

---

# **omxplayer-wrapper Documentation**

*Release 0.3.2*

**Will Price**

**Dec 11, 2018**



---

## Contents

---

<b>1 Installation:</b>	<b>3</b>
<b>2 Examples:</b>	<b>5</b>
<b>3 Learning more:</b>	<b>7</b>
3.1 omxplayer API Docs .....	7
<b>Python Module Index</b>	<b>17</b>



omxplayer-wrapper is a project to control [OMXPlayer](#) from python over [dbus](#).



# CHAPTER 1

---

## Installation:

---

You'll need the following dependencies:

- *libdbus-1*
- *libdbus-1-dev*

OS pre-requisite installation

```
$ sudo apt-get update && sudo apt-get install -y libdbus-1{,-dev}
```

With `pipenv`

```
$ pipenv install omxplayer-wrapper
```

With `Pip`

```
$ pip install omxplayer-wrapper
```



## CHAPTER 2

---

### Examples:

---

Listing 1: Playing local video file

```
#!/usr/bin/env python3

from omxplayer.player import OMXPlayer
from pathlib import Path
from time import sleep

VIDEO_PATH = Path("../tests/media/test_media_1.mp4")

player = OMXPlayer(VIDEO_PATH)

sleep(5)

player.quit()
```

Listing 2: Playing RTSP stream

```
#!/usr/bin/env python3

from omxplayer.player import OMXPlayer
from time import sleep

STREAM_URI = 'rtsp://184.72.239.149/vod/mp4:BigBuckBunny_175k.mov'

player = OMXPlayer(STREAM_URI)

sleep(8)

player.quit()
```

Listing 3: Advanced usage

```
#!/usr/bin/env python3

from omxplayer.player import OMXPlayer
from pathlib import Path
from time import sleep
import logging
logging.basicConfig(level=logging.INFO)

VIDEO_1_PATH = "../tests/media/test_media_1.mp4"
player_log = logging.getLogger("Player 1")

player = OMXPlayer(VIDEO_1_PATH,
                   dbus_name='org.mpris.MediaPlayer2.omxplayer1')
player.playEvent += lambda _: player_log.info("Play")
player.pauseEvent += lambda _: player_log.info("Pause")
player.stopEvent += lambda _: player_log.info("Stop")

# it takes about this long for omxplayer to warm up and start displaying a picture on_
↪ a rpi3
sleep(2.5)

player.set_position(5)
player.pause()

sleep(2)

player.set_aspect_mode('stretch')
player.set_video_pos(0, 0, 200, 200)
player.play()

sleep(5)

player.quit()
```

---

Learning more:

---

API docs:

## 3.1 omxplayer API Docs

### 3.1.1 omxplayer.player

**exception** omxplayer.player.**FileNotFoundError**

Bases: Exception

**class** omxplayer.player.**OMXPlayer**(*source*, *args=None*, *bus\_address\_finder=None*, *Connection=None*, *dbus\_name=None*, *pause=False*)

Bases: object

OMXPlayer controller

This works by speaking to OMXPlayer over Dbus sending messages.

#### Parameters

- **source** (*str*) – Path to the file (as ~/Videos/my-video.mp4) or URL you wish to play
- **args** (*list/str*) – used to pass option parameters to omxplayer. see: <https://github.com/popcornmix/omxplayer#synopsis>

Multiple argument example:

```
>>> OMXPlayer('path.mp4', args=['--no-osd', '--no-keys', '-b'])
>>> OMXPlayer('path.mp4', args='--no-osd --no-keys -b')
>>> OMXPlayer('path.mp4', dbus_name='org.mpris.MediaPlayer2.omxplayer2')
```

**action** (*code*)

Executes a keyboard command via a code

**Parameters** **code** (*int*) – The key code you wish to emulate refer to `keys.py` for the possible keys

**aspect\_ratio()**

**Returns** aspect ratio

**Return type** float

**can\_control()**

**Returns** whether the player can be controlled

**Return type** bool

**can\_go\_next()**

**Returns** whether the player can move to the next item in the playlist

**Return type** bool

**can\_go\_previous()**

**Returns**

**whether the player can move to the previous item in the playlist**

**Return type** bool

**can\_pause()**

**Returns** whether the player can pause

**Return type** bool

**can\_play()**

**Returns** whether the player can play

**Return type** bool

**can\_quit()**

**Returns** whether the player can quit or not

**Return type** bool

**can\_raise()**

**Returns** whether the player can raise the display window atop of all other windows

**Return type** bool

**can\_seek()**

**Returns** whether the player can seek

**Return type** bool

**can\_set\_fullscreen()**

**Returns** whether the player can go fullscreen

**Return type** bool

**duration()**

**Returns** duration in seconds

**Return type** float

**exitEvent = None**

Event called on exit callback(player, exit\_status)

**fullscreen()**

**Returns** whether the player is fullscreen or not

**Return type** bool

**get\_filename()**

**Returns** source currently playing

**Return type** str

Deprecated since version 0.2.0: Use: *get\_source()* instead.

