Switch 2.0 Tutorial: Pre-Tutorial Setup

> Before you begin the tutorial, please complete the steps described here to install Switch on your computer and learn some background information. Sections 1.5, 2.2 and 2.3 are optional. You will need an Internet connection while you do these. Required steps are marked by a blue bar in the margin like this paragraph. Explanatory text and optional steps don't have this mark.

These instructions are simplified to minimize the number of elements you have to download and install. They use data and solver programs that you will find in an accompanying zip file, and they use a simplified method to install Switch. For a more common (but slower) way to install Switch, see the tutorial at https://switch-model.org.

1 Software setup (approx. 1 hour; requires 4-5 GB disk space)

This section describes how to install Switch on your computer so it is ready to solve power system planning problems. Switch depends on a collection of mostly open-source software:

- **Visual Studio Code (VS Code)** is an open-source editor and terminal program which is useful for editing input files and typing commands to run Switch. (You can use a different programming-oriented editor if you prefer.)
- **Anaconda** provides an easy, standardized way to install Switch and the software that it depends on (described below).
- **Switch** is a program that can be used to define and solve a power system optimization model using inputs that you provide. It is made up of a collection of modules written in the Python language, each of which describes a different aspect of the power system.
- **Pyomo** is a general-purpose optimization modeling framework for Python. Switch uses Pyomo to define the elements of your optimization model and call a solver.
- Pyomo converts the Switch model into a standardized, computer-readable form and sends it to an **external solver** (e.g., glpk, cbc, cplex or gurobi). The external solver does the intense computation required to find an optimal plan.

The instructions below will show you how to setup all of the components described above. All of these tools are open-source and cross-platform, so you should be able to use them on any computer.

Disk usage: Switch itself is quite small, but it depends on the Anaconda distribution (0.3-0.5 GB), and a number of Python packages (0.5-1.5 GB). We will also use the Visual Studio Code text editor for this tutorial (0.3 GB), and install some tutorial data (0.3 GB). Note that the higher disk usage numbers are for Windows installations.

1.1 Installing Anaconda and Python

> Download and install the Miniforge3 version of Anaconda from https://github.com/conda-forge/miniforge?tab=readme-ov-file. (If that link doesn't work, you can also try miniconda3 from https://docs.anaconda.com/free/miniconda/; they should both work equally well.) We recommend selecting the latest 64-bit version for your platform and processor (other versions will probably work too). On a Mac, the ".pkg" installer is easier to use than the "bash" one. When installing, you can use the recommended options (install just for me, create start menu shortcut, don't add to PATH environment variable).

If you like, you can complete the Anaconda tutorial offered at the end of the installer, but it is not needed for this tutorial.

If you already use a different version of Anaconda, miniconda or Miniforge, that should work fine for this tutorial.

1.2 Installing Visual Studio Code text editor

For this tutorial, we assume you are using the Visual Studio Code (VS Code) text editor to view and edit code and data files. You can use a different text editor if you like, but it should be capable of doing programming-oriented tasks, like quickly adjusting the indentation of many lines in a text file. If you prefer, you can also open the .csv data files directly in your spreadsheet software instead of using VS Code.

> Download and install the Visual Studio Code text editor from https://code.visualstudio.com/. (On Windows, you can also find Visual Studio Code in the Microsoft Store.)

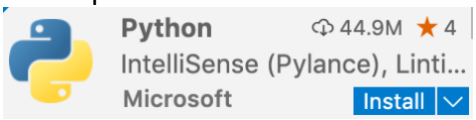
If you need more information on installing VS Code, see https://code.visualstudio.com/docs/setup/setup-overview. (On a Mac you may need to double-click on the downloaded zip file to uncompress it, then use the Finder to move the "Visual Studio Code" app from your download folder to your Applications folder.)

If you'd like a quick introduction to VS Code, see https://code.visualstudio.com/docs (not needed for this tutorial).

> Launch Visual Studio Code from the Start menu (Windows) or Applications folder (Mac). You can choose a color theme and/or work your way through the "Get Started" steps (it's a scrollable list), or you can skip them if you don't want to do that now.

