
        90                                                       method), 91
test_settings_module_initializes_MESSAGES_key($settings_validate_user_settings_returns_only_v
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_settings.TestSettingsModule
        method), 90                                               method), 91
test_settings_module_initializes_MESSAGES_key($settings_validate_user_settings_returns_only_v
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_settings.TestSettingsModule
        method), 90                                               method), 91
test_settings_module_initializes_SETTINGS_key(SettingsMetaclass__getattr__calls_validate_u
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_settings.TestSettingsModule
        method), 90                                               method), 90
test_settings_module_SETTINGS_for_value_type(SettingsMetaclass__getattr__not_changing_co
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_settings.TestSettingsModule
        method), 90                                               method), 90
test_settings_module_SETTINGS_has_valid_forma(SettingsMetaclass__getattr__returns_None_for
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_settings.TestSettingsModule
        method), 90                                               method), 90
test_settings_module_SETTINGS_is_dictionary(SettingsMetaclass__getattr__uses_SETTINGS_fo
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_settings.TestSettingsModule
        method), 90                                               method), 90
test_settings_read_user_settings_calls_json(SettingsMetaclass__getattr__uses_user_settin
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_settings.TestSettingsModule
        method), 90                                               method), 90
test_settings_read_user_settings_calls_pickle(SettingsMetaclass__getattr__uses_user_settin
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_settings.TestSettingsModule
        method), 90                                               method), 90
test_settings_read_user_settings_checks_if(SettingsMetaclass__getattr__uses_user_settin
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_settings.TestSettingsModule
        method), 90                                               method), 90
test_settings_read_user_settings_returns_test_theme_option_init_sets_initial_BG_from_Settings
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_options.TestThemeOption
        method), 90                                               method), 87
test_settings_read_user_settings_returns_test_theme_option_init_sets_initial_FG_from_Settings
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_options.TestThemeOption
        method), 90                                               method), 87
test_settings_read_user_settings_returns_test_theme_option_exists(class_of_ColorOption()
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_options.TestThemeOption
        method), 90                                               method), 87
test_settings_read_user_settings_returns_test_data_get_snapping_source_by_ordinal_ordinal
        (tests.unit.test_settings.TestSettingsModule              (tests.unit.test_utils.TestUtils method), 87
        method), 90
test_Settings_setting_type_returns_None_for_invalid(tests.unit.test_utils.TestUtils method), 87

```



`test_utils_quarter_by_smaller()` (*tests.unit.test\_utils.TestUtils method*), 89  
`test_utils_quarter_by_smaller_out_of_range()` (*tests.unit.test\_utils.TestUtils method*), 89  
`test_utils_Rectangle_is_namedtuple_class()` (*tests.unit.test\_utils.TestUtils method*), 87  
`test_utils_set_icon_calls_get_resource_path()` (*tests.unit.test\_utils.TestUtils method*), 89  
`test_utils_set_icon_calls_PhotoImage()` (*tests.unit.test\_utils.TestUtils method*), 89  
`test_utils_set_icon_calls_tk_call()` (*tests.unit.test\_utils.TestUtils method*), 89  
`test_view_CornerWidget_anchor_functionality()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_get_place_parameters()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_hide_corner_hides_frame()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_init_calls_setup_widgets()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_init_sets_attributes()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_inits_attributes()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_issubclass_of_object()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_max_box_functionality()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_max_box_is_property()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_max_xy_functionality()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_max_xy_is_property()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_set_corner_calls_framepack()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_set_corner_calls_get_widget_parameters()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_set_corner_calls_get_widget_parameters_of_Frame()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_setup_widgets_calls_set_corner()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_CornerWidget_setup_widgets_instantiates_widgets()` (*tests.unit.test\_view.TestCornerWidget method*), 76  
