UF UNIVERSITY of FLORIDA
Herbert Wertheim College of Engineering

CIS4930

Ex0: Python Toolchain Setup

Overview

This exercise is intended to help students set up your toolchain so that you can build projects the course. This will involve setting up the runtime environments and IDE for Python 3.8. You are not required to use the IDE we specify, though it is recommended. There are no deliverables and there is no grade for this exercise.

The environment specified in this document will be the exclusive one used for testing student project. Note that these instructions are written for Windows 10 users, but students using Ubuntu 20.04 and other Linux distributions should be able to set up their environments for the course. With that said, please note that there is no instructional support for other operating systems. Macintosh computer users should install BootCamp and Windows 10 (which is free to UF students).

Structure

This exercise is broken into these steps:

a) Install Chocolatey package manager and other tools
b) Setup X11 Server for Windows (VcXsrv)
c) Setup WSL Ubuntu with Python 3.8
d) Setup PyCharm (or IntelliJ) and CLion IDEs

Chocolatey Package Manager

To install Chocolatey, open PowerShell (🔎 → "**`PowerShell`**") as an Administrator (right-click → "Run as Administrator"). From there, run this (copy-paste):

```
Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

You can test this with the command "**`choco -?`**". Once it is ready, install the other tools:

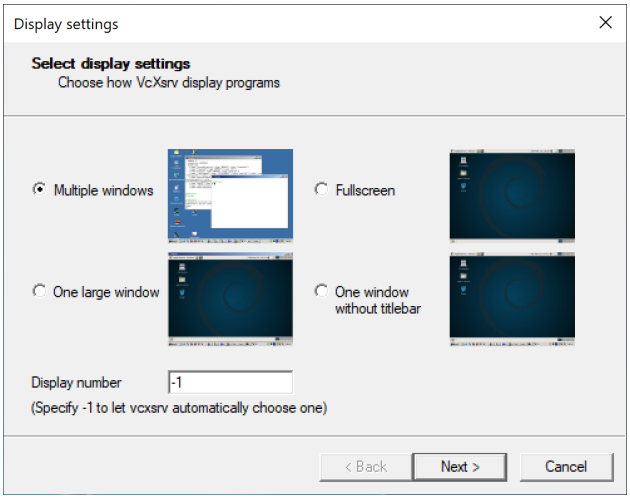
```
choco install vcxsrv wsl-ubuntu clion-ide pycharm
```

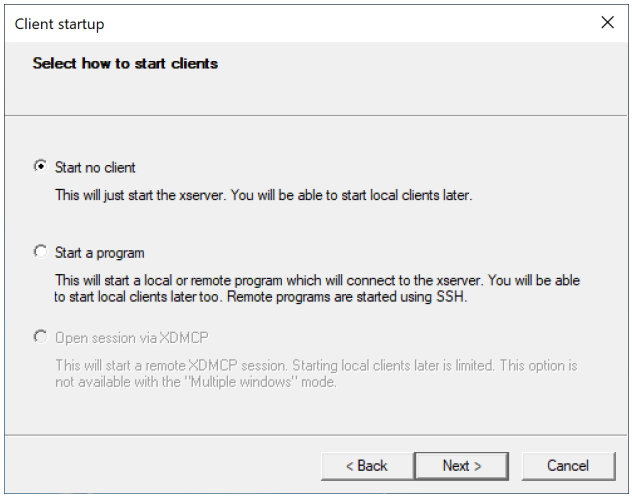
NOTE: If you prefer to use IntelliJ + Python Plugin instead, you may leave out the pycharm package.

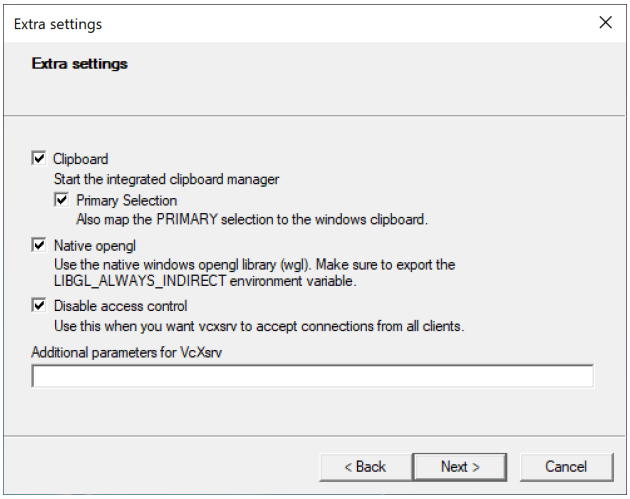
As of the time of this writing, the **`wsl-ubuntu`** package was not working. It can also be installed at this link: https://www.microsoft.com/en-us/p/ubuntu/9nblggh4msv6

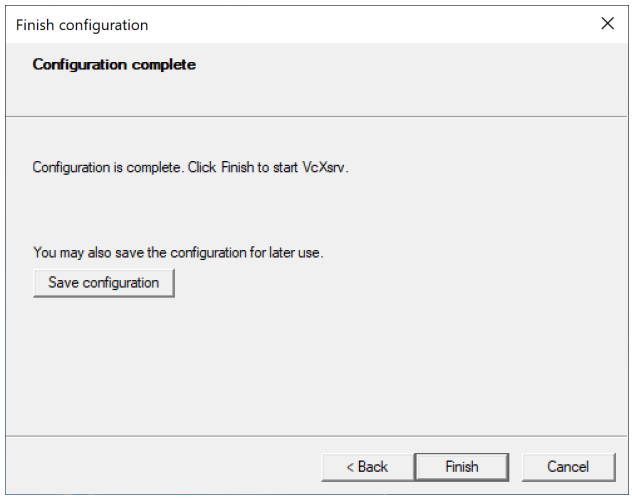
X-Server Setup

Start ***XLaunch*** with the following settings:









Save the configuration – perhaps on the Desktop – with the "xlaunch" extension. Be sure to allow **connections from all networks – private and public** – when asked about Windows Firewall.

WSL Ubuntu Setup

Open Ubuntu **from the start menu** to select a username / complete installation. Then update and install tools:

