 fcharlie	Fix bela redefined STT_GNU_IFUNC ✓	83c71a0 · 2 months ago	🕒 550 Commits
📁 .github/workflows	switch to actions/checkout v4	3 months ago	
📁 .vscode	Sync bela and wsudo add --retain	3 months ago	
📁 AppExec	Fixed the portable version loading confi...	last year	
📁 Privexec	Fixed the portable version loading confi...	last year	
📁 cmake/modules	cleanup version and update AppExec/P...	4 years ago	
📁 config	Privexec 3.0 release	4 years ago	
📁 docs	wsudo: 5.0.1 some behavioral correctio...	2 years ago	
📁 include	json.hpp 3.11.3	last year	
📁 lib	Sync bela and wsudo add --retain	3 months ago	
📁 resources	wsudo icon	5 years ago	
📁 script	Rewrite shell menu script and replace w...	4 years ago	
📁 test	sync bela	4 years ago	
📁 tools	Release 5.0.3 Fix duplicate capabilities	last year	
📁 vendor	Fix bela redefined STT_GNU_IFUNC	2 months ago	
📁 wsudo	Sync bela and wsudo add --retain	3 months ago	
📄 .clang-format	Limit line length to 120 columns	4 years ago	
📄 .clang_complete	wsudo use libExec and wsudo-tie supp...	4 years ago	
📄 .cmake-format.json	update cmake LTO option	4 years ago	
📄 .gitignore	Windows ARM64 constexpr error	4 years ago	
📄 CMakeLists.txt	Update cmake	3 months ago	
📄 LICENSE	Copyright © 2022	3 years ago	
📄 ReadMe.md	docs: add --retain	3 months ago	
📄 ReadMe.zh-CN.md	docs: add --retain	3 months ago	
📄 build.ps1	build.ps1: rename Exec to Execute	3 years ago	

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

-  fcharlie Force Charlie
-  WildByDesign WildByDesign

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 Exeplorer or 7z and other tools to extract and then use Privexec, download link: <https://github.com/M2Team/Privexec/releases/latest>

