Nexus-e

Release 0.0.1

Nexus-e

Apr 17, 2023

CONTENTS

Nexus-e

CHAPTER

ONE

SETUP

1.1 Get permissions

Before setting up, ask for permission from nexus-e@ethz.ch to access the codes and the input database.

Note: Currently, the codes and databases are only accessible with an ETH account.

1.2 Local setup

Even though the current Nexus-e platform is **not** yet suitable to run locally because of high consumption of resources such as memory, we do recommend to at least get the codes and connect to the database locally in order to view, understand, and edit them easily.

Most the instructions below have been tested both on **Windows** and on **Mac**.

1.2.1 1. Get the codes

Make sure you have installed git.

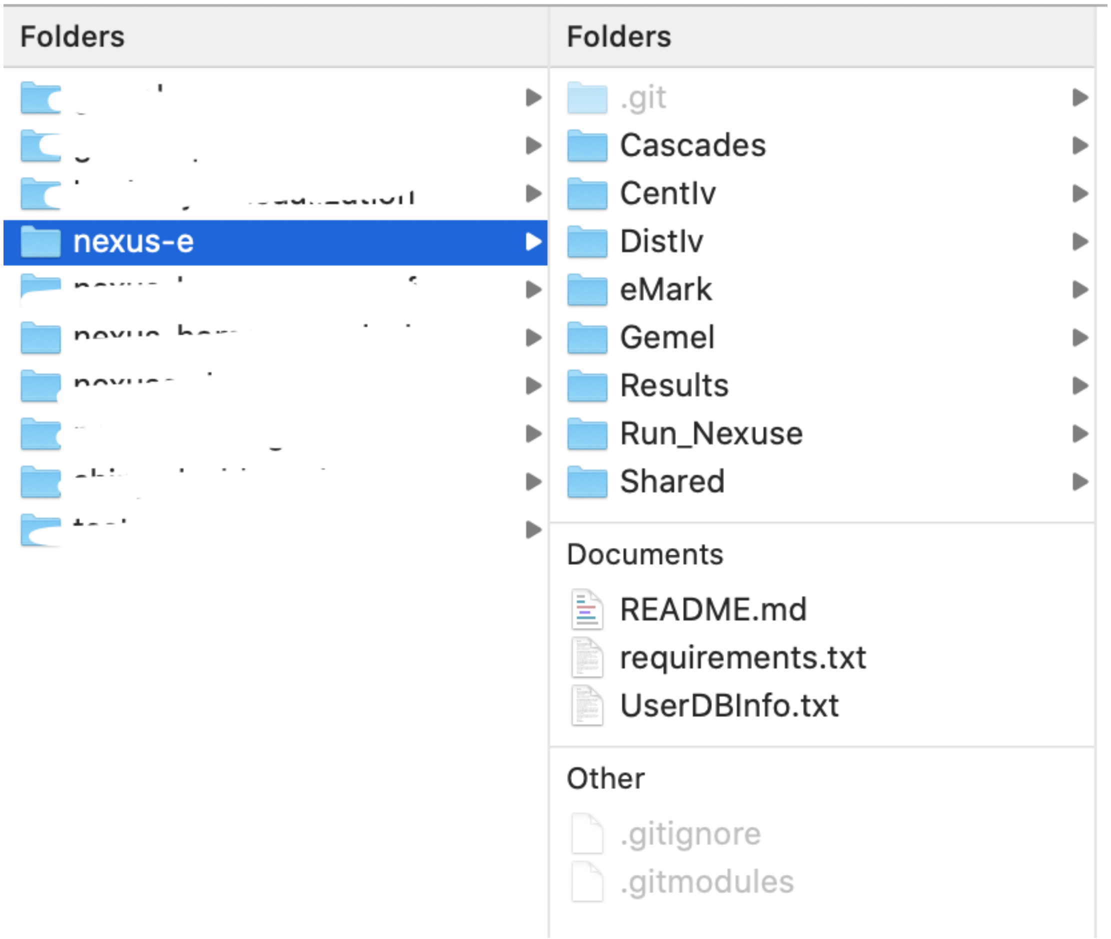
The instructions here use **command lines**. If you have a GUI tool for git, you could use the corresponding operations in the GUI tool instead. An example using a GUI tool “GitKraken” can be found in section Nexus-e repository instructions.

- Create a folder where you want to put the codes
- cd to that folder
- Clone the overarching repository with this command:

  ```
  git clone --recursive https://gitlab.ethz.ch/nexus-e/nexus-e-framework.git
  ```

  You will be asked for your username and password for the repository and its submodules.

In the end, the folder will look like the screenshot below. Note: You might not see the hidden files whose names start with a dot, and the `Results` folder will be automatically generated when you run Nexus-e.



Folders

[redacted]

nexus-e

[redacted]

Folders

- .git
- Cascades
- Centlv
- Distlv
- eMark
- Gemel
- Results
- Run_Nexuse
- Shared

Documents

- README.md
- requirements.txt
- UserDBInfo.txt

Other

- .gitignore
- .gitmodules

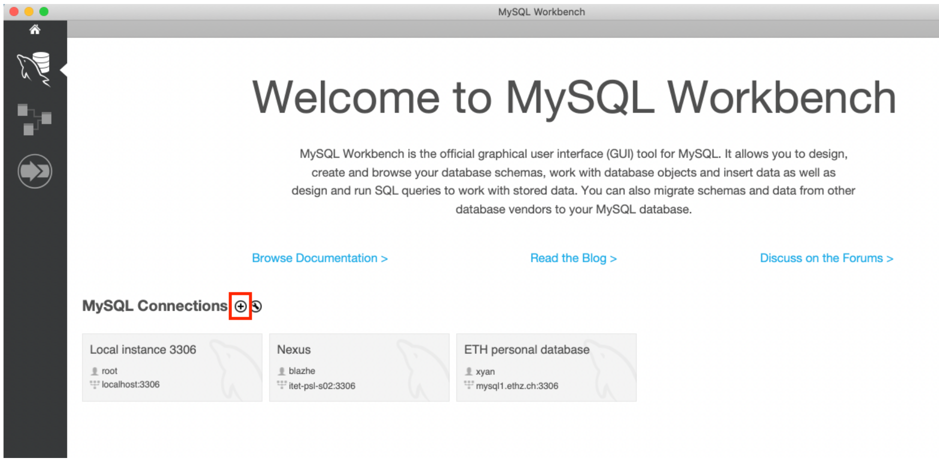
1.2.2 2. Connect with the input database

- Find information to connect with the database in `UserDBInfo.txt` in the repository. The file contains four lines (for confidentiality the username and password is not written here, but can be found in the file):

```
itet-psl-s02
3306
DATABASE_USERNAME
DATABASE_PASSWORD
```

- View the database
  - Install MySQL (Version 8.0 is recommended) and MySQL Workbench. For Windows, follwing this instruction. For MacOS, follow this instruction.
  - Connect to ETH VPN
  - Connect to the database with MySQL Workbench

- Add a new MySQL connection



MySQL Workbench

Welcome to MySQL Workbench

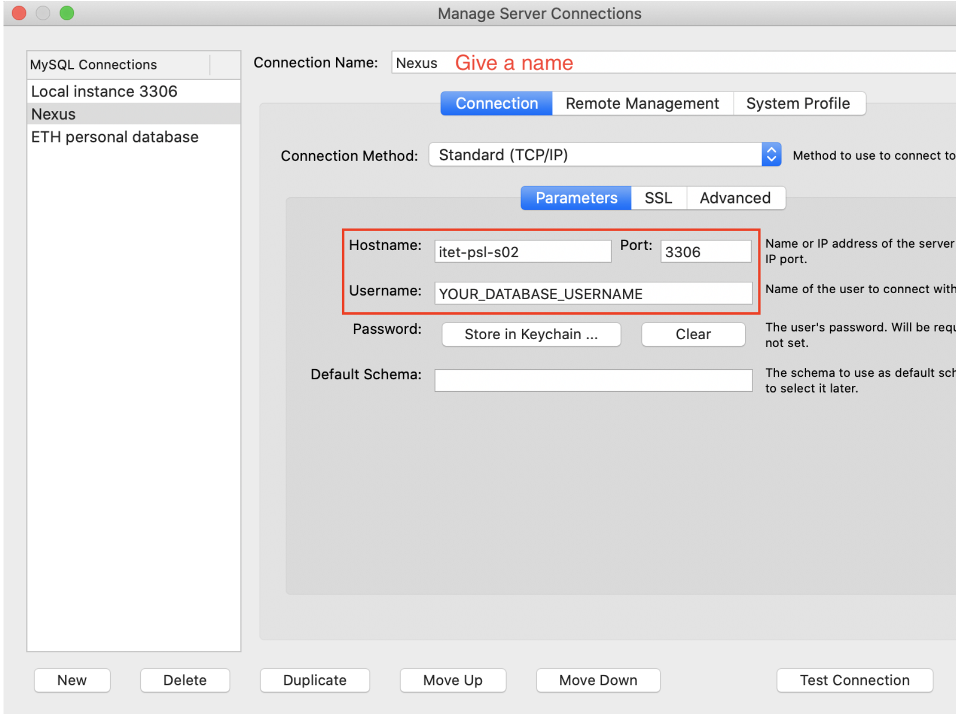
MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

Browse Documentation > Read the Blog > Discuss on the Forums >

MySQL Connections ⊕ ⊘

Local instance 3306	Nexus	ETH personal database
root localhost:3306	blazhe itet-psl-s02:3306	xyan mysql1.ethz.ch:3306

- Input connection information



Manage Server Connections

MySQL Connections
- Local instance 3306
- Nexus
- ETH personal database

Connection Name: Nexus Give a name

Connection Method: Standard (TCP/IP) Method to use to connect to

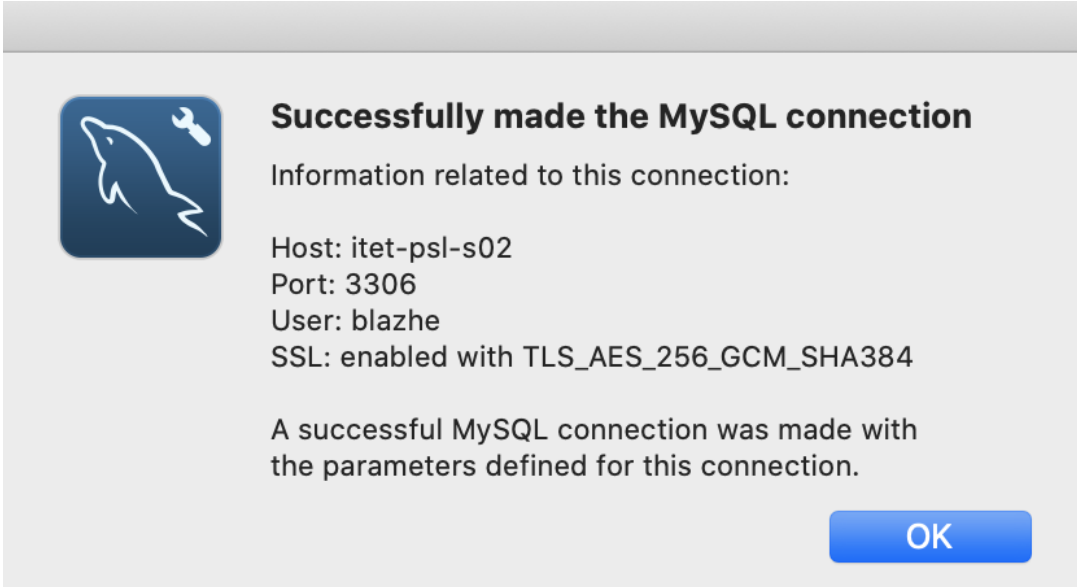
Hostname: itet-psl-s02 Port: 3306 Name or IP address of the server IP port.

Username: YOUR_DATABASE_USERNAME Name of the user to connect with

Password: Store in Keychain ... Clear The user's password. Will be requ not set.

Default Schema: The schema to use as default sch to select it later.

- Click the button `Test Connection`. You will be asked to input your database password. Then you will see a seccess message:



Successfully made the MySQL connection

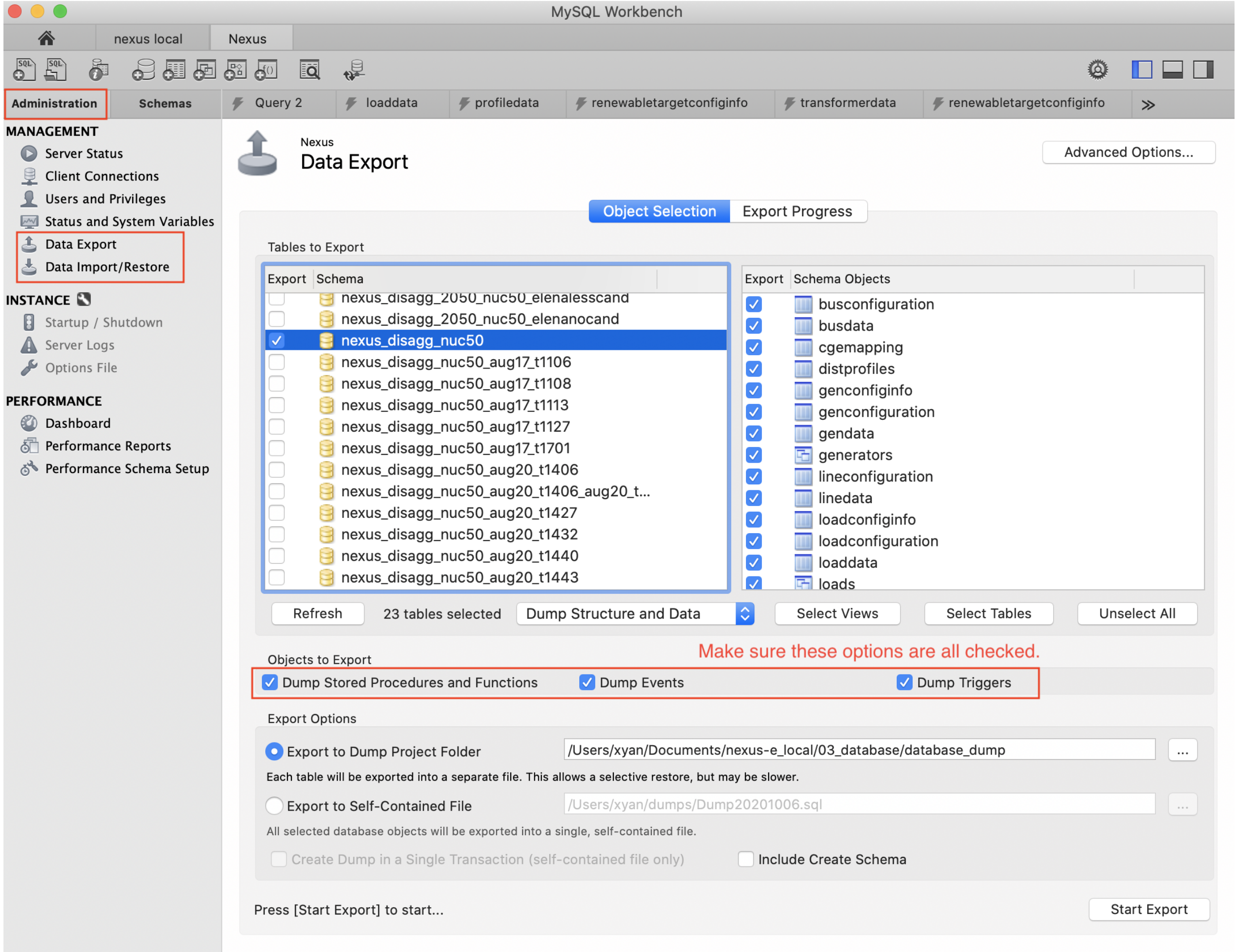
Information related to this connection:

Host: itet-psl-s02
Port: 3306
User: blazhe
SSL: enabled with TLS_AES_256_GCM_SHA384

A successful MySQL connection was made with the parameters defined for this connection.