> Follow these steps to install the Python extension for VS Code:
>
> - Click on the Extensions icon on the Activity Bar on the left side of the Visual Studio Code window (or choose View > Extensions from the menu). The icon looks like four squares:
>
> 

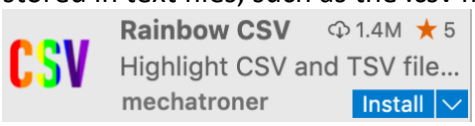
- This will open the Extensions pane on the left side of the window. Type “Python” in the search box, then click on “Install” next to the Python extension that lists Microsoft as the developer:



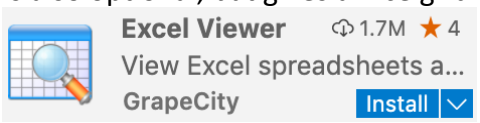
- After installing the Python extension, you will see a “Get started with Python development” tab and a “Get started with Jupyter Notebooks” tab. You can close these.

Follow these steps to install two more extensions that will be useful for this tutorial. These are optional, but they make it easier to read and edit data stored in text files, such as the .csv files used by Switch:

- Type “rainbow csv” in the search box in the Extensions pane, then click on “Install” next to the Rainbow CSV extension (this is optional, but makes it easier to read and edit data stored in text files, such as the .csv files used by Switch):



- Type “excel viewer” in the search box, then click to install the Excel Viewer extension (this is also optional, but gives a nice grid view of .csv files):



If you are running on Windows, choose File > Preferences > Settings, then search for “default profile” and choose “Integrated > Default Profile: Windows”. Then change the setting to “Command Prompt” instead of “null” or “PowerShell”. Then close the Settings tab. (In some Windows configurations, PowerShell doesn’t run correctly inside VS Code or doesn’t find the conda environment, and it also has a built-in “switch” command that may hide the “switch” command used to run Switch. So we use the “Command Prompt” profile instead.)

If you are running on macOS, choose Code > Preferences > Settings, then search for “terminal inherit” and uncheck the box next to “Terminal › Integrated: Inherit Env”. This can be helpful for making sure terminal panes in VS Code work correctly.

We also recommend that you open File > Preferences > Settings (Windows) or Code > Preferences > Settings (Mac), then search for “open folder” and set the option to “on” for “Window: Open Folders in New Window.”

Now you can close VS Code or leave it open. You can click again on the Extensions icon to make the Extensions pane disappear; we won’t need it again in this tutorial.

1.3 Decompress tutorial data, Switch source code and solvers

Find the .zip file of tutorial data that was sent to you along with these instructions. If you did not receive the .zip file or you need one for a different operating system, you can find one at https://switch-model.org/tutorial.

Decompress the .zip file into a convenient location, such as your home directory.

Next, in Visual Studio Code, choose File > Open Folder and choose the folder you created. If you are asked whether you trust the authors of the files in the folder (i.e., yourself), choose "yes".

1.4 Installing Switch and Pyomo

In Visual Studio Code, choose View > Command Palette… from the menubar, then search for the option "Python: Select Interpreter". Wait a little while for the full list of interpreters to appear. Click on the one that mentions "('base')" and has "miniforge3" in the path name. The display won't update much, but VS Code is now configured to use the copy of Python you installed earlier.

Then choose Terminal > New Terminal from the menu bar. This will open a pane at the bottom of the VS Code window where you can type commands to the operating system.

You should see a command line with "(base)" in the prompt. If you don't see these, you may need to repeat the "Select Interpreter" command above and then open a new terminal pane, or you may need to use a different terminal (see end of this section).

Next choose Terminal > New Terminal and run "pip install switch-model" at the command line. This will install Switch and all its dependencies into your "base" environment.

You can check that Switch has been installed by running the commands below at the command prompt:

```
switch --version
pyomo --version
```

You should see version numbers for each one and no errors. (Switch should be version 2.0.7.)

If your computer reports that these commands are not found, try choosing View > Command Palette again, then Python: Select Interpreter and choose the "base" environment again. Then choose Terminal > New Terminal and try the commands above again. If this still doesn't work, you can try opening Terminal.app (on a Mac) or Miniforge Prompt (on Windows) instead of VS Code, and then running the commands there. If that works, you can then run all the commands listed in square boxes in this tutorial in Terminal.app or the Miniforge Prompt instead of in the VS Code terminal pane.

Note: you now have two copies of Switch on your system—one for running (inside the conda base environment), and one for viewing (inside your tutorial directory). Any changes you make to the copy inside your tutorial directory will not affect the copy you use to solve models.

1.5 Installing a faster solver (optional)

The instructions above installed the open-source glpk solver along with Switch. This is able to solve small test cases, but is not fast enough to solve large models. You can complete this tutorial using only glpk, but if you would like to experiment on your own with the larger models, you may need to install a faster solver.

For moderate to large models, especially ones without integer decisions, you may find that the open-source COIN CBC solver works better. You can install cbc with this command:

```
conda install -c conda-forge coincbc
```

(This may report compatibility errors on Windows. Please contact Matthias Fripp <mfripp@edf.org> if you need advice on resolving them.)

