

Forum: [Hack Forums](#)

Thread: [Getting console output when there is no console](#)

Rorikstead - February 17, 2015, 5:46 am

The title might be kind of confusing, I apologize for that.

So basically, I'm trying to get the output of a cmd command that's run through a python script.

I know there are a few ways to do this, like `subprocess.check_output`, `subprocess.Popen`, etc. But everything I've tried so far does NOT work when my file is converted to a 'console-less' executable. It will always return an empty string.

Is there any way I can get the output of the console command without actually showing a console on-screen?

Thanks everyone.

bytecoder - February 17, 2015, 5:59 am

Maybe you can save the output to a file and retrieve the data through it?

Can you provide some code to look further into the problem?

Rorikstead - February 17, 2015, 6:06 am

(Feb 17, 2015 06:59 AM)bytecoder Wrote: [\[CITE\]](#) Maybe you can save the output to a file and retrieve the data through it?

Can you provide some code to look further into the problem?

Thanks, this could possibly be what I'm looking for, I'll test it out and get back to you.

And here is the code:

```
Code serial = check_output(["wmic","csproduct","get","UUID"])[41:77]
print serial
```

Works fine in the interpreter, but as soon as I disable that window from appearing, `check_output()` as well as everything else I've tried, just returns "".

bytecoder - February 17, 2015, 6:23 am

The easiest would be just save it to a file and read it. Simple. Fast.

There is a very good tutorial for IO:

tutorialspoint.com/python/python_files_io.htm

Rorikstead - February 17, 2015, 6:26 am

(Feb 17, 2015 07:23 AM)bytecoder Wrote: [\[CITE\]](#) The easiest would be just save it to a file and read it. Simple. Fast.

There is a very good tutorial for IO:

tutorialspoint.com/python/python_files_io.htm

The issue is that there is nothing to save.

When I run any sort of command without a console window, the output is always an empty string. I need a way to get the output of the command, without having a console screen show up.

bytecoder - February 17, 2015, 6:39 am

Then there is something wrong with your output in general.

If you have

Code serial = check_output(["wmic","csproduct","get","UUID"])[41:77]
you will get some information, which you can save. When you get an empty string, then there is something else bad.

Rorikstead - February 17, 2015, 6:47 am

(Feb 17, 2015 07:39 AM)bytecoder Wrote: [CITE](#) Then there is something wrong with your output in general.

If you have

Code serial = check_output(["wmic","csproduct","get","UUID"])[41:77]
you will get some information, which you can save. When you get an empty string, then there is something else bad.

I do, normally.

I'm using pyinstaller to compile this script into an exe, and there is an option to disable the console window. Once it is disabled, check_output() no longer returns anything. I can run the same exact executable with a console window, and it'll work.

I think I may have found a way around this though, I'll post my code if it does end up working correctly.

[EDIT]

I have found a solution! It's not the cleanest, but it does the job. If anyone can find a better way to do this please let me know.

```
Code call('wmic csproduct get UUID > '+ uuidpath, shell=True)
f = open(uuidpath,"r")
utfData = f.read()
f.close()
```

serial = str(utfData.decode('utf-16-le').split()[1])
Explanation
Code call('wmic csproduct get UUID > '+ uuidpath, shell=True)Calls the command and sticks the output in my previously defined uuidpath. Note that putting the output into a file is actually part of the command, instead of part of the python script.

```
Code f = open(uuidpath,"r")
utfData = f.read()
f.close()
```

This opens the file that the output was stored in, and reads it, saving the info to utfData.

Code serial = str(utfData.decode('utf-16-le').split()[1])
The text is actually stored in a UTF-16 format by the console, so here we decode it and store the relevant information (the serial number) in serial.

MR.FREE - February 18, 2015, 8:52 am

why not just make it a function that returns a value? that what you want right, one function to take output from another?

```
Code def func()
    do
    return did
```

bytecoder - February 18, 2015, 8:58 am

MR.FREE is right.

Rorikstead if you can provide me your whole code, I can look over it and maybe fix things right.

(But I still like it very much, how you solved it.)