`test_view_get_screenshot_widget_calls_label_place()` (*tests.unit.test\_view.TestViewFunctions method*), 79  
`test_view_get_screenshot_widget_initializes_Label()` (*tests.unit.test\_view.TestViewFunctions method*), 79  
`test_view_get_screenshot_widget_returns_label_instance()` (*tests.unit.test\_view.TestViewFunctions method*), 79  
`test_view_get_tkinter_root_calls_set_icon()` (*tests.unit.test\_view.TestViewFunctions method*), 79  
`test_view_get_tkinter_root_initializes_Tk()` (*tests.unit.test\_view.TestViewFunctions method*), 79  
`test_view_get_tkinter_root_returns_Tk_instance()` (*tests.unit.test\_view.TestViewFunctions method*), 79  
`test_view_get_tkinter_root_sets_title()` (*tests.unit.test\_view.TestViewFunctions method*), 79  
`test_view_get_tkinter_root_withdraw_root()` (*tests.unit.test\_view.TestViewFunctions method*), 79  
`test_view_ListedWindow_get_icon_image_calls_ImageTk()` (*tests.unit.test\_view.TestListedWindow method*), 76  
`test_view_ListedWindow_init_calls_get_icon_image()` (*tests.unit.test\_view.TestListedWindow method*), 76  
`test_view_ListedWindow_init_calls_setup_bindings()` (*tests.unit.test\_view.TestListedWindow method*), 76  
`test_view_ListedWindow_init_calls_setup_widgets()` (*tests.unit.test\_view.TestListedWindow method*), 76  
`test_view_ListedWindow_init_calls_super_with_master()` (*tests.unit.test\_view.TestListedWindow method*), 76  
`test_view_ListedWindow_init_sets_attributes()` (*tests.unit.test\_view.TestListedWindow method*), 76  
`test_view_ListedWindow_with_maxabbx_as_empty()` (*tests.unit.test\_view.TestListedWindow method*), 76  
`test_view_ListedWindow_with_hsbbox_of_Frame()` (*tests.unit.test\_view.TestListedWindow method*), 76



test\_view\_ListedWindow\_on\_widget\_enter\_returns\_view(PropertyIcon\_on\_widget\_leave\_configures\_ (tests.unit.test\_view.TestListedWindow (tests.unit.test\_view.TestPropertyIcon method), method), 76 77

test\_view\_ListedWindow\_on\_widget\_enter\_sets\_view(PropertyIcon\_on\_widget\_leave\_returns\_bre (tests.unit.test\_view.TestListedWindow (tests.unit.test\_view.TestPropertyIcon method), method), 76 77

test\_view\_ListedWindow\_on\_widget\_leave\_returns\_view(PropertyIcon\_set\_value\_calls\_config() (tests.unit.test\_view.TestListedWindow (tests.unit.test\_view.TestPropertyIcon method), method), 76 77

test\_view\_ListedWindow\_on\_widget\_leave\_sets\_view(PropertyIcon\_set\_value\_sets\_value\_attrib (tests.unit.test\_view.TestListedWindow (tests.unit.test\_view.TestPropertyIcon method), method), 76 77

test\_view\_ListedWindow\_setup\_bindings\_callbacks(PropertyIcon\_setup\_bindings\_callbacks() (tests.unit.test\_view.TestListedWindow (tests.unit.test\_view.TestPropertyIcon method), method), 77 77

test\_view\_ListedWindow\_setup\_bindings\_labels\_master(PropertyIcon\_setup\_bindings\_labels\_maste (tests.unit.test\_view.TestListedWindow (tests.unit.test\_view.TestPropertyIcon method), method), 77 77

test\_view\_ListedWindow\_setup\_widgets\_calls\_configure(PropertyIcon\_setup\_widgets\_configs\_label (tests.unit.test\_view.TestListedWindow (tests.unit.test\_view.TestPropertyIcon method), method), 77 77

test\_view\_ListedWindow\_setup\_widgets\_calls\_set\_color(PropertyIcon\_setup\_widgets\_sets\_colorize (tests.unit.test\_view.TestListedWindow (tests.unit.test\_view.TestPropertyIcon method), method), 77 77

test\_view\_ListedWindow\_setup\_widgets\_sets\_view(PropertyIcon\_setup\_widgets\_sets\_icon\_imag (tests.unit.test\_view.TestListedWindow (tests.unit.test\_view.TestPropertyIcon method), method), 77 77

test\_view\_ListedWindow\_setup\_widgets\_sets\_view(Resizable\_init\_calls\_super\_with\_master\_ar (tests.unit.test\_view.TestListedWindow (tests.unit.test\_view.TestResizable method), 77 method), 77 test\_view\_Resizable\_inits\_attr\_as\_empty()

test\_view\_PropertyIcon\_init\_calls\_setup\_binding(tests.unit.test\_view.TestResizable method), 77 test\_view\_Resizable\_inits\_image\_name() (tests.unit.test\_view.TestResizable method), 77 77

test\_view\_PropertyIcon\_init\_calls\_setup\_widgets(test\_view\_Resizable\_issubclass\_of\_PropertyIcon() (tests.unit.test\_view.TestPropertyIcon method), (tests.unit.test\_view.TestResizable method), 77 77 test\_view\_Restored\_init\_calls\_super\_with\_master\_and (tests.unit.test\_view.TestPropertyIcon method), (tests.unit.test\_view.TestRestored method), 78 78

test\_view\_PropertyIcon\_init\_calls\_super\_with\_master\_and(tests.unit.test\_view.TestRestored method), 78 test\_view\_Restored\_inits\_attr\_as\_empty() (tests.unit.test\_view.TestRestored method), 78 78

test\_view\_PropertyIcon\_init\_sets\_attributes(test\_view\_Restored\_inits\_image\_name() (tests.unit.test\_view.TestPropertyIcon method), (tests.unit.test\_view.TestRestored method), 78 77 test\_view\_Restored\_issubclass\_of\_PropertyIcon() (tests.unit.test\_view.TestPropertyIcon method), (tests.unit.test\_view.TestRestored method), 78 78