**get\_source()**

Get the source URI of the currently playing media

**Returns** source currently playing

**Return type** str

**has\_track\_list()**

**Returns** whether the player has a track list or not

**Return type** bool

**height()**

**Returns** video height in px

**Return type** int

**hide\_subtitles()**

Hide subtitles

**hide\_video()**

Hides the video overlays

**identity()**

**Returns** Returns *omxplayer*, the name of the player

**Return type** str

**is\_playing()**

**Returns** Whether the player is playing

**Return type** bool

**list\_audio()**

**Returns**

**A list of all known audio streams, each item is in the format:**

`<index>:<language>:<name>:<codec>:<active>`

**Return type** [str]

**list\_subtitles()**

**Returns**

**A list of all known subtitles, each item is in the format:** `<index>:<language>:<name>:<codec>:<act`

**Return type** [str]

**list\_video()**

### Returns

A list of all known video streams, each item is in the format:

<index>:<language>:<name>:<codec>:<active>

**Return type** [str]

**load** (*source*, *pause=False*)

Loads a new source (as a file) from *source* (a file path or URL) by killing the current omxplayer process and forking a new one.

**Parameters** **source** (*string*) – Path to the file to play or URL

**maximum\_rate** ()

**Returns** maximum playback rate (as proportion of normal rate)

**Return type** float

**metadata** ()

**Returns** containing track information ('URI', 'length')

**Return type** dict

### Examples

```
>>> player.metadata()
{
  'mpris:length': 19691000,
  'xesam:url': 'file:///home/name/path/to/media/file.mp4'
}
```

**minimum\_rate** ()

**Returns** minimum playback rate (as proportion of normal rate)

**Return type** float

**mute** ()

Mute audio. If already muted, then this does not do anything

**next** ()

Skip to the next chapter

**Returns** Whether the player skipped to the next chapter

**Return type** bool

**pause** ()

Pause playback

**pauseEvent = None**

Event called on pause callback (player)

**play** ()

Play the video asynchronously returning control immediately to the calling code

**playEvent = None**

Event called on play callback (player)

**play\_pause** ()

Pause playback if currently playing, otherwise start playing if currently paused.

**play\_sync()**

Play the video and block whilst the video is playing

**playback\_status()**

**Returns** one of (“Playing” | “Paused” | “Stopped”)

**Return type** str

**position()**

**Returns** position in seconds

**Return type** int

**positionEvent = None**

Event called on setting position `callback(player, absolute_position)`

**previous()**

Skip to the previous chapter

**Returns** Whether the player skipped to the previous chapter

**Return type** bool

**quit()**

Quit the player, blocking until the process has died

**rate()**

**Returns** playback rate, 1 is the normal rate, 2 would be double speed.

**Return type** float

**seek(*relative\_position*)**

Seek the video by *relative\_position* seconds

**Parameters** **relative\_position** (*float*) – The position in seconds to seek to.

**seekEvent = None**

Event called on seek `callback(player, relative_position)`

**select\_audio(*index*)**

Select audio stream specified by the index of the stream in *list\_audio*

**Parameters** **index** (*int*) – index of audio stream returned by *list\_audio*

**select\_subtitle(*index*)**

Enable a subtitle specified by the index it is listed in *list\_subtitles*

**Parameters** **index** (*int*) – index of subtitle listing returned by *list\_subtitles*

**set\_alpha(*alpha*)**

Set the transparency of the video overlay

**Parameters** **alpha** (*float*) – The transparency (0..255)

**set\_aspect\_mode(*mode*)**

Set the aspect mode of the video

**Parameters** **mode** (*str*) – One of (“letterbox” | “fill” | “stretch”)

**set\_position(*position*)**

Set the video to playback position to *position* seconds from the start of the video

**Parameters** **position** (*float*) – The position in seconds.

**set\_rate** (*rate*)

Set the playback rate of the video as a multiple of the default playback speed

### Examples

```
>>> player.set_rate(2)
# Will play twice as fast as normal speed
>>> player.set_rate(0.5)
# Will play half speed
```

**set\_video\_crop** (*x1, y1, x2, y2*)

#### Parameters

- **x1** (*int*) – Top left x coordinate (px)
- **y1** (*int*) – Top left y coordinate (px)
- **x2** (*int*) – Bottom right x coordinate (px)
- **y2** (*int*) – Bottom right y coordinate (px)

