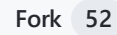




Sign in

M2Team / Privexec Public



[Code](#)
[Issues](#)
[Pull requests](#)
[Actions](#)
[Projects](#)
[Security](#)
[Insights](#)

 master ▼

Go to file



About

Run the program with the specified permission level (C++20 required)

windows



Report repository

Releases 25



5.1.0

Latest



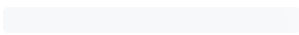
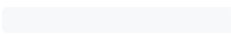




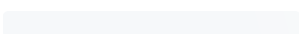
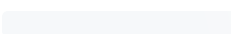

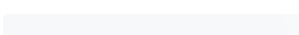
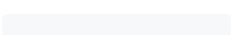



















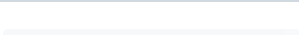
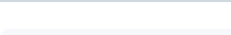












on Apr 3, 2023










+ 24 releases


Packages


No packages published

Contributors 2

			
 .github/workflows			
 .vscode			
 AppExec			
 Privexec			
 cmake/modules			
 config			
 docs			
 include			
 lib			
 resources			
 script			
 test			
 tools			
 vendor			
 wsudo			

 .clang-format		
 .clang_complete		
 .cmake-format.json		
 .gitignore		
 CMakeLists.txt		
 LICENSE		
 ReadMe.md		
 ReadMe.zh-CN.md		
 build.ps1		





Languages




📖 README

📄 MIT license

☰

Privexec

license MIT

 CI passing

downloads@latest 1.5k

downloads 4.3k

link 996.icu


[简体中文](#)

Run the program with the specified permission level

Install

Install Privexec by [baulk](#)

```
baulk install wsudo  
wsudo --version
```



Or you can download it directly, use Explorer or 7z and other tools to extract and then use Privexec, download link:

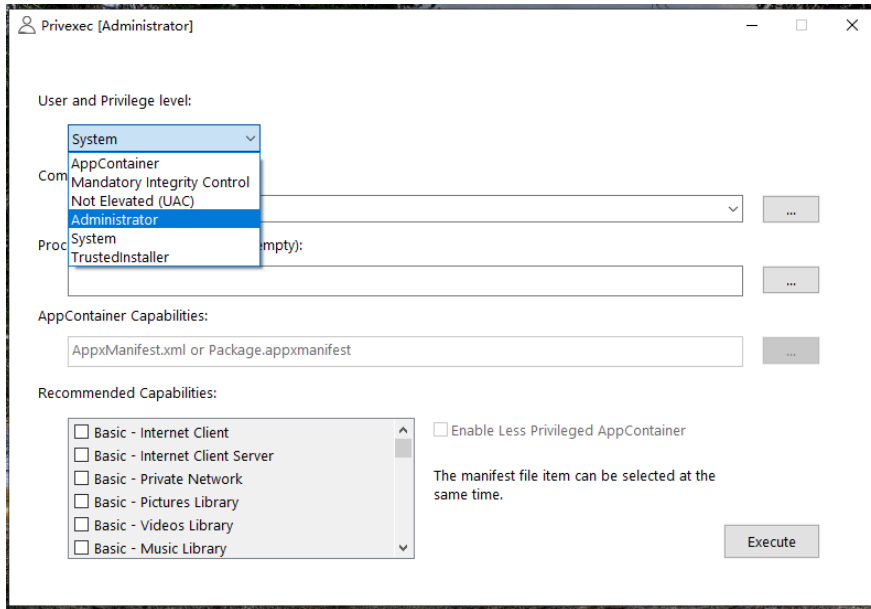
<https://github.com/M2Team/Privexec/releases/latest>

