WinRun v3.0.8 -- Windows "Run" Utility
C.L.Distefano rev. 2025-06-22
https://ammaze.net/xywwweb/dls/WinRun30.zip
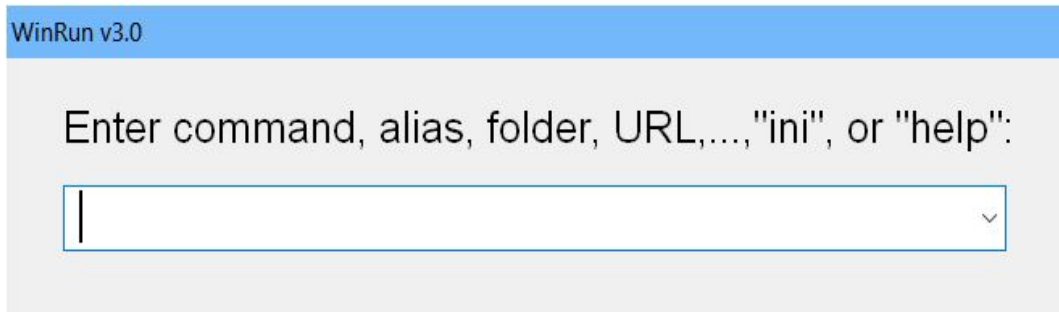
WinRun is a command-line utility, similar to the Windows Run dialog but more versatile. It allows you to use the keyboard to:

- Launch programs, open folders, and visit websites
- Switch focus to any open window by entering a fragment of the window title or process name
- Minimize, maximize, restore, or close the current window
- Send keystrokes to running programs
- Run single commands or open a cmd.exe window
- Create and run batch files on the WinRun command-line
- Create and run AutoIt macros on the fly
- Save and run complex commands as "aliases" that you create yourself
- Run any WinRun command from the Windows (cmd.exe) command line

If you're a command-oriented user keen on escaping the tyranny of the mouse, give WinRun a try.

WinRun and its related executables were created with with the Windows scripting language AutoIt, version 3.3.14.5 (https://www.autoitscript.com/). WinRun also makes use of Wengier Wu's freeware start-command substitute start.exe, included in his vDosPlus DOS emulator (http://www.vdosplus.org).

Included Files

```
WinRun.exe ............. The program
WinRunSetHotkey.exe .... Set a hotkey to launch WinRun
WinRunClearHotkey.exe .. Clear the WinRun hotkey
WinRun.ini ........... Settings and alias definitions
WinRun.pdf ........... This documentation
WR.bat ............... Batch file for running WinRun commands
                           from the Windows command line (created the
                           first time you run WinRun.exe)
```

Quick-Start Instructions

- Unzip `WinRun30.zip` (all files) into a directory of your choosing.
- Run `WinRunSetHotkey.exe`; this sets <F5> as the hotkey to call up the WinRun dialog. If <F5> is unavailable or you prefer to use a different key, open `WinRun.ini` in a text editor and, in the **[Settings]** section, modify the value of "WRhotkey". Shifted keys are named in the format "!a" (Alt+A), "^b" (Ctrl+B), "+c" (Shift+C), "#d" (Winkey+D), etc.; function keys are designated in curly braces as "{f1}"..."{f12}", Alt+F1 is "!{f1}", and so forth. For example, to set Ctrl+F5 as the hotkey, write:

WRhotkey=^{f5}

  - For a full listing of key names, see https://www.autoitscript.com/autoit3/docs/appendis/SendKeys.htm
- Press <F5> or your chosen hotkey to call up the WinRun command line. Try entering commands, folder names, or URLs as you would in the Windows Run dialog.
  - <Up>|<Down> arrow keys scroll through previous commands.
  - <F4> lists all previous commands.
- Enter `ini` in the WinRun dialog to review/edit `WinRun.ini`. The **[Settings]** section includes *x,y* coordinates to position the WinRun dialog on the screen. The **[Aliases]** section contains examples of customized commands available to WinRun users. Try some of these – and see below for instructions on creating your own aliases.
- Enter `help` in WinRun to review this documentation.
- Run `WinRunClearHotkey.exe` to release the hotkey.

