

"Register" an .exe so you can run it from any command line in Windows



How can you make a .exe file accessible from any location in the Windows command window? Is there some registry entry that has to be entered?

128

windows



27

edited Jun 13 '13 at 1:48

asked Jan 27 '11 at 21:59



HK1

7,868

10

55

93

6 [superuser.com](#) – David Heffernan Jan 27 '11 at 22:27

15 Answers



You need to make sure that the exe is in a folder that's on the `PATH` environment variable.

78

You can do this by either installing it into a folder that's already on the `PATH` or by adding your folder to the `PATH`.



You can have your installer do this - but you will need to restart the machine to make sure it gets picked up.



answered Jan 27 '11 at 22:02



ChrisF ♦

117k

25

223

295

This answer is almost always the wrong one. At the very least it is incomplete. Specifically, this approach works fine for INSTALLING, but not good for uninstalling. Most uninstallers will WHIPE out the entire `PATH` variable when uninstalling the application associated with this. Therefor, if you write an install action to add the dir to `PATH`, you should make it a CUSTOM install action that isn't automatically uninstalled by the generic remove package.

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up


OR SIGN IN WITH



Google

Facebook

VARIABLES), it seems prudent to explain it. If one follows the directions exactly, they will always run into this issue, which means, the solution should probably always include that caveat. No install packages take care of installing and uninstalling sections of Registry entries or ENV variables that I know of, although please let me know if there is some I don't know of. – [Kim Gentes](#) Mar 6 '15 at 21:40

Instead of "installing" the environmental use a batch file with the SETX command: SETX PATH "C:\Windows" ----- and youre done. – [Stavm](#) Mar 10 '15 at 17:47 

- 3 SETX is also dangerous because it has a much lower length limit than the PATH variable itself, and will truncate instead of failing. – [Ben Voigt](#) Oct 1 '15 at 17:37

You can add the following registry key:

43

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\myexe.exe

In this key, add the default string value containing the path to the exe file.

answered Jan 27 '11 at 22:36





[Andreas Rejbrand](#)

73.8k 6 219 308

- 10 This works for Windows Shell, but not for command prompt. – [TLama](#) Nov 28 '14 at 0:02

This works in command line in Windows 10 – [sherdim](#) Sep 18 '18 at 11:24

- 1 @sherdim: I highly doubt that. However, it has always worked with the `start` command, so you can type `start myexe` in a cmd.exe prompt. But `myexe` alone shouldn't work, I think (unless you are in the directory of `myexe` or that directory is in the `PATH` environment variable). – [Andreas Rejbrand](#) Sep 18 '18 at 11:30 

I simply make a small `myprog.cmd` file with the following content: `@START myprog.exe %*` This invokes the App Paths entry for `myprog.exe` and passes all command arguments. The file should be created somewhere accessible in `PATH` – [Stavr00](#) Oct 2 '18 at 17:21 

@sherdim @andreas-rejbrand I can confirm this does not work on windows 10 if myexe is not in `PATH`. `start myexe` does work with this registry change, *without* adding myexe to `PATH`. – [davenpcj](#) May 7 at 16:02

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



answered Jan 27 '11 at 22:04



darioo

39.5k 9 66 96

2 You may also want to try Eveditor (eveditor.com) – Dmitry Apr 8 '12 at 8:12

28

Rather than putting the executable into a directory on the path, you should create a batch file in a directory on the path that launches the program. This way you don't separate the executable from its supporting files, and you don't add other stuff in the same directory to the path unintentionally.

Such batch file can look like this:

```
@echo off
start "" "C:\Program Files (x86)\Software\software.exe" %*
```

edited Apr 14 '15 at 11:54



Daniel Schneller

11.2k 5 37 67

answered Jan 27 '11 at 22:22



Ben Voigt

240k 30 326 592

1 Do you mind adding an example of such an batch-file - would be nice? – petermeissner Aug 13 '13 at 8:45

I feel like this is the best option, I didn't want to add too many stuff to path variable. I'll edit into this answer example of such batch file. – Dino Apr 14 '15 at 11:43

1 The problem I'm having with this is it starts a new shell. I'm trying to get a cygwin binary to launch in the same shell, which it does if I reference it directly as C:\cygwin\bin\grep.exe – Johann Oct 1 '15 at 16:55

4 @Johann: Take out the start "" and just begin with the path to the executable. Also, for use in cygwin, you might want a cygwin shell script, or a symlink. Or a shell alias. – Ben Voigt Oct 1 '15 at 17:36

Perfect, thank you! I agree with your other options for within cygwin, but in this case I'm trying to use a cygwin binary from powershell. – Johann Oct 1 '15 at 19:27

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



19

1. Type `Edit environment variables`
2. Open the option `Edit the system environment variables`
3. Click `Environment variables...` button
4. There you see two boxes, in `System Variables` box find `path` variable
5. Click `Edit`
6. a window pops up, click `New`
7. Type the **Directory** path of your `.exe` or `batch` file (**Directory** means exclude the file name from path)
8. Click `ok` on all open windows and ~~restart your system~~ **restart the command prompt.**

edited Aug 16 '18 at 5:38



ThatGuyRob

50 11

answered Jan 27 '17 at 13:32



AmiNadimi

1,909 1 18 33

-
- 1 This is the only answer that helped me. I would add that "restarting the system" isn't necessary. Simply restarting the cmd would suffice. – [Joffrey Baratheon](#) Feb 5 '18 at 23:35
-

