

Join the Stack Overflow Community

Stack Overflow is a community of 6.8 million programmers, just like you, helping each other.
Join them; it only takes a minute:

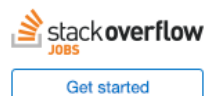
[Sign up](#)

How do I get all of the output from my .exe using subprocess and Popen?

```

36 if (dev.isBored() || job.sucks()) {
37     searchJobs({flexibleHours: true, companyCulture: 100});
38 }
39 // A career site that's by developers, for developers.

```



I am trying to run an executable and capture its output using `subprocess.Popen` ; however, I don't seem to be getting all of the output.

```

import subprocess as s
from subprocess import Popen
import os

ps = Popen(r'C:\Tools\Dvb_pid_3_0.exe', stdin = s.PIPE, stdout = s.PIPE)
print 'popen done..'

while:

    line = ps.stdout.readline()
    print line

```

It prints two line less than the original exe file when opened manually.

I tried an alternative approach with the same result:

```

f = open('myprogram_output.txt', 'w')
proc = Popen('C:\Tools\Dvb_pid_3_0.exe ', stdout =f)

line = proc.stdout.readline()
print line
f.close()

```

Can anyone please help me to get the full data of the exe?

As asked by Sebastian:

Original exe file last few lines o/p:

-Gdd : Generic count (1 - 1000)

-Cdd : Cut start at (0 - 99) -Edd : Cut end at (1 - 100)

Please select the stream file number below:

1 - .\pdsx100-bcm7230-squashfs-sdk0.0.0.38-0.2.6.0-prod.sao.ts

The o/p I get after running:

```

-P0xYYYY      : Pid been interested

-S0xYYYY : Service ID been interested
-T0xYYYY : Transport ID been interested
-N0xYYYY : Network ID been interested
-R0xYYYY : A old Pid been replaced by this PID
-Gdd : Generic count (1 - 1000)

```

So we can see some lines missing. I have to write 1 and choose value after please select the file number below appears.

I tried to use `ps.stdin.write('1\n')`. It didn't print the value in the exe file

New code:

```
#!/usr/bin/env python
from subprocess import Popen, PIPE

cmd = r'C:\Tools\Dvb_pid_3_0.exe'
p = Popen(cmd, stdin=PIPE, stdout=None, stderr=None, universal_newlines=True)
stdout_text, stderr_text = p.communicate(input="1\n\n")

print("stdout: %r\nstderr: %r" % (stdout_text, stderr_text))
if p.returncode != 0:
    raise RuntimeError("%r failed, status code %d" % (cmd, p.returncode))
```

Thanks Sebastian. I am able to see the entire output but not able to feed in any input with the current code.

[python](#) [windows](#) [subprocess](#) [popen](#)

edited Sep 28 '12 at 10:06

asked Sep 26 '12 at 11:39



Ajith

26 1 6

1 What are the two lines in question? Are you sure they're send to `stdout` and not `stderr` ? – [Pierre GM](#) Sep 26 '12 at 11:46

1 Typo in your first example, indentation after the `while` , or is your code really like that? – [cdarke](#) Sep 26 '12 at 11:48

@cdarke, Yes typo it is while true: while editing it was removed – [Ajith](#) Sep 27 '12 at 7:37

2 Answers

To get all stdout as a string:

```
from subprocess import check_output as qx

cmd = r'C:\Tools\Dvb_pid_3_0.exe'
output = qx(cmd)
```

To get both stdout and stderr as a single string:

```
from subprocess import STDOUT

output = qx(cmd, stderr=STDOUT)
```

To get all lines as a list:

```
lines = output.splitlines()
```

To get lines as they are being printed by the subprocess:

```
from subprocess import Popen, PIPE

p = Popen(cmd, stdout=PIPE, bufsize=1)
for line in iter(p.stdout.readline, ''):
    print line,
p.stdout.close()
if p.wait() != 0:
    raise RuntimeError("%r failed, exit status: %d" % (cmd, p.returncode))
```

Add `stderr=STDOUT` to the `Popen()` call to merge stdout/stderr.

