

Hands On Exercise 1: Installing MUSE

Short description

This hands-on exercise will allow you to install MUSE on your computer. We will then take you through an example to run and visualise a default MUSE example.

If at any point you get stuck with these hands-on, feel free to post a question in the purpose made MUSE google group if your question hasn't already been answered there!:

<https://groups.google.com/g/muse-model>

Learning objectives

- Install MUSE
- Run an example
- Visualise the results of the example

Exercise content

For Windows users only

Windows users and developers may need to install `windows Build Tools`. These tools include C/C++ compilers which are needed to build some python dependencies.

MacOS includes compilers by default, hence no action is needed for Mac users.

Linux users may need to install a C compiler, whether GNU gcc or Clang, as well python development packages, depending on their distribution.

If you have MacOS or Linux you can skip this section and head to the next section below [here](#).

1. Install Visual Studio from the following link:
<https://visualstudio.microsoft.com/downloads/>
2. Select your preferred edition. Although, the “Community” version is free and contains what is required.
3. Install Visual Studio by selecting the default options.
4. Download the Microsoft Visual C++ Build Tools from the following link by downloading Visual Studio 2019: <https://visualstudio.microsoft.com/downloads/>
5. Select your preferred edition. The “Community” is free and contains what is required.
6. Run the installer
7. Select: Workloads → Desktop development with C++.
8. Install options: select only the “Windows 10 SDK” (assuming the computer is Windows 10). This will come up on the right hand side of the screen.

The installation screen should look similar to the following:

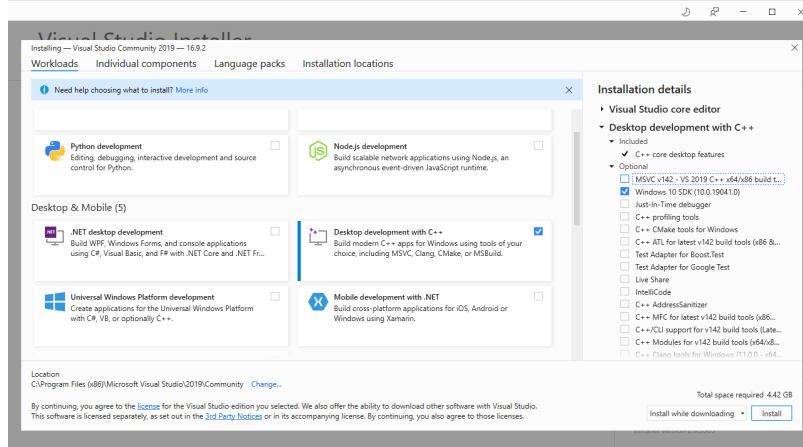


Figure 1.1: Visual Studio Installer window

For further information, see this link: <https://www.scivision.dev/python-windows-visual-c-14-required>

Installing MUSE

MUSE is developed using python, an open-source programming language, which means that there are two steps to the installation process. First, python should be installed. Then so should MUSE.

The simplest method to install python is by downloading the Anaconda distribution. Make sure to choose the appropriate operating system (e.g. windows), python version 3.8, and the 64 bit installer. Once this has been done follow the steps for the anaconda installer, as prompted.

After python is installed we can install MUSE. MUSE can be installed via the Anaconda Prompt (or any terminal on Mac and Linux). This is a command-line interface to python and the python eco-system. In the anaconda prompt, run:

```
python -m pip install --user
git+https://github.com/SGIModel/MUSE_OS
```

It should now be possible to run muse. Again, this can be done in the anaconda prompt as follows:

```
python -m muse --help
```

Running your first example

In this section we run an example simulation of MUSE, in the next section we will visualise the results.

First we need to download the MUSE source code. To do that navigate to the MUSE GitHub repository: https://github.com/SGIModel/MUSE_OS

Click on the green Code button in the top right hand corner and then click on Download ZIP. Figure 2.1 shows how to do this, once you are on the relevant page.

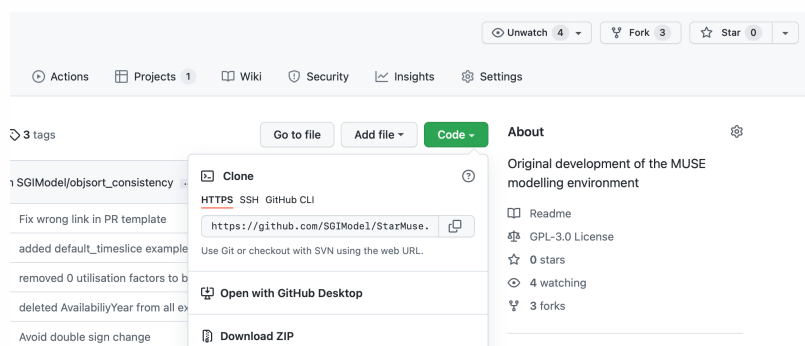


Figure 2.1: How to download MUSE

Once you have downloaded the source code, unzip the folder and move it to a location that is convenient for you.

