

---

# shortcut Documentation

**Martin O'Hanlon**

**Apr 02, 2022**



---

## Contents

---

<b>1</b>	<b>Documentation</b>	<b>3</b>
<b>2</b>	<b>Install</b>	<b>5</b>
<b>3</b>	<b>Status</b>	<b>7</b>
<b>4</b>	<b>Table of Contents</b>	<b>9</b>
4.1	shortcut app . . . . .	9
4.2	API . . . . .	10
4.3	Operating Systems . . . . .	11
4.4	Build . . . . .	11
4.5	Change log . . . . .	13
	<b>Python Module Index</b>	<b>15</b>
	<b>Index</b>	<b>17</b>



Shortcut is a cross platform command line application and API for creating shortcuts.

`shortcut` will do its best to find your app, searching for the usual suspects in the usual places (i.e. those in the `PATH`), or you can give it a full path.

To create desktop and menu shortcuts for `python`:

- Using the app:

```
shortcut python
```

- Using the Python API:

```
from shortcut import ShortCutter
s = ShortCutter()
s.create_desktop_shortcut("python")
s.create_menu_shortcut("python")
```

It was created to solve a simple problem - if you install a python package using `pip` there is no simple way of creating a shortcut to the program it installs.



# CHAPTER 1

---

## Documentation

---

There is comprehensive documentation at [shortcut.readthedocs.io](https://shortcut.readthedocs.io).





Shortcut is available on [pypi](#) and can be installed using pip:

- Windows

```
pip install shortcut
```

- MacOS

```
pip3 install shortcut
```

- Linux

```
sudo pip3 install shortcut
```



## CHAPTER 3

---

### Status

---

Alpha - tested and works but [issues](#) maybe experienced and API changes are possible.

It should work with Windows, MacOS and Linux operating systems.



### 4.1 shortcut app

The command line application `shortcut` creates desktop and menu shortcuts.

`shortcut` will search for the app or use a full path if given.

#### 4.1.1 Usage

To create desktop and menu shortcuts for the *python* application:

```
shortcut python
```

#### 4.1.2 Options

The `-h` or `--help` option will display the help:

```
shortcut --help
```

```
usage: shortcut [-h] [-d] [-m] target

Auto shortcut creator

positional arguments:
target                The target executable to create Desktop and Menu shortcuts

optional arguments:
-h, --help            show this help message and exit
-d, --desktop         Only create a desktop shortcut
-m, --menu            Only create a menu shortcut
```

To only create a desktop shortcut use `--desktop`:

```
shortcut --desktop python
```

Or just a menu shortcut using `--menu`:

```
shortcut --menu python
```

## 4.2 API

### 4.2.1 ShortCutter

**class** `shortcut.base.ShortCutter`

Creates application shortcuts for Windows, MacOS and Linux operating systems.

To create desktop and menu shortcuts to *python*:

```
from shortcut import ShortCutter
s = ShortCutter()
s.create_desktop_shortcut("python")
s.create_menu_shortcut("python")
```

**create\_desktop\_shortcut** (*target*)

Creates a desktop shortcut to a target.

**Parameters** **target** (*str*) – The target to create a shortcut for, it can be a fully qualified file path */path/to/my\_program* or a simple application name *my\_program*.

Returns a tuple of (target\_name, target\_path, shortcut\_file\_path)

**create\_menu\_shortcut** (*target*)

Creates a menu shortcut to a target.

**Parameters** **target** (*str*) – The target to create a shortcut for, it can be a fully qualified file path */path/to/my\_program* or a simple application name *my\_program*.

Returns a tuple of (target\_name, target\_path, shortcut\_file\_path)

**create\_shortcut** (*target, shortcut\_directory*)

Creates a shortcut to a target.

**Parameters**

- **target** (*str*) – The target to create a shortcut for, it can be a fully qualified file path */path/to/my\_program* or a simple application name *my\_program*.
- **shortcut\_directory** (*str*) – The directory path where the shortcut should be created.

Returns a tuple of (target\_name, target\_path, shortcut\_file\_path)

**desktop\_directory**

Sets or returns the directory used when creating desktop shortcuts.

**find\_target** (*target*)

Finds a file path for a target application.

**Parameters** **target** (*str*) – The target to find, it can be a fully qualified file path */path/to/my\_program* or a simple application name *my\_program*.

Returns a single target file path or `None` if a path can't be found.

**menu\_directory**

Sets or returns the directory used when creating menu shortcuts.

**search\_for\_target** (*target*)

Searches for a target application.

**Parameters** **target** (*str*) – The target to find.

Returns a list of potential target file paths, if no paths are found an empty list is returned.

## 4.3 Operating Systems

Shortcut support Windows, MacOS and Linux.

The way shortcuts are provided and applications launched depends on the operating system.

### 4.3.1 Windows

.lnk files pointing directly to the application paths are created in the User Desktop (CSIDL\_DESKTOPDIRECTORY) and Program (CSIDL\_PROGRAMS) folders.

### 4.3.2 MacOS

MacOS applications are created which run the application via a terminal and copied to the User Desktop (~/Desktop) and Launchpad (/Applications).

### 4.3.3 Linux

.desktop files are created which start the application via the shell and created in the User Desktop (~/Desktop) and applications menu (~/.local/share/applications).

## 4.4 Build

Instructions for developing and building shortcut.

### 4.4.1 Windows

#### Develop

Installing tools:

```
pip install pip --upgrade
pip install setuptools wheel twine
```

Get repo and install for dev:

```
git clone https://github.com/martinohanlon/shortcut
cd shortcut
git checkout dev
python setup.py develop
```

### Deploy

Create `.pypirc` credentials file:

```
notepad %HOMEPATH%\pypirc

[distutils]
index-servers =
    pypi

[pypi]
username:
password:
```

Build for deployment:

```
python setup.py sdist
python setup.py bdist_wheel
```

Deploy to [PyPI](#):

```
twine upload dist/* --skip-existing
```

## 4.4.2 Linux

### Develop

Installing tools:

```
sudo pip3 install pip --upgrade
sudo pip3 install setuptools wheel twine
```

Get repo and install for dev:

```
git clone https://github.com/martinohanlon/shortcut
cd shortcut
git checkout dev
sudo python3 setup.py develop
```

### Deploy

Create `.pypirc` credentials file:

```
nano ~/.pypirc

[distutils]
index-servers =
    pypi

[pypi]
username:
password:
```

Build for deployment:



```
python3 setup.py sdist
python3 setup.py bdist_wheel
```

Deploy to [PyPI](#):

```
twine upload dist/* --skip-existing
```

## 4.5 Change log

### 4.5.1 0.0.2 - 2018-02-25

- added `desktop_directory` and `menu_directory` properties
- documentation updates

### 4.5.2 0.0.1 - 2018-02-25

- initial alpha release
- support for Windows, MacOS & Linux
- `shortcut app` and `ShortCutter API`



**S**

`shortcut.base`, 10



## C

`create_desktop_shortcut()` (*shortcut.base.ShortCutter* method), 10  
`create_menu_shortcut()` (*shortcut.base.ShortCutter* method), 10  
`create_shortcut()` (*shortcut.base.ShortCutter* method), 10

## D

`desktop_directory` (*shortcut.base.ShortCutter* attribute), 10

## F

`find_target()` (*shortcut.base.ShortCutter* method), 10

## M

`menu_directory` (*shortcut.base.ShortCutter* attribute), 10

## S

`search_for_target()` (*shortcut.base.ShortCutter* method), 11  
`shortcut.base` (module), 10  
`ShortCutter` (class in *shortcut.base*), 10