test\_view\_PropertyIcon\_inits\_attr\_as\_empty(test\_view\_Statusbar\_init\_calls\_setup\_widgets() (tests.unit.test\_view.TestPropertyIcon method), (tests.unit.test\_view.TestStatusbar method), 78 77 test\_view\_Statusbar\_init\_calls\_super\_with\_master\_and (tests.unit.test\_view.TestPropertyIcon method), (tests.unit.test\_view.TestStatusbar method), 78 78

test\_view\_PropertyIcon\_issubclass\_of\_Label(test\_view\_Statusbar\_init\_configures\_background() (tests.unit.test\_view.TestPropertyIcon method), (tests.unit.test\_view.TestStatusbar method), 78 77 test\_view\_Statusbar\_init\_configures\_background() (tests.unit.test\_view.TestPropertyIcon method), (tests.unit.test\_view.TestStatusbar method), 78 78

test\_view\_PropertyIcon\_on\_widget\_enter\_configures(tests.unit.test\_view.TestStatusbar method), 78 test\_view\_Statusbar\_init\_sets\_attributes() (tests.unit.test\_view.TestPropertyIcon method), (tests.unit.test\_view.TestStatusbar method), 78 77 test\_view\_Statusbar\_init\_sets\_attributes() (tests.unit.test\_view.TestPropertyIcon method), (tests.unit.test\_view.TestStatusbar method), 78 78

test\_view\_PropertyIcon\_on\_widget\_enter\_returns\_view(test\_view\_Statusbar\_inits\_attributes() (tests.unit.test\_view.TestPropertyIcon method), (tests.unit.test\_view.TestStatusbar method), 78 77 test\_view\_Statusbar\_issubclass\_of\_Frame() (tests.unit.test\_view.TestPropertyIcon method), (tests.unit.test\_view.TestStatusbar method), 78 77







<i>(tests.unit.test_viewapplication.TestViewApplication method), 82</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_ViewApplication_startup_calls_focus_set_window_model_frame_clear_changed_sets_changed_to_empty (tests.unit.test_viewapplication.TestViewApplication method), 82</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_ViewApplication_startup_calls_place_test_view_model_clear_changed_sets_changed_ws_to_1 (tests.unit.test_viewapplication.TestViewApplication method), 82</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_ViewApplication_startup_calls_show_test (WindowModel_h_gets_height_from_rect () (tests.unit.test_viewapplication.TestViewApplication method), 82</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_ViewApplication_update_widgets_calls_test_enable_window_model_image_serialization_calls_setup () (tests.unit.test_viewapplication.TestViewApplication method), 82</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_ViewApplication_update_widgets_calls_test_set_window_model_value (\$attr_as_None_or_empty_tuple (tests.unit.test_viewapplication.TestViewApplication method), 82</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_ViewApplication_update_widgets_calls_test_set_window_model_value (\$_changed_as_empty_tuple () (tests.unit.test_viewapplication.TestViewApplication method), 82</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_ViewApplication_update_widgets_calls_test_remove_window_model_element_if_changed_ws_as_None () (tests.unit.test_viewapplication.TestViewApplication method), 82</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_ViewApplication_update_widgets_set_test_window_model_is_changed_functionality () (tests.unit.test_viewapplication.TestViewApplication method), 82</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_ViewApplication_update_widgets_set_test_window_model_is_ws_changed_functionality () (tests.unit.test_viewapplication.TestViewApplication method), 82</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_ViewApplication_update_widgets_set_test_window_model_set_changed_creates_empty_tuple_for (tests.unit.test_viewapplication.TestViewApplication method), 82</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_WindowModel_changed_h_gets_h_from_changed_window_model_set_changed_creates_empty_tuple_in (tests.unit.test_data.TestWindowModel method), 74</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_WindowModel_changed_h_gets_h_from_rect (test)WindowModel_set_changed_creates_from_rect () (tests.unit.test_data.TestWindowModel method), 74</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_WindowModel_changed_w_gets_w_from_changed_window_model_set_changed_creates_from_rect_element (tests.unit.test_data.TestWindowModel method), 74</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_WindowModel_changed_w_gets_w_from_rect (test)WindowModel_set_changed_creates_from_rect_element (tests.unit.test_data.TestWindowModel method), 74</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_WindowModel_changed_x_gets_x_from_changed_window_model_set_changed_not_changing_same_value (tests.unit.test_data.TestWindowModel method), 74</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_WindowModel_changed_x_gets_x_from_rect (test)WindowModel_set_changed_sets_changed_ws_and_ch (tests.unit.test_data.TestWindowModel method), 74</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_WindowModel_changed_y_gets_y_from_changed_window_model_set_changed_sets_changed_ws_for_pi (tests.unit.test_data.TestWindowModel method), 74</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>
