Installing Windows Subsystem for Linux and integrating it as your default shell in Visual Studio Code.

The following instructions will walk you through the process of installing the Windows Subsystem for Linux on your Windows computer to give you the full power of the bash shell and all Linux command line inside Windows. The Bash shell is a significant improvement over the built-in Windows Command Line or PowerShell. This should take no longer than an hour to complete, and by the end you will have a functional linux shell integrated into VSCode to improve your daily workflow.

1. Prerequisites
    a. Have Windows 10 - build 16215 or later
        i. To check your build number:
            1. Hit Win key
            2. In the search box, type “Settings”
            3. Click on “System”
            4. Click on “About” from left-bar
            5. Scroll down to “Windows specifications”
            6. OS build must be 16215 or greater

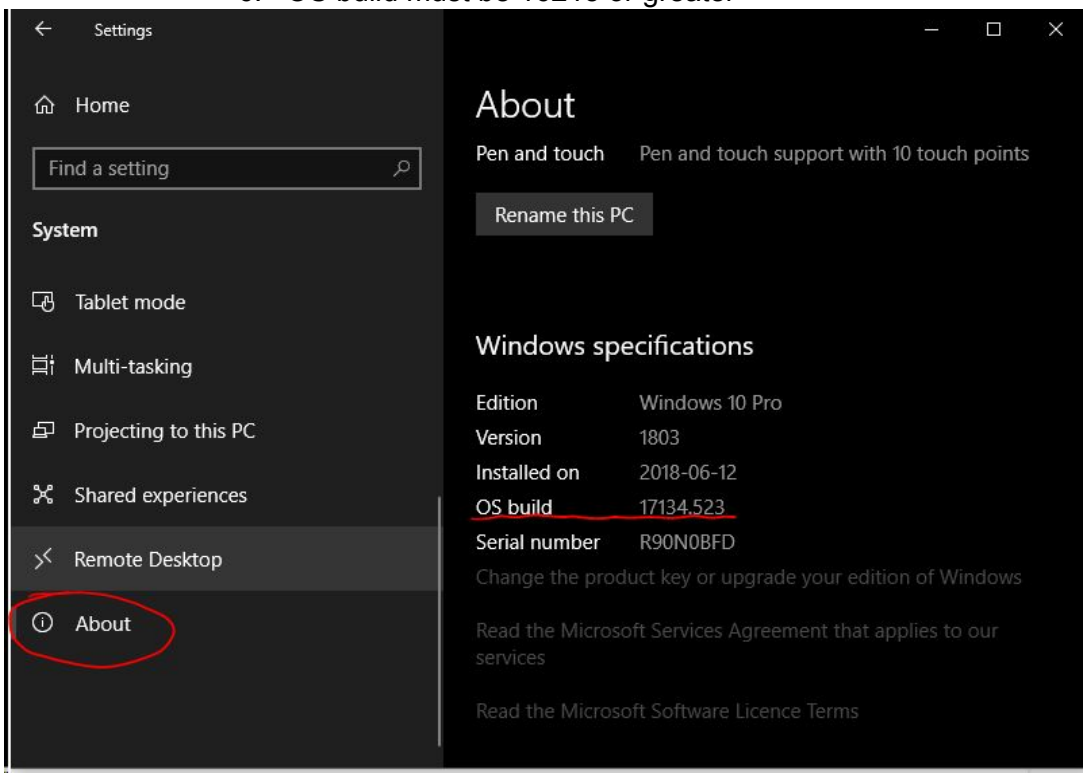


Figure 1: Check build number on Windows 10

b. Being an administrator on your computer
c. Have Windows PowerShell2.0 installed
d. Have Visual Studio installed
e. Basic knowledge of Linux command line

2. Setup
    a. Enable Windows Subsystem for Linux on your computer
        i. Hit Win key
        ii. In the search box, type “Powershell”
        iii. Right-click on “Powershell.exe”
        iv. Click “Run as administrator”
        v. Inside PowerShell, execute the command `Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux`
        vi. Once this command finishes running, exit PowerShell. You may need to restart your computer.