1.2.2.1 2.1. (Optional) Make a local copy of the database

By default, everyone has write permission to the database. Therefore, to be safe, always make a local copy of the database before playing with it - i.e., first “Dump” it to a local folder, then import it to your personal MySQL account. These can be done in this window:



Tables to Export

Export	Schema
□	nexus_disagg_2050_nuc50_elenalesscand
□	nexus_disagg_2050_nuc50_elenanocand
☑	nexus_disagg_nuc50
□	nexus_disagg_nuc50_aug17_t1106
□	nexus_disagg_nuc50_aug17_t1108
□	nexus_disagg_nuc50_aug17_t1113
□	nexus_disagg_nuc50_aug17_t1127
□	nexus_disagg_nuc50_aug17_t1701
□	nexus_disagg_nuc50_aug20_t1406
□	nexus_disagg_nuc50_aug20_t1406_aug20_t...
□	nexus_disagg_nuc50_aug20_t1427
□	nexus_disagg_nuc50_aug20_t1432
□	nexus_disagg_nuc50_aug20_t1440
□	nexus_disagg_nuc50_aug20_t1443

Export	Schema Objects
☑	busconfiguration
☑	busdata
☑	cgemapping
☑	distprofiles
☑	genconfiginfo
☑	genconfiguration
☑	gendata
☑	generators
☑	lineconfiguration
☑	linedata
☑	loadconfiginfo
☑	loadconfiguration
☑	loaddata
☑	loads

23 tables selected

Dump Structure and Data

Objects to Export

Make sure these options are all checked.

☑ Dump Stored Procedures and Functions ☑ Dump Events ☑ Dump Triggers

Export Options

● Export to Dump Project Folder /Users/xyan/Documents/nexus-e_local/03_database/database_dump

Each table will be exported into a separate file. This allows a selective restore, but may be slower.

○ Export to Self-Contained File /Users/xyan/dumps/Dump20201006.sql

All selected database objects will be exported into a single, self-contained file.

□ Create Dump in a Single Transaction (self-contained file only) □ Include Create Schema

Press [Start Export] to start...

You might encounter the following problems when you run Nexus-e with a local database, follow the link for possible solutions:(1) Error: "Access denied for user 'your_username'@'your_host_address' . . ."(2) Error: "The server time zone value 'xxx' is unrecognized . . ."(3) Error: "The user specified as a definer ('xxx'@'%') does not exist . . ."

1.2.3 3. Prepare software & license

Apart from MySQL introduced above, we also need the following software to run Nexus-e locally. This will be introduced in this section. You could skip this section if you don't plan to run the whole Nexus-e platform locally.

- Matlab
- Python
- Gurobi
- GAMS
- Latex
- ImageMagick

1.2.3.1 Matlab

- Download Matlab (R2020a or higher). Available at ETH IT shop.
- During installation, you need to install the "Database Toolbox", the "Financial Toolbox", and the "Communication Toolbox" as well by ticking the box for each toolbox (This option is available if you downloaded Matlab through the ETH IT shop). If this is not an option during installation, you can download it seperately here. If possible install all toolboxes to avoid any inconveniences.
- Set up database connector
  - Download the **MySQL connector** from here. **Unzip** the file. Before downloading, select "Platform Independent" from the "Select Operating System" drop-down list.
  - Copy the MySQL connector folder (e.g. mysql-connector-java-8.0.18) into a folder at your preference. We recommend to put it into the **Matlab preferences folder**, which you can find by typing in Matlab Command Window `prefdir`.
  - Create a **javaclasspath.txt** file in the **Matlab preferences folder**.
  - In the **javaclasspath.txt** file, write the path to the connector .jar file that you just copied. E.g., On a Windows computer the path is similar to `C:\Users\user\AppData\Roaming\MathWorks\MATLAB\R2018a\mysql-connector-java-8.0.18\mysql-connector-java-8.0.18.jar`.
  - Reload Matlab
  - Test whether a database connector is set up successfully. Write the following commands in Matlab (substitue `YOUR_USERNAME` and `YOUR_PASSWORD` with your credentials for the database). If the second command returns `1`, it means success.

```
conn = database("sys", YOUR_USERNAME, YOUR_PASSWORD, 'Vendor', 'MySQL', 'Server',
↪'itet-psl-s02');
isopen(conn)
```

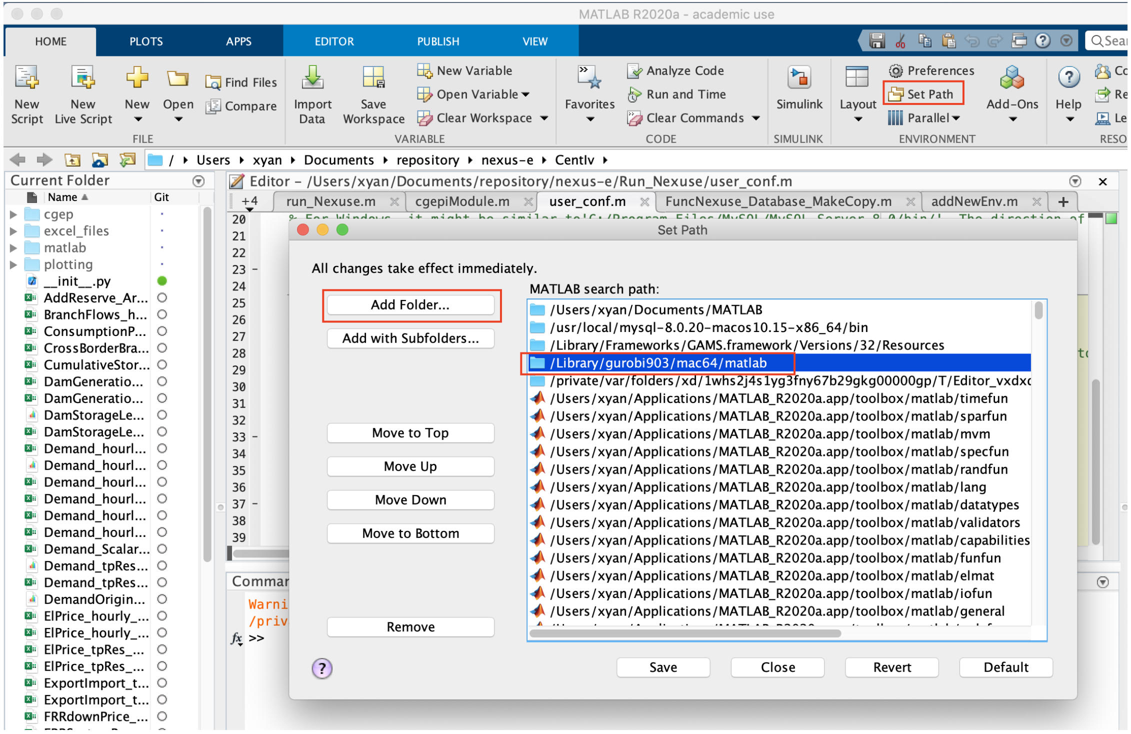
1.2.3.2 Python

- Download Python. Version 3.8.6 is recomended. Python 3.9 is not compatible with Gurobi 9.0.x yet.
- (Optional) If you (will) use Python for multiple projects, it is recommended to create a virtual environment for Nexus-e (for example, using virtualenv).
- Install the required python packages listed in `requirements.txt` (e.g., with command `pip install -r requirements.txt`).

1.2.3.3 Gurobi

- Download Gurobi (9.0 is recomended) and set up a free Gurobi academic license from here.
- Connect Gurobi with Python following this instruction.
- Save Gurobi path for Matlab
  - Find out the path of `gurobi_setup.m`. (E.g., for mac, it is similar to `/Library/gurobi903/mac64/matlab`.)
  - Open Matlab; under the tab "Home", click **Set Path**; select **Add Folder…**; browse to the folder containing `gurobi_setup.m`; click **Open**.

– A new entry of the selected path will appear on the right side of the **Set Path** window; click **Save**.



Editor – /Users/xyan/Documents/repository/nexus-e/Run_Nexuse/user_conf.m

All changes take effect immediately.

MATLAB search path:

/Users/xyan/Documents/MATLAB
/usr/local/mysql-8.0.20-macos10.15-x86_64/bin
/Library/Frameworks/GAMS.framework/Versions/32/Resources
/Library/gurobi903/mac64/matlab
/private/var/folders/xd/1whs2j4s1yg3fny67b29gkg00000gp/T/Editor_vxdxc
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/timefun
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/sparfun
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/mvm
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/specfun
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/randfun
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/lang
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/datatypes
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/validators
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/capabilities
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/funfun
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/elmat
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/iofun
/Users/xyan/Applications/MATLAB_R2020a.app/toolbox/matlab/general

1.2.3.4 GAMS

- Download GAMS. Version 32.2 is recomended. Our current license doesn't support versions newer than 32.
- Set GAMS License
  - Use the `gamslice.txt` provided in polybox `02_Model/02_Model_Setup_Instruction`.
  - Install the license: here or here or (for MacOS) here.
- Save GAMS path for Matlab
  - Find out your GAMS path. It varies largely depending on your PC's operating system and the version of GAMS. For example, GAMS 32 on Mac has the path: `/Library/Frameworks/GAMS.framework/Versions/32/Resources`.
  - Similar to how you configured Gurobi for Matlab: Open Matlab; under the tab "Home", click **Set Path**; select **Add Folder…**; browse to the GAMS path; click **Open**.
  - A new entry of the selected path will appear on the right side of the **Set Path** window; click **Save**.
- Connect GAMS with Python
  - Follow the instruction here.

1.2.3.5 Latex

- Download Latex.

1.2.3.6 ImageMagick

- Instructions for Windows:
  - Download ImageMagick from here. Install ImageMagick from the .exe file. Tick **all** boxes when asked which packages to be installed (by default only the first three are selected).
  - Download dependable software "Ghostscript" from here. Select Public License Ghostscript. Install the Ghostscript.
  - Optional (if ImageMagick does not work), download and install Visual studio from here.
- Instructions for Mac:
  - Install ImageMagick from here
- Test if ImageMagick works:
  - Open Windows command prompt and browse to folder that contains a pdf file, e.g., testch.pdf
  - Use the following command to convert file: `convert testch.pdf testch.jpg`

1.2.4 4. Edit conf_local.m

- Make a copy of `example_conf_local.m` in the folder `Run_Nexuse/conf_local`. Name the copy as `conf_local.m`. (Note: You should only work with the local copy, because the original `example_conf_local.m` will be synchronized to git and it shouldn't be changed.)
- In `conf_local.m` uncomment and modify the following variables based on your local computer's settings. More instructions can be found in the script `conf_local.m` itself.
  - PATH_MYSQL
  - PATH_PYTHON
  - PATH_GUROBI_FOR_PYTHON
  - PATH_LATEX
  - PATH_CONVERT
  - PATH_GAMS

Note: You don't necessarily need all software paths here. For example, if you only want to run the postprocess scripts, `PATH_GUROBI_FOR_PYTHON` is not needed; if you only want to run DBcreation scripts, only `DB_INFO_FILE` and `PATH_MYSQL` are needed.

- (Optional) If you want to work with a *local copy of the database*
  - Make a copy of `example_UserDBInfo_local.txt` in the folder `Run_Nexuse/conf_local`. Name the copy as `UserDBInfo_local.txt`. (Note: You should only work with the local copy, because the original `example_UserDBInfo_local.txt` will be synchronized to git and it shouldn't be changed.)
  - Edit `example_UserDBInfo_local.txt` based on your local settings:

```
127.0.0.1 (or "localhost")
3306
YOUR_LOCAL_USERNAME (e.g. "root")
YOUR_LOCAL_PASSWORD
```

  - Uncomment the variable `DB_INFO_FILE` in `conf_local.m`.
- Test `conf_local.m`: in Matlab, run the script `conf_local/conf_test.m`.
  - If your `conf_local.m` is set up correctly, you should see outputs like "xxx works."
  - If you see "xxx FAILED", there will be information in the output showing why xxx failed. However, this is not necessarily a problem. Because as mentioned above, you don't always need all software. For example, if you only want to run postprocess scripts, `PATH_GUROBI_FOR_PYTHON` is not needed; if you only want to run DBcreation scripts, only `DB_INFO_FILE` and `PATH_MYSQL` are needed.

1.2.5 5. Run Nexus-e

- Connect to ETH VPN (in order to connect with the database)
- Open Matlab
- (Optional) Run `bench_Nexuse.m`. This is required before you run `run_Nexuse.m` for the first time, or after major changes have been made to eMark and/or CentIv. Running `bench_Nexuse.m` will create a file `/Gemel/results/benchmark_2015.mat` which is necessary to run `run_Nexuse.m`.
- Run the script `run_Nexuse.m` in the `Run_Nexuse` folder. There are a few parameters that you can customize in the beginning of this run script:
  - You could change the variables `scen`, `scenShortName`, `tpRes`, `limDifference`, and `dbSuffix` as instructed in the script
  - If you don't want to run all 4 years, adjust `endY` for the case you are running. E.g., if endY=staY, it will only run one year.
- Consult the Nexus-e team if you want to run other scripts in the `Run_Nexuse` folder.

1.3 Software for interacting with Euler from local machine

In our experience, it is usefull to have two tools available for

- transferring files between Euler and your local machine and
- interacting with the command prompt on Euler.

GUIs for file transfer

For transferring files between Euler and your personal computer, a graphical user interface (GUI) is useful. We suggest FileZilla since it is available for "all" platforms. The scicomp wiki also has information on this.

Command prompts for secure shell connections

Unix systems (Linux/Mac) come with shells (command prompts) from which useful interactions with Euler are possible. E.g., type `ssh <username>@euler.ethz.ch` to make a "secure shell" connection to Euler and issue commands to the Euler command prompt. Windows users are less fortunate and have to install additional software to achieve this functionality. Examples are

- putty.exe and
- MobaXterm.

We can highly recommend MobaXTerm.

Login

- Log-in with `ssh` on Unix-based operating systems:

  ```
  ssh <ethz-username>@euler.ethz.ch
  ```

  Lacking SSH keys, ssh will ask you for your ETH (LDAP) password and after that your following commands are issued to Euler. See scicomp wiki for instructions for setting up SSH keys if desired.

- File transfers between local and Euler in FileZilla can be made through `sftp` (secure file transfer protocol). For this, enter `sftp://euler.ethz.ch` in the 'server' field and your ETH-user name and password in the respective fields, then hit 'connect'.

- Access to Euler: since May 2020, first time access to Euler is only granted upon request. ETH account holders just write an email to cluster-support@id.ethz.ch; If you used the Euler cluster before the 15. May 2020, then you would need to change your LDAP password to get again access to the Euler cluster (wait for 1 day until the change takes effect). (see https://scicomp.ethz.ch/wiki/Getting_started_with_clusters for more info).

1.4 Euler setup

In order to run the full Nexus-e platform efficiently, we use Euler, an ETH cluster for High Performance Computing. To set up the Nexus-e platform on Euler, you first need to access Euler. Instructions on accessing Euler and commands for Euler can be found here.

1.4.1 1. Join the Nexus-e user group

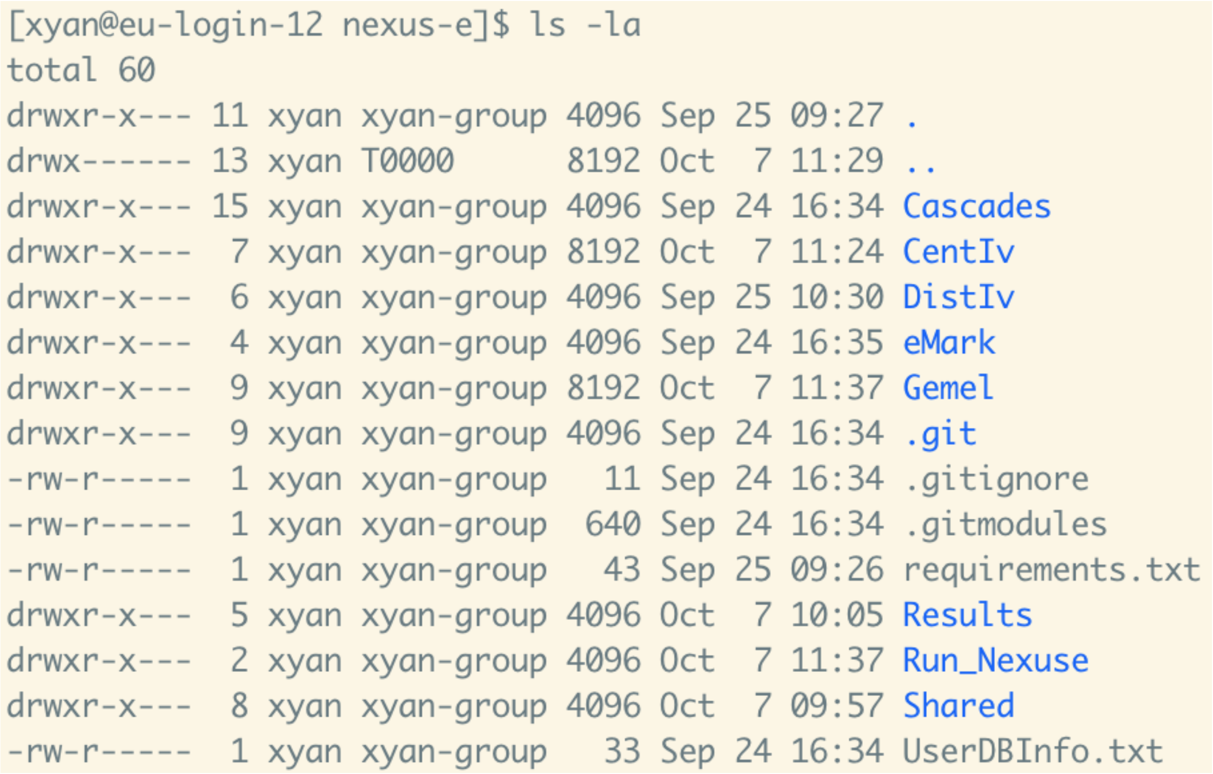
We have a user group for access to Nexus-e specific licenses (e.g. GAMS) and the Euler pre-paid share. The user group is called: **MAVT-esc-nexus-e**, and is set and maintained by D-MAVT IT. To join the user group, ask Blazhe Gjorgiev gblazhe@ethz.ch to send an email to `servicedesk@mavt.ethz.ch`.

