daemonize Documentation

Release 2.4.7

Ilya Otyutskiy

Mar 23, 2018

Contents

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.

CHAPTER 1

Dependencies

It is tested under following Python versions:

- 2.6
- 2.7
- 3.3
- 3.4
- 3.5

CHAPTER 2

Installation

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

```
$ pip install daemonize
```

CHAPTER 3

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()
```

CHAPTER 4

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()
```

CHAPTER 5

API

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.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

d

Index