

---

# **daemonize Documentation**

*Release 2.4.7*

**Ilya Otyutskiy**

July 28, 2016



<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>File descriptors</b>	<b>9</b>
<b>5</b>	<b>API</b>	<b>11</b>
<b>6</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>



**daemonize** is a library for writing system daemons in Python. It is distributed under MIT license. Latest version can be downloaded from [PyPI](#). Full documentation can be found at [ReadTheDocs](#).



---

## Dependencies

---

It is tested under following Python versions:

- 2.6
- 2.7
- 3.3
- 3.4
- 3.5





---

## Installation

---

You can install it from Python Package Index (PyPI):

```
$ pip install daemonize
```



---

## Usage

---

```
from time import sleep
from daemonize import Daemonize

pid = "/tmp/test.pid"

def main():
    while True:
        sleep(5)

daemon = Daemonize(app="test_app", pid=pid, action=main)
daemon.start()
```



---

## File descriptors

---

Daemonize object's constructor understands the optional argument **keep\_fds** which contains a list of FDs which should not be closed. For example:

```
import logging
from daemonize import Daemonize

pid = "/tmp/test.pid"
logger = logging.getLogger(__name__)
logger.setLevel(logging.DEBUG)
logger.propagate = False
fh = logging.FileHandler("/tmp/test.log", "w")
fh.setLevel(logging.DEBUG)
logger.addHandler(fh)
keep_fds = [fh.stream.fileno()]

def main():
    logger.debug("Test")

daemon = Daemonize(app="test_app", pid=pid, action=main, keep_fds=keep_fds)
daemon.start()
```



---

```
class daemonize.Daemonize(app, pid, action, keep_fds=None, auto_close_fds=True, privileged_action=None, user=None, group=None, verbose=False, logger=None, foreground=False, chdir='/')
```

Daemonize object.

Object constructor expects three arguments.

#### Parameters

- **app** – contains the application name which will be sent to syslog.
- **pid** – path to the pidfile.
- **action** – your custom function which will be executed after daemonization.
- **keep\_fds** – optional list of fds which should not be closed.
- **auto\_close\_fds** – optional parameter to not close opened fds.
- **privileged\_action** – action that will be executed before drop privileges if user or group parameter is provided. If you want to transfer anything from privileged\_action to action, such as opened privileged file descriptor, you should return it from privileged\_action function and catch it inside action function.
- **user** – drop privileges to this user if provided.
- **group** – drop privileges to this group if provided.
- **verbose** – send debug messages to logger if provided.
- **logger** – use this logger object instead of creating new one, if provided.
- **foreground** – stay in foreground; do not fork (for debugging)
- **chdir** – change working directory if provided or /

**exit** ()

Cleanup pid file at exit.

**sigterm** (*signum, frame*)

These actions will be done after SIGTERM.

**start** ()

Start daemonization process.





---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



**d**

daemonize, [11](#)



## D

Daemonize (class in daemonize), 11  
daemonize (module), 11

## E

exit() (daemonize.Daemonize method), 11

## S

sigterm() (daemonize.Daemonize method), 11  
start() (daemonize.Daemonize method), 11