Note: if `cmd` uses block-buffering in the non-interactive mode then lines won't appear until the buffer flushes. [winexpect](#) module might be able to get the output sooner.

To save the output to a file:

```
import subprocess

with open('output.txt', 'wb') as f:
    subprocess.check_call(cmd, stdout=f)

# to read line by line
with open('output.txt') as f:
    for line in f:
        print line,
```

If `cmd` always requires input even an empty one; set `stdin` :

```
import os

with open(os.devnull, 'rb') as DEVNULL:
    output = qx(cmd, stdin=DEVNULL) # use subprocess.DEVNULL on Python 3.3+
```

You could combine these solutions e.g., to merge stdout/stderr, and to save the output to a file, and to provide an empty input:

```
import os
from subprocess import STDOUT, check_call as x

with open(os.devnull, 'rb') as DEVNULL, open('output.txt', 'wb') as f:
    x(cmd, stdin=DEVNULL, stdout=f, stderr=STDOUT)
```

To provide all input as a single string you could use `.communicate()` method:

```
#!/usr/bin/env python
from subprocess import Popen, PIPE

cmd = ["python", "test.py"]
p = Popen(cmd, stdin=PIPE, stdout=PIPE, stderr=PIPE, universal_newlines=True)
stdout_text, stderr_text = p.communicate(input="1\n\n")

print("stdout: %r\nstderr: %r" % (stdout_text, stderr_text))
if p.returncode != 0:
    raise RuntimeError("%r failed, status code %d" % (cmd, p.returncode))
```

where `test.py` :

```
print raw_input('abc')[::-1]
raw_input('press enter to exit')
```

If your interaction with the program is more like a conversation than you might need [winexpect](#) module. Here's an [example from pexpect docs](#):

```
# This connects to the openbsd ftp site and
# downloads the recursive directory listing.
from winexpect import winspawn as spawn

child = spawn ('ftp ftp.openbsd.org')
child.expect ('Name .*: ')
child.sendline ('anonymous')
child.expect ('Password:')
child.sendline ('noah@example.com')
child.expect ('ftp> ')
child.sendline ('cd pub')
child.expect('ftp> ')
child.sendline ('get ls-lR.gz')
child.expect('ftp> ')
child.sendline ('bye')
```

To send special keys such as `F3` , `F10` on Windows you might need [SendKeys](#) module or its pure Python implementation [SendKeys-ctypes](#) . Something like:

```
from SendKeys import SendKeys

SendKeys(r"""
{LWIN}
{PAUSE .25}
r
C:\Tools\Dvb_pid_3_0.exe{ENTER}
{PAUSE 1}
1{ENTER}
{PAUSE 1}
2{ENTER}
{PAUSE 1}
{F3}
{PAUSE 1}
{F10}
""")
```

It doesn't capture output.

edited Oct 3 '12 at 8:22

answered Sep 26 '12 at 16:31



J.F. Sebastian

194k 48 378 553

from subprocess import check_output as qx What does `qx` stand for? – [Piotr Dobrogost](#) Sep 26 '12 at 18:16

`x` eXecute, `q` quote: origin `` backquotes in bash, `qx''` operator in perl. – [J.F. Sebastian](#) Sep 26 '12 at 18:42

Hi Sebastian, Many thanks for taking the time and effort to provide the answers. I tried your script 1)To get lines as they are printed by the subprocesses, 2)To save the output to a file. Both the scripts gave me the same

response as before which is two lines less. Couldn't understand how to write inputs using your script .Can you please explain.. Many thanks again – [Ajith](#) Sep 27 '12 at 8:05