To check whether you are added to the user group, use command line ssh to Euler (type `ssh username@euler.ethz.ch`), then type `groups` to show a list of groups that your user account is linked to, which should contain "MAVT-esc-nexus-e".

1.4.2 2. Get the codes

- Create a folder where you want to put the codes
- cd to that folder
- Clone the overarching repository with this command: `git clone --recursive https://gitlab.ethz.ch/nexus-e/nexus-e-framework.git` You will be asked for your username and password for the repository and its submodules.

In the end, the folder will look like the screenshot below. Note: tThe `Results` folder will be automatically generated when you run Nexus-e.



```
[xyan@eu-login-12 nexus-e]$ ls -la
total 60
drwxr-x--- 11 xyan xyan-group 4096 Sep 25 09:27 .
drwx------ 13 xyan T0000      8192 Oct  7 11:29 ..
drwxr-x--- 15 xyan xyan-group 4096 Sep 24 16:34 Cascades
drwxr-x---  7 xyan xyan-group 8192 Oct  7 11:24 CentIv
drwxr-x---  6 xyan xyan-group 4096 Sep 25 10:30 DistIv
drwxr-x---  4 xyan xyan-group 4096 Sep 24 16:35 eMark
drwxr-x---  9 xyan xyan-group 8192 Oct  7 11:37 Gemel
drwxr-x---  9 xyan xyan-group 4096 Sep 24 16:34 .git
-rw-r-----  1 xyan xyan-group   11 Sep 24 16:34 .gitignore
-rw-r-----  1 xyan xyan-group  640 Sep 24 16:34 .gitmodules
-rw-r-----  1 xyan xyan-group   43 Sep 25 09:26 requirements.txt
drwxr-x---  5 xyan xyan-group 4096 Oct  7 10:05 Results
drwxr-x---  2 xyan xyan-group 4096 Oct  7 11:37 Run_Nexuse
drwxr-x---  8 xyan xyan-group 4096 Oct  7 09:57 Shared
-rw-r-----  1 xyan xyan-group   33 Sep 24 16:34 UserDBInfo.txt
```

1.4.3 3. Prepare dependencies

Unlike setting up Nexue-s locally, we don't need to download any software or license on Euler. Instead, they are all prepared for all users in the user group **MAVT-esc-nexus-e**. You should already be a member of the group if you have followed the *first step*.

But still, we need to explicitely load all the dependencies, including modules (an Euler term for "software") and python packages.

- Load modules
  - Option 1: Manually load modules every time before running Nexus-e

  Copy the following commands into Euler:

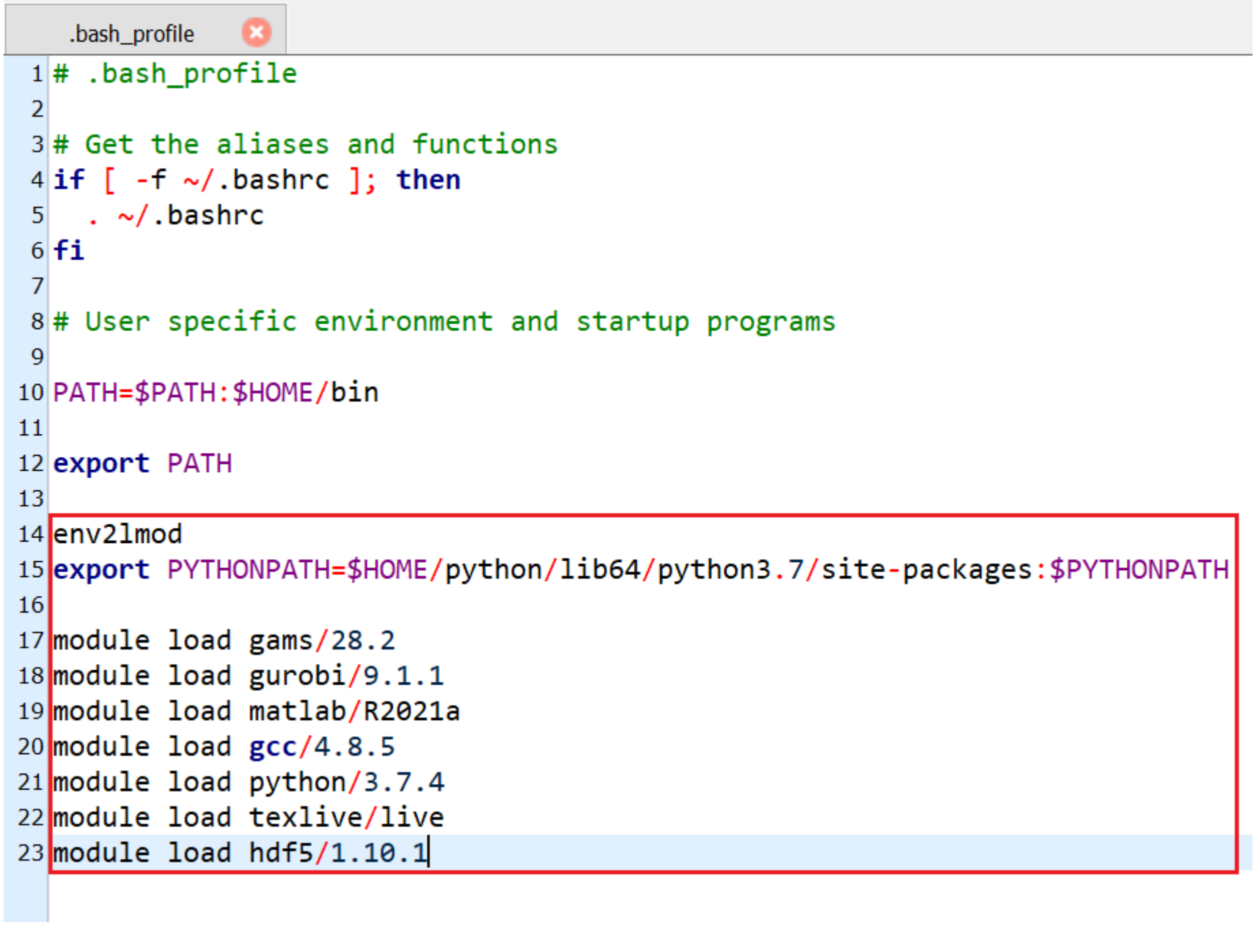
```
env2lmod
export PYTHONPATH=$HOME/python/lib64/python3.7/site-packages:$PYTHONPATH

module load gams/28.2
module load gurobi/9.1.1
module load matlab/R2020b
module load gcc/4.8.5
module load python/3.7.4
module load texlive/live
module load hdf5/1.10.1
```

env2lmod serves as a switch from the old to the new Euler software stack. Alterntievly one can type in `comand set_software_stack.sh new` to permently switch to the new software stack.

- Option 2: Edit `.bash_profile` or `.bashhrc` so that the modules are automatically loaded upon log in to Euler

Write the same commands at the end of `.bash_profile`. This is a hidden file in your Euler's home directory. After adding these commands, your `.bash_profile` should look similar to this:



```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
  . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH

env2lmod
export PYTHONPATH=$HOME/python/lib64/python3.7/site-packages:$PYTHONPATH

module load gams/28.2
module load gurobi/9.1.1
module load matlab/R2021a
module load gcc/4.8.5
module load python/3.7.4
module load texlive/live
module load hdf5/1.10.1
```

Test: To test whether the modules have been successfully loaded, type `module list` in the commaand line. It should then list all loaded modules.

- Install python packages

  This you only need to do it **once**, i.e. you don't need to do it everytime when you run Nexus-e.

  In command line under your home directory,

  - `cd` to the folder where you have stored the Nexus-e codes
  - type `pip install --user -r requirements.txt`

  You might encounter software compatibility problems when running Nexus-e on Euler, follow the link for possible solutions: Matlab & Python compatibility error: "ImportError: …pyexpat…: undefined symbol: XML_SetHashSalt"

1.4.4 4. Connect with the input database

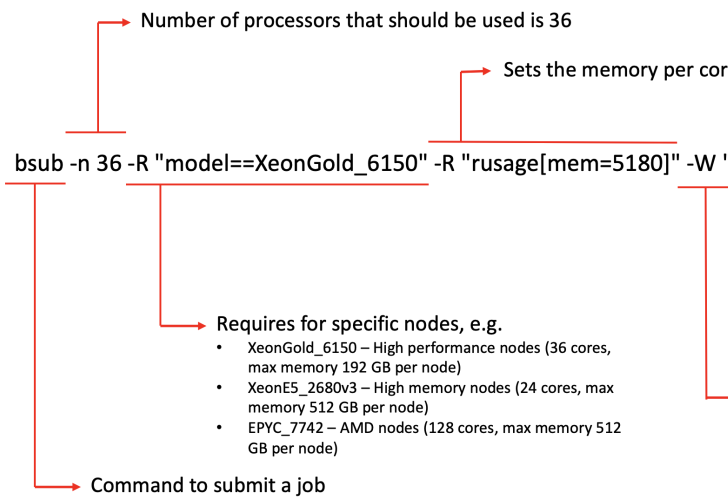
- Download the **MySQL connector** from here. **Unzip** the file.
- Copy the MySQL connector folder into the hidden .matlab folder (e.g., .matlab/2020a).
  - Tip: To see all folders including the hidden ones, use the command `ls -a`.
  - If you don't see the .matlab folder, run Matlab once: type in `matlab` -> you will see Matlab is started in the terminal -> type in `exit` -> Matlab will stop -> you should then see the .matlab folder.
- Create a **javaclasspath.txt** file in the same folder as above.
- In the **javaclasspath.txt** file, write the path to the connector .jar file that you just copied. E.g., On Blazhe's Euler account the path is `/cluster/home/gblazhe/.matlab/R2018a/mysql-connector-java-8.0.18/mysql-connector-java-8.0.18.jar`.
- Logoff and re-login to Euler

1.4.5 5. Run Nexus-e

- `cd` to the `Run_Nexuse` folder
- (Optional) Run `bench_Nexuse.m`. This is required before you run `run_Nexuse.m` for the first time, or after major changes have been made to eMark and/or CentIv. Running `bench_Nexuse.m` will create a file `/Gemel/results/benchmark_2015.mat` which is necessary to run `run_Nexuse.m`.
- Customize the run script `run_Nexuse.m`.
  - You could change the variables `scen`, `scenShortName`, `tpRes`, `limDifference`, and `dbSuffix` as instructed in the script. (Recommendation: For quick test, use `tpRes = 8` and `limDifference = 0.1`.)
  - If you don't want to run all 4 years, adjust `endY` for the case you are running. E.g., if endY=staY, it will only run one year.
- Submit the job (the standard run script is `run_Nexuse.m`) with a command such as (Note: Don't type `.m` after the script name.)

```
bsub -n 36 -R "model==XeonGold_6150" -R "rusage[mem=5180]" -W "10:00" matlab -r run_
↪Nexuse
```

The parameters in this command can be costomized. Experience with resource usage to run Nexus-e with different time resolution and convergence criteria can be found here. General instructions on bash and Euler commands can be found here.



```
bsub -n 36 -R "model==XeonGold_6150" -R "rusage[mem=5180]" -W '
```

Number of processors that should be used is 36
Sets the memory per cor
Requires for specific nodes, e.g.
• XeonGold_6150 – High performance nodes (36 cores, max memory 192 GB per node)
• XeonE5_2680v3 – High memory nodes (24 cores, max memory 512 GB per node)
• EPYC_7742 – AMD nodes (128 cores, max memory 512 GB per node)
Command to submit a job

Here we give a short explanation to the example command above:

After submission, Euler will respond by telling you what the jobID is.

- Consult Nexus-e team if you want to run other scripts in the `Run_Nexuse` folder.

NOTE: For submissions that require large memory, it is recommended to select EPYC_7742, for submission with smaller memory requirements either select EPYC_7742 or XeonGold_6150.

We have experienced issues with the database connector when computing node is not selected on Euler.

CHAPTER

TWO

EULER INSTRUCTIONS

2.1 Useful bash commands

Euler's command line interface (shell) is called "bash". The scicomp wiki gives a good overview over a list of bash commands. We just give a short list of useful applications of these commands:

- `Ctrl-p` and `Ctrl-n` give previous and next commands in the command history.
- `Ctrl-r <expression>` searches for `<expression>` in the command history.
- Use `cd` for **c**hanging **d**irectories. E.g., `cd nexus-e/Run_Nexuse` to go to the directory where the main matlab script `run_Nexuse.m` lives.

  Note: avoid white space in directory or file names **like the plague**.
- Use `ls` for **lis**ting the contents of the current directory
- Use `rm <filename>` for deleting file with name filename
- Use `rm -r <foldername>` for recursively deleting the contents of folder
- Use `pwd` for printing the current working directory
- `grep <expression> <filename>` prints all lines of file(s) `filename` in which `expression` appears. Use option `-i` for case-**i**nsensitive search.
- `<command1> | <command2>` employs a "**pipe**" (`|`) to redirect output of command1 as input to command2. E.g., `ls | grep <filename>` sends the listing of the current directory contents to grep, which looks for lines in which filename appears. If this command does not return anything, filename does not exist in the current directory.
- Wildcards: bash accepts several wildcards for file name or string completion. E.g.,
  - `*` stands for a sequence of characters of arbitrary length
  - `?` stands for one character

  `ls lsf.o14*`, e.g., will list all files with names starting with *lsf.o14*.
- Use `scp` for making **s**ecure (remote) **cop**ies of files and folders. E.g. enter

  ```
  scp ./run_Nexuse.m <username>@euler.ethz.ch:/cluster/home/<username>/
  nexus-e/Run_Nexuse/run_Nexuse.m
  ```

  To transfer a local copy of `run_Nexuse.m` to `~/nexus-e/Run_Nexuse/run_Nexuse.m` on Euler. Issue this command on your local machine!

  This command may come in handy for scripting certain things but is unnecessary if you are happy to make all file transfers using FileZilla.

2.2 Euler specific commands

2.2.1 Account information

To access information about your account on Euler

- `lquota` checks the amount of data and files you have on the cluster
- `busers -w` shows resource usage
- `my_share_info` returns your user group

2.2.2 Modules

Diverse commands exist for organizing and checking the modules that Euler has loaded in your environment. Generally, running Nexus-e should work fine if the modules listed under Setup are loaded. Here's a list anyway (see scicomp wiki for more explanation around the commands).

- `module load new` loads all new modules
- `module list` shows currently loaded modules
- `module avail` shows all available modules
- `module help <module_name>` brief description
- `module show <module_name>` what the module would do
- `module load <module_name>` load a module
- `which icc` check the compiler
- `module unload <module_name>` unload module

2.2.3 Batch system: How to run Nexus-e

On Euler, users are asked to run large jobs using Euler's batch system. Scicomp gives an extensive description of this system. Here we summarize what is useful for running Nexus-e on Euler.

Generally, commands like

```
bsub -n 36 -R "model==EPYC_7742" -R "rusage[mem=5180]" -W "10:00" matlab -r
run_Nexuse_platform
```

will be used to run Nexus-e from folder nexus-e/Run_Nexuse_platform on Euler.