We will place ours on the desktop for simplicity, but feel free to make a folder in your documents or otherwise.

To run MUSE, we must open anaconda prompt for Windows machines or terminal if on MacOS or Linux. Then we must navigate to the directory using the prompt or terminal to find the MUSE examples. Ours is in Desktop, so we will run the following command:

```
cd ~/Desktop/StarMuse/run/example/default/
```

But yours could be in another location, so fill in the {MUSE_download_location} with your path (This could be /Users/{my_name}/Documents for instance):

```
cd
{MUSE_download_location}/StarMuse/run/example/default/
```

Once we have navigated to the directory containing the example settings settings.toml we can run the simulation using the following command in the anaconda prompt or terminal:

```
python -m muse settings.toml
```

If running correctly, your prompt should output text similar to the following:

```
-- 2020-11-03 15:58:29 - muse.sectors.register -
INFO
Sector legacy registered.

-- 2020-11-03 15:58:29 - muse.sectors.register -
INFO
Sector preset registered, with alias presets.

-- 2020-11-03 15:58:29 - muse.sectors.register -
INFO
Sector default registered.

-- 2020-11-03 15:58:29 - muse.readers.toml - INFO
Reading MUSE settings

-- 2020-11-03 15:58:29 - muse.readers.toml - INFO
Default input values used:
carbon_budget_control.commodities,
carbon_budget_control.method,
carbon_budget_control.debug,
carbon_budget_control.control_undershoot,
carbon_budget_control.control_overshoot,
carbon_budget_control.method_options
```

Results

If the default MUSE example has run successfully, you should now have a folder called `Results` in the same directory as `settings.toml`.

This directory should contain results for each sector (Gas, Power and Residential) as well as results for the entire simulation in the form of `MCACapacity.csv` and `MCAPrices.csv`.

`MCACapacity.csv` contains information about the capacity each agent has per technology per benchmark year. Each benchmark year is the modelled year in the `settings.toml` file. In our example, this is 2020, 2025, ..., 2050.

`MCAPrices.csv` has the converged price of each commodity per benchmark year and timeslice. eg. the cost of electricity at night for electricity in 2020.

Within each of the sector result folders, there is an output for Capacity for each commodity in each year. The years into the future, which the simulation has not run to, refers to the capacity as it retires. Within the Residential folder there is also a folder for Supply within each year. This refers to how much end-use commodity was output.

Visualisation

There are many different ways to visualise the results of MUSE. For example, you could use a programming language such as python or R. In this course, however, we will use Excel for simplicity.

There are also many different variables and combinations of data that we can plot. In this course we will primarily explore the capacity installed over the time horizon (2020-2050). Through this, we can see which of the technologies are invested in and understand the competition between technologies.

To start the visualisation process of the default example, navigate to the folder where you run the `default` example. For instance:

```
cd
{MUSE_download_location}/StarMuse/run/example/default/
```

Go into the folder called `/Results/` and right click on the file called `MCACapacity.csv` and open it with Microsoft Excel. Once you've opened the file with Excel, we can begin the data visualisation process.

First, select the PivotChart button under the insert menu. You should then see a box open up such as that shown in Figure 1.3, below:

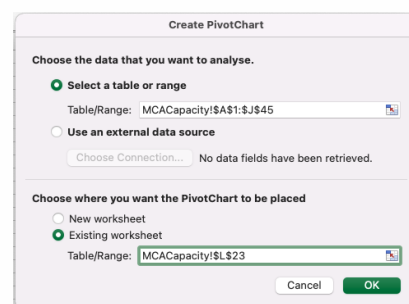


Figure 1.3: Insert Create PivotChart

Ensure that the “Select a table or range” highlights all of the data, and the “Choose where you want the PivotChart to be placed” is selected to “Existing worksheet” like in the figure above. Then within the “Table/Range” box, click the cell where you would like the figure to be placed.