Alias

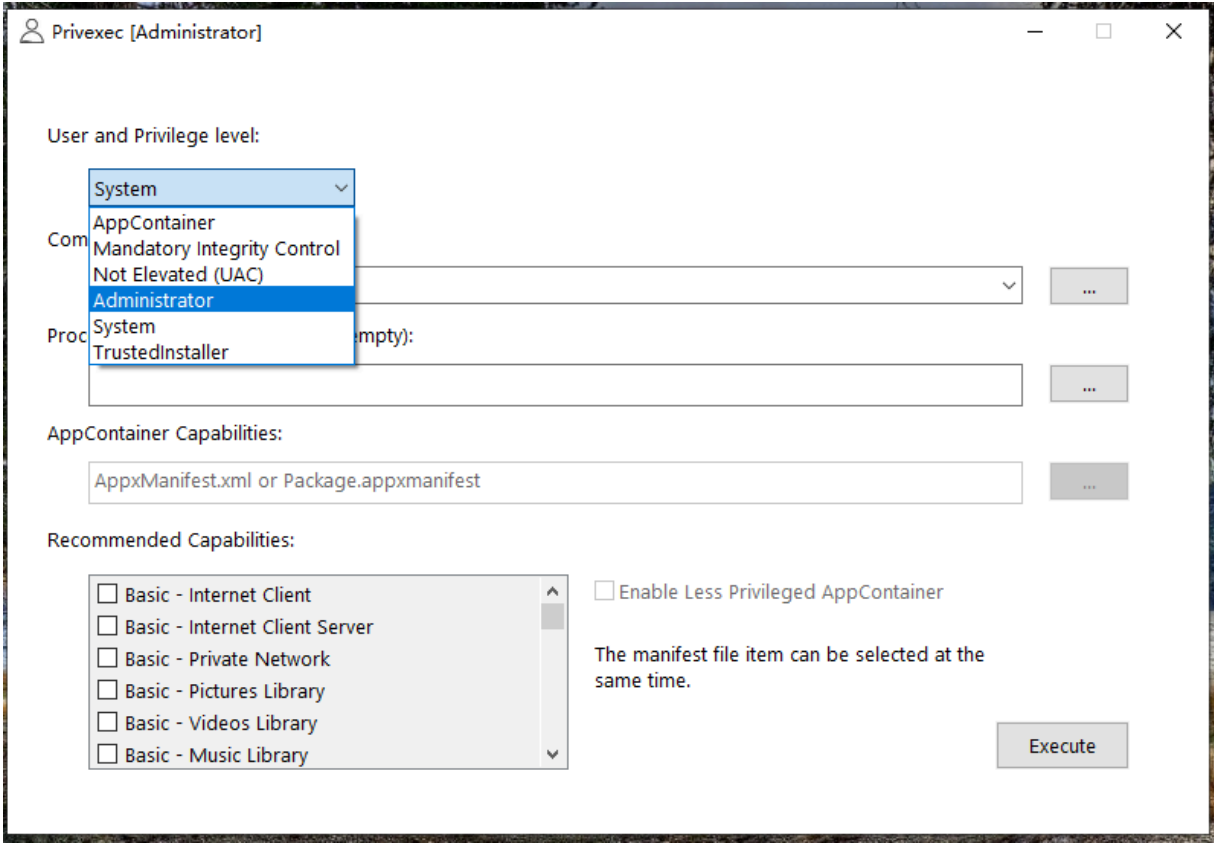
Privexec and wsudo can resolve aliases. In addition, wsudo adds or deletes aliases. It is also a good choice to use vscode to edit `Privexec.json` to modify aliases. When Privexec is installed via baulk, the storage directory of `Privexec.json` is `$BAULK_ROOT/bin/etc` . If Privexec Download and unzip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

