M2Team / **Privexec** Public

master | Go to file | Code

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

About

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

windows

Readme
MIT license
Activity
Custom properties
325 stars
19 watching
52 forks
Report repository

Releases 25

5.1.0 Latest
on Apr 3, 2023

+ 24 releases

Packages

No packages published

Contributors 2

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

README | MIT license

Privexec

简体中文

Run the program with the specified permission level

Install

Install Privexec by baulk

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

Languages

C++ 97.9% Other 2.1%

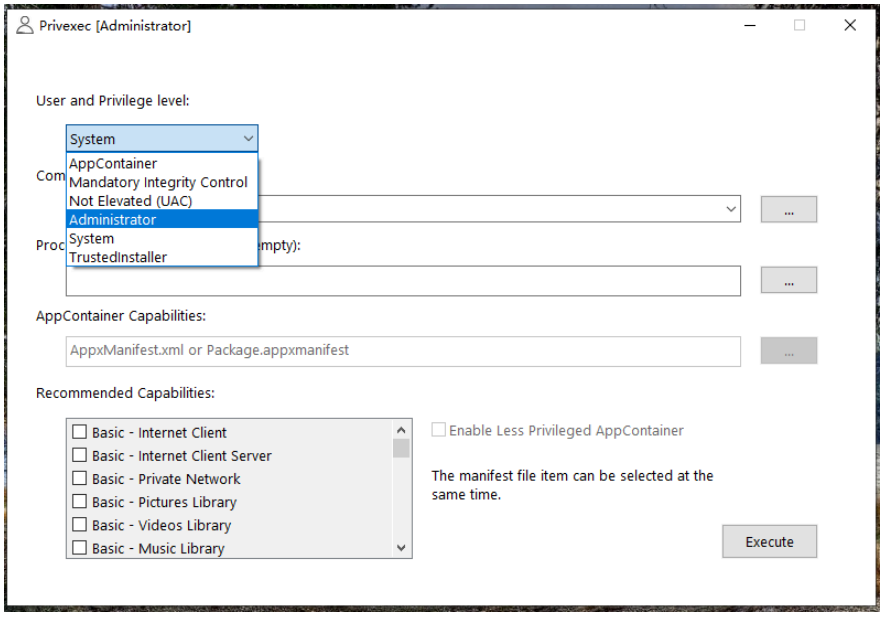
Or you can download it directly, use Exeplorer or 7z and other tools to extract and then use Privexec, download link: https://github.com/M2Team/Privexec/releases/latest

Alias

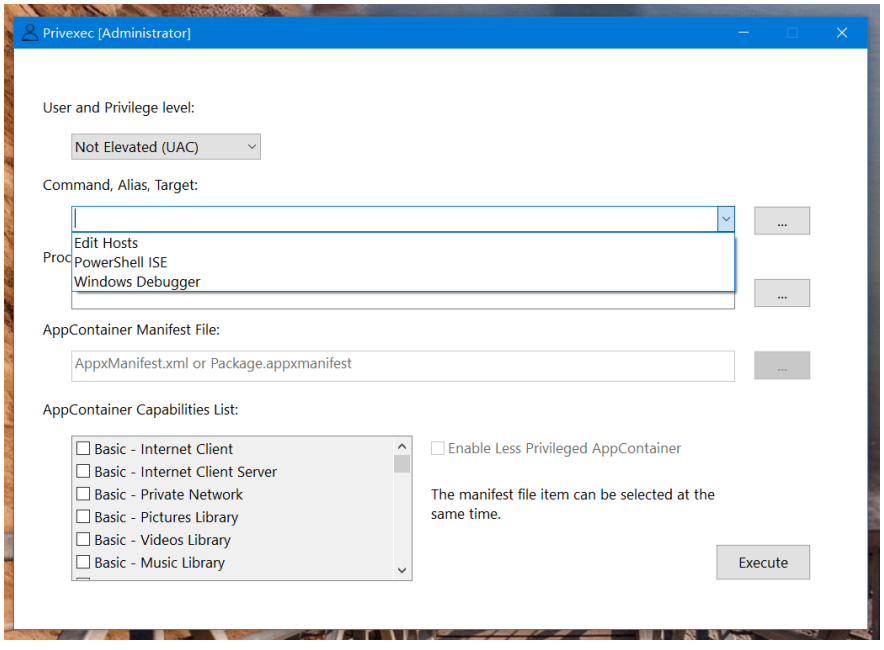
Privexec and wsudo can resolve aliases. In addition, wsudo adds or deletes aliases. It is also a good choice to use vscode to edit `Privexec.json` to modify aliases. When Privexec is installed via baulk, the storage directory of `Privexec.json` is `$BAULK_ROOT/bin/etc`. If Privexec Download and unzip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