- `-W "10:00"` gives Euler 10 hours to run the process
- `-R "rusage[mem=5180]"` tells Euler to allocate 5180 MB of RAM per core (default is 1GB)
- `-R "model==EPYC_7742"` tells Euler to use the EPYC_7742 nodes Available nodes:
  - **XeonE5_2680v3** High memory nodes (24 cores, max memory 512 GB per node) - Recomended!
  - **XeonGold_6150** High performance nodes (36 cores, max memory 192 GB per node)
  - **EPYC_7742** AMD nodes (128 cores, max memory 512 GB per node) Note that requiring for specific nodes may imply longer queuing times.
- `-n 36` tells Euler to use 36 processors
- `matlab -r run_Nexuse` tells Euler to run the script run_Nexuse using matlab

Other useful options are the following.

- Add `-nojvm` to matlab command to prevent Java from being used.

  ```
  bsub [...] matlab -nojvm -r run_Nexuse
  ```

- Add `-nodisplay` to matlab command to explicitly tell matlab that no graphical interface is available.

  ```
  bsub [...] matlab -nodisplay -r run_Nexuse
  ```

- Set the output filename using `-o <output_filename>`. Default is `lsf.o<JobID>`.
- Add `-B` and/or `-N` to be notified via email when your job begins and/or ends (reference). Also, you need to have a file `.forward` in your home directory containing your email address.

  ```
  bsub -B -N [...] matlab -r run_Nexuse
  ```

- For parallel computation using OpenMP, the number of processor cores available needs to be specified using the environmental variable OMP_NUM_THREADS. Set this with the command

  ```
  export OMP_NUM_THREADS=<number_of_cores>
  ```

  before issuing your `bsub ...` command. Consider writing this to .bash_profile if you don't want to repeat this each session.

- To use job arrays for parallel calculation, use option `-J "arrayname[1-10]" ./program [arguments]`

  If some jobs fail, it's possible to rerun only those:

  ```
  brequeue -e <JOBID>
  ```

2.2.4 Batch system: How to check the status of your jobs

- `bjobs` lists jobs
- `bjobs -p` lists only pending jobs
- `bjobs -l` lists jobs with more details
- `bkill <JOBID>` kills job with JOBID
- `bpeek <JOBID>` checks the output of a particular running job
- `bbjobs <JOBID>` checks resources used
- `bjobs -l -aff` is another method for checking the resources used

2.3 Recommendations for running Nexus-e

2.3.1 Experience with resource usage

(1) [For final run] 2-day time resolution with convergence criterium 0.1 percent

For this, set `tpRes = 2;` and `limDifference = 0.001` in run_Nexuse.m. The following command works fine.

```
bsub -n 36 -R "model==XeonGold_6150" -R "rusage[mem=5180]" -W "350:00" matlab
-r run_Nexuse
```

(2) [For quick test] 8-day time resolution with convergence criterium 10 percent

For this, set `tpRes = 8;` and `limDifference = 0.1` in run_Nexuse.m. The following command works fine.

```
bsub -n 36 -R "rusage[mem=2180]" -W "15:00" matlab -r run_Nexuse
```

(3) 8-day time resolution with convergence criterium 2 percent

For this, set `tpRes = 8;` iand `limDifference = 0.02` in run_Nexuse.m. The following command works fine.

```
bsub -n 36 -R "rusage[mem=2180]" -W "30:00" matlab -r run_Nexuse
```

(4) 8-day time resolution with convergence criterium 0.1 percent

For this, set `tpRes = 8;` and `limDifference = 0.001` in run_Nexuse.m. The following command works fine. (This is still being tested as of October 22 2020).

```
bsub -n 36 -R "rusage[mem=2180]" -W "60:00" matlab -r run_Nexuse
```

(5) [For benchmark] Run the benchmark script bench_Nexuse.m

As instructed in the setup procedure, you sometimes need to run the script `bench_Nexuse.m` to (re-)calibrate GemEl. This script runs rather quickly because it doesn't run the entire energy-economic loop and has a low time resolution. The following command works fine.

```
bsub -n 36 -W "2:00" matlab -r bench_Nexuse
```

2.3.2 Finding your way around output on Euler

After you issue a batch job through `bsub`, Euler will respond by telling you what the jobID is. After completion, the 'standard output' of running Nexus-e will be written to lsf.jobID and can inspected there using the text *editor of your choice*.

While the job is running, however, standard output can be accessed by means of the command

```
bpeek
```

which in turn writes to standard output of your console. To more conveniently browse this, several options exist:

- write the output of bpeek to a file:

  ```
  bpeek>yourfilename.txt
  ```

  and then look at it in the text *editor of your choice*.

- use a pipe ('|') to find lines including certain 'patterns' with grep:

  ```
  bpeek | grep <pattern>
  ```

  for example,

  ```
  bpeek | grep 'maximum difference'
  ```

  will print all lines that contain information about Gemel's convergence criterion in given iterations.

- use a pipe ('|') to display the output of bpeek using the 'less' command:

  ```
  bpeek | less
  ```

  This will bring up an interface that lets the user browse through the output of bpeek without using the mouse. Some basic functions of the interface:

  - press "Space" to go down one page,
  - press "q" to **q**uit the "less" interface,
  - press "/" to enter a pattern to search for.
    - Pressing "n" (n for next) after having searched for a pattern jumps to the next instance of the pattern;
    - pressing "N" jumps to the previous.

  - A relative to “/” (forward search) is “?” (backward search).
  - “G” goes to the end of bpeek’s output, “g” to the beginning.

2.3.2.1 Finding stuff in lsf.o<JobID> files

- `grep -i <expression> lsf*` searches for `<expression>` in all files starting with lsf. E.g.,

  `grep -inH nexus_disagg_nuc50_oct20 lsf*`

  searches for copies of `database nexuse_disagg_nuc50` mentioned in lsf* files that where made on October 20. This may be helpful for identifying the copies of the database that can safely be removed (“dropped”) from the PSL server.

2.4 Viewing output

The output of Matlab is directed to files `lsf.o<JobID>`. To inspect this output, open it in your editor of choice.

- Several editors are available on Euler, e.g.,
  - emacs
  - vim
  - nano

  nano may be a good choice if you don’t know emacs or vim (if you do know emacs or vim, you will have strong opinions on which one to use). Nano gives some on-screen instructions on basic key combinations. (E.g., `^G` means *type g while holding Ctrl pressed*).
- For better user experience, copy files to your local computer for viewing with a GUI editor (e.g., using FileZilla).

CHAPTER THREE

REPOSITORY INSTRUCTIONS

For a full instruction to work with git repository, please see here.

This document is only used to demonstrate git operations for certain nexus-e-related questions and use cases:

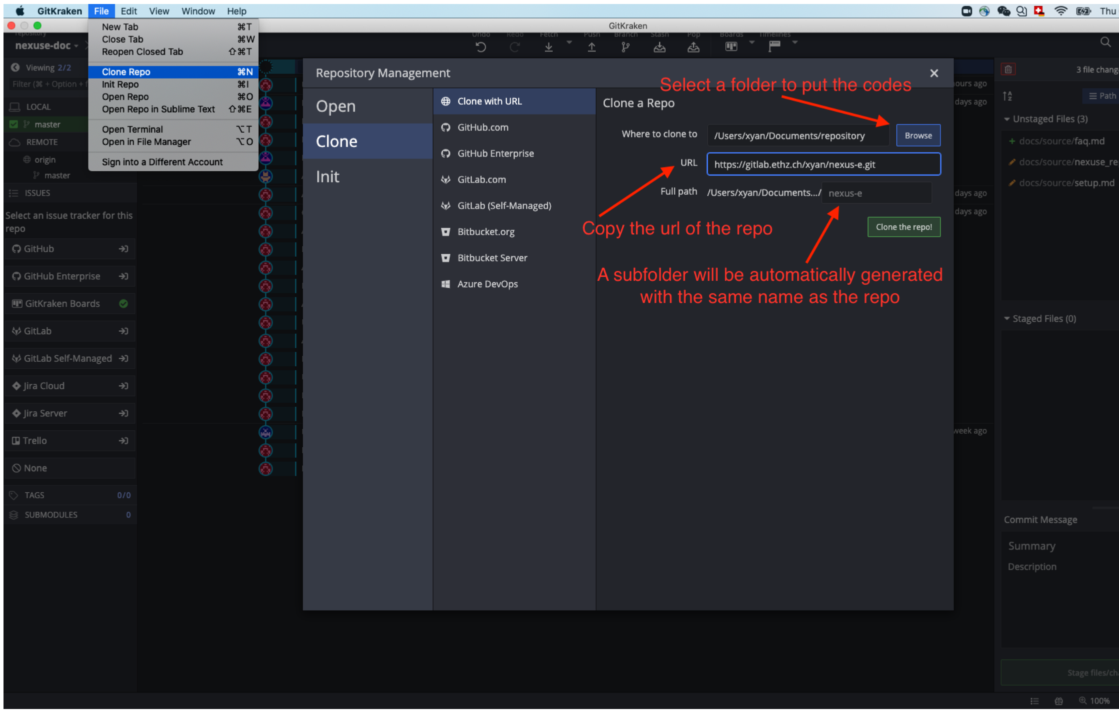
- *How do I get the repository?*
  - Short answer: you "git clone" it from https://gitlab.ethz.ch/nexus-e/nexus-e-framework.git to a local folder.
- *What is the structure of the repository?*
  - Short answer: it is an overarching repository called "nexus-e" containing sub-repositories ("submodules" is the official term in git).
- *I want to make changes in my own module. What should I do first?*
  - Short answer: be aware of where you are in other modules. If needed, pull changes from other submodules.
- *I have made some changes in my own module and tested them in a small scale. How do I test them with the full Nexus-e platform?*
  - Short answer: depending on the scale of your change - if small, you might want to copy the changed file(s) to Euler; if large, you might want to push the changes first then pull it on Euler.
- *I have fully tested my changes in my own module. What should I do now?*
  - Short answer: push what you have tested (maybe not only your own module) into the overarching repo.

We recommend to use a git GUI (e.g. GitKraken, Git Extensions, SourceTree, and many more) to work with git locally. However, Euler doesn't have a GUI, therefore you have to use command line there. In this documentation, we demonstrate with the GUI "GitKraken". You can get a pro account for free as a researcher. The screenshots are taken from MacOS. There should be equivalent operations on other operating systems.

Why GitKraken:

- It is stable in all operating systems (Linux, Mac, Windows)
- It provides a good overview for the status of submodules, which is the key structure of our Nexus-e repo.
- It has an embeded & very handy merging conflict resolution tool.
- It is mature and well-documented.

3.1 How do I get the repository?



Clone a Repo

Where to clone to: /Users/xyan/Documents/repository

URL: https://gitlab.ethz.ch/xyan/nexus-e.git

Full path: /Users/xyan/Documents.../ nexus-e

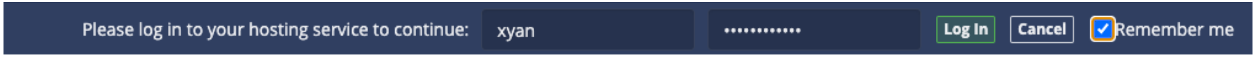
Select a folder to put the codes

Copy the url of the repo

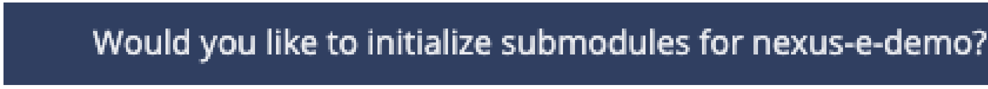
A subfolder will be automatically generated with the same name as the repo

(1) Clone the overarching repo

(2) Put in your credentials for the repo (normally your ETH account)



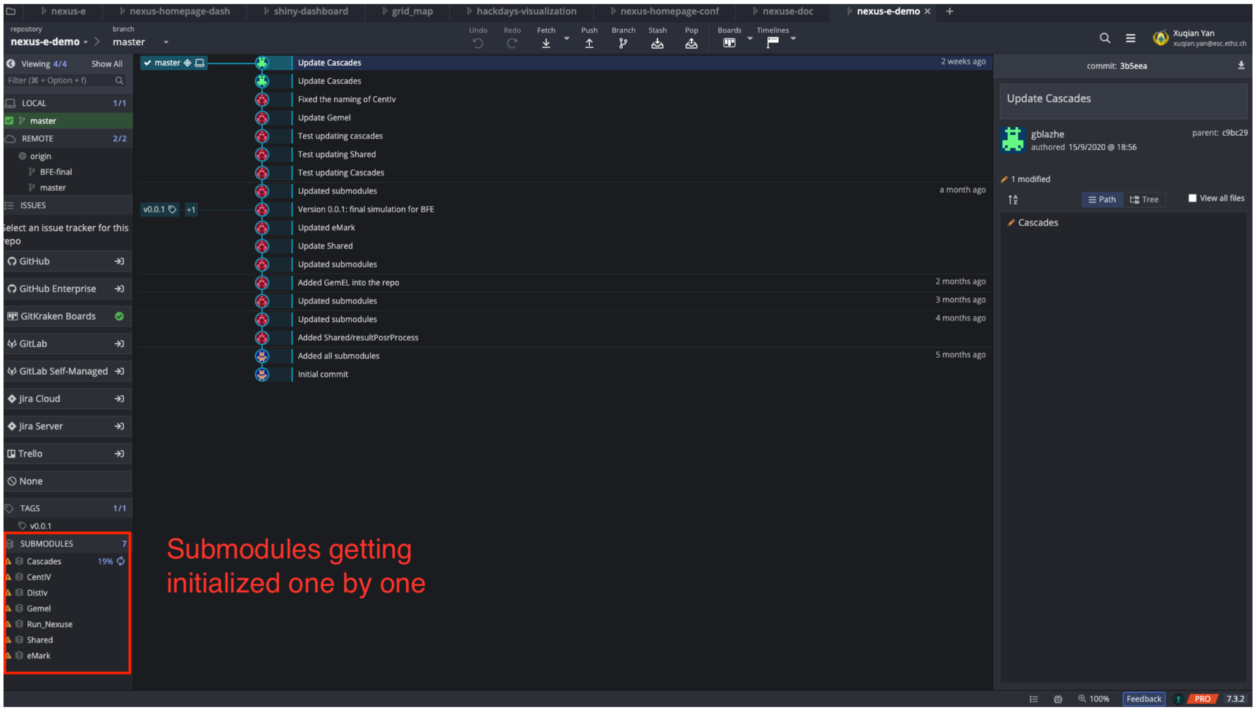
Please log in to your hosting service to continue: xyan ············ Log In Cancel ☑ Remember me



Would you like to initialize submodules for nexus-e-demo?