```

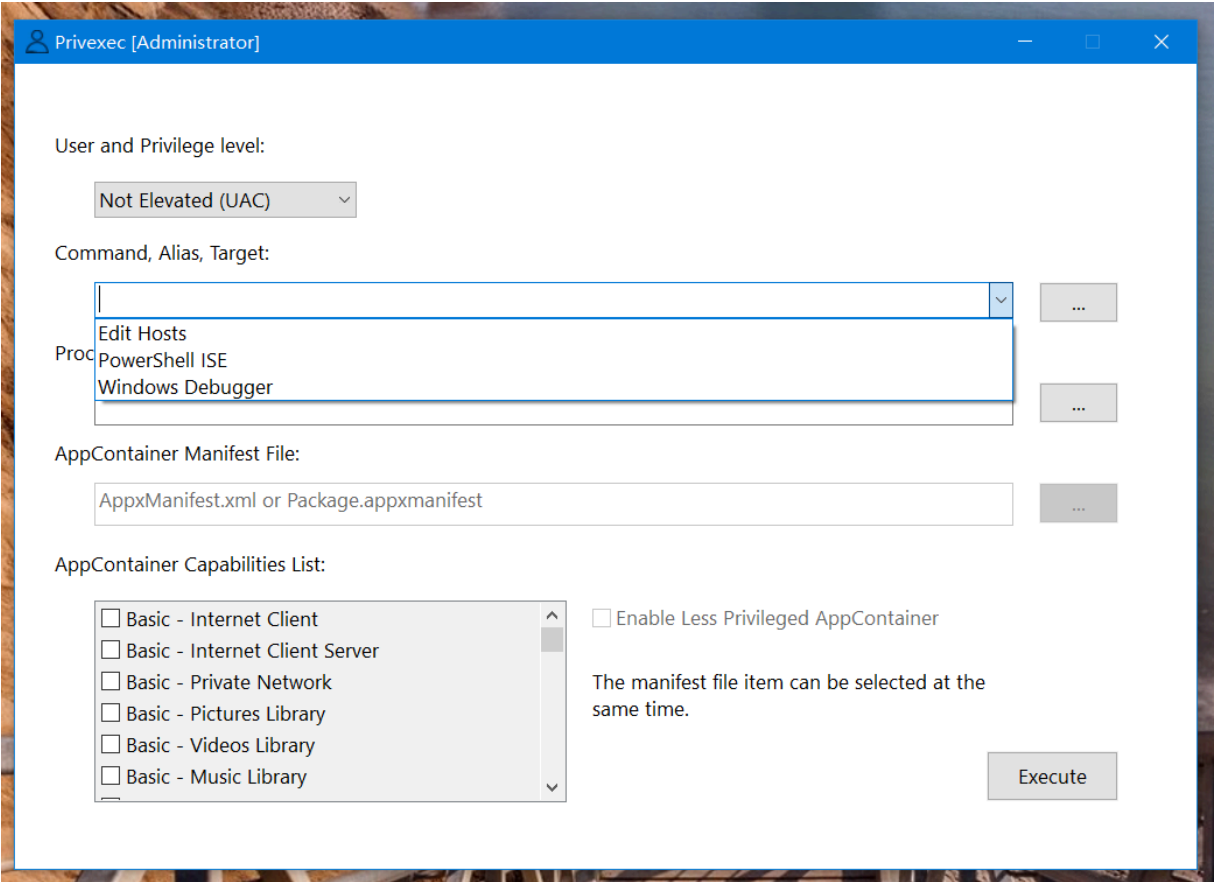
{
  "alias": [
    {
      "description": "Edit Hosts",
      "name": "edit-hosts",
      "target": "Notepad %windir%\\System32\\Drivers\\etc\\hos
    },
    {
      "description": "Windows Debugger",
      "name": "windbg",
      "target": "\"%ProgramFiles(x86)%\\Windows Kits\\10\\Debu
    }
  ]
}

