

# Windows

## *Before you Start*

- Save and close any other work (you will have to reboot at least once)
- If you have a pending Windows Update, do that before you start
- You will need ~5 GB of hard drive space and a stable internet connection
- If you have an operating system installed using the **VirtualBox** software, converting to WSL **may destroy your VM**. Make sure you back up your files if you are using that software.

## *Install Windows Subsystem for Linux (WSL) and a Linux Distribution*

- Open the Windows Command Prompt (search for “cmd”)
- Run the command: `wsl --list --online` to list available Linux distributions
- Once you have chosen a distribution, run the command:  
`wsl --install -d <distribution_name>`
  - If unsure which distribution to choose, choose Ubuntu
- Windows will install WSL and your chosen distribution. This will take a minute or two.
- Restart your system when prompted.
- That's it. Access your chosen Linux distribution at will by running `wsl` through `cmd`
  - The first time you run WSL you will be prompted to choose a username and password as shown in the figure below.
  - This process should also install executable programs named WSL and whatever your distribution was (e.g. Ubuntu) which you can use to run as an alternative to opening the command prompt

If this command does not work for some reason, troubleshooting options can be found at:

<https://learn.microsoft.com/en-us/windows/wsl/install>

```
Ubuntu is already installed.
Launching Ubuntu...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: cbozhart
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.146.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/cbozhart/.hushlogin file.
```

### *Setup Shortcut to WSL Home (Optional But Recommended)*

Sometimes it will be useful to access files inside your Linux operating system from Windows (e.g. to email them, edit an image in Photoshop, etc.). To access the root directory of your WSL operating system(s) through the Windows Explorer, you can type `\\wsl$` into the Windows Explorer. This will lead to a directory containing subdirectories for each WSL Linux distribution you have installed. Each of these directories is the root directory of that Linux distribution.

It is very convenient to create a shortcut to the `home/<username>` directory of your Linux distribution.

- Navigate to `\\wsl$/<distribution>/home/` in the Windows Explorer
- Right click the `<username>` directory
- Choose “show more options” → “create shortcut”
  - You will be prompted to create the shortcut on your desktop

## **Linux**

### *Installing Packages*

For those unfamiliar with Linux, software is often installed through a package management system through the command line. This tutorial will assume the user has installed Ubuntu and is using `apt` to install packages.

If you are on a new Linux installation - including an installation through WSL on Windows - first run

```
sudo apt-get update
```

While you're at it, it's a good idea to update your OS itself

```
sudo apt update && sudo apt full-upgrade
```

This command (and any command that starts with `sudo`) will usually require a password authentication (the same password that you chose when you set up your Linux OS). This is the equivalent of running a program as administrator on Windows.

The general command for installing a package on Ubuntu using `apt` is:

```
sudo apt-get install <package-name>
```

Usually `package-name` is quite self explanatory (e.g. `git` or `gcc`) but if that isn't cutting it, you can search for a package using:

```
apt search <keyword>
```

Tip: for better searching, install the `aptitude` package and then search using `aptitude search <keyword>`

### *Text Editing*

The default text editors on Ubuntu are `vim` and `nano`. If you have never heard of or used either of these, `nano` is probably the right call. `vim` is a very useful (and common) text editor on Linux platforms, but it does have a significant learning curve.

### *Required Packages*

The following packages are needed to run Proto (as well as do any general C++ development). Note that if you are running Linux through WSL and have previously installed any of these programs, you will have to reinstall them on your Linux platform.

- `make`: the GNU make system
- `cmake`: cross platform alternative to make
- `build-essential`: includes - among other things - the `g++` compiler
- `gfortran`: Fortran compiler. The default version should be 11
- `gcc`: this might be included in build-essential, but try installing it anyway
- `tar`: programming for zipping files into compressed tarball archives
- `openmpi-bin`: library for parallel programming through MPI
- `git`: version control and interface with github
- `libhdf5-dev`: library for interfacing with HDF5 files in serial
- `libhdf5-mpi-dev`: MPI parallel version of the above
- `hdf5-tools`: Command line tools for examining HDF5 files
- `python3-pip`: Installs Python 3 and Python's own package manager "`pip`"
- `libblas-dev`: Basic Linear Algebra Subroutines
- `liblapack-dev`: Linear Algebra Packages
- `qtbase5-dev`: Needed by VisIt
- `xterm`: Needed by VisIt
- `x11-apps`: Needed by VisIt

## *Installing Proto*

Proto is installed through git. The github repository is located at:

<https://github.com/applied-numerical-algorithms-group-lbnl/proto>

To download Proto using git, navigate to the directory where proto should be installed and use the following command:

```
git clone https://github.com/applied-numerical-algorithms-group-lbnl/proto.git
```

Once Proto is downloaded, the `blt` and `blis` submodules need to be initialized and updated:

```
cd proto
git submodule init
git submodule update
```

## *Installing VisIt*

Installation:

- Install the qt, x11, and xterm packaged noted in the *Required Packages* section
- Navigate to the VisIt releases page:  
<https://visit-dav.github.io/visit-website/releases-as-tables/>
- Download the tgz file for your distribution
- Download the `visit-install` script
  - If you installed Linux using WSL, move these downloaded files into your Linux home directory. See “set up shortcut to WSL home above”
- Follow the installation instructions at:  
[https://visit-sphinx-github-user-manual.readthedocs.io/en/v3.2.0/gui\\_manual/Intro/Installing\\_VisIt.html](https://visit-sphinx-github-user-manual.readthedocs.io/en/v3.2.0/gui_manual/Intro/Installing_VisIt.html)
  - Choose configuration file number 2 (ANAG/APDEC) when running the `visit-install` script unless you have a reason to do otherwise.
- VisIt should be installed in `/usr/local/visit`. The VisIt GUI can be opened with `/usr/local/visit/bin/visit`

Create A VisIt Alias (Optional But Recommended):

- Open the file `~/.bashrc`
- At the bottom of this file add the following line:  
`alias visit="/usr/local/visit/bin/visit"`
- Save and close the file and restart. You can now open VisIt by typing `visit`