(3) Initialize the submodules (select "Yes")

Afterwards, you will see submodules getting initialized one by one (In the screenshot below there are 7 submodules. Since January 2021, it is reduced to 5 submodules.):



Commit message	Time
Update Cascades	2 weeks ago
Update Cascades	
Fixed the naming of CentIv	
Update Gemel	
Test updating cascades	
Test updating Shared	
Test updating Cascades	
Updated submodules	a month ago
Version 0.0.1: final simulation for BFE	
Updated eMark	
Update Shared	
Updated submodules	
Added GemEL into the repo	2 months ago
Updated submodules	3 months ago
Updated submodules	4 months ago
Added Shared/resultPosrProcess	
Added all submodules	5 months ago
Initial commit	

commit: 3b5eea

Update Cascades

gblazhe authored 15/9/2020 @ 18:56 parent: c9bc29

1 modified

Cascades

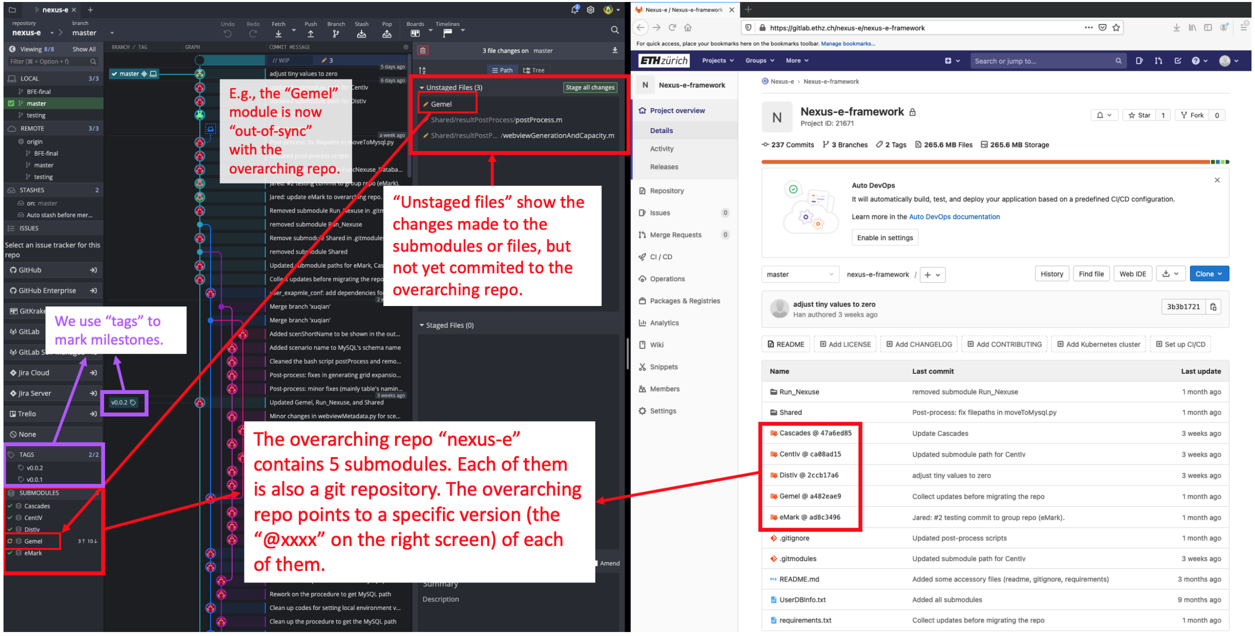
SUBMODULES 7

- Cascades 19%
- CentIV
- DistIv
- Gemel
- Run_Nexuse
- Shared
- eMark

Submodules getting initialized one by one

3.2 What is the structure of the repository?

To help with understanding, we show both interfactes in GitKraken (left) and in the repository (right).



E.g., the "Gemel" module is now "out-of-sync" with the overarching repo.

"Unstaged files" show the changes made to the submodules or files, but not yet commited to the overarching repo.

We use "tags" to mark milestones.

The overarching repo "nexus-e" contains 5 submodules. Each of them is also a git repository. The overarching repo points to a specific version (the "@xxxx" on the right screen) of each of them.

Unstaged Files (3)

- Gemel
- Shared/resultPostProcess/postProcess.m
- Shared/resultPostP... /webviewGenerationAndCapacity.m

Staged Files (0)

TAGS 2/2

- v0.0.2
- v0.0.1

SUBMODULES

- Cascades
- CentIv
- DistIv
- Gemel
- eMark

Nexus-e-framework

Project ID: 21671

237 Commits 3 Branches 2 Tags 265.6 MB Files 265.6 MB Storage

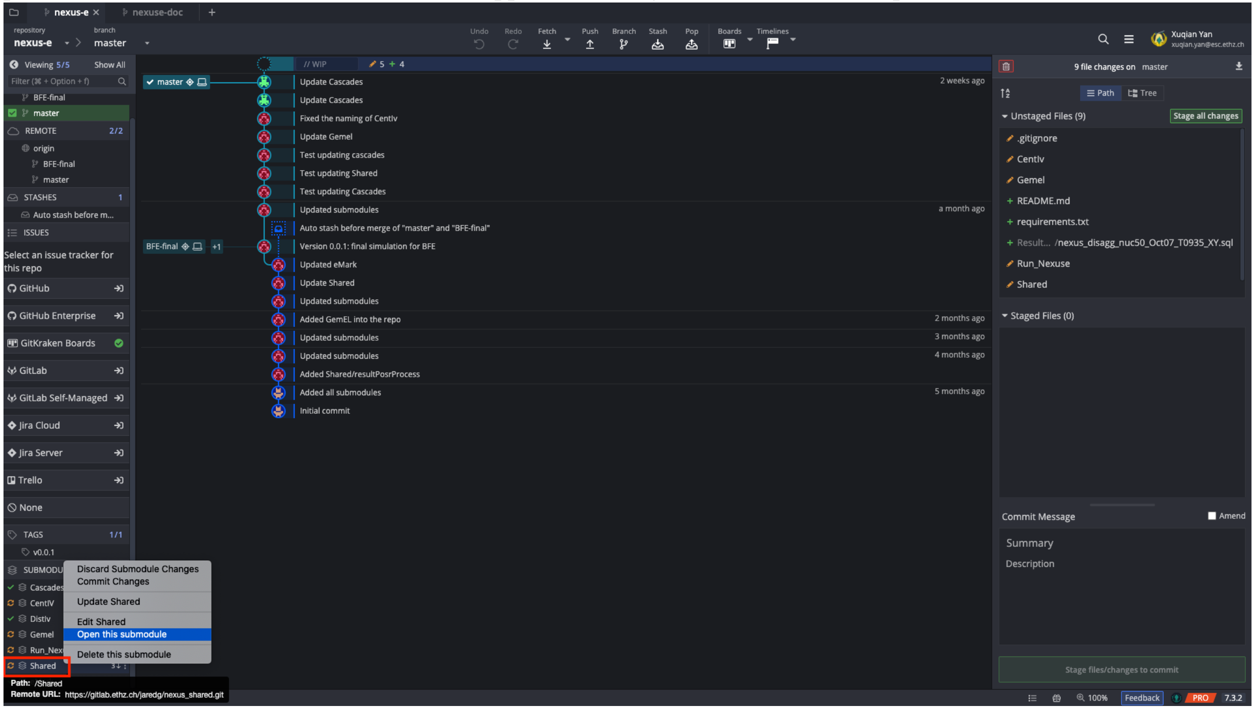
adjust tiny values to zero

Han authored 3 weeks ago 3b3b1721

Name	Last commit	Last update
Run_Nexuse	removed submodule Run_Nexuse	1 month ago
Shared	Post-process: fix filepaths in moveToMysql.py	1 month ago
Cascades @ 47a6ed85	Update Cascades	3 weeks ago
CentIv @ ca08ad15	Updated submodule path for CentIv	3 weeks ago
DistIv @ 2ccb17a6	adjust tiny values to zero	3 weeks ago
Gemel @ a482eae9	Collect updates before migrating the repo	1 month ago
eMark @ ad8c3496	Jared: #2 testing commit to group repo (eMark).	1 month ago
.gitignore	Updated post-process scripts	1 month ago
.gitmodules	Updated submodule path for CentIv	3 weeks ago
README.md	Added some accessory files (readme, gitignore, requirements)	3 months ago
UserDBInfo.txt	Added all submodules	9 months ago
requirements.txt	Collect updates before migrating the repo	1 month ago

3.3 I want to make changes in my own module. What should I do first?

For example, you want to make changes in the "Shared" module. Note: Since January 2021, "Shared" is not a submodule anymore but a folder directly under the Nexus-e repository, but the procedures demonstrated here can be applied to any submodules.(1) Open the submodule "Shared".

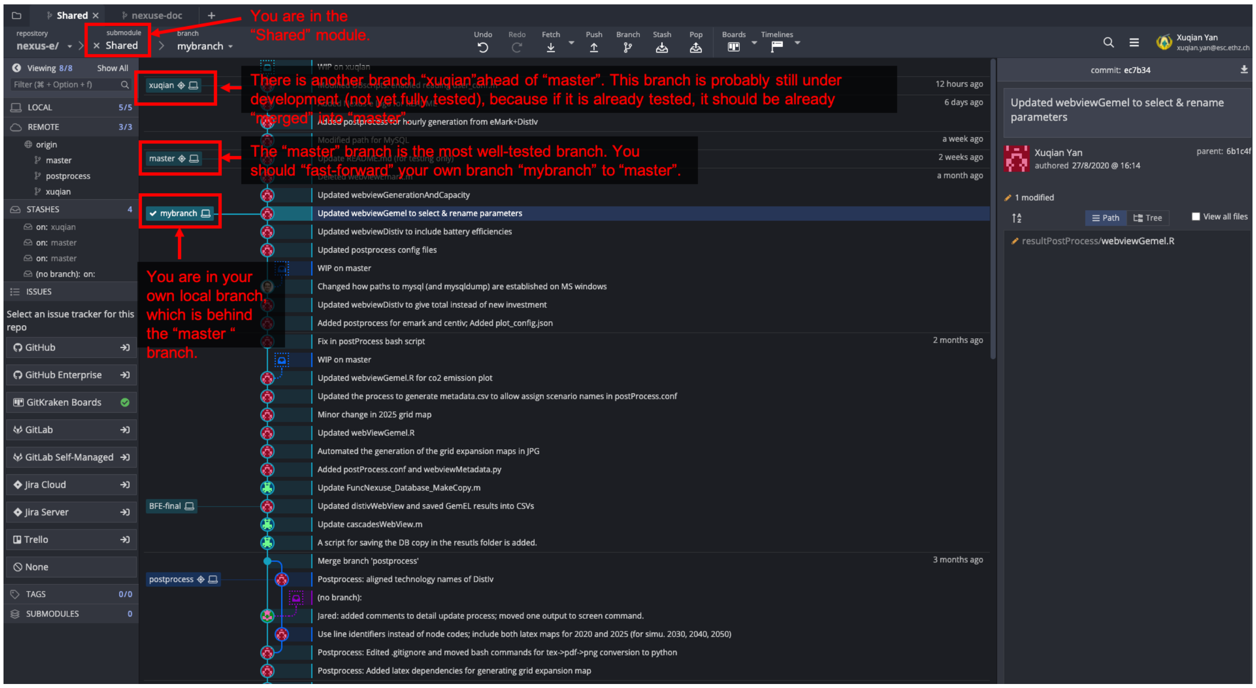


Update Cascades — 2 weeks ago
Update Cascades
Fixed the naming of Centlv
Update Gemel
Test updating cascades
Test updating Shared
Test updating Cascades
Updated submodules — a month ago
Auto stash before merge of "master" and "BFE-final"
Version 0.0.1: final simulation for BFE
Updated eMark
Update Shared
Updated submodules
Added GemEL into the repo — 2 months ago
Updated submodules — 3 months ago
Updated submodules — 4 months ago
Added Shared/resultPosrProcess
Added all submodules — 5 months ago
Initial commit

Discard Submodule Changes
Commit Changes
Update Shared
Edit Shared
Open this submodule
Delete this submodule

Path: /Shared
Remote URL: https://gitlab.ethz.ch/jaredg/nexus_shared.git

(2) After entering the submodule "Shared", you might see such a structure.



You are in the "Shared" module.

There is another branch "xuqian"ahead of "master". This branch is probably still under development (not yet fully tested), because if it is already tested, it should be already "merged" into "master".

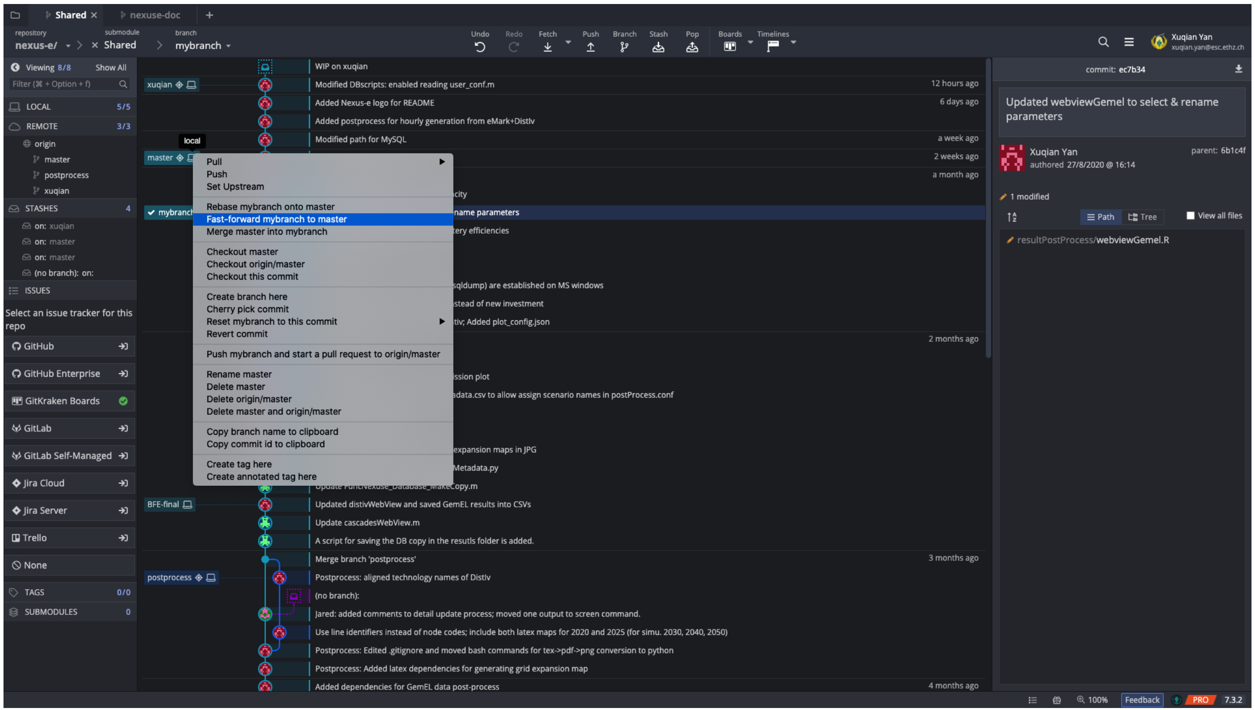
The "master" branch is the most well-tested branch. You should "fast-forward" your own branch "mybranch" to "master".

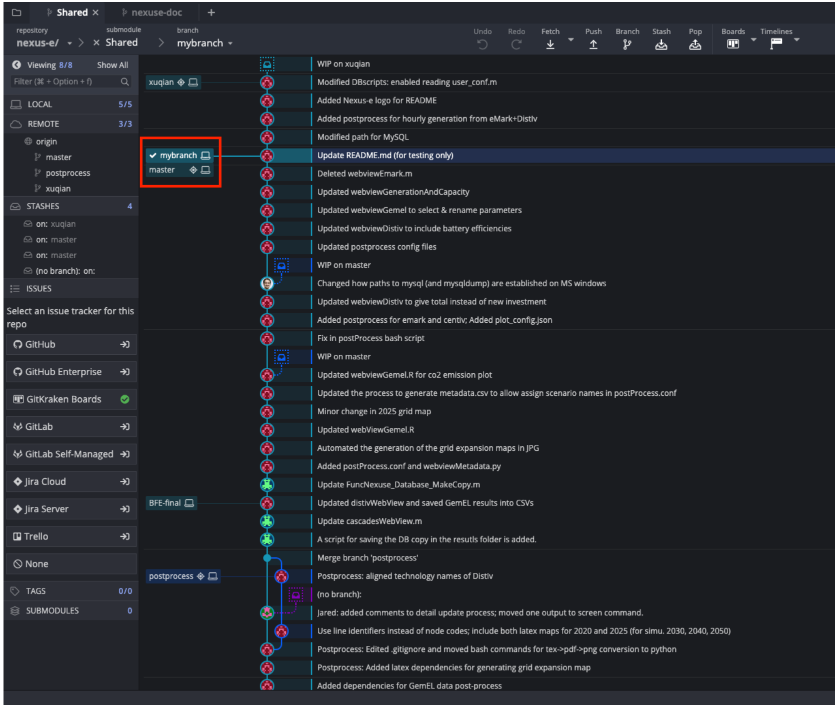
You are in your own local branch, which is behind the "master " branch.