```

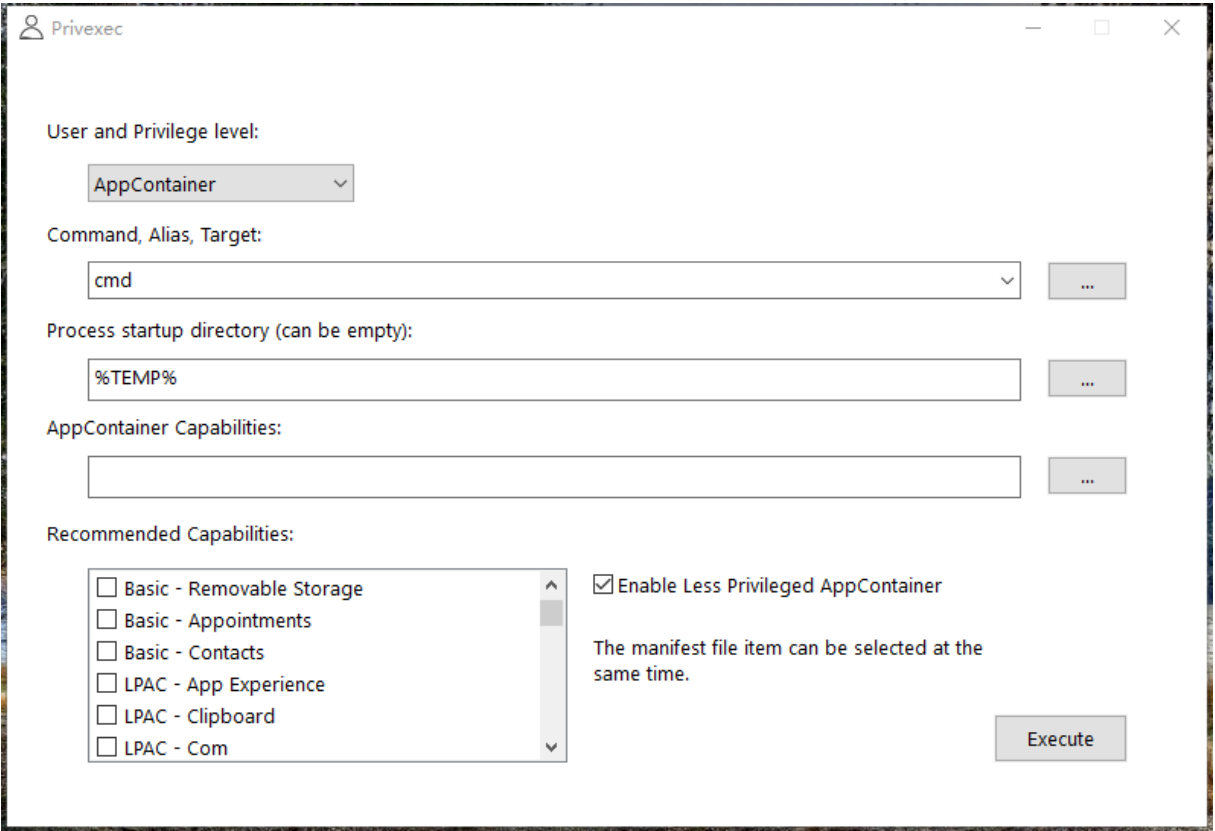
Screenshot



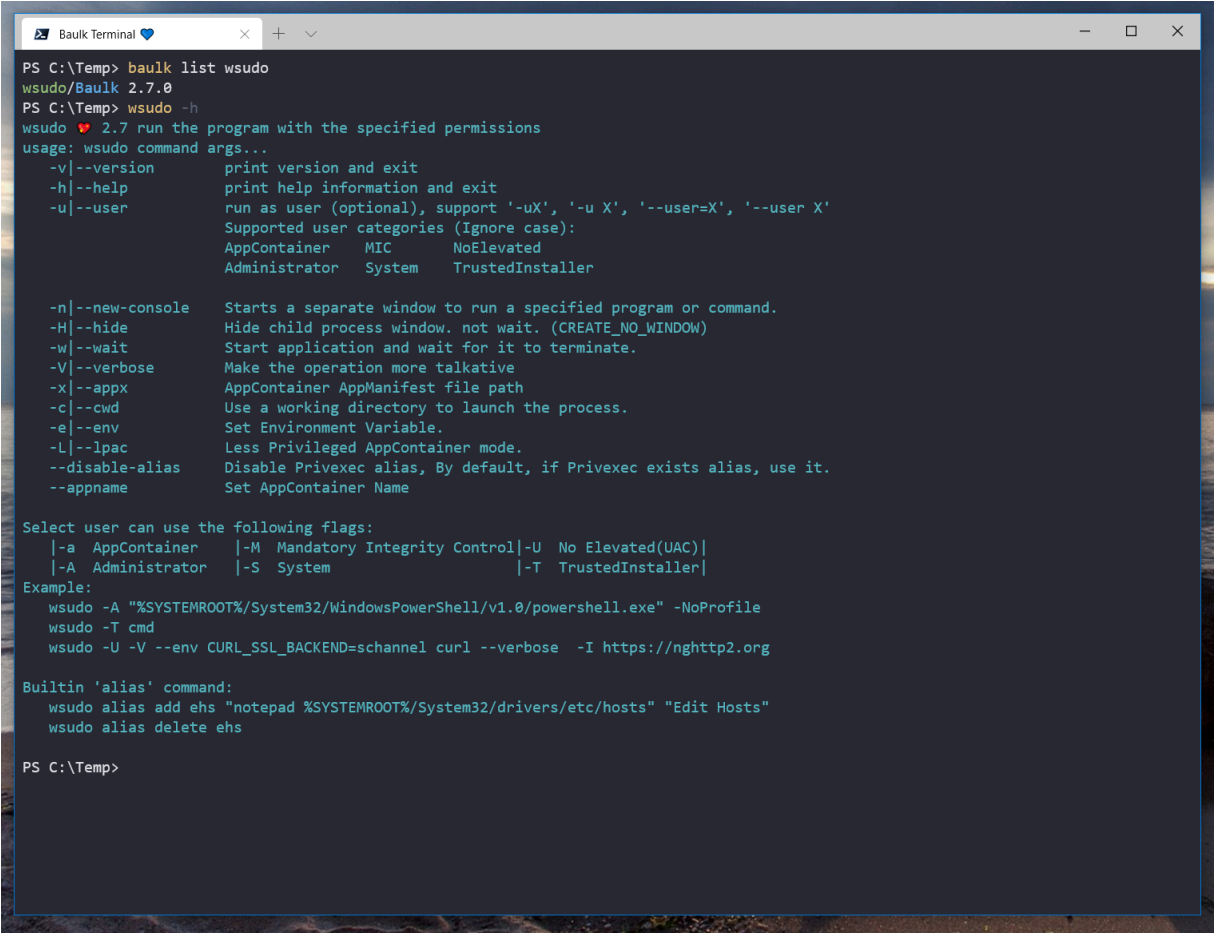
Alias:



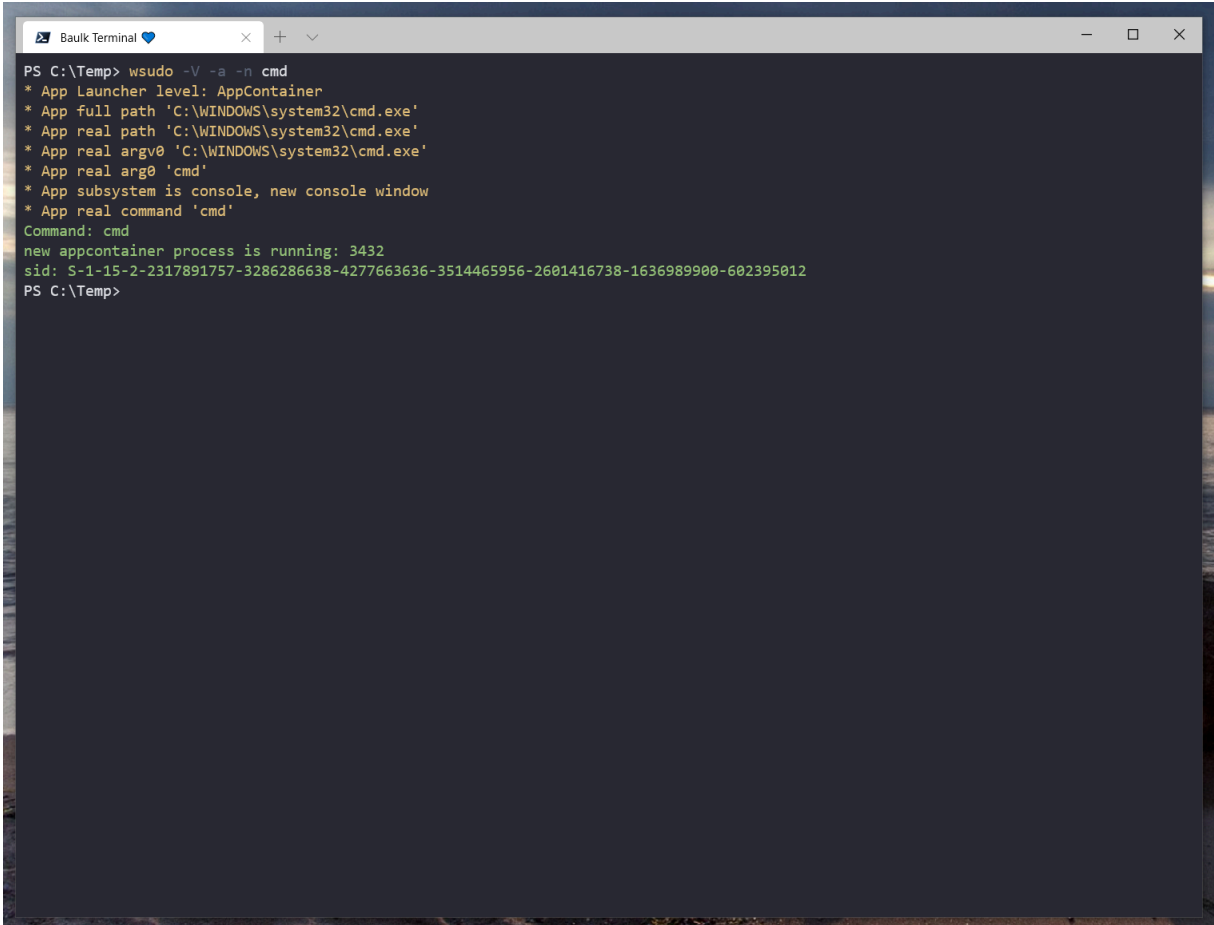
AppContainer:



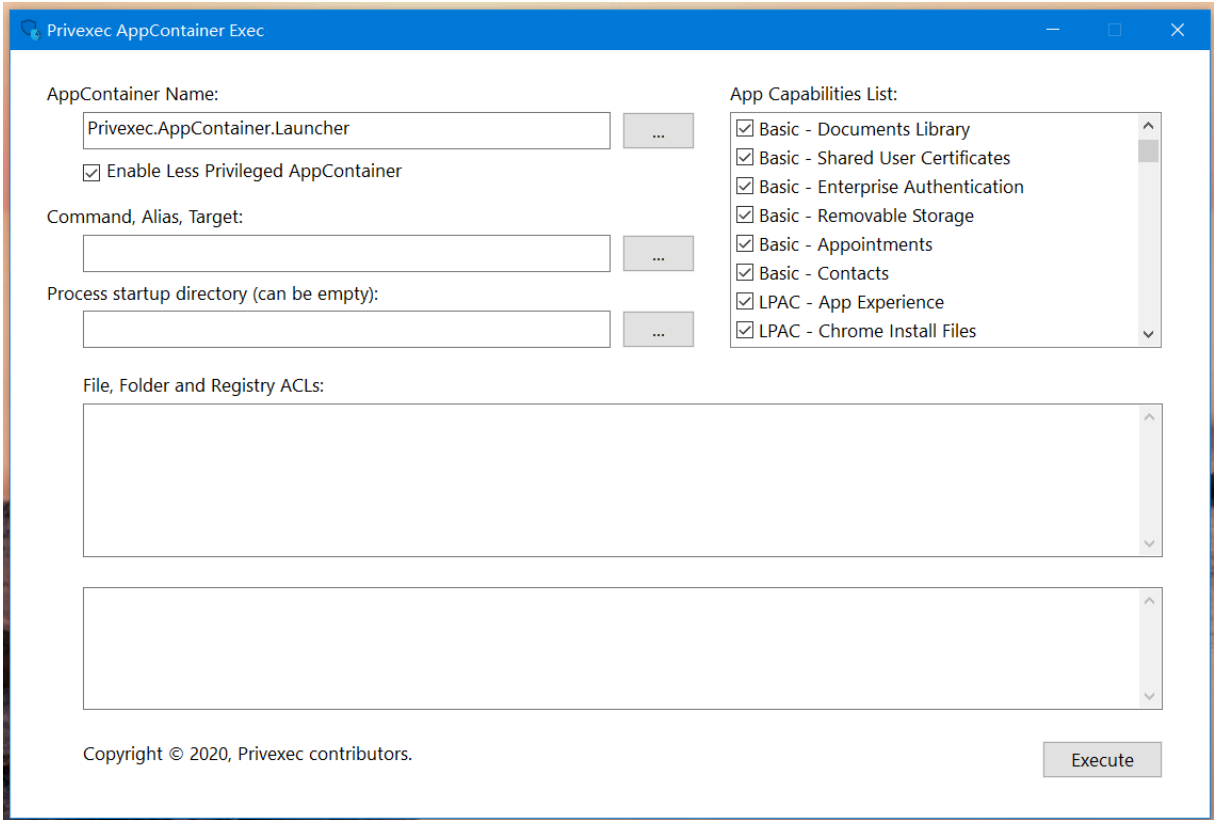
wsudo usage:



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 permissions
usage: wsudo command args...
-v|--version          print version and exit
-h|--help             print help information and exit
-V|--verbose          Make the operation more talkative
-c|--cwd              Use a working directory to launch the process
-e|--env              Use specific environment variables to start cl
```

```

-n|--nui           Starts a separate window to run a specified pr
-H|--hide          Hide child process window. not wait. (CREATE_I
-w|--wait          Start application and wait for it to terminat
-u|--user           run as user (optional), support '-uX', '-u X'
                    Supported user categories (Ignore case):
                    AppContainer  MIC              Basic
                    Standard      Administrator  System
                    TrustedInstaller

-x|--appx           AppContainer AppManifest file path
-L|--lpac           Less Privileged AppContainer mode.
--disable-alias     Disable Privexec alias, By default, if Privex
--appid             Set AppContainer ID name (compatible --appnam
--retain            Retain AppContainer Profile (experimental)