**set\_video\_pos** (*x1, y1, x2, y2*)

Set the video position on the screen

#### Parameters

- **x1** (*int*) – Top left x coordinate (px)
- **y1** (*int*) – Top left y coordinate (px)
- **x2** (*int*) – Bottom right x coordinate (px)
- **y2** (*int*) – Bottom right y coordinate (px)

**set\_volume** (*volume*)

**Parameters** **float** – volume in the interval [0, 10]

**show\_subtitles** ()

Shows subtitles after *hide\_subtitles*

**show\_video** ()

Shows the video (to undo a *hide\_video*)

**stop** ()

Stop the player, causing it to quit

**stopEvent** = **None**

Event called on stop callback (player)

**supported\_uri\_schemes** ()

**Returns** list of supported URI schemes

**Return type** str

### Examples

```
>>> player.supported_uri_schemes()
["file", "http", "rtsp", "rtmp"]
```

**unmute()**

Unmutes the video. If already unmuted, then this does not do anything

**video\_pos()**

**Returns**

Video spatial position (x1, y1, x2, y2) where (x1, y1) is top left, and (x2, y2) is bottom right. All values in px.

**Return type** (int, int, int, int)

**video\_stream\_count()**

**Returns** number of video streams

**Return type** int

**volume()**

**Returns** current player volume

**Return type** float

**width()**

**Returns** video width in px

**Return type** int

**exception** omxplayer.player.OMXPlayerDeadError

Bases: Exception

### 3.1.2 omxplayer.bus\_finder

**class** omxplayer.bus\_finder.BusFinder (path=None)

Bases: object

**find\_address\_file()**

Finds the OMXPlayer Dbus connection Assumes there is an alive OMXPlayer process. :return:

**get\_address()**

**wait\_for\_dbus\_address\_to\_be\_written\_to\_file()**

**wait\_for\_file()**

**wait\_for\_path\_to\_exist()**

### 3.1.3 omxplayer.dbus\_connection

**class** omxplayer.dbus\_connection.DBusConnection (bus\_address, dbus\_name=None)

Bases: object

**proxy**

The proxy object by which one interacts with a dbus object, this makes communicating with a similar to that of communicating with a POJO.

**root\_interface**

org.mpris.MediaPlayer2 interface proxy object

**player\_interface**

org.mpris.MediaPlayer2.Player interface proxy object

**exception** `omxplayer.dbus_connection.DBusConnectionError`

Bases: `Exception`

Connection error raised when `DBusConnection` can't set up a connection

**class** `omxplayer.dbus_connection.DbusObject` (*object\_proxy, property\_manager, interface\_name, methods, properties*)

Bases: `object`

### 3.1.4 `omxplayer.keys`

`omxplayer.keys.BLANK = 24`

`omxplayer.keys.DECREASE_SPEED = 1`

`omxplayer.keys.DECREASE_SUBTITLE_DELAY = 13`

`omxplayer.keys.DECREASE_VOLUME = 17`

`omxplayer.keys.EXIT = 15`

`omxplayer.keys.FAST_FORWARD = 4`

`omxplayer.keys.HIDE_SUBTITLES = 30`

`omxplayer.keys.HIDE_VIDEO = 28`

`omxplayer.keys.INCREASE_SPEED = 2`

`omxplayer.keys.INCREASE_SUBTITLE_DELAY = 14`

`omxplayer.keys.INCREASE_VOLUME = 18`

`omxplayer.keys.MOVE_VIDEO = 27`

`omxplayer.keys.NEXT_AUDIO = 7`

`omxplayer.keys.NEXT_CHAPTER = 9`

`omxplayer.keys.NEXT_SUBTITLE = 11`

`omxplayer.keys.PAUSE = 16`

`omxplayer.keys.PREVIOUS_AUDIO = 6`

`omxplayer.keys.PREVIOUS_CHAPTER = 8`

`omxplayer.keys.PREVIOUS_SUBTITLE = 10`

`omxplayer.keys.REWIND = 3`

`omxplayer.keys.SEEK_ABSOLUTE = 26`

`omxplayer.keys.SEEK_BACK_LARGE = 21`

`omxplayer.keys.SEEK_BACK_SMALL = 19`

`omxplayer.keys.SEEK_FORWARD_LARGE = 22`

`omxplayer.keys.SEEK_FORWARD_SMALL = 20`

`omxplayer.keys.SEEK_RELATIVE = 25`

`omxplayer.keys.SET_ALPHA = 32`

`omxplayer.keys.SHOW_INFO = 5`

`omxplayer.keys.SHOW_SUBTITLES = 31`

```
omxplayer.keys.STEP = 23  
omxplayer.keys.TOGGLE_SUBTITLE = 12  
omxplayer.keys.UNHIDE_VIDEO = 29
```