You should then see the boxes appear shown below in Figure 1.4 in Excel:

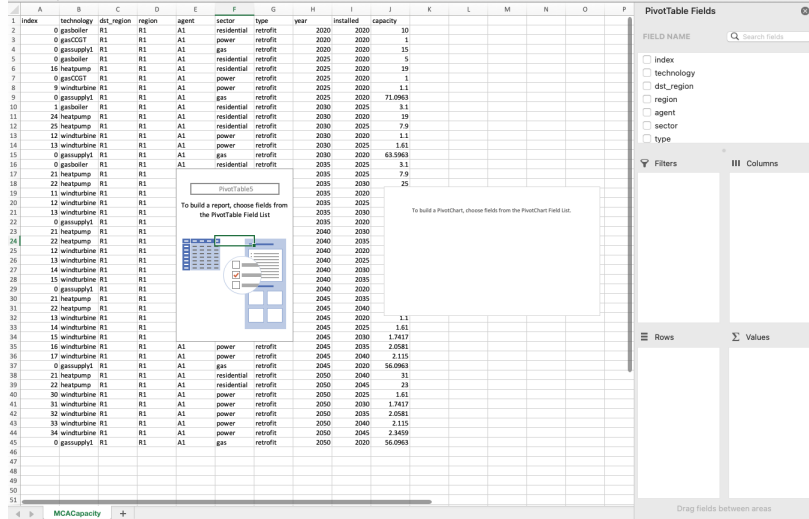


Figure 1.4: Inserted PivotChart

Next we want to view how the capacity changes per technology over the technology horizon for the power sector.

To do this, we must drag the relevant Fields from the PivotTable to the Filters, Columns, Rows and values boxes.

First, we will drag the capacity field to the values boxes. These are the values that we want to plot. Then we will drag the year field to the Axis (Categories) box. This will start to populate a table and a graph. Finally, we want to filter for the residential sector. To do that we will drag the sector field to the Filters box.

We should then end up with the chart, table and PivotTable fields shown in Figure 1.5.

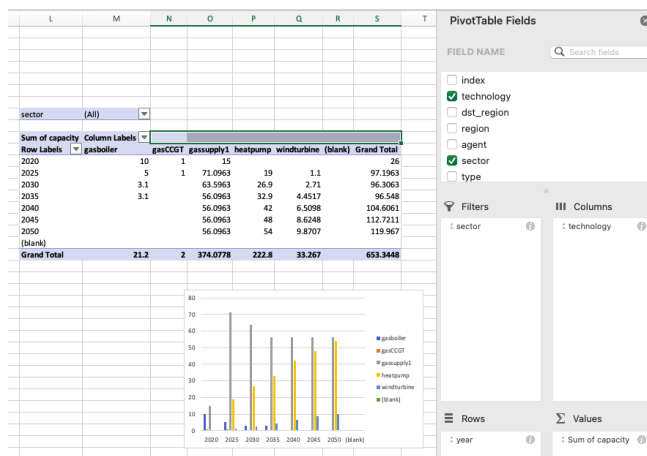


Figure 1.5: Default visualisation example.

However, this is displaying all sectors currently, so to filter for solely the residential sector, we must click the dropdown arrow next to the “sector, (All)” cells.

This can be shown in Figure 1.6, below:

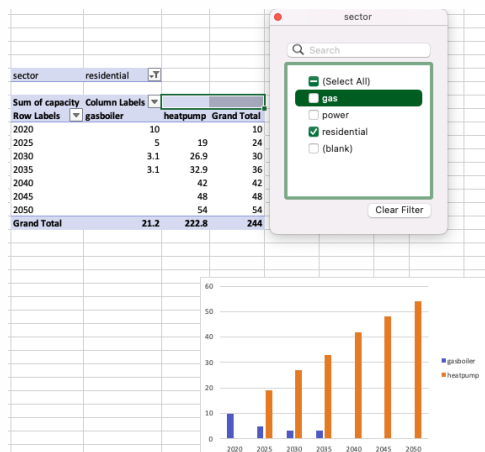


Figure 1.6: Residential sector visualisation for default example.

We have unchecked all sectors apart from the residential sector to generate the results shown in Figure 1.6.

We can see that gas boilers are the main technology in 2020, but this reduces to zero by 2040. Heatpumps take over between 2020 and 2050, increasing significantly as demand increases.

Summary

In this hands-on we have installed MUSE, learnt how to run a demo example and visualised the results of this demo example. In the next lectures and hands-on we will learn in detail the fundamentals of MUSE and how to edit the demo example for a case study of our choice.