Rorikstead - February 18, 2015, 11:00 am

(Feb 18, 2015 09:52 AM)MR.FREE Wrote: [\[CITE\]](#) why not just make it a function that returns a value? that what you want right, one function to take output from another?

```
Code def func()
    do
    return did
```

Because the command isn't within python, I'm trying to get the output of a command that I sent to the windows console.

There are modules to check the output of a command like that, but none of them work when there is no python window visible.

I'm sorry that I can't explain it very well, the code is kind of complicated.
But trust me, if it were as easy as you said, it would've been the first thing I tried!

(Feb 18, 2015 09:58 AM)bytecoder Wrote: [\[CITE\]](#) MR.FREE is right.

Rorikstead if you can provide me your whole code, I can look over it and maybe fix things right.

(But I still like it very much, how you solved it.)

Thanks for the help, but I have it all figured it.

MR.FREE - February 18, 2015, 12:05 pm

(Feb 17, 2015 06:46 AM)Rorikstead Wrote: [\[CITE\]](#) The title might be kind of confusing, I apologize for that.

So basically, I'm trying to get the output of a cmd command that's run through a python script.#!

```
Code #!/usr/bin/python
import sys
```

```
def func(x):
    for arg in sys.argv:
        print arg
```

```
func(sys.argv)console
```

```
Code mrfree@orphan:~/Desktop/Scripts % python consoleargs.py butt plug help
```

```
consoleargs.py
```

```
butt
```

```
plug
```

```
help
```

mrfree@orphan:~/Desktop/Scripts %

Im confused as to what your doing tho, if you want the output of a cmd to be stored I would do it in a function using sys instead of a file, unless your taking in databases thru the cmd then I would use a file(no I would use a database module like pySQL or something)

Rorikstead - February 20, 2015, 4:44 am

(Feb 18, 2015 01:05 PM)MR.FREE Wrote: [CITE](#) #!

Code #!/usr/bin/python

import sys

def func(x):

for arg in sys.argv:

print arg

func(sys.argv)console

Code mrfree@orphan:~/Desktop/Scripts % python consoleargs.py butt plug help

consoleargs.py

butt

plug

help

mrfree@orphan:~/Desktop/Scripts %

Im confused as to what your doing tho, if you want the output of a cmd to be stored I would do it in a function using sys instead of a file, unless your taking in databases thru the cmd then I would use a file(no I would use a database module like pySQL or something)

It's more like this.

Say I want to run the command 'ipconfig' like in a cmd window.

I could do os.system('ipconfig') which would work fine.

Now let's say I want to get the output of that command. os.system() will not provide that. And things like subprocess.check_output() only work if the python console is visible, and in my case it is not.

Again, sorry for being confusing. I really wasn't sure how to word my problem.

MR.FREE - February 21, 2015, 11:05 am

Thanks for clarifying. I see what the problem is now, let me know if you find a working solution plz.

Thanks

Edit:

Maybe something like this, we can also set it up so you can view while its running etc...

Code import subprocess

def cmd():

cmd = raw_input("Please Input cmd: ")

subprocess.call("%s > out.txt" % cmd, shell = True)

with open("/usr/home/mrfree/Desktop/Scripts/out.txt","r") as log:

print log.read()

if __name__ == "__main__":

cmd()

If that didn't get it, let me know and I'll try to help you figure it out if you want

Rorikstead - February 22, 2015, 5:35 am

(Feb 21, 2015 12:05 PM)MR.FREE Wrote: [\[CITE\]](#) Thanks for clarifying. I see what the problem is now, let me know if you find a working solution plz. Thanks

Edit:

Maybe something like this, we can also set it up so you can view while its running etc...

Code

```
import subprocess
```

```
def cmd():  
    cmd = raw_input("Please Input cmd: ")  
    subprocess.call("%s > out.txt" % cmd, shell = True)  
    with open("/usr/home/mrfree/Desktop/Scripts/out.txt","r") as log:  
        print log.read()
```

```
if __name__ == "__main__":  
    cmd()
```

If that didn't get it, let me know and I'll try to help you figure it out if you want

Aha! That code is exactly perfect. Thanks for putting your own time into this, your code does exactly what I was looking for!

A very elegant solution indeed. Definitely solves the problem I was having.

I declare this thread, closed!