WIP on xuqian
Added postprocess for hourly generation from eMark+Distlv
Updated webviewGenerationAndCapacity
Updated webviewGemel to select & rename parameters
Updated webviewDistlv to include battery efficiencies
Updated postprocess config files
WIP on master
Changed how paths to mysql (and mysqldump) are established on MS windows
Updated webviewDistlv to give total instead of new investment
Added postprocess for emark and centiv; Added plot_config.json
Fix in postProcess bash script — 2 months ago
WIP on master
Updated webviewGemel.R for co2 emission plot
Updated the process to generate metadata.csv to allow assign scenario names in postProcess.conf
Minor change in 2025 grid map
Updated webViewGemel.R
Automated the generation of the grid expansion maps in JPG
Added postProcess.conf and webviewMetadata.py
Update FuncNexuse_Database_MakeCopy.m
Updated distivWebView and saved GemEL results into CSVs
Update cascadesWebView.m
A script for saving the DB copy in the resutls folder is added.
Merge branch 'postprocess' — 3 months ago
Postprocess: aligned technology names of Distlv
(no branch):
Jared: added comments to detail update process; moved one output to screen command.
Use line identifiers instead of node codes; include both latex maps for 2020 and 2025 (for simu. 2030, 2040, 2050)
Postprocess: Edited .gitignore and moved bash commands for tex->pdf->png conversion to python
Postprocess: Added latex dependencies for generating grid expansion map

commit: ec7b34
Updated webviewGemel to select & rename parameters
Xuqian Yan
authored 27/8/2020 @ 16:14
parent: 6b1c4f
1 modified
resultPostProcess/webviewGemel.R

(3) Fast-forward your branch to "master" to make sure your local codes are up-to-date.





```
WIP on xuqian
Modified DBscripts: enabled reading user_conf.m
Added Nexus-e logo for README
Added postprocess for hourly generation from eMark+Distlv
Modified path for MySQL
Update README.md (for testing only)
Deleted webviewEmark.m
Updated webviewGenerationAndCapacity
Updated webviewGemel to select & rename parameters
Updated webviewDistiv to include battery efficiencies
Updated postprocess config files
WIP on master
Changed how paths to mysql (and mysqldump) are established on MS windows
Updated webviewDistlv to give total instead of new investment
Added postprocess for emark and centiv; Added plot_config.json
Fix in postProcess bash script
WIP on master
Updated webviewGemel.R for co2 emission plot
Updated the process to generate metadata.csv to allow assign scenario names in postProcess.conf
Minor change in 2025 grid map
Updated webViewGemel.R
Automated the generation of the grid expansion maps in JPG
Added postProcess.conf and webviewMetadata.py
Update FuncNexuse_Database_MakeCopy.m
Updated distivWebView and saved GemEL results into CSVs
Update cascadesWebView.m
A script for saving the DB copy in the resutls folder is added.
Merge branch 'postprocess'
Postprocess: aligned technology names of Distlv
(no branch):
Jared: added comments to detail update process; moved one output to screen command.
Use line identifiers instead of node codes; include both latex maps for 2020 and 2025 (for simu. 2030, 2040, 2050)
Postprocess: Edited .gitignore and moved bash commands for tex->pdf->png conversion to python
Postprocess: Added latex dependencies for generating grid expansion map
Added dependencies for GemEL data post-process
```

(4) Now your own branch is at the same level as "master".

(5) Sometimes you might need to update other modules as well.E.g., if your change in the "Shared" module depends on the newest changes in the "CentIv" module which is in the "target_branch" branch (doesn't have to be the "master" branch), you will need to go through a similar procedure:

- Enter "CentIv" submodule.
- Fast-forward your own branch to the "target_branch".

(6) In the end, you could enter all submodules one by one to make sure that you are at the right place in all of them.

3.4 I have made some changes in my own module and tested them in a small scale. How do I test them with the full Nexus-e platform?

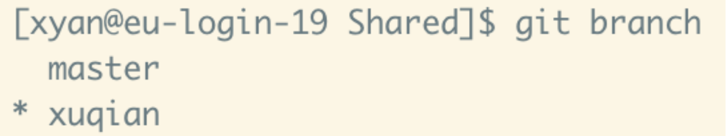
To test the full platform, you will need to move your codes with your changes to Euler and run Nexus-e on Euler.

There are two ways to get codes to Euler: (1) copy from your local and (2) use git. Here we list the pros, cons, and use cases of both ways. You could decide which way to use.

- Comparison
  - Copy from local
    - Pro: (1) This can be done with a GUI tool. (2) You don't need to push to the repository first.
    - Con: (1) Copying the whole Nexus-e folder takes time (a few minutes). (2) If you only copy selected files where you have made changes, you might by accident forget some.
    - When to use: (1) when you need to test small changes that you don't want to push to the repository yet, and (2) when you can't push to git, e.g., when you made changes in a repo where you don't have permission to push.
  - Git
    - Pro: (1) Using git gaurantees that what you run on Euler is exactly the same as in the repository. (2) It is fast because only updated files will be reloaded by git and some files (e.g. results) are ignored by git.
    - Con: (1) You need to run git commands in a terminal (no GUI). (2) You need to push your codes to the repository first and then pull it to Euler
    - When to use: (1) When you want to test what is already in the repository, and (2) when you have made lots of changes to the codes and you might forget some files when you copy them to Euler manually.
- Copy from local
  - Use a graphical tool to transfer files between Euler and your local computer.
  - Copy your whole Nexus-e folder (you don't have to copy the "Results" folder) to Euler.
  - Alternatively, you could also only copy individual files where you made changes, but you risk forgetting some files or forgetting to update some modules on Euler.
- Use gitIn this section, we assume that you already have Nexus-e codes on Euler, so you only need to update the codes where you have made changes. If this is not the case, follow the section Euler setup.
  - First, make sure to have a clean start - be aligned with the most updated commit in the overarching repository.

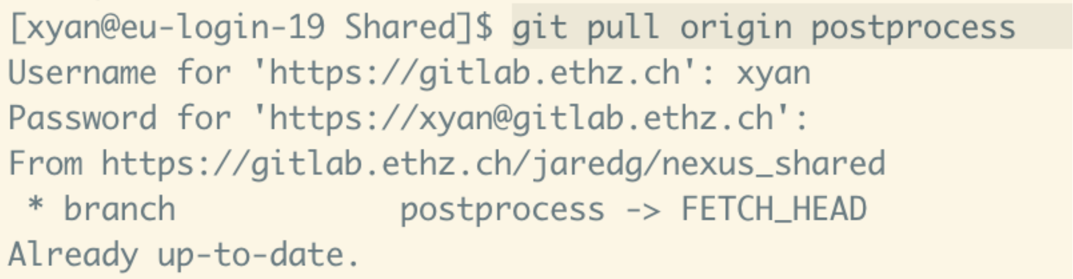
    (1) On Euler, go to the folder where you put the Nexus-e codes. e.g.,`cd nexus-e`(2) Update the codes
    - Delete your local changes (e.g., your edits in "run_Nexuse.m")`git reset --hard`
    - Pull changes from the overarching repository`git pull`
    - Update the submodules`git submodule update`
    - (Optional) Check the commit numbers of each submodule`git submodule foreach "git rev-parse HEAD"`
  - Then, go to the individual submodules where you want to pull changes that are not in the overarching repository. For example, you want to test your changes in the "Shared" submodule which is pushed to a branch named as "postprocess".

(1) Go to the folder where you want to pull the changes, e.g.,`cd Shared`(2) Check which branch you are at right now with`git branch`E.g., in the screenshot below, it is at the "xuqian" branch (seen from the "*" sign) and there is **no "postprocess" branch yet**.(Note: This is the most complicated case. For example, if you already have the "postprocess" branch and you are already at the "postprocess" branch, you could skip the following steps and only run `git pull`. See more information on switching branch with git checkout and updating the current branch with git pull.)



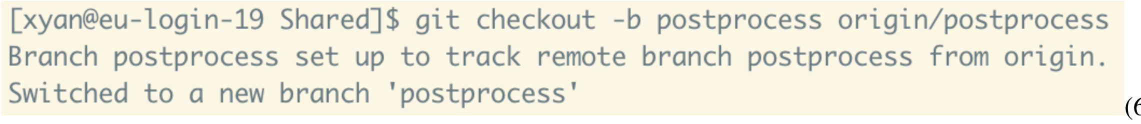
```
[xyan@eu-login-19 Shared]$ git branch
  master
* xuqian
```

(3) Fetch the origin`git fetch origin`(4) Pull changes in the "postprocess" remote branch with`git pull origin postprocess`You will see the following message asking for your credentials for the "Shared" repo.



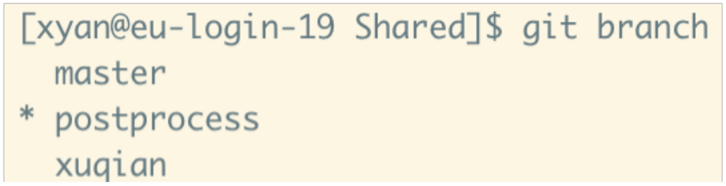
```
[xyan@eu-login-19 Shared]$ git pull origin postprocess
Username for 'https://gitlab.ethz.ch': xyan
Password for 'https://xyan@gitlab.ethz.ch':
From https://gitlab.ethz.ch/jaredg/nexus_shared
 * branch            postprocess -> FETCH_HEAD
Already up-to-date.
```

(5) Create and check out local branch "postprocess" with`git checkout -b postprocess origin/postprocess`



```
[xyan@eu-login-19 Shared]$ git checkout -b postprocess origin/postprocess
Branch postprocess set up to track remote branch postprocess from origin.
Switched to a new branch 'postprocess'
```

(6 Check your branch again with `git branch` to make sure that you are at the "postprocess" branch.

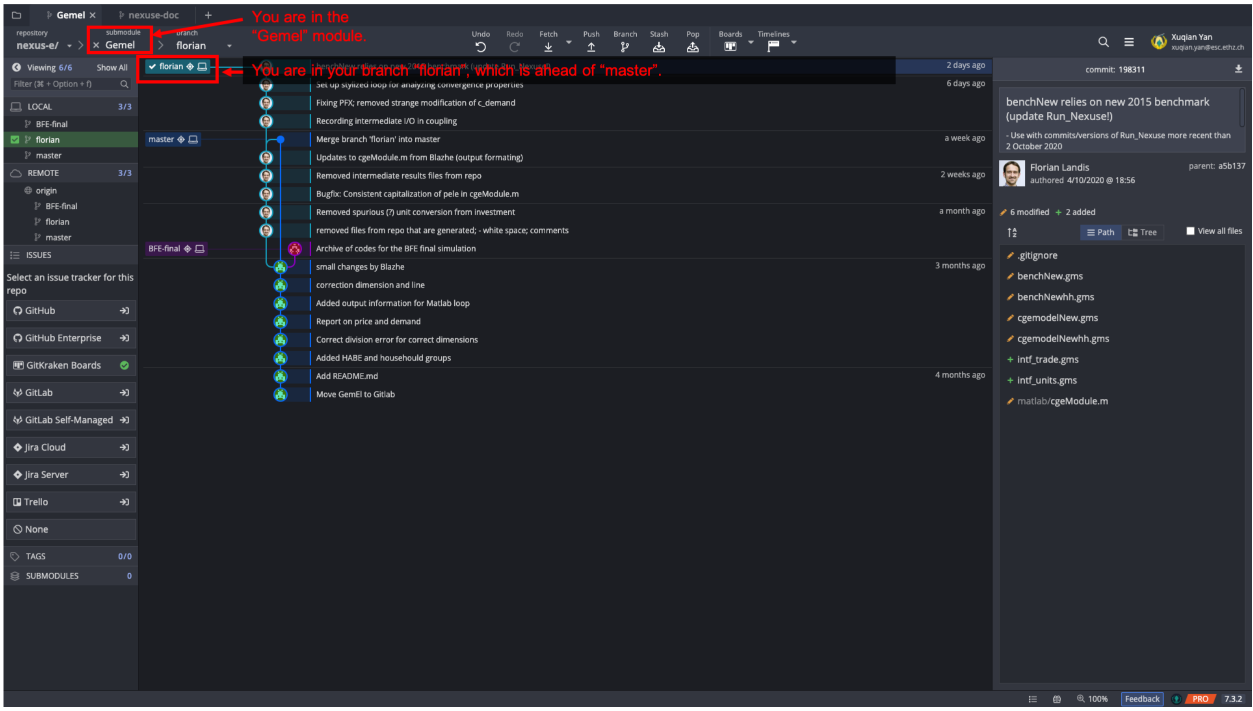


```
[xyan@eu-login-19 Shared]$ git branch
  master
* postprocess
  xuqian
```

- Sometimes you need to do the branch switching (as instructed in the step above) for other submodules as well. From *here*, you should already know which branch you are at in each submodule.

3.5 I have fully tested my changes in my own module. What should I do now?

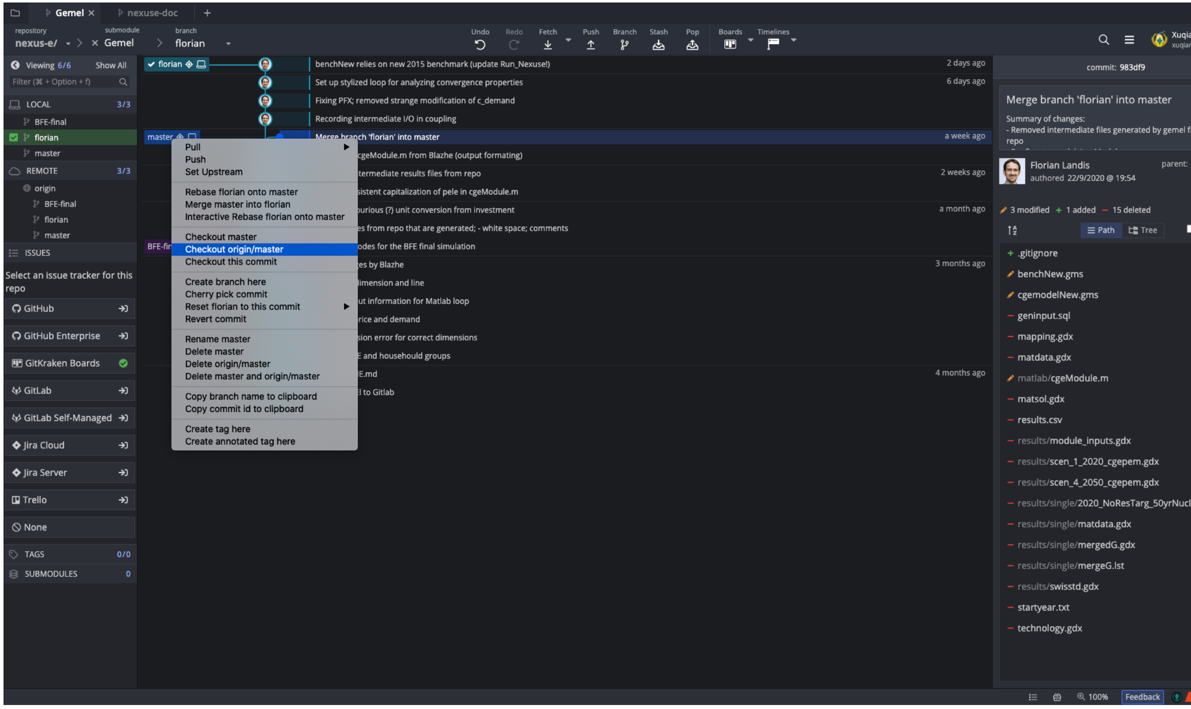
For example, you have made changes on the branch "florian" in the submodule "Gemel". The changes have been tested together with the updated module "Run_Nexuse". Now you are confident about the codes and you want other researchers also use your updates.Note: Since January 2021, "Run_Nexuse" is not a submodule anymore but a folder directly under the Nexus-e repository, but the procedures demonstrated here can be applied to any submodules.(1) Enter the "Gemel" submodule.