Select user can use the following flags:
-a|--appcontainer   AppContainer
-M|--mic            Mandatory Integrity Control
-B|--basic           Basic execution, permission inheritance mode
-U|--standard       Standard user no elevated (UAC)
-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 --verbose -I https:
wsudo -U -V CURL_SSL_BACKEND=schannel curl --verbose -I https://

Builtin 'alias' command:
wsudo alias add ehs "notepad %SYSTEMROOT%/System32/drivers/etc/ho
wsudo alias delete ehs

```

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

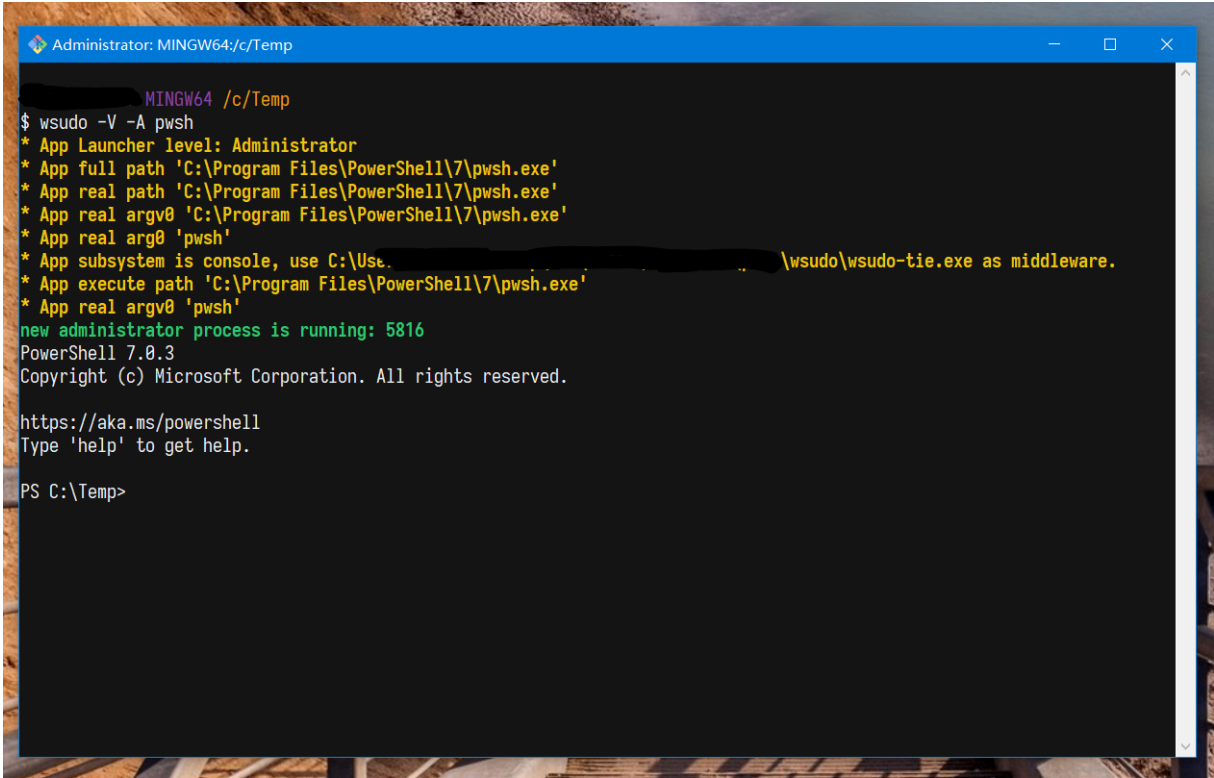
WSUDO Details

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:

```

::curl must enabled multiple SSL backends.
wsudo  -U -V --env CURL_SSL_BACKEND=schannel curl --verbose  -I http:
```

The environment variables will be deduced according to the Batch mechanism, that is, the environment variables are marked with matching `%` .

```

# powershell
.\bin\wsudo.exe -n -e 'PATH=%PATH%;%TEMP%' -U cmd
```

```

::cmd
wsudo  -e "PATH=%PATH%;%TEMP%" -n -U cmd
```

Changelog