Alias

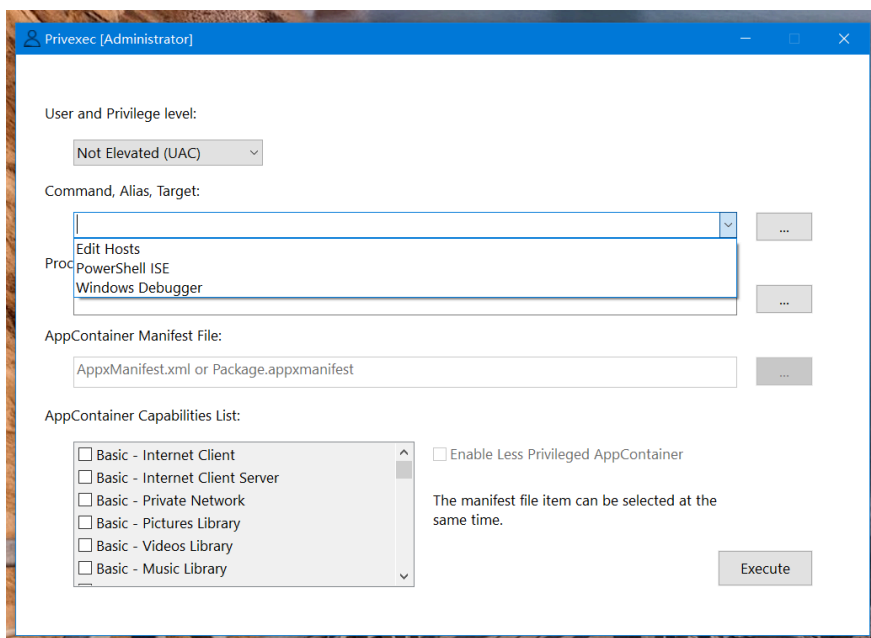
Privexec and wsudo can resolve aliases. In addition, wsudo adds or deletes aliases. It is also a good choice to use vscode to edit `Privexec.json` to modify aliases. When Privexec is installed via baulk, the storage directory of `Privexec.json` is `$BAULK_ROOT/bin/etc`. If Privexec Download and unzip directly, then `Privexec.json` will be in the same directory as `Privexec.exe`.

```
{
  "alias": [
    {
      "description": "Edit Hosts",
      "name": "edit-hosts",
      "target": "Notepad %windir%\\System:
    },
    {
      "description": "Windows Debugger",
      "name": "windbg",
      "target": "\"%ProgramFiles(x86)%\\W:
    }
  ]
}
```

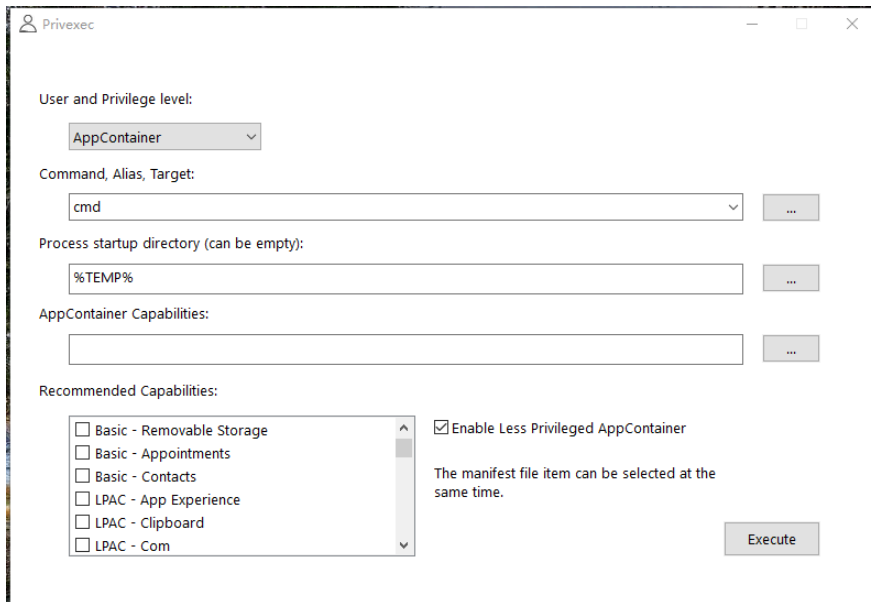
Screenshot



Alias:



AppContainer:



wsudo usage:

```
PS C:\Temp> baulk list wsudo
wsudo/Baulk 2.7.0
PS C:\Temp> wsudo -h
wsudo 2.7 run the program with the specified permissions
usage: wsudo command args...
-v|--version      print version and exit
-h|--help         print help information and exit
-u|--user         run as user (optional), support '-u X', '-u X', '--user=X', '--user X'
                  Supported user categories (Ignore case):
                  AppContainer  MIC      NoElevated
                  Administrator System  TrustedInstaller

-n|--new-console  Starts a separate window to run a specified program or command.
-H|--hide         Hide child process window. not wait. (CREATE_NO_WINDOW)
-w|--wait         Start application and wait for it to terminate.
-V|--verbose      Make the operation more talkative
-x|--appx         AppContainer AppManifest file path
-c|--cmd          Use a working directory to launch the process.
-e|--env          Set Environment Variable.
-l|--lpac         Less Privileged AppContainer mode.
--disable-alias   Disable Privexec alias, By default, if Privexec exists alias, use it.
--appname         Set AppContainer Name

Select user can use the following flags:
|-a AppContainer  |-M Mandatory Integrity Control|-U No Elevated(UAC)|
|-A Administrator|-S System              |-T TrustedInstaller|

Example:
wsudo -A "%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile
wsudo -T cmd
wsudo -U -V --env CURL_SSL_BACKEND=schannel curl --verbose -I https://nghttp2.org

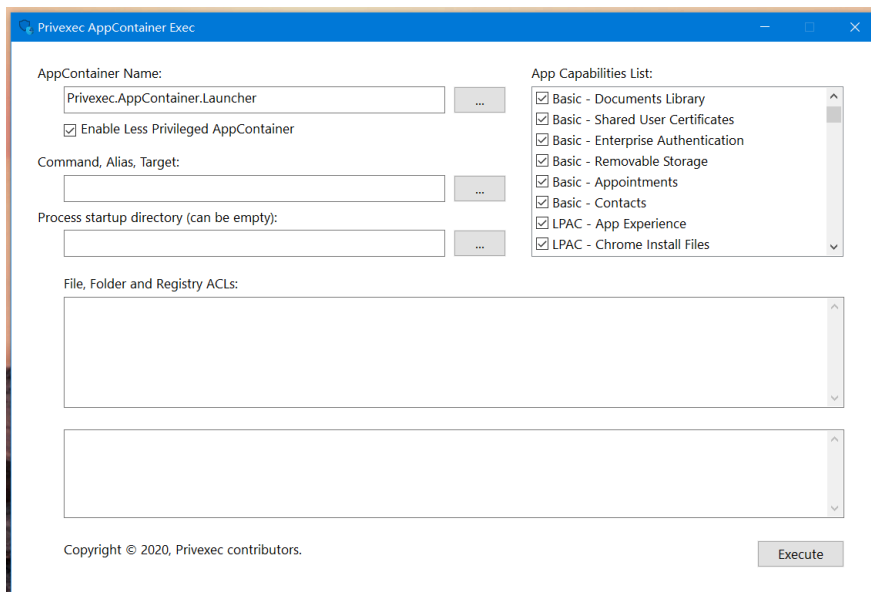
Builtin 'alias' command:
wsudo alias add ehs "notepad %SYSTEMROOT%\System32\drivers\etc\hosts" "Edit Hosts"
wsudo alias delete ehs

PS C:\Temp>
```

wsudo Verbose Mode:

```
PS C:\Temp> wsudo -V -a -n cmd
* App Launcher level: AppContainer
* App full path 'C:\WINDOWS\system32\cmd.exe'
* App real path 'C:\WINDOWS\system32\cmd.exe'
* App real argv0 'C:\WINDOWS\system32\cmd.exe'
* App real argv0 'cmd'
* App subsystem is console, new console window
* App real command 'cmd'
Command: cmd
new appcontainer process is running: 3432
sid: S-1-15-2-2317891757-3286286638-4277663636-3514465956-2601416738-1636989900-602395012
PS C:\Temp>
```

AppContainer Exec



Usage


Privexec is a GUI client. When running as a standard user, you can start the administrator process; when running as an administrator, you can elevate the privileges to `System` or `TrustedInstaller`. It should be noted that `System` or

TrustedInstaller has too many privileges, which can easily damage the system operation. Be careful when using it.

AppExec is a program that starts the AppContainer process. Some developers use this program to study the running details of Windows AppContainer and the vulnerabilities of AppContainer. UWP applications run in the AppContainer container.

wsudo is the console version of Privexec/AppExec. The detailed help is as follows:

wsudo usage:

```
wsudo ❤️ 5.0 run the program with the specified   
usage: wsudo command args...
```

```
-v|--version      print version and exit  
-h|--help         print help information at  
-V|--verbose      Make the operation more v  
-c|--cwd          Use a working directory v  
-e|--env          Use specific environment  
-n|--nui          Starts a separate window  
-H|--hide         Hide child process window  
-w|--wait         Start application and wait  
-u|--user         run as user (optional), v  
                  Supported user categories:  
                  AppContainer MIC  
                  Standard      Administrator  
                  TrustedInstaller
```

```
-x|--appx         AppContainer AppManifest  
-L|--lpac         Less Privileged AppConta  
--disable-alias   Disable Privexec alias, l  
--appid          Set AppContainer ID name  
--retain         Retain AppContainer Prof:
```

