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 though an example to run and visualise a default MUSE example.

If at any point you get stuck with these hands-on exercises, 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

For an in-depth look at the MUSE documentation have a look here: http://muse-docs.readthedocs.io

Learning objectives

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

Exercise content

In all of the hands-on, there is an accompanying video linked at the start of the lectuer. Here we show you how to do the hands-on. This should make the process simpler to follow.

Hands-on accompanying video: https://youtu.be/Gppj1Gl-ajA

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/vs/older-downloads/. Please select the 2019 version. Click on download.

2. Select the "Visual Studio Community" version. Click on "Download" and save the executable vs_Commmunity.exe.

3. Install Visual Studio by selecting the default options. You may be asked to reboot your computer to complete the installation.

4. Download the Microsoft Visual C++ Build Tools from the following link: https://visualstudio.microsoft.com/vs/older-downloads/.

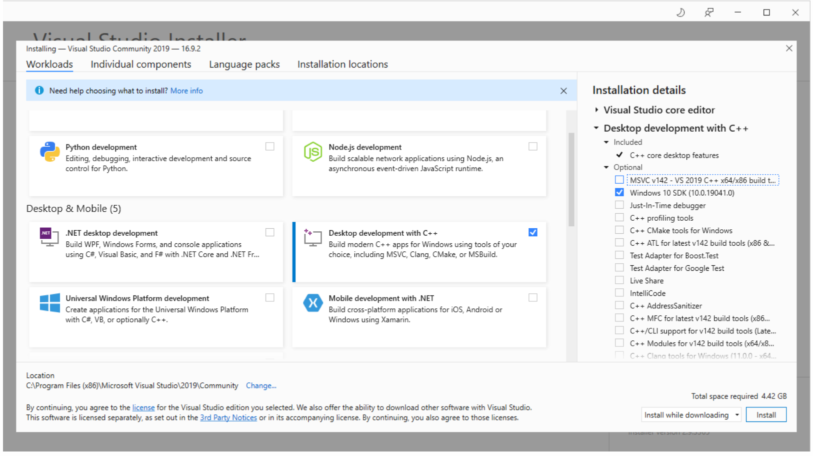
5. Please select the "Build Tools for Visual Studio 2019 (version 16.9)". Click on download. Save the vs_BuildTools.exe.

6. Run the installer

7. 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:



Installing — Visual Studio Community 2019 — 16.9.2

Workloads | Individual components | Language packs | Installation locations

Need help choosing what to install? More info

- Python development: Editing, debugging, interactive development and source control for Python. □
- Node.js development: Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime. □

Desktop & Mobile (5)

- .NET desktop development: Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET Core and .NET Fr... □
- Desktop development with C++: Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild. ☑
- Universal Windows Platform development: Create applications for the Universal Windows Platform with C#, VB, or optionally C++. □
- Mobile development with .NET: Build cross-platform applications for iOS, Android or Windows using Xamarin. □

Installation details

- Visual Studio core editor
- Desktop development with C++
  - Included
    - ✓ C++ core desktop features
  - Optional
    - □ MSVC v142 - VS 2019 C++ x64/x86 build t...
    - ☑ Windows 10 SDK (10.0.19041.0)
    - □ Just-In-Time debugger
    - □ C++ profiling tools
    - □ C++ CMake tools for Windows
    - □ C++ ATL for latest v142 build tools (x86 &...
    - □ Test Adapter for Boost.Test
    - □ Test Adapter for Google Test
    - □ Live Share
    - □ IntelliCode
    - □ C++ AddressSanitizer
    - □ C++ MFC for latest v142 build tools (x86...
    - □ C++/CLI support for v142 build tools (Late...
    - □ C++ Modules for v142 build tools (x64/x8...
    - □ C++ Clang tools for Windows (11.0.0 - x64...

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community Change...

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 4.42 GB

Figure 1.1: Visual Studio Installer window

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 3.9, and the 64 bit installer. Once this has been done follow the steps for the anaconda installer, as prompted.

Open the Anaconda prompt for Windows machines or terminal if on MacOS or Linux and create a new environment hosting python 3.8.

Activate the new environment.

```
conda create --name muse python=3.8
activate muse
```

After python is installed we can install MUSE. MUSE can be installed via the Anaconda Prompt or CMD.exe (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
```

If you get an error such as the following:

```
Cannot find command 'git' - do you have 'git'
installed and in your PATH
```

Then you have to install git in anaconda. This can be completed by running the following command:

```
conda install git pip
```

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

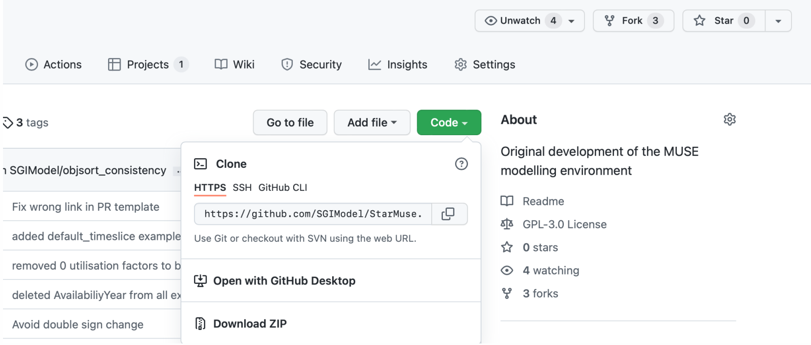
```
python -m muse --help
```

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.



SGIModel/objsort_consistency

Fix wrong link in PR template

added default_timeslice example

removed 0 utilisation factors to b

deleted AvailabiliyYear from all ex

Avoid double sign change