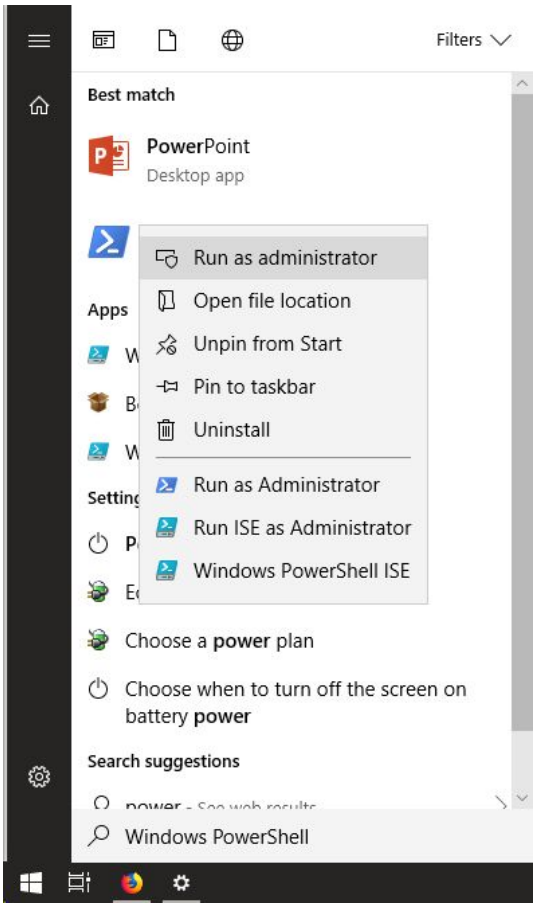


Figure 2: Running PowerShell as admin

3. Install Linux Distribution on Windows
    a. Hit Win key
    b. Type in “Store”
    c. Open “Microsoft Store”
    d. Search for “Linux” or the name of your distribution
    e. Click on the distribution of your choice
        i. The supported distributions are:
            1. Ubuntu
            2. OpenSUSE
            3. SLES
            4. Kali Linux
            5. Debian GNU/Linux

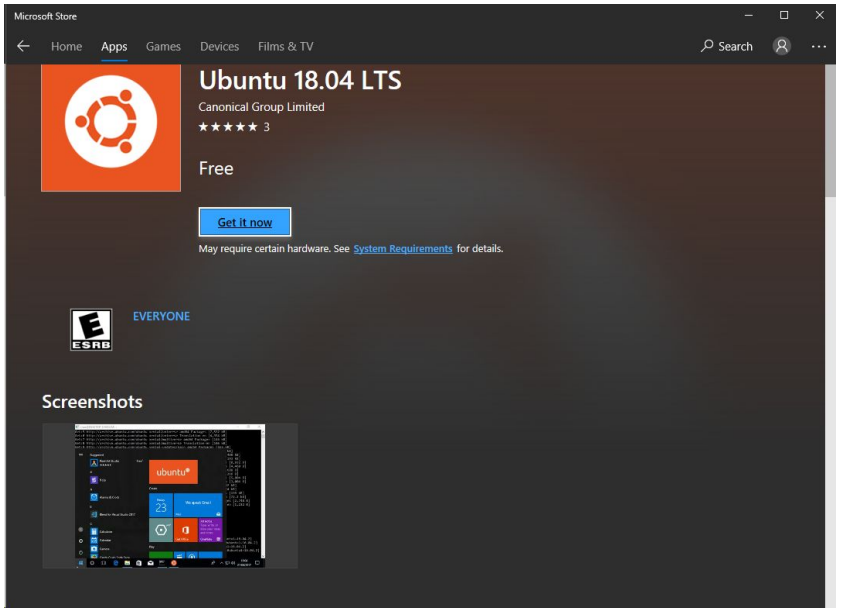
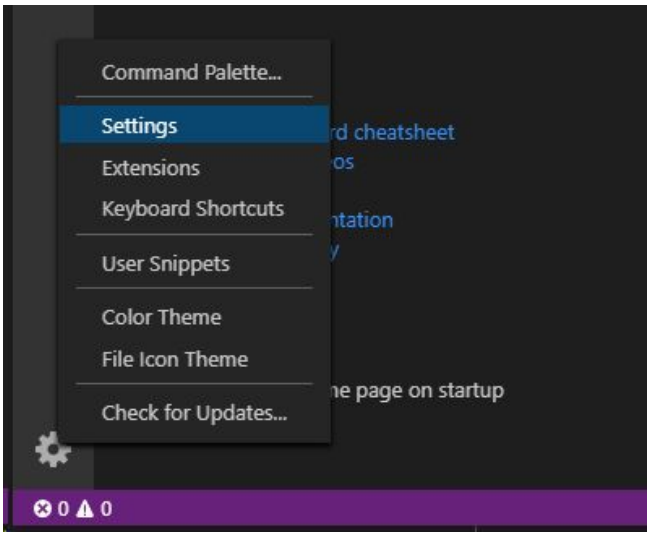