For large models with integer variables (e.g., discrete plant construction), it is generally necessary to use proprietary solvers: CPLEX or Gurobi. These are expensive for professional use, but it is possible to get a trial license before you buy a long-term one. Academics can also get full licenses for free. Note that it is important to get a license (temporary or long-term) for the *full* version of the software, not the free or community version that only supports small problem sizes.

You can obtain licenses and download these solvers from here:

Professional:
https://www.gurobi.com/products/gurobi-optimizer/
https://www.ibm.com/products/ilog-cplex-optimization-studio/pricing

Academic:
https://www.gurobi.com/academia/
https://www.ibm.com/products/ilog-cplex-optimization-studio/pricing ("Get the no-cost academic edition" button)

Once you have installed these, you can test that they are available by running one of these pairs of commands from a command prompt or terminal window:

CBC

```
cbc
quit
```

CPLEX

```
cplex
quit
```

Gurobi on Windows:

```
gurobi.bat
exit()
```

Gurobi on Mac or Linux

```
gurobi.sh
exit()
```

Once you have installed a solver, you can select it when you run Switch by specifying --solver cbc, --solver cplex or --solver gurobi on the switch command line or in options.txt (discussed in the main tutorial).

2 Introduction to Switch, Pyomo and Python (0.5 – 9 hours)

This section points you to some useful, quick introductions to Switch, Python and Pyomo. The latter two are optional, but recommended if you will be using Switch extensively or defining custom behaviors (new technologies, rules or policies).

2.1 Introduction to Switch (0.5 – 2 hours)

> For a quick overview of Switch, please read Section 2 of the paper on Switch 2.0 at https://doi.org/10.1016/j.softx.2019.100251.

The following are optional: You can read section 3 of the Switch paper (above) for an overview of the case study we'll examine during this tutorial. And if you would like more detail on Switch, please see the Supplementary Material for the Switch 2.0 paper at https://ars.els-cdn.com/content/image/1-s2.0-S2352711018301547-mmc1.pdf.

2.2 Introduction to Python (optional, 1 – 6 hours)

We recommend reading sections 3 and 4 of the Python introduction at https://docs.python.org/3/tutorial/ . If you would like a deeper understanding, sections 5 and 6 are also worth reading.

If you want to run sample code from the Python tutorial, you can do so as follows:

- Open VS Code.
- Choose File > Open Folder… to select a folder where you'd like to work (can be an empty scratch folder or the folder where you downloaded the switch data earlier).
- Choose Terminal > New Terminal.

- Type “python<enter>” in the terminal pane to start the Python interpreter.
- Type code from the Python tutorial into the interpreter as needed. Generally the code marked with “>>>” or “...” is code that you can type to the Python interpreter. But you should not copy the >>> or ... symbols themselves.

As noted earlier, if you were unable to get VS Code to communicate with the conda environment, you can instead run all these commands (starting with “python<enter>”) in Terminal.app (Mac) or the Anaconda Prompt (Windows).

2.3 Introduction to Optimization and Pyomo (optional, 1 – 3 hours)

We recommend going through the following sections at https://pyomo.readthedocs.io/ to get an introduction to Pyomo, the optimization software used by Switch.

- Pyomo Overview
- Pyomo Modeling Components

This will enable you to read and write Switch code, which is just Pyomo code applied to power system modeling. i.e., a Switch model is a Pyomo AbstractModel used to optimize the design of a power system.

Notation: In the Pyomo introduction, you will see problems with a notation like this:

min $c_1x_1 + c_2x_2$
s.t.
$a_{11}\ x_1 + a_{12}\ x_2 \geq b_1$
$a_{21}\ x_1 + a_{22}\ x_2 \geq b_2$
$x_1 \geq 0$
$x_2 \geq 0$

This is a common way to describe mathematical optimization problems. It means "find values for x_1 and x_2 that will minimize the value of $c_1x_1 + c_2x_2$, such that all the specified constraints are satisfied."

In this problem, the *x* values are called **decision variables** (these are numbers that will be chosen when the problem is run, e.g., the amount of power to produce from a project during a particular hour), the *a*, *b*, and *c* values are **parameters** (data you know when you set up the problem, e.g., the maintenance cost per MWh produced from a project), $c_1x_1 + c_2x_2$ is the **objective function** (the value to be minimized or maximized), and the other equations are the **constraints** (e.g., that power output is less than or equal to installed capacity).

Now you are ready for the Switch tutorial.