<i>test_WindowModel_changed_y_gets_y_from_rect (test)WindowModel_set_changed_sets_changed_ws_to_No (tests.unit.test_data.TestWindowModel method), 74</i>	<i>(tests.unit.test_data.TestWindowModel method), 74</i>

*(tests.unit.test\_data.TestWindowModel method), 75*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_WindowModel\_setup\_calls\_get\_value\_if\_equal\_value\_for\_helpers\_\_init\_\_calls\_platform\_s*  
*(tests.unit.test\_data.TestWindowModel method), 75*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_WindowModel\_setup\_set\_None\_or\_empty\_test\_windows\_app\_helpers\_\_init\_\_not\_calling\_\_setup*  
*(tests.unit.test\_data.TestWindowModel method), 75*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_WindowModel\_setup\_sets\_attrs\_for\_valued\_test\_windows\_api\_Helpers\_\_setup\_base\_sets\_WinDLL\_dv*  
*(tests.unit.test\_data.TestWindowModel method), 75*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_WindowModel\_setup\_sets\_attrs\_if\_provided\_test\_windows\_api\_Helpers\_\_setup\_base\_sets\_WinDLL\_k*  
*(tests.unit.test\_data.TestWindowModel method), 75*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_WindowModel\_setup\_sets\_None\_for\_valued\_test\_windows\_api\_Helpers\_\_setup\_base\_sets\_WinDLL\_p*  
*(tests.unit.test\_data.TestWindowModel method), 75*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_WindowModel\_w\_gets\_width\_from\_rect (test\_windows\_api\_Helpers\_\_setup\_base\_sets\_WinDLL\_us*  
*(tests.unit.test\_data.TestWindowModel method), 75*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_WindowModel\_ws\_is\_alias\_for\_workspatest\_test\_windows\_api\_Helpers\_\_setup\_base\_sets\_WNDENUMPI*  
*(tests.unit.test\_data.TestWindowModel method), 75*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_WindowModel\_x\_gets\_x\_from\_rect () test\_windows\_api\_Helpers\_\_setup\_common\_\_get\_windows*  
*(tests.unit.test\_data.TestWindowModel method), 75*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_WindowModel\_y\_gets\_y\_from\_rect () test\_windows\_api\_Helpers\_\_setup\_common\_helpers\_\_cl*  
*(tests.unit.test\_data.TestWindowModel method), 75*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_windows\_api\_DWM\_THUMBNAIL\_PROPERTIES\_set\_test\_windows\_api\_Helpers\_\_setup\_common\_helpers\_\_dwr*  
*(tests.unit.test\_windows\_api.TestDWM\_THUMBNAIL\_PROPERTIES.test\_windows\_api.TestWindowsApiHelpersCommon method), 99*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_windows\_api\_DWM\_THUMBNAIL\_PROPERTIES\_set\_test\_windows\_api\_Helpers\_\_setup\_common\_helpers\_\_ent*  
*(tests.unit.test\_windows\_api.TestDWM\_THUMBNAIL\_PROPERTIES.test\_windows\_api.TestWindowsApiHelpersCommon method), 99*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_windows\_api\_DWM\_THUMBNAIL\_PROPERTIES\_set\_test\_windows\_api\_Helpers\_\_setup\_common\_helpers\_\_ent*  
*(tests.unit.test\_windows\_api.TestDWM\_THUMBNAIL\_PROPERTIES.test\_windows\_api.TestWindowsApiHelpersCommon method), 99*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_windows\_api\_EventRegistrationToken\_feed\_test\_windows\_api\_Helpers\_\_setup\_common\_helpers\_\_get*  
*(tests.unit.test\_windows\_vdi.TestWindowsVdiEventRegistrationToken.test\_windows\_api.TestWindowsApiHelpersCommon method), 103*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_windows\_api\_EventRegistrationToken\_test\_test\_windows\_api\_Helpers\_\_setup\_common\_helpers\_\_get*  
*(tests.unit.test\_windows\_vdi.TestWindowsVdiEventRegistrationToken.test\_windows\_api.TestWindowsApiHelpersCommon method), 103*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_windows\_api\_Helpers\_\_init\_\_calls\_test\_setup\_helpers\_\_setup\_common\_helpers\_\_get*  
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_windows\_api\_Helpers\_\_init\_\_calls\_test\_setup\_helpers\_\_setup\_common\_helpers\_\_get*  
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_windows\_api\_Helpers\_\_init\_\_calls\_test\_setup\_helpers\_\_setup\_common\_helpers\_\_get*  
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*
*(tests.unit.test\_windows\_api.TestWindowsApiHelpersCommon method), 101*

*test\_windows\_api\_Helpers\_\_init\_\_calls\_test\_setup\_helpers\_\_setup\_common\_helpers\_\_op*



(tests.unit.test_windows_api.TestWindowsApiCustomFunction( method), 101	(tests.unit.test_windows_vdi.TestWindowsVdiApplicationViewOri method), 103
test_windows_api_platform_supports_virtual_desktops_for_event_registration_token_is_Structure( (tests.unit.test_windows_api.TestWindowsApiCustomFunction( method), 101	(tests.unit.test_windows_vdi.TestWindowsVdiEventRegistrationTo method), 103
test_windows_api_platform_supports_virtual_desktops_for_hstring_is_Structure_subclass( (tests.unit.test_windows_api.TestWindowsApiCustomFunction( method), 101	(tests.unit.test_windows_vdi.TestWindowsVdiHSTRING_ method), 104