Gemel
nexuse-doc
repository
nexus-e/
submodule
Gemel
branch
florian
Undo
Redo
Fetch
Push
Branch
Stash
Pop
Boards
Timelines
Xuqian Yan
Viewing 6/6
Show All
LOCAL 3/3
BFE-final
florian
master
REMOTE 3/3
origin
BFE-final
florian
master
ISSUES
Select an issue tracker for this repo
GitHub
GitHub Enterprise
GitKraken Boards
GitLab
GitLab Self-Managed
Jira Cloud
Jira Server
Trello
None
TAGS 0/0
SUBMODULES 0
florian
master
BFE-final
benchNew relies on new 2015 benchmark (update Run_Nexuse!)
Set up stylized loop for analyzing convergence properties
Fixing PFX; removed strange modification of c_demand
Recording intermediate I/O in coupling
Merge branch 'florian' into master
Updates to cgeModule.m from Blazhe (output formating)
Removed intermediate results files from repo
Bugfix: Consistent capitalization of pele in cgeModule.m
Removed spurious (?) unit conversion from investment
removed files from repo that are generated; - white space; comments
Archive of codes for the BFE final simulation
small changes by Blazhe
correction dimension and line
Added output information for Matlab loop
Report on price and demand
Correct division error for correct dimensions
Added HABE and household groups
Add README.md
Move GemEl to Gitlab
2 days ago
6 days ago
a week ago
2 weeks ago
a month ago
3 months ago
4 months ago
commit: 198311
benchNew relies on new 2015 benchmark (update Run_Nexuse!)
- Use with commits/versions of Run_Nexuse more recent than 2 October 2020
Florian Landis
authored 4/10/2020 @ 18:56
parent: a5b137
6 modified
2 added
Path
Tree
View all files
.gitignore
benchNew.gms
benchNewhh.gms
cgemodelNew.gms
cgemodelNewhh.gms
intf_trade.gms
intf_units.gms
matlab/cgeModule.m
100%
Feedback
PRO
7.3.2

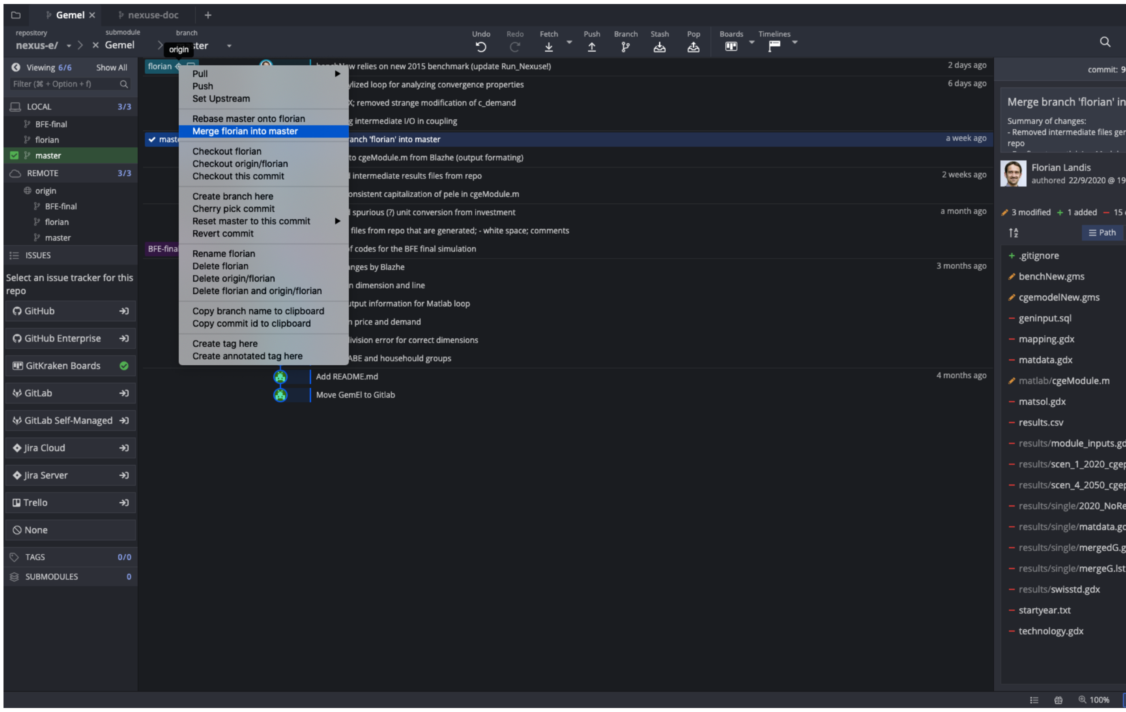
You are in the "Gemel" module.

You are in your branch "florian", which is ahead of "master".

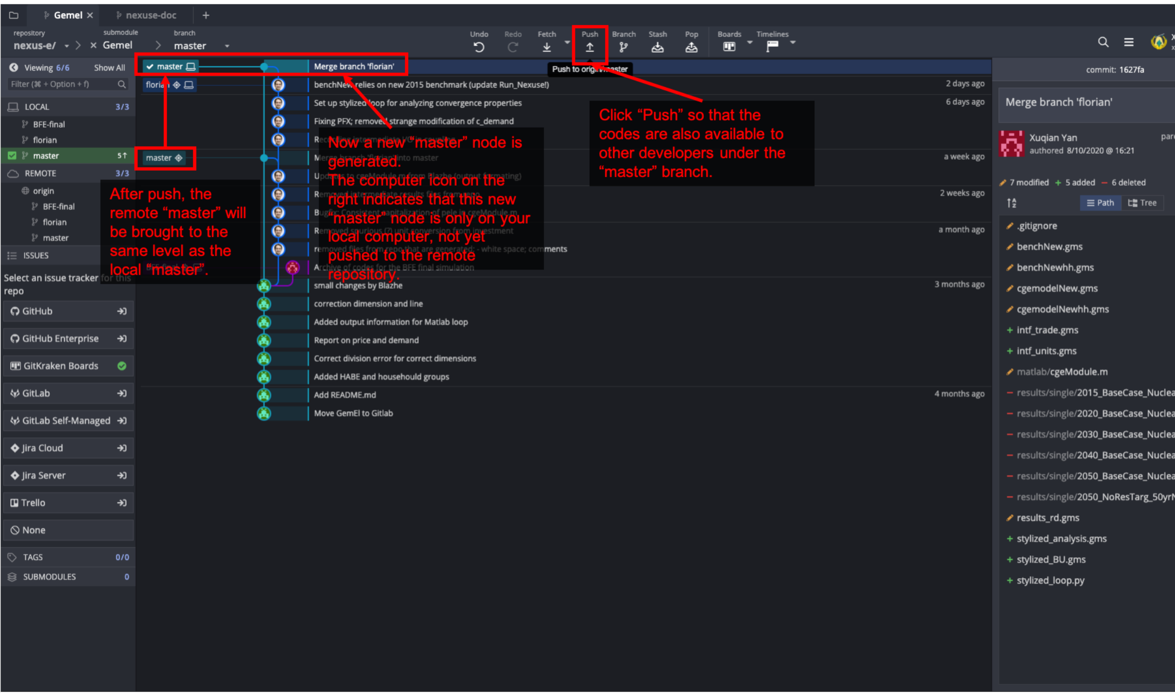
(2)



Checkout origin/master



Merge “florian” into “master”



Now a new “master” node is generated.

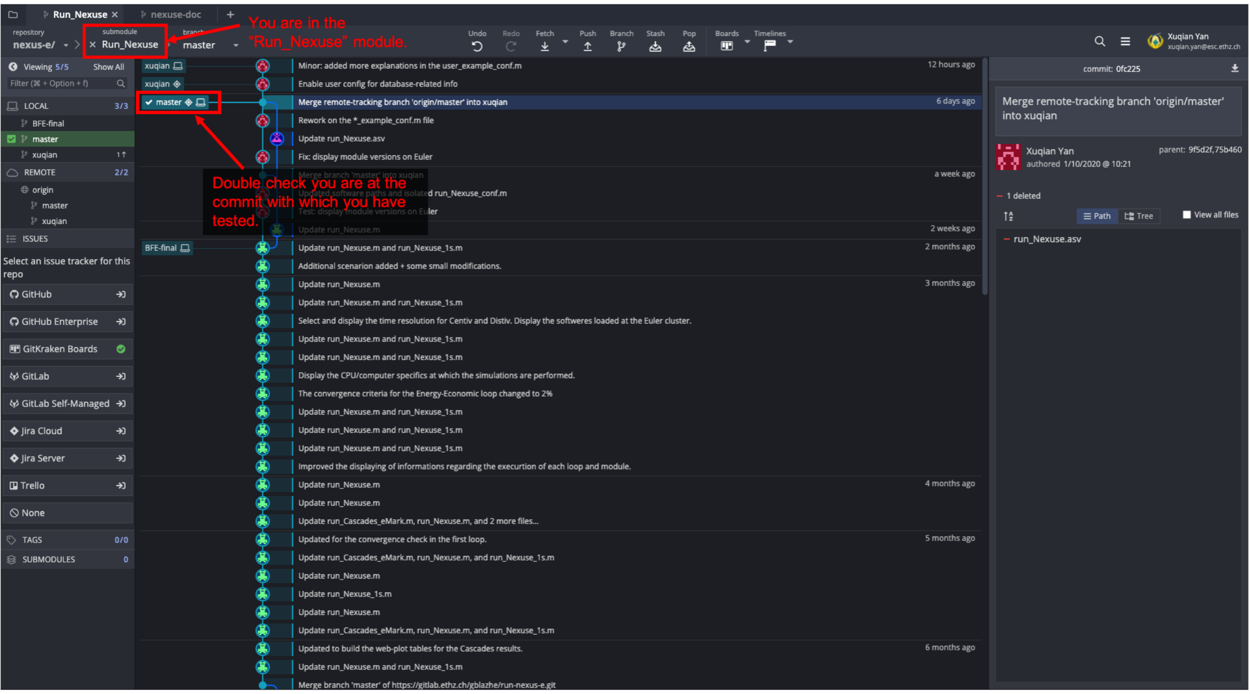
The computer icon on the right indicates that this new “master” node is only on your local computer, not yet pushed to the remote repository.

Click “Push” so that the codes are also available to other developers under the “master” branch.

After push, the remote “master” will be brought to the same level as the local “master”.

Push the “master” branch

Enter the “Run_Nexuse” submodule and make sure you are at the right



You are in the "Run_Nexuse" module.

Double check you are at the commit with which you have tested.

```
Minor: added more explanations in the user_example_conf.m                                     12 hours ago
Enable user config for database-related info
Merge remote-tracking branch 'origin/master' into xuqian                                       6 days ago
Rework on the *_example_conf.m file
Update run_Nexuse.asv
Fix: display module versions on Euler
Merge branch 'master' into xuqian                                                              a week ago
Updated software paths and isolated run_Nexuse_conf.m
Test: display module versions on Euler
Update run_Nexuse.m                                                                            2 weeks ago
Update run_Nexuse.m and run_Nexuse_1s.m                                                        2 months ago
Additional scenarion added + some small modifications.
Update run_Nexuse.m                                                                            3 months ago
Update run_Nexuse.m and run_Nexuse_1s.m
Select and display the time resolution for Centiv and Distiv. Display the softweres loaded at the Euler cluster.
Update run_Nexuse.m and run_Nexuse_1s.m
Update run_Nexuse.m and run_Nexuse_1s.m
Display the CPU/computer specifics at which the simulations are performed.
The convergence criteria for the Energy-Economic loop changed to 2%
Update run_Nexuse.m and run_Nexuse_1s.m
Update run_Nexuse.m and run_Nexuse_1s.m
Update run_Nexuse.m and run_Nexuse_1s.m
Improved the displaying of informations regarding the execurtion of each loop and module.
Update run_Nexuse.m                                                                            4 months ago
Update run_Nexuse.m
Update run_Cascades_eMark.m, run_Nexuse.m, and 2 more files...
Updated for the convergence check in the first loop.                                           5 months ago
Update run_Cascades_eMark.m, run_Nexuse.m, and run_Nexuse_1s.m
Update run_Nexuse.m
Update run_Nexuse_1s.m
Update run_Nexuse.m
Update run_Cascades_eMark.m, run_Nexuse.m, and run_Nexuse_1s.m
Updated to build the web-plot tables for the Cascades results.                                 6 months ago
Update run_Nexuse.m and run_Nexuse_1s.m
Merge branch 'master' of https://gitlab.ethz.ch/gblazhe/run-nexus-e.git
```

Merge remote-tracking branch 'origin/master' into xuqian

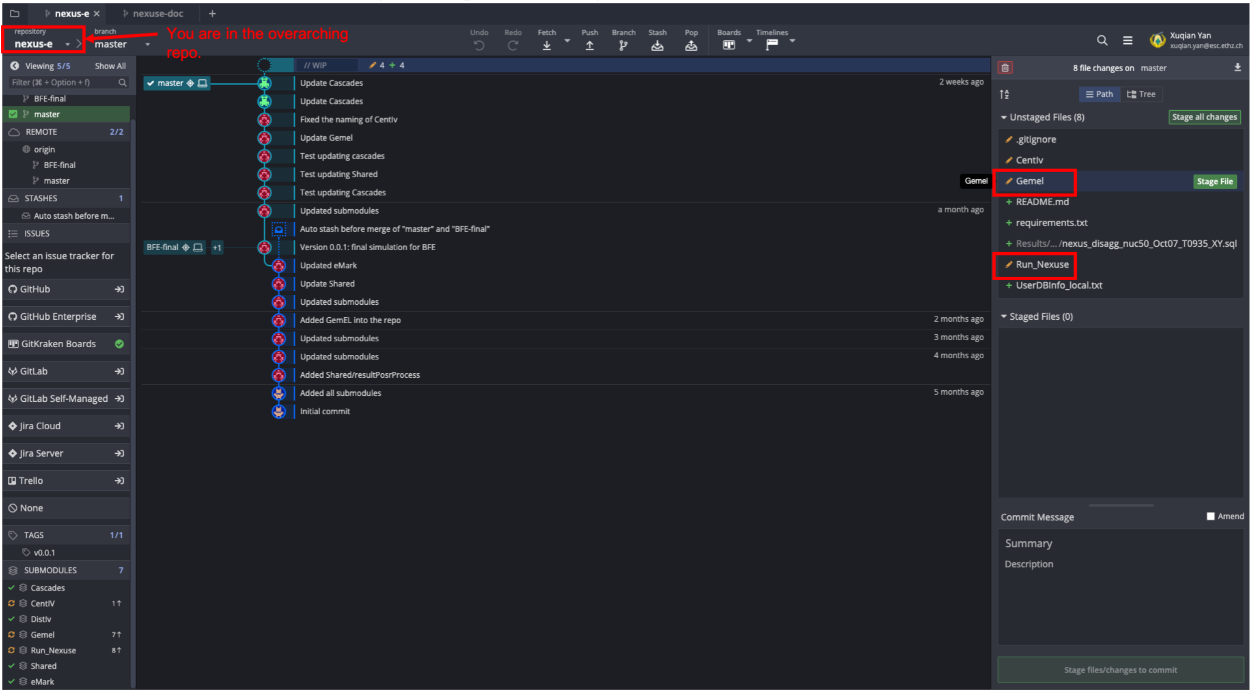
Xuqian Yan authored 1/10/2020 @ 10:21 parent: 9f5d2f,75b460

1 deleted

run_Nexuse.asv

commit (6)

Go back to the overarching repo, stage the "Gemel" and "Run_Nexuse" modules (7)



You are in the overarching repo.