```
sudo apt update
sudo apt upgrade
sudo apt install x11-apps python3-pip python3-virtualenvwrapper python3-tk python3-venv
sudo apt install cmake g++ gcc gdb build-essential
```

Once this is done, add a few settings to allow WSL, the IDEs, and other tools to connect seamlessly:

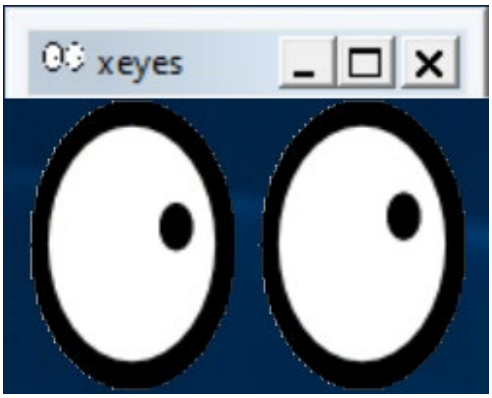
```
echo -e "\nexport WSL_HOST=\$(tail -1 /etc/resolv.conf | cut -d' ' -f2)" >> ~/.profile #1
echo -e "export DISPLAY=\$WSL_HOST:0" >> ~/.profile                                    #2
echo -e "export VIRTUALENVWRAPPER_PYTHON=$(which python3)" >> ~/.profile               #3
echo -e "source /usr/share/virtualenvwrapper/virtualenvwrapper.sh" >> ~/.profile       #4

wget https://raw.githubusercontent.com/JetBrains/clion-wsl/master/ubuntu_setup_env.sh && bash
ubuntu_setup_env.sh                                                                    #5
```

These lines do the following:

1) Identify the Windows host local IP (WSL_HOST)
2) Set up the X11 route (DISPLAY)
3) Set the python version for virtual environments (VIRTUALENVWRAPPER_PYTHON)
4) Enable virtual environment tools for Python (virtualenvwrapper.sh)
5) Set up the SSH connection for CLion and C/C++ remote execution / debugging

Close and reopen the terminal (console) and then run **`xeyes`** to test the server connection. You should see this:

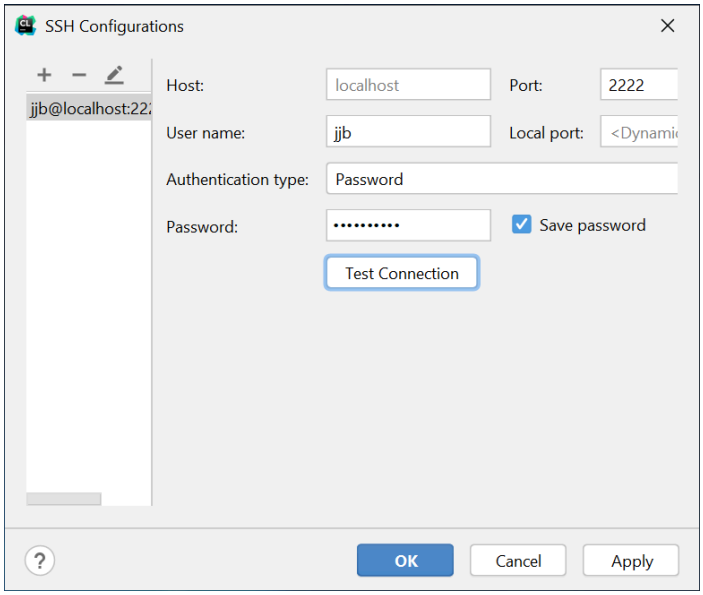


IDE Setup

This section covers the setup of PyCharm (alternatively, IntelliJ with Python Plugin) and CLion IDEs.

Setup CLion

1) In CLion, open **Configure | Settings | Build, Execution, Deployment (left) | Toolchains**.
2) Select **+ | WSL | Credentials Settings (⚙) | +**. Add your username, password, and port 2222.
3) Test the connection. Once everything is working, click "OK". The rest of the tools will be autodetected. There may be a complaint about the C/C++ ABI; this can be ignored.
4) To Test CLion, create a new executable project, then select **Tools | CMake | Reload CMake Project**. Once the project is reloaded, run it (▶) to verify that it is functioning.



Setup PyCharm

1) In PyCharm, open **Configure | Settings | Python Interpreter | ⚙ | Add | WSL**
2) Set the Python Interpreter to Python3: **`/usr/bin/python3`**
3) Click OK, then OK again.
4) Load and execute the "HelloTk" project to test the setup.