Run process as admin with subprocess.run in python

Asked 6 years, 6 months ago Modified 1 year, 5 months ago Viewed 41k times

17

Is there a way of passing some `runas=True` arg to a `subprocess.run` function in python? I want to run a process as admin (elevate it). Thanks for answers :)\

EDIT: Using Windows OS.

python windows subprocess admin

Share Edit Follow Flag

edited Jul 28, 2022 at 13:05
mklement0
415k 67 659 846

asked Nov 19, 2017 at 18:23
Jakub Bláha
1,589 7 24 43

4 Answers

Sorted by: Highest score (default)

24

+50

Windows has a command line utility "Run as", which can be used as

```
runas [{/profile | /noprofile}] [/env] [{/netonly | /savecred}] [/smartcard] [/showtrustlevels] [/trustlevel] /user:
<UserAccountName> "<ProgramName> <PathToProgramFile>"
```

for further reference https://technet.microsoft.com/en-us/library/cc771525.aspx

You can use this in code like below

```
import subprocess as sp
```

```
prog = sp.Popen(['runas', '/noprofile', '/user:Administrator',
'NeedsAdminPrivilege.exe'],stdin=sp.PIPE)
prog.stdin.write('password')
prog.communicate()
```

Share Edit Follow Flag

edited Jul 28, 2022 at 16:09
mklement0
415k 67 659 846

answered Nov 20, 2017 at 5:18
pankaj mishra
2,605 3 19 33

2 Note that this only works with the built-in `Administrator` user account, not generally with users that are part of the `Administrators` group. In practice, the built-in `Administrator` account is often disabled for security reasons. – mklement0 Jul 28, 2022 at 13:07

4

There are three approaches:

1. Using `runas` as shown in this answer. The downside of this approach is that it uses the Administrator account instead of the current user's Administrator privileges. This does not work well if you plan to deploy your software to users.
2. Use `ShellExecute` as discussed in this question to start your subprocess. The downside is that you won't be able to work with stdin/stdout/stderr.
3. Use JetBrains' `WinElevator` (signed launcher.exe and elevator.exe are available here). The downside of this approach is that you need to ship two additional ~150kb binaries, the upside is that you can interact with stdin/stdout/stderr.

Share Edit Follow Flag

answered Aug 21, 2022 at 10:51
Maximilian Hils
6,592 3 29 48

3

If you want to run a command as the same user but with admin privilege

Please refer to this solution:

```
os.system(r'''
Powershell -Command "& { Start-Process \"notepad.exe\"
```

```
    -ArgumentList @(\"C:\\Windows\\System32\\drivers\\etc\\hosts\")
    -Verb RunAs } " '''
```

The original answer can be found here https://superuser.com/a/753600/1088510

Share Edit Follow Flag

answered Dec 10, 2021 at 7:11

ZMJ
367 3 12

-3

As others have suggested you can achieve this using powershell. These are my PS functions I use to elevate to Admin:

```
========================== Admin Rights
=======================================
 $PSCommandPath


Path = $args[0]
ity.Principal.WindowsPrincipal]
oal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole]
) { Start-Process powershell.exe "-NoProfile -ExecutionPolicy
cmdPath`"" -Verb RunAs; exit }



WithSTA

Path = $args[0]
ity.Principal.WindowsPrincipal]
oal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole]
) { Start-Process powershell.exe "-sta  -NoProfile -
Bypass -File `"$cmdPath`"" -Verb RunAs; exit }
```

Share Edit Follow Flag

edited Dec 29, 2022 at 6:27

answered Dec 29, 2022 at 6:14

Geordie
2,037 2 27 37

Sorry but the OP was looking for a solution in Python – PKCS12 Jun 7, 2023 at 13:56

@PKCS12 My point was you can call powershell from Python which in turn could run another exe with elevated privileges – Geordie Jun 7, 2023 at 17:14