test_windows_api_TITLEBARINFO_field_and_type(windows_vdi_IApplicationView_field_and_value( (tests.unit.test_windows_api.TestTITLEBARINFO method), 100	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_api_TITLEBARINFO_inits__field_test(windows_vdi_IApplicationView_is_IInspectable_ (tests.unit.test_windows_api.TestTITLEBARINFO method), 100	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_api_TITLEBARINFO_is_Structure_subclass(windows_vdi_IApplicationView_method_add_Conso (tests.unit.test_windows_api.TestTITLEBARINFO method), 100	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_api_WINDOWINFO_field_and_type(windows_vdi_IApplicationView_method_get_Adjace (tests.unit.test_windows_api.TestWINDOWINFO method), 100	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_api_WINDOWINFO_inits__field_test(windows_vdi_IApplicationView_method_get_Adjace (tests.unit.test_windows_api.TestWINDOWINFO method), 100	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_api_WINDOWINFO_is_Structure_subclass(windows_vdi_IApplicationView_method_get_Id() (tests.unit.test_windows_api.TestWINDOWINFO method), 100	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_utils_module_extract_name_from_window_path(IAppBaseNameView_method_get_IsFull (tests.unit.test_windows.TestWindowsUtils method), 99	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_utils_module_extract_name_from_window_path(IAppBaseNameView_method_get_IsOnLo (tests.unit.test_windows.TestWindowsUtils method), 99	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_utils_module_extract_name_from_window_path(IApplicationView_method_get_IsScre (tests.unit.test_windows.TestWindowsUtils method), 99	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_utils_module_extract_name_from_window_path(IApplicationView_method_get_Orient (tests.unit.test_windows.TestWindowsUtils method), 99	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_utils_module_extract_name_from_window_path(IApplicationView_method_get_Title (tests.unit.test_windows.TestWindowsUtils method), 99	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_utils_module_user_data_path(windows_vdi_IApplicationView_method_put_IsScre (tests.unit.test_windows.TestWindowsUtils method), 99	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_vdi_AdjacentDesktop_field_and_type(windows_vdi_IApplicationView_method_put_Title (tests.unit.test_windows_vdi.TestWindowsVdiAdjacentDesktop method), 103	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_vdi_AdjacentDesktop_is_INT_subclass(windows_vdi_IApplicationView_method_remove_Cor (tests.unit.test_windows_vdi.TestWindowsVdiAdjacentDesktop method), 103	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationView method), 104
test_windows_vdi_ApplicationViewOrientation_test(windows_vdi_IApplicationViewCollection_field_ (tests.unit.test_windows_vdi.TestWindowsVdiApplicationViewOrientation method), 103	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationViewCo method), 104
test_windows_vdi_ApplicationViewOrientation_test(windows_vdi_IApplicationViewCollection_is_IUnl (tests.unit.test_windows_vdi.TestWindowsVdiApplicationViewCollection method), 103	(tests.unit.test_windows_vdi.TestWindowsVdiIApplicationViewCo method), 104











<i>(tests.unit.test_data.TestWindowsCollection method), 75</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollection_get_model_by_wid_val</i>	<i>test_WindowsCollector__get_image_from_icon_handle_</i>
<i>(tests.unit.test_data.TestWindowsCollection method), 75</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollection_get_windows_calls_get</i>	<i>test_WindowsCollector__get_image_from_icon_handle_</i>
<i>(tests.unit.test_data.TestWindowsCollection method), 75</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollection_get_windows_list_restr</i>	<i>test_WindowsCollector__get_image_from_icon_handle_</i>
<i>(tests.unit.test_data.TestWindowsCollection method), 75</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 96</i>
<i>test_WindowsCollection_initialization_setes</i>	<i>test_WindowsCollector__get_image_from_icon_handle_</i>
<i>(tests.unit.test_data.TestWindowsCollection method), 75</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollection_inits_____memberst</i>	<i>test_WindowsCollector__get_uwpapp_icon_calls_get_p</i>
<i>(tests.unit.test_data.TestWindowsCollection method), 75</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollection_repopulate_for_widest</i>	<i>test_WindowsCollector__get_uwpapp_icon_returns_ico</i>
<i>(tests.unit.test_data.TestWindowsCollection method), 75</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollection_size_is_property</i>	<i>test_WindowsCollector__get_uwpapp_icon_sets_api_pa</i>
<i>(tests.unit.test_data.TestWindowsCollection method), 75</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollection_size_returns__memb</i>	<i>test_WindowsCollector__get_window_geometry_calls_ex</i>
<i>(tests.unit.test_data.TestWindowsCollection method), 75</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollection_sort_functionality</i>	<i>test_WindowsCollector__get_window_geometry_calls_G</i>
