
Python MSS

Release latest

Mickaël Schoentgen

Apr 23, 2026

CONTENTS

1	Installation	2
2	Usage	2
3	Examples	2
4	Support	2
5	MSS API	2
6	Developers	2
7	Versioning	2
8	History	2
9	Who Uses it?	2
10	Indices and tables	3
	Python Module Index	5
	Index	7


```
from mss import MSS

# The simplest use, save a screenshot of the 1st monitor
with MSS() as sct:
    sct.shot()
```

An ultra fast cross-platform multiple screenshots module in pure python using ctypes.

- **Python 3.9+**, **PEP 8** compliant, no dependency, thread-safe;
- very basic, it will grab one screenshot by monitor or a screenshot of all monitors and save it to a PNG file;
- but you can use PIL and benefit from all its formats (or add yours directly);
- integrate well with Numpy and OpenCV;
- it could be easily embedded into games and other software which require fast and platform optimized methods to grab screenshots (like AI, Computer Vision);
- get the [source code on GitHub](#);
- learn with a [bunch of examples](#);
- you can [report a bug](#);
- need some help? Use the tag *python-mss* on [Stack Overflow](#);
- **MSS** stands for Multiple ScreenShots;

Content

CHAPTER
ONE

INSTALLATION

1.1 Recommended Way

Quite simple:

```
$ python -m pip install -U --user mss
```

1.1.1 Conda Package

The module is also available from Conda:

```
$ conda install -c conda-forge python-mss
```

1.2 From Sources

Alternatively, you can get a copy of the module from GitHub:

```
$ git clone https://github.com/BoBoTiG/python-mss.git  
$ cd python-mss
```

And then:

```
$ python setup.py install --user
```

CHAPTER
TWO

USAGE

2.1 Import

MSS can be used simply as:

```
from mss import MSS  
  
with MSS() as sct:  
    # ...
```

For compatibility with existing code, `mss.mss()` is still available in 10.2, but deprecated:

```
import mss
```

```
2with mss.mss() as sct: # Deprecated in 10.2  
    # ...
```

CONTENTS

For compatibility with existing code, platform-specific class names are also still available in 10.2:

```
# GNU/Linux
```

INDICES AND TABLES

- genindex
- search

PYTHON MODULE INDEX

m

mss, 2

mss.models, 2

Symbols

\$DISPLAY, 2

B

bgra (*mss.ScreenShot* property), 2

C

close() (*mss.MSS* method), 2

compression_level (*mss.MSS* attribute), 2

D

details (*mss.ScreenShotError* attribute), 2

DISPLAY, 2

E

environment variable

 \$DISPLAY, 2

 DISPLAY, 2

F

from_size() (*mss.ScreenShot* class method), 2

G

grab() (*mss.MSS* method), 2

H

height (*mss.models.Size* attribute), 2

height (*mss.ScreenShot* property), 2

L

left (*mss.models.Pos* attribute), 2

left (*mss.ScreenShot* property), 2

M

max_displays (*mss.MSS* property), 2

module

 mss, 2

 mss.models, 2

monitors (*mss.MSS* property), 2

mss

 module, 2

MSS (*class in mss*), 2

mss() (*in module mss*), 2

mss.models

 module, 2

P

performance_status (*mss.MSS* property), 2

pixel() (*mss.ScreenShot* method), 2

pixels (*mss.ScreenShot* property), 2

Pos (*class in mss.models*), 2

pos (*mss.ScreenShot* attribute), 2

primary_monitor (*mss.MSS* property), 2

Python Enhancement Proposals

 PEP 8, 1

R

raw (*mss.ScreenShot* attribute), 2

rgb (*mss.ScreenShot* property), 2

S

save() (*mss.MSS* method), 2

ScreenShot (*class in mss*), 2

ScreenShotError, 2

shot() (*mss.MSS* method), 2

Size (*class in mss.models*), 2

size (*mss.ScreenShot* attribute), 2

T

top (*mss.models.Pos* attribute), 2

top (*mss.ScreenShot* property), 2

W

width (*mss.models.Size* attribute), 2

width (*mss.ScreenShot* property), 2

with_cursor (*mss.MSS* property), 2