

Steps for Installing LROSE on Windows

1. Install the Windows Linux Subsystem manually by following the steps for the manual install here: <https://docs.microsoft.com/en-us/windows/wsl/install-win10>. For choosing your Linux distribution, I recommend Ubuntu. (**DO NOT** join the Windows Insider Program and attempt the automatic install method, it can introduce instability to your computer).

Manual Installation Steps

If you are not on a Windows Insiders build, the features required for WSL will need to be enabled manually following the steps below.

Step 1 - Enable the Windows Subsystem for Linux

You must first enable the "Windows Subsystem for Linux" optional feature before installing any Linux distributions on Windows.

Open PowerShell as Administrator and run:

```
PowerShell Copy  
  
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

We recommend now moving on to step #2, updating to WSL 2, but if you wish to only install WSL 1, you can now **restart** your machine and move on to [Step 6 - Install your Linux distribution of choice](#). To update to WSL 2, **wait to restart** your machine and move on to the next step.

Step 2 - Check requirements for running WSL 2

To update to WSL 2, you must be running Windows 10.

- For x64 systems: **Version 1903** or higher, with **Build 18362** or higher.

2. Create a folder named "git" where you wish to install LROSE and follow the instructions from here to install dependencies for LROSE: https://github.com/NCAR/lrose-core/blob/master/docs/build/lrose_package_dependencies.linux.md. Follow the manual install method and make sure to follow the instructions for the Linux distribution you chose in step 1.

Manually install packages on Debian and Ubuntu

```
apt-get update && \  
apt-get install -y \  
  tcsh git gcc g++ gfortran rsync chrpath \  
  automake make cmake mlocate libtool pkg-config python \  
  libcurl3-dev curl \  
  libf1-dev libbz2-dev libx11-dev libpng-dev \  
  libfftw3-dev libxpat1-dev \  
  qtbase5-dev qtdeclarative5-dev \  
  libeigen3-dev libzip-dev \  
  libarmadillo-dev libopenmpi-dev \  
  libnetcdf-dev libhdf5-dev hdf5-tools \  
  libcurl4-openssl-dev  
  
# create link for qmake  
  
cd /usr/bin; \  
/bin/rm -f qmake qmake-qt5; \  
ln -s /usr/lib/x86_64-linux-gnu/qt5/bin/qmake qmake; \  
ln -s /usr/lib/x86_64-linux-gnu/qt5/bin/qmake qmake-qt5
```

3. To install LROSE, follow the instructions here from step 3 onwards:
https://github.com/NCAR/lrose-core/blob/master/docs/build/LROSE_cmake_build.auto.md. (The default options worked fine for me).

3. Build automatically using scripts in the bootstrap repository

Clone the bootstrap for LROSE

```
cd ~/git
git clone https://github.com/ncar/lrose-bootstrap
```

The distribution will be in the lrose-bootstrap subdirectory:

```
cd ~/git/lrose-bootstrap
```

Run the `checkout_and_build_cmake.py` script:

To see the usage:

```
cd ~/git/lrose-bootstrap/scripts
./checkout_and_build_cmake.py --help
```

```
Usage: checkout_and_build_cmake.py [options]
Options:
  -h, --help            show this help message and exit
  --clean               Cleanup tmp build dir
  --debug              Set debugging on
  --verbose            Set verbose debugging on
  --package=PACKAGE    Package name. Options are: lrose-core (default),
```