<i>(tests.unit.test_data.TestWindowsCollection method), 75</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollector__get_application_ite</i>	<i>test_WindowsCollector__get_image_from_icon_handle_</i>
<i>(tests.unit.test_windows.TestWindowsCollector method), 96</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollector__get_application_ite</i>	<i>test_WindowsCollector__get_uwpapp_icon</i>
<i>(tests.unit.test_windows.TestWindowsCollector method), 96</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollector__get_application_ite</i>	<i>test_WindowsCollector__get_window_geometry_returns_</i>
<i>(tests.unit.test_windows.TestWindowsCollector method), 96</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollector__get_application_ite</i>	<i>test_WindowsCollector__get_uwpapp_icon</i>
<i>(tests.unit.test_windows.TestWindowsCollector method), 96</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollector__get_image_from_ict</i>	<i>test_WindowsCollector__get_image_from_ict</i>
<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollector__get_image_from_ict</i>	<i>test_WindowsCollector__get_image_from_ict</i>
<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollector__get_image_from_ict</i>	<i>test_WindowsCollector__get_image_from_ict</i>
<i>(tests.unit.test_windows.TestWindowsCollector method), 96</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollector__get_image_from_ict</i>	<i>test_WindowsCollector__get_image_from_ict</i>
<i>(tests.unit.test_windows.TestWindowsCollector method), 96</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollector__get_image_from_ict</i>	<i>test_WindowsCollector__get_image_from_ict</i>
<i>(tests.unit.test_windows.TestWindowsCollector method), 96</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>
<i>test_WindowsCollector__get_image_from_ict</i>	<i>test_WindowsCollector__get_image_from_ict</i>
<i>(tests.unit.test_windows.TestWindowsCollector method), 96</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>

<i>(tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector__is_alt_tab_applicable__is_windows_collector_get_by_application_name_function (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector__is_alt_tab_applicable__is_windows_collector_get_by_application_name_returns (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector__is_alt_tab_applicable__is_windows_collector_get_by_application_name_returns (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector__is_alt_tab_applicable__is_windows_collector_get_available_calls_and_return (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector__is_alt_tab_applicable__is_windows_collector_get_monitors_rects_calls_Enum (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector__is_cloaked_calls_and_returns_collector_get_monitors_rects_returns_list (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector__is_tool_window_calls_test_WindowsCollector_get_windows_calls_api_enum_w (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector__is_tool_window_returns_test_WindowsCollector_get_workspace_number_for_win (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector__is_tray_window_calls_test_WindowsCollector_is_applicable_calls () (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector__is_tray_window_returns_test_WindowsCollector_is_applicable_returns_False_ (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector_add_window_calls_method_test_WindowsCollector_is_applicable_returns_False_ (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector_add_window_calls_WindowsCollector__add__is_applicable_returns_False_ (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector_add_window_inits_WindowsCollector__add__is_applicable_returns_False_ (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector_check_window_calls_test_WindowsCollector_is_applicable_returns_False_ (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector_check_window_function_test_WindowsCollector_is_applicable_returns_False_ (tests.unit.test_windows.TestWindowsCollector method), 97</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector_get_application_name_test_WindowsCollector_is_applicable_returns_True () (tests.unit.test_windows.TestWindowsCollector method), 98</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector_get_application_name_test_WindowsCollector__is_applicable_calls () (tests.unit.test_windows.TestWindowsCollector method), 98</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>
<i>test_WindowsCollector_get_application_name_test_WindowsCollector__is_resizable_return () (tests.unit.test_windows.TestWindowsCollector method), 98</i>	<i>(tests.unit.test_windows.TestWindowsCollector method), 98</i>

(tests.unit.test\_windows.TestWindowsCollector method), 98  
 test\_WindowsCollector\_is\_restored\_calls() (tests.unit.test\_windows.TestWindowsCollector method), 98  
 test\_WindowsCollector\_is\_restored\_return() (tests.unit.test\_windows.TestWindowsCollector method), 98  