@Ajith: Have you tried `stderr=STDOUT` as shown in the 2nd example? Are these two missing lines something like password prompts (they might be written directly to the console (outside stdout, stderr))? Do you want to provide all input at once and read the whole output from the program or is the interaction more like a conversation (back and forth)? How large are the input/output? [Update your question with the info.](#) You could leave a comment if you want me to be notified. – [J.F. Sebastian](#) Sep 27 '12 at 9:12

Second example. The script just waits for something after execution. Error message after Keyboard interrupt- C:\Tools>python DVB.py Traceback (most recent call last): File "DVB.py", line 16, in <module> output = qx(cmd, stderr=STDOUT) File "C:\Python27\lib\subprocess.py", line 538, in check_output output, unused_err = process.communicate() File "C:\Python27\lib\subprocess.py", line 746, in communicate stdout = _eintr_retry_call(self.stdout.read) File "C:\Python27\lib\subprocess.py", line 478, in _eintr_retry_call return func(*args) KeyboardInterrupt – [Ajith](#) Sep 27 '12 at 12:13

Traceback (most recent call last): File "C:\Tools\Dvb_Pid.py", line 12, in <module> ps.stdin.write("1\n") IOError: [Errno 22] Invalid argument – [Ajith](#) Sep 27 '12 at 13:09

@Ajith: I've added examples on how to provide non-empty input to my answer. – [J.F. Sebastian](#) Sep 27 '12 at 13:11

Ignore the previous comment. Please also let me know how to input some values as well. I have tried `p.stdin.write("1\n")` and `p.communicate()[0]`. Nothing works. I wonder why the two lines are not read and not able to see whatever I gave as an input – [Ajith](#) Sep 27 '12 at 13:22

This is the output I get. Traceback (most recent call last): File "C:\Tools\Last.py", line 10, in <module> raise RuntimeError("%r failed, status code %d" % (cmd, p.returncode))... RuntimeError: 'C:\Tools\Dvb_pid_3_0.exe' failed, status code -1073741510 the Dvb_pid.exe screen opens and cursor starts to blink. Once the exe window is closed it prints the output till GDD in python shell and gave the above error – [Ajith](#) Sep 27 '12 at 13:35

@Ajith: Either update your question or ask a new one. Include a minimal complete example (the code) that produces the error and the corresponding traceback (format it properly (there is an orange help button with ? when you edit a SO question)). Describe in plain English how you interact with `Dvb_pid.exe` when you start it manually from a console. Is `Dvb_pid.exe` a command-line utility or it opens its own window when you start it from `cmd.exe` window? What input does it expect? – [J.F. Sebastian](#) Sep 27 '12 at 14:15

Apologise .My mistake. – [Ajith](#) Sep 27 '12 at 14:27

`Dvb_pid.exe` is a command line utility. It doesn't start its own window in `cmd.exe`. It populates the .ts files present in the folder and ask for user input as Please select the stream file number below: 1 - .\pdsx100-bcm7230-squashfs-sdk0.0.0.38-0.2.6.0-prod.sao.ts – [Ajith](#) Sep 27 '12 at 14:31

@Ajith: You should add this info to the question. Do you see "Please select.." in `stdout_text` or `stderr_text` ? Try to experiment with combinations of `universal_newlines=True`, `universal_newlines=False` and removing `\n` or replacing it with `\r\n`. Do you press just 1 when it displays the prompt or do you press 1 then Enter when you run it manually? Do you press anything else after that until the program completes? – [J.F. Sebastian](#) Sep 27 '12 at 14:50

I press 1 and enter . It will pop up another menu and I need to press 2 and enter. Then need to press F3 and F10 .It will generate a table which I need to put into a file. Thanks for the help . Kindly check this out.python-forum.org/pythonforum/viewtopic.php?f=15&t=22871 [python-forum.org/pythonforum/...](http://python-forum.org/pythonforum/) is a similar exe like `dvb.exe`. – [Ajith](#) Sep 27 '12 at 17:04