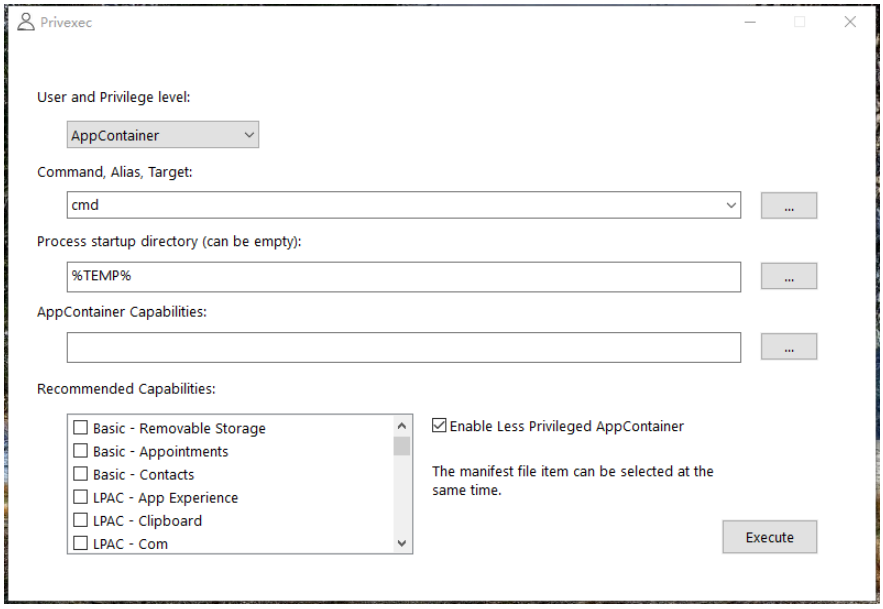
Screenshot



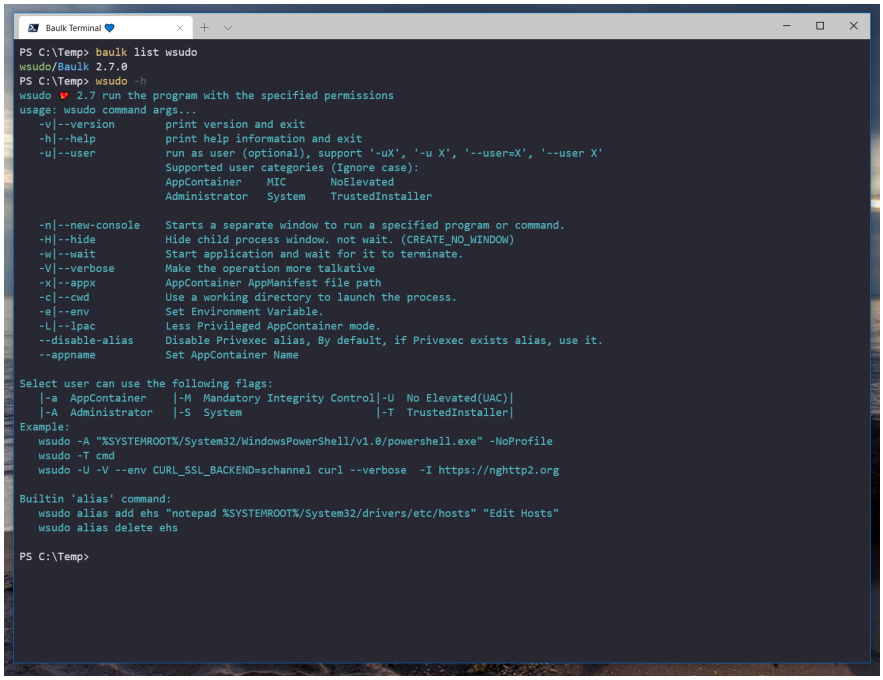
Alias:



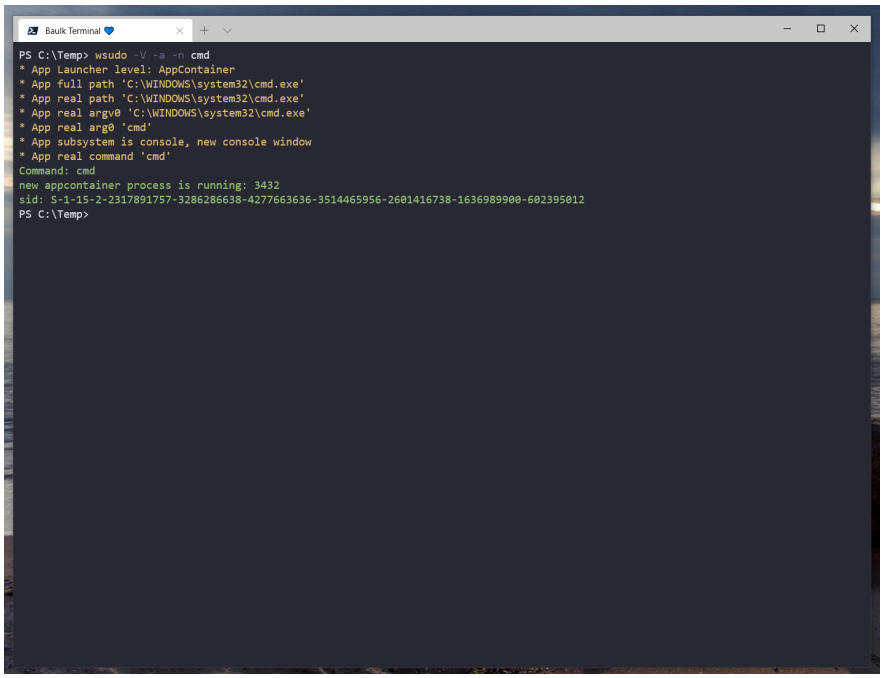
AppContainer:



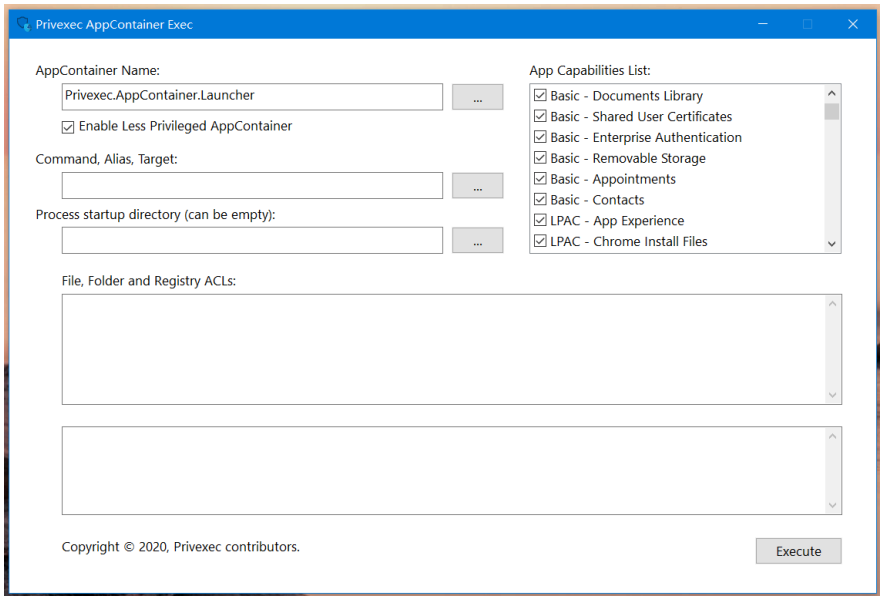
wsudo usage:



wsudo Verbose Mode:



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 AppContaner. 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 a
   -V|--verbose        Make the operation more
   -c|--cwd            Use a working directory
   -e|--env            Use specific environment
   -n|--nui            Starts a separate window
   -H|--hide           Hide child process windo
   -w|--wait           Start application and wa
   -u|--user           run as user (optional), s
                       Supported user categorie
                       AppContainer  MIC
                       Standard      Administra
                       TrustedInstaller

   -x|--appx           AppContainer AppManifest
   -L|--lpac           Less Privileged AppConta
   --disable-alias     Disable Privexec alias, I
   --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

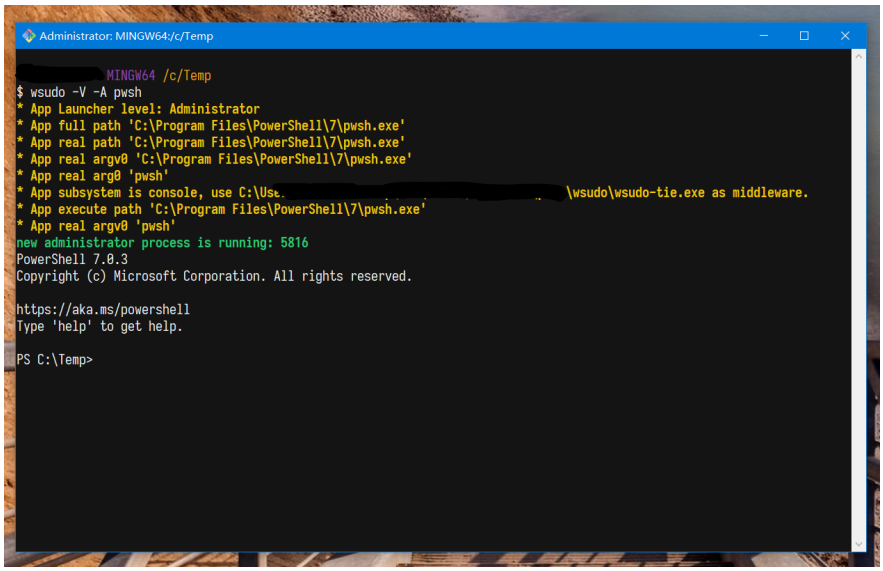
The wsudo visible and wait related parameters are `--hide` `--wait` `--new-console`. The corresponding situation is as follows:

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 `-wait`	wait/Inheritance console	wait/New console	wait/No console
Windows GUI `-wait`	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):



WSUDO Environment

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