**O**

`omxplayer.bus_finder`, 13  
`omxplayer.dbus_connection`, 13  
`omxplayer.keys`, 14  
`omxplayer.player`, 7



**A**

action() (*omxplayer.player OMXPlayer method*), 7  
 aspect\_ratio() (*omxplayer.player OMXPlayer method*), 7

**B**

BLANK (*in module omxplayer.keys*), 14  
 BusFinder (*class in omxplayer.bus\_finder*), 13

**C**

can\_control() (*omxplayer.player OMXPlayer method*), 8  
 can\_go\_next() (*omxplayer.player OMXPlayer method*), 8  
 can\_go\_previous() (*omxplayer.player OMXPlayer method*), 8  
 can\_pause() (*omxplayer.player OMXPlayer method*), 8  
 can\_play() (*omxplayer.player OMXPlayer method*), 8  
 can\_quit() (*omxplayer.player OMXPlayer method*), 8  
 can\_raise() (*omxplayer.player OMXPlayer method*), 8  
 can\_seek() (*omxplayer.player OMXPlayer method*), 8  
 can\_set\_fullscreen() (*omxplayer.player OMXPlayer method*), 8

**D**

DBusConnection (*class in omxplayer.dbus\_connection*), 13  
 DBusConnectionError, 13  
 dbusObject (*class in omxplayer.dbus\_connection*), 14  
 DECREASE\_SPEED (*in module omxplayer.keys*), 14  
 DECREASE\_SUBTITLE\_DELAY (*in module omxplayer.keys*), 14  
 DECREASE\_VOLUME (*in module omxplayer.keys*), 14  
 duration() (*omxplayer.player OMXPlayer method*), 8

**E**

EXIT (*in module omxplayer.keys*), 14

exitEvent (*omxplayer.player OMXPlayer attribute*), 8

**F**

FAST\_FORWARD (*in module omxplayer.keys*), 14  
 FileNotFoundError, 7  
 find\_address\_file() (*omxplayer.bus\_finder BusFinder method*), 13  
 fullscreen() (*omxplayer.player OMXPlayer method*), 8

**G**

get\_address() (*omxplayer.bus\_finder BusFinder method*), 13  
 get\_filename() (*omxplayer.player OMXPlayer method*), 9  
 get\_source() (*omxplayer.player OMXPlayer method*), 9

**H**

has\_track\_list() (*omxplayer.player OMXPlayer method*), 9  
 height() (*omxplayer.player OMXPlayer method*), 9  
 HIDE\_SUBTITLES (*in module omxplayer.keys*), 14  
 hide\_subtitles() (*omxplayer.player OMXPlayer method*), 9  
 HIDE\_VIDEO (*in module omxplayer.keys*), 14  
 hide\_video() (*omxplayer.player OMXPlayer method*), 9

**I**

identity() (*omxplayer.player OMXPlayer method*), 9  
 INCREASE\_SPEED (*in module omxplayer.keys*), 14  
 INCREASE\_SUBTITLE\_DELAY (*in module omxplayer.keys*), 14  
 INCREASE\_VOLUME (*in module omxplayer.keys*), 14  
 is\_playing() (*omxplayer.player OMXPlayer method*), 9

## L

`list_audio()` (*omxplayer.player OMXPlayer method*), 9  
`list_subtitles()` (*omxplayer.player OMXPlayer method*), 9  
`list_video()` (*omxplayer.player OMXPlayer method*), 9  
`load()` (*omxplayer.player OMXPlayer method*), 10

## M

`maximum_rate()` (*omxplayer.player OMXPlayer method*), 10  
`metadata()` (*omxplayer.player OMXPlayer method*), 10  
`minimum_rate()` (*omxplayer.player OMXPlayer method*), 10  
`MOVE_VIDEO` (*in module omxplayer.keys*), 14  
`mute()` (*omxplayer.player OMXPlayer method*), 10

## N

`next()` (*omxplayer.player OMXPlayer method*), 10  
`NEXT_AUDIO` (*in module omxplayer.keys*), 14  
`NEXT_CHAPTER` (*in module omxplayer.keys*), 14  
`NEXT_SUBTITLE` (*in module omxplayer.keys*), 14