 test\_WindowsCollector\_is\_valid\_state\_calls\_\_is\_activable() (tests.unit.test\_windows.TestWindowsCollector method), 98  
 test\_WindowsCollector\_is\_valid\_state\_calls\_\_is\_unpacked() (tests.unit.test\_windows.TestWindowsCollector method), 98  
 test\_WindowsCollector\_is\_valid\_state\_return\_value() (tests.unit.test\_windows.TestWindowsCollector method), 98  
 test\_WindowsCollector\_is\_valid\_state\_returns\_cloaked() (tests.unit.test\_windows.TestWindowsCollector method), 98  
 test\_WindowsCollector\_run\_functionality() (tests.unit.test\_windows.TestWindowsCollector method), 98  
 test\_WindowsController\_inits\_screenshot\_when\_package\_has\_inversion() (tests.unit.test\_windows.TestWindowsController method), 98  
 test\_WindowsController\_setup\_root\_window\_calls\_\_is\_cloaked() (tests.unit.test\_windows.TestWindowsController method), 98  
 test\_WindowsController\_setup\_root\_window\_calls\_\_is\_property\_id() (tests.unit.test\_windows.TestWindowsController method), 99  
 TestAboutDialog (class in tests.unit.test\_options), 83  
 TestBaseApp (class in tests.unit.test\_base), 63  
 TestBaseCollector (class in tests.unit.test\_base), 65  
 TestBaseController (class in tests.unit.test\_basecontroller), 69  
 TestBaseControllerDomainLogic (class in tests.unit.test\_basecontroller\_domain\_logic), 66  
 TestBaseMouse (class in tests.unit.test\_base), 66  
 TestCheckOption (class in tests.unit.test\_options), 83  
 TestColorOption (class in tests.unit.test\_options), 84  
 TestCornerWidget (class in tests.unit.test\_view), 76  
 TestDarwinApp (class in tests.unit.test\_darwin), 91  
 TestDarwinCollector (class in tests.unit.test\_darwin), 91  
 TestDarwinUtils (class in tests.unit.test\_darwin), 92  
 TestDummyVirtualDesktops (class in tests.unit.test\_windows\_api), 99  
 TestDWM\_THUMBNAIL\_PROPERTIES (class in tests.unit.test\_windows\_api), 99  
 TestFiles (class in tests.unit.test\_setup), 63  
 TestFloatScaleOption (class in tests.unit.test\_options), 84  
 TestLinuxApp (class in tests.unit.test\_linux), 92  
 TestLinuxCollector (class in tests.unit.test\_linux), 93  
 TestLinuxController (class in tests.unit.test\_linux), 95  
 TestLinuxUnpacked (class in tests.unit.test\_linux), 95  
 TestListedWindow (class in tests.unit.test\_view), 76  
 TestOptionsDialog (class in tests.unit.test\_options), 84  
 TestOptionsModule (class in tests.unit.test\_options), 86  
 TestPACKAGE\_FOR\_Dcloaked() (class in tests.unit.test\_windows\_api), 99  
 TestPACKAGE\_INFO (class in tests.unit.test\_windows\_api), 99  
 TestPACKAGE\_INFO\_REFERENCE (class in tests.unit.test\_windows\_api), 100  
 TestPACKAGE\_INVERSION (class in tests.unit.test\_windows\_api), 100  
 TestPACKAGE\_VERSION (class in tests.unit.test\_windows\_api), 100  
 TestPACKAGE\_VERSION\_U (class in tests.unit.test\_windows\_api), 100  
 TestPropertyId (class in tests.unit.test\_view), 77  
 TestResizable (class in tests.unit.test\_view), 77  
 TestRestored (class in tests.unit.test\_view), 78  
 tests (module), 63  
 tests.functional (module), 112  
 tests.unit (module), 63  
 tests.unit.fixtures (module), 112  
 tests.unit.mock\_helpers (module), 112  
 tests.unit.nested\_helper (module), 112  
 tests.unit.test\_base (module), 63  
 tests.unit.test\_basecontroller (module), 69  
 tests.unit.test\_basecontroller\_domain\_logic (module), 66  
 tests.unit.test\_darwin (module), 91  
 tests.unit.test\_data (module), 74  
 tests.unit.test\_linux (module), 92  
 tests.unit.test\_options (module), 83  
 tests.unit.test\_settings (module), 89  
 tests.unit.test\_setup (module), 63  
 tests.unit.test\_utils (module), 87  
 tests.unit.test\_view (module), 76  
 tests.unit.test\_viewapplication (module), 80  
 tests.unit.test\_windows (module), 95

tests.unit.test\_windows\_api (module), 99  
 tests.unit.test\_windows\_api\_api (module), 108  
 tests.unit.test\_windows\_vdi (module), 103  
 tests.vm (module), 112  
 TestScaleOption (class in tests.unit.test\_options), 86  
 TestSettings (class in tests.unit.test\_settings), 89  
 TestSettingsModule (class in tests.unit.test\_settings), 90  
 TestSetup (class in tests.unit.test\_setup), 63  
 TestStatusBar (class in tests.unit.test\_view), 78  
 TestStructure (class in tests.unit.test\_setup), 63  
 TestThemeOption (class in tests.unit.test\_options), 87  
 TestTITLEBARINFO (class in tests.unit.test\_windows\_api), 100  
 TestToolBar (class in tests.unit.test\_view), 78  
 TestUtils (class in tests.unit.test\_utils), 87  
 TestViewApplication (class in tests.unit.test\_viewapplication), 80  
 TestViewFunctions (class in tests.unit.test\_view), 78  
 TestWINDOWINFO (class in tests.unit.test\_windows\_api), 100  
 TestWindowModel (class in tests.unit.test\_data), 74  
 TestWindowsApiApiPrivate (class in tests.unit.test\_windows\_api\_api), 108  
 TestWindowsApiApiPrivateWin8 (class in tests.unit.test\_windows\_api\_api), 108  
 TestWindowsApiApiPublic (class in tests.unit.test\_windows\_api\_api), 109  
 TestWindowsApiApiPublicWin8 (class in tests.unit.test\_windows\_api\_api), 111  
