

---

## Running T<sub>E</sub>X under Windows PowerShell

Adeline Wilcox

When I need to use T<sub>E</sub>X with Microsoft Windows, I prefer compiling my (L<sup>A</sup>)T<sub>E</sub>X code from the command-line shell called PowerShell to using `cmd.exe`.

After running, say, `pdflatex mypaper.tex` (file extension optional), correcting L<sup>A</sup>T<sub>E</sub>X code with gVim (my editor of choice) and writing the file, typing `r` at the PowerShell prompt reruns the last command given to PowerShell.

Further, if a command has already been used in the current PowerShell session, executing PowerShell's `Get-History` cmdlet gives its Id number. For example, if `bibtex mypaper` was previously run and the command Id is 8, B<sub>I</sub>B<sub>T</sub>E<sub>X</sub> can be rerun by typing the short command `r 8`.

At the PowerShell prompt, typing '`Get-Help <cmdlet-name>`' works rather like the Unix man pages.

PowerShell also knows some Unix aliases. Instead of typing the PowerShell cmdlet

```
Move-Item oldfile.tex newfile.tex
```

the same thing can be done with

```
mv oldfile.tex newfile.tex
```

An object-based shell, PowerShell can be maddening to experienced Unix shell programmers. But as long as one does not attempt too much with PowerShell, it can still be a handy way to run L<sup>A</sup>T<sub>E</sub>X.

◇ Adeline Wilcox  
Department of Veterans Affairs  
810 Vermont Ave, NW  
Washington, DC, USA  
`adeline dot wilcox (at) va dot gov`