```
https://github.com/SGIModel/StarMuse.
```

Use Git or checkout with SVN using the web URL.

Original development of the MUSE modelling environment

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 the 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:

```
python -m muse -model --default
```

Alternatively, 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 contained within this example (Gas, Power and Residential) as well as results for the entire simulation in the form of the files `MCACapacity.csv` and `MCAPrices.csv`.

`MCACapacity.csv` contains information about the energy 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. For example, it has the cost of electricity at night for electricity in 2020, and other similar results.

Within each of the sector result folders, there is an output for capacity for each commodity in each year. Future years, which the simulation has not run to, refers to the technology 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.

Some of these terms will not be familiar to you yet, but do not worry about this for now. This hands-on is just guiding you the basic process from installation to data visualisation and later hands-on material will give more information.

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 to 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 in anaconda prompt or CMD.exe. For instance:

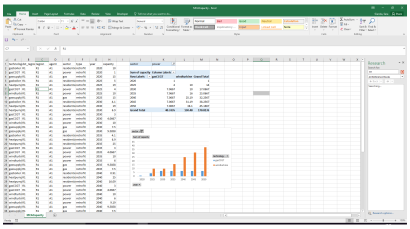
```
cd {MUSE_download_location}/StarMuse/run/example/default/
```

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. 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. Click "OK" when finished.

You can choose to filter with "sector", add "technology" to columns, "year" to rows, and display "capacity" as sum.

You can then display a barchart like the one in Figure 1.3, below:



	A	B	C	D	E	F	G	H
1	technolog	dst_regior	region	agent	sector	type	year	capacity
2	gasboiler	R1	R1	A1	residentia	retrofit	2020	10
3	gasCCGT	R1	R1	A1	power	retrofit	2020	1
4	gassupply	R1	R1	A1	gas	retrofit	2020	15
5	gasboiler	R1	R1	A1	residentia	retrofit	2025	5
6	heatpum	R1	R1	A1	residentia	retrofit	2025	19
7	gasCCGT	R1	R1	A1	power	retrofit	2025	4
8	windturbi	R1	R1	A1	power	retrofit	2025	10
9	gassupply	R1	R1	A1	gas	retrofit	2025	15
10	gasboiler	R1	R1	A1	residentia	retrofit	2030	4.1
11	heatpum	R1	R1	A1	residentia	retrofit	2030	19
12	heatpum	R1	R1	A1	residentia	retrofit	2030	6.9
13	gasCCGT	R1	R1	A1	power	retrofit	2030	3
14	gasCCGT	R1	R1	A1	power	retrofit	2030	4.0667
15	windturbi	R1	R1	A1	power	retrofit	2030	10
16	gassupply	R1	R1	A1	gas	retrofit	2030	7.5
17	gassupply	R1	R1	A1	gas	retrofit	2030	9.5858
18	gasboiler	R1	R1	A1	residentia	retrofit	2035	4.1
19	heatpum	R1	R1	A1	residentia	retrofit	2035	6.9
20	heatpum	R1	R1	A1	residentia	retrofit	2035	25
21	gasCCGT	R1	R1	A1	power	retrofit	2035	3
22	gasCCGT	R1	R1	A1	power	retrofit	2035	4.0667
23	windturbi	R1	R1	A1	power	retrofit	2035	10
24	windturbi	R1	R1	A1	power	retrofit	2035	6
25	gassupply	R1	R1	A1	gas	retrofit	2035	9.5858
26	gassupply	R1	R1	A1	gas	retrofit	2035	7.5
27	gasboiler	R1	R1	A1	residentia	retrofit	2040	0.91
28	heatpum	R1	R1	A1	residentia	retrofit	2040	25
29	heatpum	R1	R1	A1	residentia	retrofit	2040	16.09
30	gasCCGT	R1	R1	A1	power	retrofit	2040	3
31	gasCCGT	R1	R1	A1	power	retrofit	2040	4.0667
32	windturbi	R1	R1	A1	power	retrofit	2040	10
33	windturbi	R1	R1	A1	power	retrofit	2040	6
34	windturbi	R1	R1	A1	power	retrofit	2040	9.19
35	gassupply	R1	R1	A1	gas	retrofit	2040	9.5858
36	gassupply	R1	R1	A1	gas	retrofit	2040	7.5

sector	power		
Sum of capacity	Column Labels		
Row Labels	gasCCGT	windturbine	Grand Total
2020	1		1
2025	4	10	14
2030	7.0667	10	17.0667
2035	7.0667	16	23.0667
2040	7.0667	25.19	32.2567
2045	7.0667	31.19	38.2567
2050	7.0667	38.1	45.1667
Grand Total	40.3335	130.48	170.8135

Figure 1.3: Insert -> Create PivotChart

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.