Select user can use the following flags:

```
-a|--appcontainer AppContainer  
-M|--mic          Mandatory Integrity Cont  
-B|--basic        Basic execution, permiss  
-U|--standard     Standard user no elevate  
-A|--administrator Administrator  
-S|--system       System  
-T|--ti           TrustedInstaller
```

```

Example:
wsudo -A pwsh -NoProfile
wsudo -T cmd
wsudo -U -V -eCURL_SSL_BACKEND=schannel curl
wsudo -U -V CURL_SSL_BACKEND=schannel curl -...

Builtin 'alias' command:
wsudo alias add ehs "notepad %SYSTEMROOT%/Sy
wsudo alias delete ehs

```

When Privexec, AppExec, wsudo launch commands, the command line and launch directory support deduction via `ExpandEnvironmentString`.

WSUDO Details

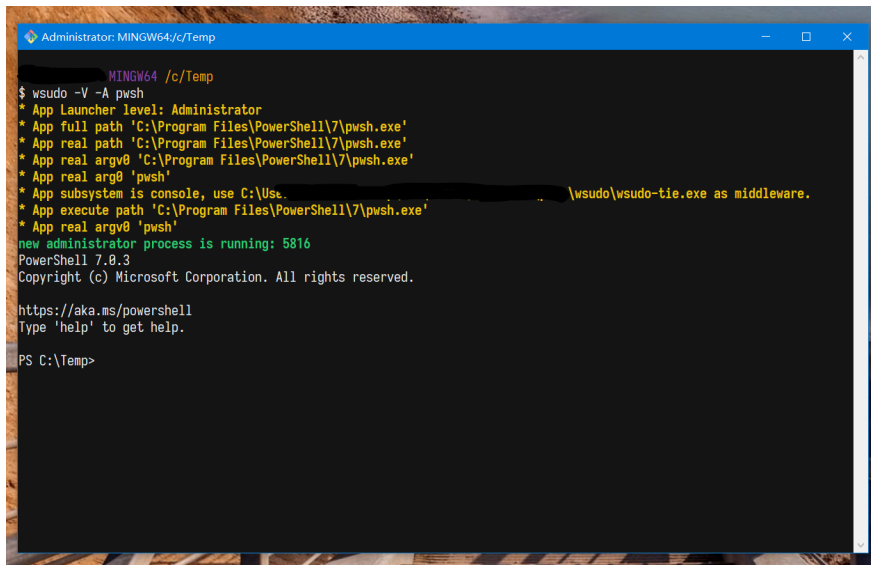
The wsudo visible and wait related parameters are `--hide` `--wait` `--new-console`. The corresponding situation is as follows:

PE Subsystem	No relevant parameters	--new-console	--hide
Windows CUI	wait/Inheritance console	no wait/New console	no wait/No console
Windows GUI	no wait/New UI	no wait/New UI	no wait/No window
Windows CUI <code>-wait</code>	wait/Inheritance console	wait/New console	wait/No console
Windows GUI <code>-wait</code>	wait/New UI	wait/New UI	wait/No window

When wsudo starts the administrator process as a standard user, if it is currently running in the console, it supports inheriting the console window. If it is not running in the

console, it can do nothing. The newer Cygwin currently supports the newer Windows 10 ConPty starts the console, so it can inherit the console window, which is the terminal. The picture below is the proof.

wsudo exec administrator process under mintty (Turn on ConPty):



```
Administrator: MINGW64/c/Temp
MINGW64 /c/Temp
$ wsudo -V -A pwsh
* App Launcher level: Administrator
* App full path 'C:\Program Files\PowerShell\7\pwsh.exe'
* App real path 'C:\Program Files\PowerShell\7\pwsh.exe'
* App real argv0 'C:\Program Files\PowerShell\7\pwsh.exe'
* App real argv0 'pwsh'
* App subsystem is console, use C:\Use... \wsudo\wsudo-tie.exe as middleware.
* App execute path 'C:\Program Files\PowerShell\7\pwsh.exe'
* App real argv0 'pwsh'
new administrator process is running: 5816
PowerShell 7.0.3
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/powershell
Type 'help' to get help.

PS C:\Temp>
```

WSUDO Environment

wsudo support `-e/--env` to set environment. such as:

[Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

 © 2024 GitHub, Inc.