## O

`OMXPlayer` (*class in omxplayer.player*), 7  
`omxplayer.bus_finder` (*module*), 13  
`omxplayer.dbus_connection` (*module*), 13  
`omxplayer.keys` (*module*), 14  
`omxplayer.player` (*module*), 7  
`OMXPlayerDeadError`, 13

## P

`PAUSE` (*in module omxplayer.keys*), 14  
`pause()` (*omxplayer.player OMXPlayer method*), 10  
`pauseEvent` (*omxplayer.player OMXPlayer attribute*), 10  
`play()` (*omxplayer.player OMXPlayer method*), 10  
`play_pause()` (*omxplayer.player OMXPlayer method*), 10  
`play_sync()` (*omxplayer.player OMXPlayer method*), 10  
`playback_status()` (*omxplayer.player OMXPlayer method*), 11  
`player_interface` (*omxplayer.dbus\_connection DBusConnection attribute*), 13  
`playEvent` (*omxplayer.player OMXPlayer attribute*), 10  
`position()` (*omxplayer.player OMXPlayer method*), 11  
`positionEvent` (*omxplayer.player OMXPlayer attribute*), 11

`previous()` (*omxplayer.player OMXPlayer method*), 11  
`PREVIOUS_AUDIO` (*in module omxplayer.keys*), 14  
`PREVIOUS_CHAPTER` (*in module omxplayer.keys*), 14  
`PREVIOUS_SUBTITLE` (*in module omxplayer.keys*), 14  
`proxy` (*omxplayer.dbus\_connection DBusConnection attribute*), 13

## Q

`quit()` (*omxplayer.player OMXPlayer method*), 11

## R

`rate()` (*omxplayer.player OMXPlayer method*), 11  
`REWIND` (*in module omxplayer.keys*), 14  
`root_interface` (*omxplayer.dbus\_connection DBusConnection attribute*), 13

## S

`seek()` (*omxplayer.player OMXPlayer method*), 11  
`SEEK_ABSOLUTE` (*in module omxplayer.keys*), 14  
`SEEK_BACK_LARGE` (*in module omxplayer.keys*), 14  
`SEEK_BACK_SMALL` (*in module omxplayer.keys*), 14  
`SEEK_FORWARD_LARGE` (*in module omxplayer.keys*), 14  
`SEEK_FORWARD_SMALL` (*in module omxplayer.keys*), 14  
`SEEK_RELATIVE` (*in module omxplayer.keys*), 14  
`seekEvent` (*omxplayer.player OMXPlayer attribute*), 11  
`select_audio()` (*omxplayer.player OMXPlayer method*), 11  
`select_subtitle()` (*omxplayer.player OMXPlayer method*), 11  
`SET_ALPHA` (*in module omxplayer.keys*), 14  
`set_alpha()` (*omxplayer.player OMXPlayer method*), 11  
`set_aspect_mode()` (*omxplayer.player OMXPlayer method*), 11  
`set_position()` (*omxplayer.player OMXPlayer method*), 11  
`set_rate()` (*omxplayer.player OMXPlayer method*), 11  
`set_video_crop()` (*omxplayer.player OMXPlayer method*), 12  
`set_video_pos()` (*omxplayer.player OMXPlayer method*), 12  
`set_volume()` (*omxplayer.player OMXPlayer method*), 12  
`SHOW_INFO` (*in module omxplayer.keys*), 14  
`SHOW_SUBTITLES` (*in module omxplayer.keys*), 14  
`show_subtitles()` (*omxplayer.player OMXPlayer method*), 12

`show_video()` (*omxplayer.player OMXPlayer method*), 12  
`STEP` (*in module omxplayer.keys*), 14  
`stop()` (*omxplayer.player OMXPlayer method*), 12  
`stopEvent` (*omxplayer.player OMXPlayer attribute*), 12  
`supported_uri_schemes()` (*omxplayer.player OMXPlayer method*), 12

## T

`TOGGLE_SUBTITLE` (*in module omxplayer.keys*), 15

## U

`UNHIDE_VIDEO` (*in module omxplayer.keys*), 15  
`unmute()` (*omxplayer.player OMXPlayer method*), 12

## V

`video_pos()` (*omxplayer.player OMXPlayer method*), 13  
`video_stream_count()` (*omxplayer.player OMXPlayer method*), 13  
`volume()` (*omxplayer.player OMXPlayer method*), 13

## W

`wait_for_dbus_address_to_be_written_to_file()` (*omxplayer.bus\_finder BusFinder method*), 13  
`wait_for_file()` (*omxplayer.bus\_finder BusFinder method*), 13  
`wait_for_path_to_exist()` (*omxplayer.bus\_finder BusFinder method*), 13  
`width()` (*omxplayer.player OMXPlayer method*), 13