Basic Commands

Launch an app:
> `firefox.exe`

If a folder or file name contains spaces, enclose it in double quotes:

> `"C:\Program Files (x86)\Windows Media Player\wmplayer.exe"`

Open a folder in Windows Explorer:
> `C:\Windows\System32`

Visit a website:
> [`https://www.google.com`](https://www.google.com)

Tips:

- If the program or folder does not open with any of the above types of commands, WinRun provides an alternate way to launch, by prefixing the command with "`start `".

- The pulldown menu in the WinRun dialog displays the command history. Down arrow|Up arrow scrolls through the history. Command history is stored in `WinRunCm.txt`, located in the directory with `WinRun.exe`.

Special-Purpose Commands

Issue a single DOS (cmd.exe) command:
> `cmd dir "C:\Program Files (x86)\Celestia\*.exe"`

Note: WinRun inserts a pause to allow viewing of command output

Issue a DOS command and keep the cmd.exe window open:
> `kmd help findstr|more`

Tip: `cmd` or `kmd` (no argument) opens a cmd.exe window.

Create and run batch files "on the fly" (use "\n" for the newline character):
> `bat @echo off\necho This is a short batch file!\npause\n exit\n`

DuckDuckGo|Google searches:
> `ddg Babe Ruth bows out`
> `ggl daphne mezereum invasive`

Switch focus to an open window identified by all or part of the window title or process name:

> `sw thunderbird`

Switch back to the window that had focus when WinRun was launched:

> `sw back`

See also Rule 3 under Aliases, below.

Run an AutoIt script (.au3 or .a3x extension):

> `au3 MyScript.au3 [args]`

Send keystrokes to the current window; command starts with `":"`. For example, to send Ctrl+F5:

> `:^{f5}`

Note: Parentheses may be used instead of curly braces in key names:

> `:^(f5)`

Run lines of AutoIt code "on the fly" (use "\n" for the newline character).
For example, switch to Notepad if open, otherwise start Notepad:

> `au Opt("WinTitleMatchMode", -2)\n`
> `If ProcessExists("notepad.exe") Then\n`
> `WinActivate("notepad")\nElse\nRun("notepad.exe")\nEndIf\n`

Note: The above would be entered into the WinRun dialog as one unbroken line.

Insert a pause (milliseconds) in a string of commands (minimum 20 ms):

> `pause n`

or

> `wait n`

View/Edit `WinRun.ini`:

> `ini`

View this help documentation:

> `help`

Window Management

The following commands are established as aliases in `WinRun.ini`:

Minimize the active window:

> `m`

Minimize all windows:
> `mall`

Maximize the active window:
> `mm`

Restore the active window:
> `r`

Restore all windows:
> `rall`

Close the active window:
> `c`

Close the active browser tab:
> `cw`

Move a window:
> `mw <title>,x,y`

Set topmost flag for active window:
> `ontop`

Clear topmost flag for active window:
> `notontop`

View window information (PID, hWnd, Title, Position, Size):
> `winfo [window_title|substring]`
(default = active window)

Window Position and Size

The following commands are internal to `WinRun.exe` and operate on the **active window** only:

Move window n pixels to the left or right:
> `left n`
> `right n`

Move window n pixels up or down:
> `up n`
> `down n`

Increase | Decrease window size by n **percent**:
> `more n`
> `less n`

Increase | Decrease window width by n pixels:

> `wmore` *n* [or: `xmore` *n*]

> `wless` *n* [or: `xless` *n*]

Increase|Decrease window height by *n* pixels:

> `hmore` *n* [or: `ymore` *n*]

> `hless` *n* [or: `yless` *n*]

System Control

The following self-explanatory commands are established as aliases in
`WinRun.ini`:

> `logoff`

> `forcelogoffg`

> `shutdown`

> `forceshutdown`

> `reboot`

> `forcereboot`

> `powerdown`

> `forcepowerdown`

> `standby`

> `sleep`

> `hibernate`

Chained Commands

WinRun commands can be strung together, separated by semicolons (or
semicolon+space). For example, to switch to Notepad, minimize it, then switch to
Firefox:

> `sw notepad; m; sw firefox`

Or, switch to Chrome and open a new tab:

> `sw chrome; :^t`

Move and resize the active window:

> `left 25; up 20; wmore 40`

Aliases

You can define your own commands ("aliases") in `WinRun.ini`. Each alias
occupies a separate line in the `[Aliases]` section of `WinRun.ini`, and takes the

form "`alias=command[; command; …]`", where "`command`" is any valid WinRun statement. For example:

```
edge=msedge.exe
```

creates a (case-*in*sensitive) alias to launch the Microsoft Edge browser. As a more elaborate example, here is an alias to turn the "Switch to|Run Notepad" script above into an alias called "notepad":

```
notepad=au Opt("WinTitleMatchMode", -2)\nIf
ProcessExists("notepad.exe") Then\nWinActivate("notepad")
\nElse\nRun("notepad.exe")\nEndIf\n
```

Rules:

1. The following WinRun reserved keywords may not be used as alias names:

```
au, au3, bat, cmd, ddg, down, ggl, help, hless, hmore, ini, kmd,
left, less, more, right, start, sw, up, wait, wless, wmore, xless,
xmore, yless, ymore
```

2. An alias definition may include the symbol `%*` to indicate replaceable parameter(s), i.e., the alias accepts one or more arguments. When the alias is invoked in WinRun, `%*` will be replaced by arguments, if any, supplied in the command. For example, the following definition supports commands such as `oed mayhem` and `oed /x`:

```
oed=oed.bat %*
```

3. An alias definition may call another alias. For demonstration purposes, the following uses the `clear` (clear the WinRun hotkey) and `set` (set the hotkey) aliases established in `WinRun.ini`:

```
resethotkey=clear;wait 1500;:{enter};set;wait 1500;
:{enter}
```

4. The string `sw back` can be used as the final command in an alias definition to switch focus back to the window that had focus immediately befor+e the alias was executed. For example, to switch focus to Chrome, open a new tab, then return to the previous window:

```
sw chrome; :^t; sw back
```

5. To promote portability of `WinRun.ini`, the string `[IniDir]` can be used in alias definitions as a stand-in for the WinRun program directory. For example:

```
clear=[IniDir]\WinRunClearHotkey.exe
```

6. Maximum line length in `WinRun.ini` is 255 characters. For longer aliases, numbered keyname stems will allow the definition to span multiple lines, as in the following alias `checknet`, a batch file that checks for an active internet connection:

```
checknet.1=bat @echo off\necho Checking connection, please wait ... \n
checknet.2=ping 8.8.8.8 | find /i "reply from " > nul\n
checknet.3=if errorlevel 0 (\necho You have an active conn
checknet.4=ection to the internet\n) else (\n
checknet.5=echo You are NOT connected to the internet!)\n
checknet.6=echo/&pause\nexit\n
```

See `WinRun.ini` for examples of useful aliases and then -- create your own!

Command-Line Usage

WinRun commands, including aliases, can be issued from the Windows (cmd.exe) command line, as an argument to `WinRun.exe`. The first time you run `WinRun.exe`, it creates `WR.bat`, which can be used to issue WinRun commands (including aliases) with less typing. For ease of use, copy `WR.bat` to a directory in the Windows path. The usage is:

```
wr <command|alias>[; <command|alias>;  ... ]
```

Freeware/Disclaimer

`WinRun.exe` and its associated programs are freeware. Use them at your own risk; there are no warranties of any kind.

Appendix: Revision History

Appendix

Revision History

2025-06-22   Documentation of `sleep` command (established in `winrun.ini`) was inadvertently omitted from this help file: fixed

2025-01-19   In special-purpose command `cmd`, improved handling of multiple Windows commands concatenated with `&`

2024-07-05   Minor adjustment to WinRun input box

2024-06-28   Before switching focus to a window, reset it to a visible state (in case it was hidden)

2024-06-15   Technical fix to undocumented command `CM` (send command keystrokes to a running XyWrite session) ensures that characters `!`, `#`, `+`, `^`, and `{tab}` are transmitted intact

2024-06-05   Built-in `sleep` command renamed to `pause`; synonym `wait` is unchanged

2024-03-08   WinRun not responding to setting MouseMove = 1 (move mouse pointer out of the way): fixed. If enabled, mouse pointer is moved to upper left corner of screen (position 0,0) when command is executed

2024-03-30   Added built-in `sleep` command (synonym for `wait`) to insert milliseconds pause in a string of WinRun commands