3

- If you want to be able to run it inside `cmd.exe` or batch files you need to add the directory the `.exe` is in to the `%path%` variable ([System or User](#))
- If you want to be able to run it in the Run dialog (`Win+R`) or any application that calls [ShellExecute](#), adding your exe to the [app paths key](#) is enough (This is less error prone during install/uninstall and also does not clutter up the path variable)

answered Jan 27 '11 at 22:34



Anders

73k 10 78 129

it's amazing there's no simple solution for such a simple task on windows, I created this little cmd script that you can use to define aliases on windows (instructions are at the file header itself).

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



Facebook

answered Nov 19 '13 at 19:36



Benja

3,405

1

21

25

Nice idea, but the implementation has many bugs. – [Roland Illig](#) Oct 29 '16 at 1:56



You may also permanently (after reboots) add to the *Path* variable this way:

3

Right click *My Computer* -> Click *Properties* -> Click *Advanced system settings* -> Click *Environment Variables*



Reference: [Change System/User Variables](#)

answered Jul 15 '14 at 4:02



JoePC

60

14



Put it in the c:\windows directory or add your directory to the "path" in the environment-settings (windows-break - tab advanced)

1

regards, //t



answered Jan 27 '11 at 22:01



Teson

4,058

5

34

59

Instead of placing custom executables into C:\Windows, one should rather add a custom directory to the PATH environment variable. – [Roland Illig](#) Oct 29 '16 at 1:49

It depends. Putty.exe I put in /windows. – [Teson](#) Oct 31 '16 at 8:59

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook

run the bat file

Now place your .exe in c:\windows, and you're done.

you may type the 'exename' in command-line and it'll run it.

answered Mar 10 '15 at 17:45



Stavm

2,790

1

19

41

-
- 2 Instead of placing custom executables into C:\Windows, one should rather add a custom directory to the PATH environment variable. – Roland Illig Oct 29 '16 at 1:48
-

Simple Bash-like aliases in Windows

1

To get global bash-like aliases in Windows for applications not added to the path automatically without manually adding each one to the path, here's the cleanest solution I've come up with that does the least amount of changes to the system and has the most flexibility for later customization:

"Install" Your Aliases Path

```
mkdir c:\aliases
setx PATH "c:\aliases;%PATH%"
```

Add Your Alias

Open in New Shell Window

To start C:\path to\my program.exe, passing in all arguments, opening it in a new window, create c:\aliases\my program.bat file with the following contents(see [NT Start Command](#) for details on the start command):

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook

To start `c:\path to\my program.exe` , passing in all arguments, but running it in the same window (more like how bash operates) create `c:\aliases\my program.bat` file with the following contents:

```
@echo off
pushd "C:\path to\"
"my program.exe" %*
popd
```

Execute in Current Shell Window 2

If you don't need the application to change the current working directory at all in order to operate, you can just add a symlink to the executable inside your aliases folder:

```
cd c:\aliases\
mklink "my program.exe" "c:\path to\my program.exe"
```

answered Jun 16 '17 at 15:56



Erasmus

1,239 9 4

Add to the PATH, steps below (Windows 10):

0

1. Type in search bar "environment..." and choose Edit the system environment variables which opens up the System Properties window
2. Click the Environment Variables... button
3. In the Environment Variables tab, double click the Path variable in the System variables section
4. Add the path to the folder containing the .exe to the Path by double clicking on the empty line and paste the path.
5. Click ok and exit. Open a new cmd prompt and hit the command from any folder and it should work.

edited Jul 2 '18 at 18:47

answered Jul 2 '18 at 18:26

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook



0

Another way could be through adding .LNK to your \$PATHEX. Then just create a shortcut to your executable (ie: yourshortcut.lnk) and put it into any of the directories listed within \$PATH.



WARNING NOTE: Know that any .lnk files located in any directories listed in your \$PATH are now "PATH'ed" as well. For this reason, I would favor the batch file method mentioned earlier to this method.

answered Sep 9 '18 at 5:06



Hept0p

1 1



-1

Should anyone be looking for this after me here's a really easy way to add your Path.

Send the path to a file like the image shows, copy and paste it from the file and add the specific path on the end with a preceding semicolon to the new path. It may be needed to be adapted prior to windows 7, but at least it is an easy starting point.



[Command Prompt Image to Export PATH to text file](#)

answered Apr 28 '16 at 17:06



Kevin

1

The essential part of an answer should be in a text, not in an external screenshot. – [Roland Illig](#) Oct 29 '16 at 1:50



-3

The best way to do this is just install the .EXE file into the windows/system32 folder. that way you can run it from any location. This is the same place where .exe's like ping can be found



answered Oct 20 '16 at 0:19



Matthew

1 3

Join Stack Overflow to learn, share knowledge, and build your career.

[Email Sign Up](#)

OR SIGN IN WITH



Google



Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

 Google

Facebook 