@Ajith: sending F3, F10 could require Windows-specific code. It might be worth to ask it as a separate question. Does the program receive any input at all when you pass it via `.communicate` (it might not read its input from the pipe provided by `subprocess` module)? About the code on python-forum.org: don't use `p.stdin.write`, `p.stdout.read` directly unless you know what you are doing. – [J.F. Sebastian](#) Sep 27 '12 at 20:14

The program doesn't receive any inputs via `.communicate` or I couldn't see what's happening. All I got was a black screen with `.communicate`. – [Ajith](#) Sep 28 '12 at 8:41

@Ajith: actually it is a good sign. It means that the stdout/stderr redirection works. Set stdout/stderr to `None` temporarily to see whether the menu navigation works via `.communicate("1\n2\n")`. Though it won't help with F3,F10 issue. Can you navigate the menu using only printable characters? – [J.F. Sebastian](#) Sep 28 '12 at 9:12

Able to see the entire output..But the input 1 is not seen...You can see the code in the question to confirm – [Ajith](#) Sep 28 '12 at 10:03

But not sure it runs as an exe because when I ran the script in idle .I got black screen with o/p in shell as `stdout:None and stderr:None>>> stdout: None stderr: None` Traceback (most recent call last): File "C:\Tools\Last.py", line 10, in <module> raise RuntimeError("%r failed, status code %d" % (cmd, p.returncode)) RuntimeError: 'C:\Tools\Dvb_pid_3_0.exe' failed, status code -1073741510 – [Ajith](#) Sep 28 '12 at 10:13

@Ajith: run your script from console (`cmd.exe`) to avoid interference with idle – [J.F. Sebastian](#) Sep 28 '12 at 10:18

Yes from console ..it works as how .exe file works and waits for the input 1.. – [Ajith](#) Sep 28 '12 at 10:20

Hi Sebastian, Anything wrong with the code. Please reply. – [Ajith](#) Sep 28 '12 at 12:14

@Ajith: what is true: it waits for the input or it prints the whole output? Are you able to provide input from keyboard when you run the Python script and it waits for input (i.e., does it ignore `stdin=PIPE` ?) . To be clear *none of the examples in my answer* can pass function keys (F*) on Windows. – [J.F. Sebastian](#) Sep 28 '12 at 12:33

It waits for the input just like how I open it manually. Yes I was able to provide input from keyboard. – [Ajith](#) Oct 3 '12 at 7:14

@Ajith: If it ignores `stdin=PIPE` then you need Windows specific configuration. I've added `SendKeys` example to the answer. Have you tried `winexpect`? – J.F. Sebastian Oct 3 '12 at 8:26

If the called exe returns a error code, this fails with `raise.CalledProcessError(retcode,process.args,output=output)` ... returned non-zero `exit status 2` in my example. Any way to prevent that and still capture the screen output? – 576i Jul 2 '15 at 13:17

@576i: see [subprocess.check_output\(\) : show output on failure](#). If something is unclear ask a new StackOverflow question – J.F. Sebastian Jul 2 '15 at 14:06



The indentation of your question threw me off a bit, since Python is particular about that. Have you tried something as so:

```
import subprocess as s
from subprocess import Popen
import os

ps = Popen(r'C:\Tools\Dvb_pid_3_0.exe', stdin = s.PIPE, stdout = s.PIPE)
print 'popen done..'

(stdout, stderr) = ps.communicate()
print stdout
```

I think that `stdout` will be one single string of whatever you return from your command, so this may not be what you desire, since `readLine()` presumes you want to view output line by line.

Would suggest poking around <http://docs.python.org/library/subprocess.html> for some uses that match what you are up to.

answered Sep 26 '12 at 12:05



[terryjbates](#)

163 7

Hi, Thanks for replying but the code just waits for something after executing. – Ajith Sep 27 '12 at 7:54

Thanks again. Very helpful but it didnt work – Ajith Sep 27 '12 at 17:03