
pk3proc

Release 0.2.1

May 22, 2020

Contents:

1	Documentation for the Code	3
1.1	Exceptions	3
1.2	Functions	4
2	Indices and tables	7
	Python Module Index	9
	Index	11

pykit3 is is a collection of toolkit in python3.

pk3proc is utility to create sub process.

Execute a shell script:

```
import pk3proc

# execute a shell script

returncode, out, err = pk3proc.shell_script('ls / | grep bin')
print returncode
print out
# output:
# > 0
# > bin
# > sbin
```

Run a command:

```
# Unlike the above snippet, following statement does not start an sh process.
returncode, out, err = pk3proc.command('ls', 'a*', cwd='/usr/local')
```

1.1 Exceptions

exception `pk3proc.CalledProcessError` (*returncode, out, err, cmd, options*)

It is sub class of *subprocess.CalledProcessError*.

It is raised if a sub process return code is not 0. Besides *CalledProcessError.args*, extended from super class *Exception*, it has 6 other attributes.

returncode

process exit code.

Type int

stdout

stdout in one string.

Type str

stderr

stderr in one string.

Type str

out

stdout in list.

Type list

err

stderr in list.

Type list

cmd

the command a process *exec()*.

Type list

options

other options passed to this process. Such as *close_fds*, *cwd* etc.

Type dict

pk3proc.ProcError

alias of `pk3proc.proc.CalledProcessError`

exception pk3proc.TimeoutExpired (*cmd*, *timeout*, *output=None*, *stderr=None*)

This exception is raised when the timeout expires while waiting for a child process.

cmd, **output**, **stdout**, **stderr**, **timeout**

1.2 Functions

pk3proc.command (*cmd*, **arguments*, *bufsize=-1*, *close_fds=True*, *creationflags=0*, *cwd=None*, *encoding=None*, *env=None*, *errors=None*, *executable=None*, *pass_fds=()*, *pre_exec_fn=None*, *restore_signals=True*, *shell=False*, *start_new_session=False*, *startupinfo=None*, *stderr=None*, *stdin=None*, *stdout=None*, *text=None*, *universal_newlines=None*, *input=None*, *check=False*, *inherit_env=None*, *timeout=None*, *capture=None*, *tty=None*)

Run a *cmd* with arguments *arguments* in a subprocess. It blocks until sub process exit or timeout.

**options* are the same as *subprocess.Popen*. Only those differ from *subprocess.Popen* are listed.

Parameters

- **cmd** (*list*, *tuple*, *str*) – The path of executable to run.
- **arguments** (*list*, *tuple*) – arguments passed to *cmd*.
- **encoding** – by default is the system default encoding.
- **env** – by default inherit from parent process.
- **check=False** – if *True*, raise *CalledProcessError* if returncode is not 0. By default it is *False*.

- **capture=True** – whether to capture stdin, stdout and stderr. Otherwise inherit these fd from current process.
- **inherit_env=True** – whether to inherit environment vars from current process.
- **input=None** – input to send to stdin, if it is not None.
- **timeout=None** – seconds to wait for sub process to exit. By default it is None, for waiting for ever.
- **tty=False** – whether to create a pseudo tty to run sub process so that the sub process believes it is in a tty(just like controlled by a human).

Returns

- *returncode*: sub process exit code.
- *out*: sub process stdout.
- *err*: sub process stderr.

Return type (int, str, str)

Raises

- *CalledProcessError* – If the sub process exit with non-zero and *check=True*.
- *TimeoutExpired* – If *timeout* is not *None* and expires before sub process exit.

`pk3proc.command_ex(cmd, *arguments, **options)`

This is a shortcut of *command* with *check=True*: if sub process exit code is not 0, it raises exception *CalledProcessError*.

`pk3proc.shell_script(script_str, **options)`

This is a shortcut of *command*("sh", *input=script_str*).

Run a shell script:

```
shell_script('ls | grep foo.txt')
```

`pk3proc.start_process(cmd, target, env, *args)`

Create a child process and replace it with *cmd*. Besides *stdin*, *stdout* and *stderr*, all file descriptors from parent process will be closed in the child process. The parent process waits for the child process until it is completed.

Parameters

- **cmd** (*str*) – The path of executable to run. Such as *sh*, *bash*, *python*.
- **target** (*str*) – The path of the script.
- **env** (*dict*) – pass environment variables to the child process.
- ***args** – The arguments passed to the script. Type of every element must be *str*.

CHAPTER 2

Indices and tables

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

p

pk3proc, 3

C

`CalledProcessError`, 3
`cmd` (*pk3proc.CalledProcessError attribute*), 4
`command()` (*in module pk3proc*), 4
`command_ex()` (*in module pk3proc*), 5

E

`err` (*pk3proc.CalledProcessError attribute*), 4

O

`options` (*pk3proc.CalledProcessError attribute*), 4
`out` (*pk3proc.CalledProcessError attribute*), 4

P

`pk3proc` (*module*), 3
`ProcError` (*in module pk3proc*), 4

R

`returncode` (*pk3proc.CalledProcessError attribute*), 3

S

`shell_script()` (*in module pk3proc*), 5
`start_process()` (*in module pk3proc*), 5
`stderr` (*pk3proc.CalledProcessError attribute*), 4
`stdout` (*pk3proc.CalledProcessError attribute*), 3

T

`TimeoutExpired`, 4