figure 3: Linux in the Microsoft Store

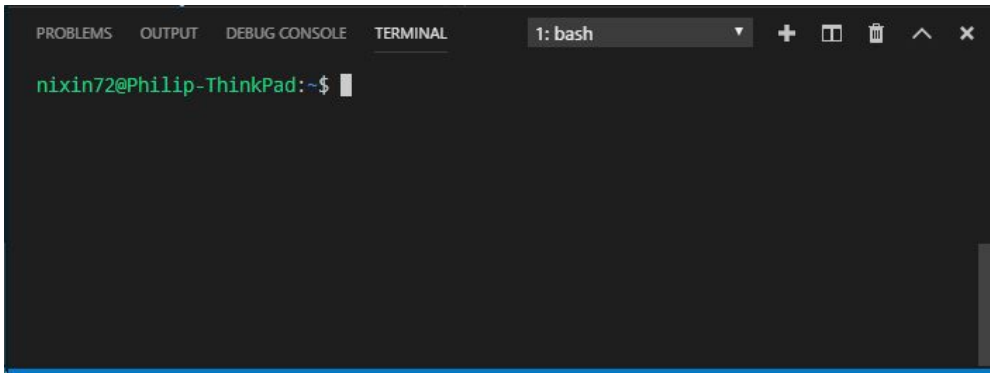
f. Click on “Get it now” in the Microsoft Store
g. The linux distribution will begin to install
h. This can take a while to download and install, and will vary depending on your internet speed.

4. Set Bash to be your default shell in Visual Studio Code
    a. If you don’t already have VSCode, install it from here.
    b. Open VSCode.
    c. In the bottom right, click the gear icon
    d. Click the “settings” option
    e. In the search bar, type in `terminal.integrated.shell.windows`
    f. The next step will change if you’re editing your settings through the GUI or through JSON.
        i. GUI:
            1. Simply type in the path to your bash shell
            2. This should be at `C:/Windows/System32/bash.exe`
            3. Save your settings and open a new terminal. The default should now be bash.
        ii. JSON:
            1. In the left panel, click “Replace in settings”
            2. On the right, type in the path to your bash shell
            3. This should be at `C:/Windows/System32/bash.exe`
            4. Save your settings and open a new terminal. The default should now be bash.



Congratulation!

You now have Bash set as the default shell in VSCode. Whenever you open a terminal now, you should see something like this:



Optional next steps: Getting a better default shell

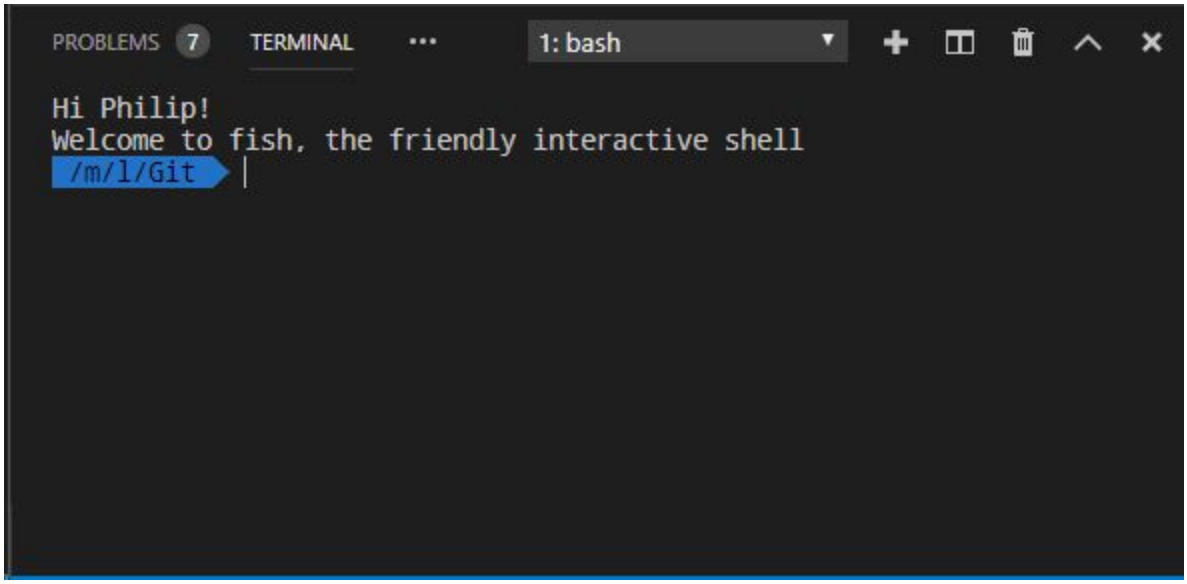
The default shell in Visual Studio Code will now be set to use the Windows Subsystem for Linux shell, with whichever operating system you chose to install. For setting it up, the distribution shouldn't matter, but make sure you're familiar with whatever you install.

Now, the default bash shell isn't very nice looking and isn't the most functional, so these steps are optional, but will give you a much better experience using your shell. We're going to install the Fish shell inside WSL and use that as the default shell inside the integrated terminal. This section is going to assume you're using Ubuntu, but the commands will be similar, just depends on which package manager you're using.

1. Open a bash shell either from inside VSCode or just standalone.
2. Install the Fish shell in Linux
    a. Type the command `sudo apt-get install fish`
    b. You can also compiler from source following their GitHub repository.
3. Install the Fish shell package manager
    a. Type the command `curl -L https://get.oh-my.fish | curl`
4. Install themes
    a. Type the command `omf install <theme-name>`
    b. My personal preference is "agnoster", but you can check out a bunch of themes here.
5. Install PowerLine fonts on *Windows,* not on Linux
    a. Make sure you have git installed on your local machine.
    b. Install and build from source - this can take a while since there's a lot of fonts.
    ```
    > git clone https://github.com/powerline/fonts.git --depth=1
    > ./fonts/install.ps1
    ```
    c. Again, make sure you run this from inside Windows Powershell, *not* a bash shell.
    d. Or if you only want a specific font, you can check out all the fonts at the Git repo
6. Use Powerline fonts in VSCode
    a. Open VSCode
    b. Click on the gear icon in the bottom left hand corner
    c. Click on "Settings"
    d. Search for "`terminal.integrated.fontFamily`"
    e. Change the value to the name of the font you want to use
        i. Ex. "Droid Sans Mono Dotted for Powerline"
    f. Save the changes to your settings.
7. Set the integrated terminal to use Fish as the default shell
    a. In the settings, search for `terminal.integrated.shellArgs.windows`
    b. Add the two arguments "`-c`" and "`fish`"
    c. This will make bash be executed as `bash -c fish`

8. Setting the theme once they're installed
    a. Run the command `omf theme <theme-name>`
    b. Ex. `omf theme agnoster`

Now when you open your terminal, you'll see something more like this:



Which is much nicer than the default bash shell, and gives you some extra utility like syntax highlighting, code completion, and smart git integration.