 TestWindowsApiCustomFunctions (class in tests.unit.test\_windows\_api), 101  
 TestWindowsApiHelpersCommon (class in tests.unit.test\_windows\_api), 101  
 TestWindowsApiHelpersWin8 (class in tests.unit.test\_windows\_api), 102  
 TestWindowsApiPackage (class in tests.unit.test\_windows\_api), 102  
 TestWindowsApp (class in tests.unit.test\_windows), 95  
 TestWindowsCollection (class in tests.unit.test\_data), 75  
 TestWindowsCollector (class in tests.unit.test\_windows), 96  
 TestWindowsController (class in tests.unit.test\_windows), 98  
 TestWindowsList (class in tests.unit.test\_view), 79  
 TestWindowsUtils (class in tests.unit.test\_windows), 99  
 TestWindowsVdiAdjacentDesktop (class in tests.unit.test\_windows\_vdi), 103  
 TestWindowsVdiApplicationViewOrientation (class in tests.unit.test\_windows\_vdi), 103  
 TestWindowsVdiEventRegistrationToken (class in tests.unit.test\_windows\_vdi), 103  
 TestWindowsVdiHSTRING\_\_ (class in tests.unit.test\_windows\_vdi), 103  
 TestWindowsVdiIApplicationView (class in tests.unit.test\_windows\_vdi), 104  
 TestWindowsVdiIApplicationViewCollection (class in tests.unit.test\_windows\_vdi), 104  
 TestWindowsVdiIApplicationViewConsolidatedEventArgs (class in tests.unit.test\_windows\_vdi), 104  
 TestWindowsVdiIInspectable (class in tests.unit.test\_windows\_vdi), 105  
 TestWindowsVdiIObjectArray (class in tests.unit.test\_windows\_vdi), 105  
 TestWindowsVdiIServiceProvider (class in tests.unit.test\_windows\_vdi), 105  
 TestWindowsVdiIVirtualDesktop (class in tests.unit.test\_windows\_vdi), 105  
 TestWindowsVdiIVirtualDesktopManager (class in tests.unit.test\_windows\_vdi), 105  
 TestWindowsVdiIVirtualDesktopManagerInternal (class in tests.unit.test\_windows\_vdi), 106  
 TestWindowsVdiModuleUids (class in tests.unit.test\_windows\_vdi), 106  
 TestWindowsVdiTrustLevel (class in tests.unit.test\_windows\_vdi), 106  
 TestWindowsVdiVirtualDesktopsWin10 (class in tests.unit.test\_windows\_vdi), 106  
 TestWorkspace (class in tests.unit.test\_view), 79  
 TestWorkspacesCollection (class in tests.unit.test\_view), 80  
 ThemeOption (class in arrangeit.options), 43  
 thumbnails (arrangeit.windows.app.App attribute), 54  
 timer (arrangeit.base.BaseController attribute), 29  
 timer (arrangeit.options.OptionsDialog attribute), 43  
 title (arrangeit.data.WindowModel attribute), 32  
 title (arrangeit.view.ListedWindow attribute), 35  
 title\_info\_state() (arrangeit.windows.api.Api method), 57  
 Toolbar (class in arrangeit.view), 37

## U

unregister\_thumbnail() (arrangeit.windows.api.Api method), 57  
 update() (arrangeit.base.BaseController method), 29  
 update\_positioning() (arrangeit.base.BaseController method), 29  
 update\_resizing() (arrangeit.base.BaseController method), 30



update\_widgets() (*arrangeit.view.ViewApplication*  
*method*), 38  
 user\_data\_path() (*in module arrangeit.darwin.utils*), 50  
 user\_data\_path() (*in module arrangeit.linux.utils*),  
 53  
 user\_data\_path() (*in module arrangeit.windows.utils*), 62  
 user\_settings (*arrangeit.settings.Settings*  
*attribute*), 47

## V

validate\_user\_settings() (*in module arrangeit.settings*), 47  
 value (*arrangeit.view.PropertyIcon* attribute), 36  
 var (*arrangeit.options.CheckOption* attribute), 41  
 var (*arrangeit.options.ColorOption* attribute), 41  
 vdi (*arrangeit.windows.api.Api* attribute), 57  
 view (*arrangeit.base.BaseController* attribute), 30  
 view\_collection (*arrangeit.windows.vdi.VirtualDesktopsWin10*  
*attribute*), 62  
 ViewApplication (*class in arrangeit.view*), 37  
 VirtualDesktopsWin10 (*class in arrangeit.windows.vdi*), 61

## W

w (*arrangeit.data.WindowModel* attribute), 32  
 wid (*arrangeit.data.WindowModel* attribute), 32  
 wid (*arrangeit.view.ListedWindow* attribute), 35  
 widget\_class\_from\_name() (*arrangeit.options.OptionsDialog*  
*method*), 43  
 width (*arrangeit.view.CornerWidget* attribute), 34  
 window\_info\_extended\_style() (*arrangeit.windows.api.Api*  
*method*), 57  
 WINDOW\_MODEL\_RECT\_ELEMENTS (*arrangeit.settings.Settings*  
*attribute*), 47  
 WINDOW\_MODEL\_TYPES (*arrangeit.settings.Settings*  
*attribute*), 47  
 WindowModel (*class in arrangeit.data*), 31  
 WindowsCollection (*class in arrangeit.data*), 32  
 WindowsList (*class in arrangeit.view*), 38  
 workspace (*arrangeit.data.WindowModel* attribute),  
 32  
 Workspace (*class in arrangeit.view*), 39  
 workspace\_activated() (*arrangeit.base.BaseController*  
*method*), 30  
 workspace\_activated\_by\_digit() (*arrangeit.base.BaseController*  
*method*), 30  
 WorkspacesCollection (*class in arrangeit.view*),  
 39  
 ws (*arrangeit.data.WindowModel* attribute), 32

## X

x (*arrangeit.data.WindowModel* attribute), 32  
 x0 (*arrangeit.utils.Rectangle* attribute), 43  
 x1 (*arrangeit.utils.Rectangle* attribute), 43

## Y

y (*arrangeit.data.WindowModel* attribute), 32  
 y0 (*arrangeit.utils.Rectangle* attribute), 43  
 y1 (*arrangeit.utils.Rectangle* attribute), 43