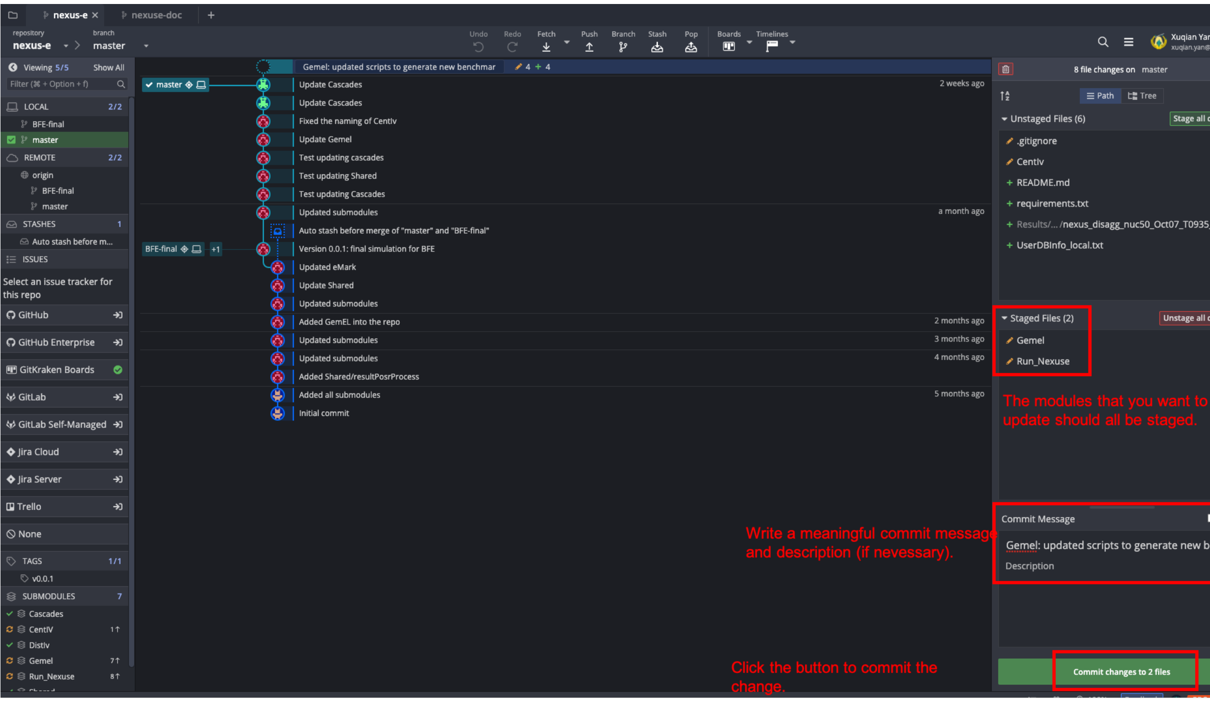
```
Update Cascades                                                2 weeks ago
Update Cascades
Fixed the naming of CentIv
Update Gemel
Test updating cascades
Test updating Shared
Test updating Cascades
Updated submodules                                             a month ago
Auto stash before merge of "master" and "BFE-final"
Version 0.0.1: final simulation for BFE
Updated eMark
Update Shared
Updated submodules
Added GemEL into the repo                                      2 months ago
Updated submodules                                             3 months ago
Updated submodules                                             4 months ago
Added Shared/resultPosrProcess
Added all submodules                                           5 months ago
Initial commit
```

Unstaged Files (8)

- .gitignore
- CentIv
- Gemel
- README.md
- requirements.txt
- Results/... /nexus_disagg_nuc50_Oct07_T0935_XY.sql
- Run_Nexuse
- UserDBInfo_local.txt

Staged Files (0)

Commit Message

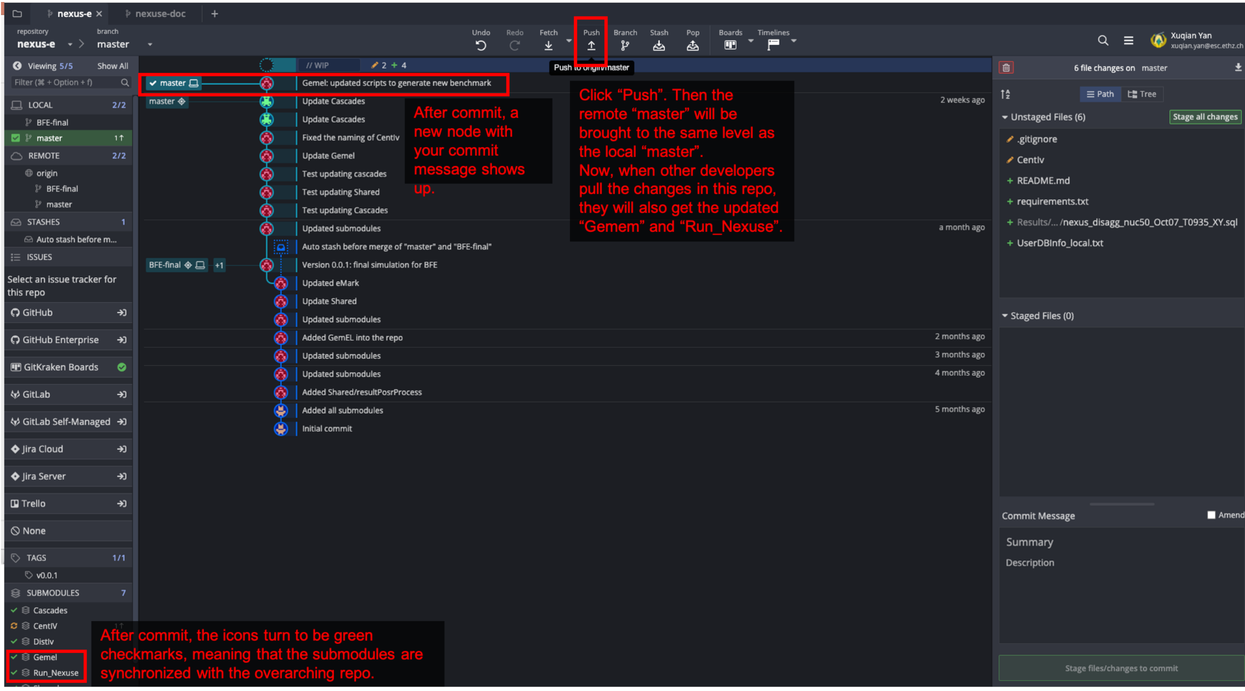


The modules that you want to update should all be staged.

Write a meaningful commit message and description (if nevessary).

Click the button to commit the change.

Commit your changes



After commit, a new node with your commit message shows up.

Click "Push". Then the remote "master" will be brought to the same level as the local "master". Now, when other developers pull the changes in this repo, they will also get the updated "Gemem" and "Run_Nexuse".

After commit, the icons turn to be green checkmarks, meaning that the submodules are synchronized with the overarching repo.

Push your changes

CHAPTER

FOUR

MODIFY INPUT DATA

Input data for Nexus-e are organized in two ways: *(1) Collected in one common MySQL database* and *(2) Stored separately in each module*. When modifying input data, you first need to figure out where the data is stored, and then take the corresponding approach to modify.

4.1 Where the input data are stored

Input data for Nexus-e are organized in the following two ways.

4.1.1 (1) Stored in one common MySQL database

By gathering input data in one common database, (1) it alignes the data used by different modules, (2) it easies the procedure to tune the parameters, and (3) the data become more transparent and have a clearer structure. Data that are collected in the database include - data that are used by multiple modules (e.g., grid parameters) - data that vary across scenarios (e.g., macroeconomic assumptions)

The structure of our current database can be found here.

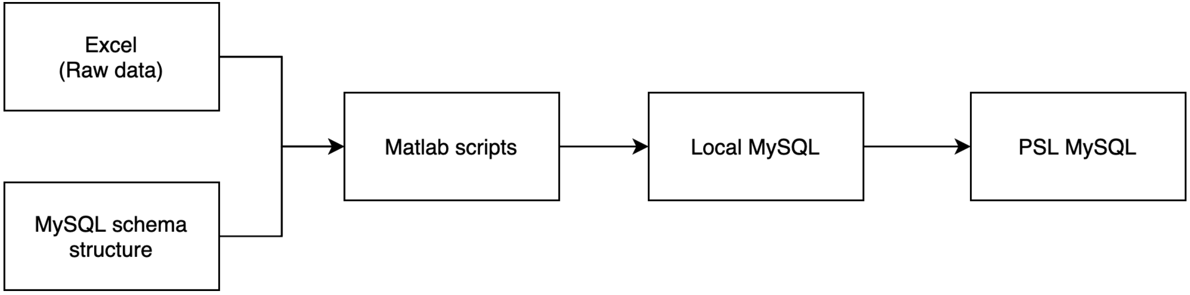
4.1.2 (2) Stored separately in each module

Storing data separately saves the effort to transfer the data into/out of the database. This is especially beneficial for complex data formats, because MySQL is only suitable for table-like data formats and other small & simple data formats that can be transformed into a BLOB. Data that are stored separately include - data that are only used for one module - data that do not need to be changed frequently - data with a complex format

4.2 How to modify

- If the data is stored within a certain module, contact the module owner (i.e., the administrator of the module's repository).
- If the data is stored in the input database, tutorial videos on how the database is created and how one can modify the input data can be found in Polybox `00_Administration/05_Internal`

`Meetings/2021/2021.05.03_MySQL-Tutorial`. Here we briefly describe the workflow:



- The ending point: The input database of Nexus-e is served by a MySQL server from the Power Systems Laboratory (PSL) at ETH Zurich. This is the **PSL MySQL** block shown on the right side of the diagram above, i.e., the ending point of the workflow.
- The starting point: The left side of the diagram shows the starting point of the workflow. It contains (1) one **Excel** file of the raw data and (2) a **MySQL schema structure** (i.e., an empty schema without data but with the strucutre). the Excel is shared through Git and is available in `\scenario_data\source_excel`, while the schema structure can be found in the folder `Shared/DBcreation/schema_structure`.
  - What to do: (1) get the Excel file (either find it on Git (as instructed above) or ask the Nexus-e team (nexus-e@ethz.ch) for it) and (2) upload the schema structure to your local MySQL server (instructions on how to work with the local MySQL).
- The intermediate steps: The raw data in the Excel is organized and populated into the MySQL schema structure in your local MySQL. This is conducted with the **Matlab script** `Shared/DBcreation/addNexusExcel2dataBase...`. This script calls other Matlab functions. Basically (but not always) each function takes care of one sheet in the Excel or one table in the schema. After running the script and making sure that the schema in your **local MySQL** is correctly populated, you can move the populated database to the **PSL MySQL** server.
  - What to do: (1) At the beginning of the `addNexusExcel2dataBase...` script (look for the sign `###>---<###`) edit the inputs as instructed in the script, (2) in `Run_Nexuse/conf_local` provide the credentials to your local MySQL in `UserDBInfo_local.txt`, (3) edit the `DB_INFO_FILE` and `PATH_MYSQL` variables in `conf_local.m` (instructions), (4) run the script `addNexusExcel2dataBase...`, (5) when ready, export your local database and import it to the PSL MySQL (instructions on how to import/export a database).
- In summary, if you want to modify the data stored in the database:
  - To **change** a data that is already in the input database, you only need to change the data in the Excel file and run the Matlab script with the existing schema structure. (You could theoretically also directly change the MySQL database, but to keep the workflow consistent, we recommend to always start from the Excel file.)
  - To **add new** data to the input database, you will also need to adjust the schema structure and the Matlab script.

CHAPTER FIVE

CONNECT NEW MODULE

- Before connecting a new module to the Nexus-e platform, the answer of following questions is needed:
  1. Is it going to be a substitute/replacement for an existing module?
  2. Is it an additional module that will be part of the existing or new convergence loops?
- In case question one is true, the existing interfaces can be used, including the convergence criteria that already exists. Furthermore, these interfaces may be extended if new points of connections are identified. This process requires consultation with the involved modelers.
- In case question two is true, clear interfaces has to be defined, i.e., the data exchange between the new module and the existing modules. Also, if the new module is part of any of the existing loops, the convergence criterion may need to be revised. Otherwise, new loop and convergence criteria need to be defined. These processes require consultation with the involved modelers.
- Each model within Nexus-e is connected to the platform using a wrapper, i.e., Matlab class. As an example use the existing wrappers, i.e., emModule.m (\eMark\electricity_markets\marketsModule) or CascadesModule.m (\Cascades\03_Run_models&methods). The class contains of properties and methods.
- In properties, assign your property attributes, e.g., wkspace (workspace information).
- In methods, we suggest that you set three types of functions:
  - Object constructor, which creates the module object. There you can have, e.g., additional paths to your subfolders, reed your input parameters, assign data structures (e.g., input parameters) to the module object, etc.
  - Run method, which executes the module. This function, besides the module object, as an inputs can use data from other modules with which interfaces are established.
  - Interface functions, which gather the data which will be stored and later send to other modules. You can have multiple interface functions. The number of these functions depends on the number of modules and type of information your module is sending data to. It is not necessary that the data exchange between two modules is going in two directions.
- Note that the aforementioned instructions are general directions, and therefore, the class is subject to customization based on the needs and preferences of the modelers.

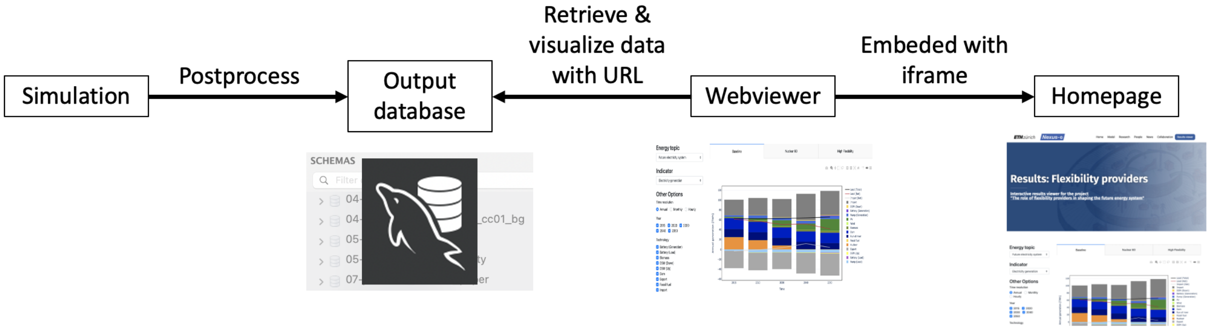
RESULTS VISUALIZATION

The simulation results from Nexus-e can be interactively viewed through a visualization tool (also called the “**webviewer**”), developed with Python Dash.

- Repository for the tool: https://gitlab.ethz.ch/nexus-e/nexus-homepage-dash.git
- A frontend example: https://nexus-e.org/results-flexibility-providers/

This documentation only introduces the general workflow of developing the visualization tool. Detailed instructions can be found in the `README.md` files in the associated repositories.

The visualization of the simulation results is achieved in the workflow shown in the figure. The key processes will be explained in the sub-sections below.



6.1 Postprocess

After a simulation finishes, selected results are **postprocessed** and uploaded to the output database. This postprocess procedure is carried out in the following two ways. The two ways are designed for different purposes and both are needed.

1. **Embedded** in the `run_Nexuse.m` script
   - **Purpose**: Automatically process the results when finishing a simulation, so that the reseacher could promtly see the results of this simulation from the webviewer.
   - **How it work**: After the simulation finishes, `run_Nexuse.m` calls the MATLAB script `Shared/resultPostProcess/postProcess.m`. This scripts calls other MATLAB/Python scripts to post-process the results.
2. **Stand-alone**
   - **Purpose**: Batch-process multiple simulations, fine-tune/modify/test/debug the post-process scripts.

- **How it works**: Run the MATLAB script `postProcess_batch.m` locally. This script loops through multiple simulations and runs only the required post-process scripts (e.g., we don't have to run the complete post-process for quick debugging). Which simulations to loop through and which post-processes to run can be configured at the "To be edited" section in the script. Details can be found in the comments in the script.

Note: To properly run the `postProcess_batch.m` script, you need to set up the paths for the required software (e.g., Python, GAMS). To do this, follow the instructions in `Run_Nexuse/user_example_conf.m` and the online documentation.

6.2 Output database

In the output database (host: itet-psl-s02, port: 3307), each simulation's results is stored in one **schema** named with the simulation **submission date** and the **scenario name** (e.g., the `scenShortName` variable in `run_Nexuse.m`), which looks like "01-dec-2020_baseline".

The output database can be viewed with MySQL Workbench in the same way as introduced here.

6.3 Retrieve & visualize data with a URL

After the postprocessed data has been uploaded to the output database, the data can be retrieved and visualized by **provideing a URL through the webviewer**. Three types of URL are accepted.

1. https://nexus-e.org/results/**JOB-SUBMISSION-DATE_SCENARIO_NAME**, e.g., https://nexus-e.org/results/01-dec-2020_baseline. This will retrieve and visualize only the data from the **exact** MySQL schema named as "JOB-SUBMISSION-DATE_SCENARIO_NAME".
2. https://nexus-e.org/results/**JOB-SUBMISSION-DATE**, e.g. https://nexus-e.org/results/01-dec-2020. This will retrieve and visualize the data from **all** the MySQL schemas whose name start with "JOB-SUBMISSION-DATE". In other words, results from all simulations submitted on the same day will be shown on the website.
3. https://nexus-e.org/results/**MANUALLY_DEFINED_URL**, e.g. https://nexus-e.org/results/project. This will retrieve and visualize the results from **all** simulations associated with the MANUALLY_DEFINED_URL. This is particularly useful if we run multiple scenarios for one project. The mapping of URL and simulations is specified in the `constants.py` in the repository for the webviewer.

Note: Synchronizing data between the web server and the database

Not all changes in the output database will be automatically sychronized to the webviewer. The reason lies in how the data is downloaded from the database to the web server: based on the given URL, the webviewer checks the output database and decides which simulation(s) will be shown, i.e., which schema(s) should be downloaded to the web server; the webviewer then checks (by comparing the existing folder names and the schema name) whether the needed schema has already been downloaded; if the viewer finds a folder with the same name as the needed schema, it won't re-download the schema.

Therefore, to show a changed data in the output database (e.g., after changing the postprocess script, added new data), you need to **first delete the corresponding data folder on the web server** so that the webviewer will re-download the data from the database.

6.4 Embed the webviewer into the homepage

The webviewer itself is plainly a Python Dash application without much Nexus-e branding. To change this, it is embedded into the homepage with iframe - a html element to display a nested webpage within another webpage. By doing so, we can easily have the same header, footer and menubar for the visualization tool as in other Nexus-e webpages.

The embedding can only be done by the administrators of the homepage. If you are an administrator, you could login to the homepage and check the backend of this page to learn how it is done.

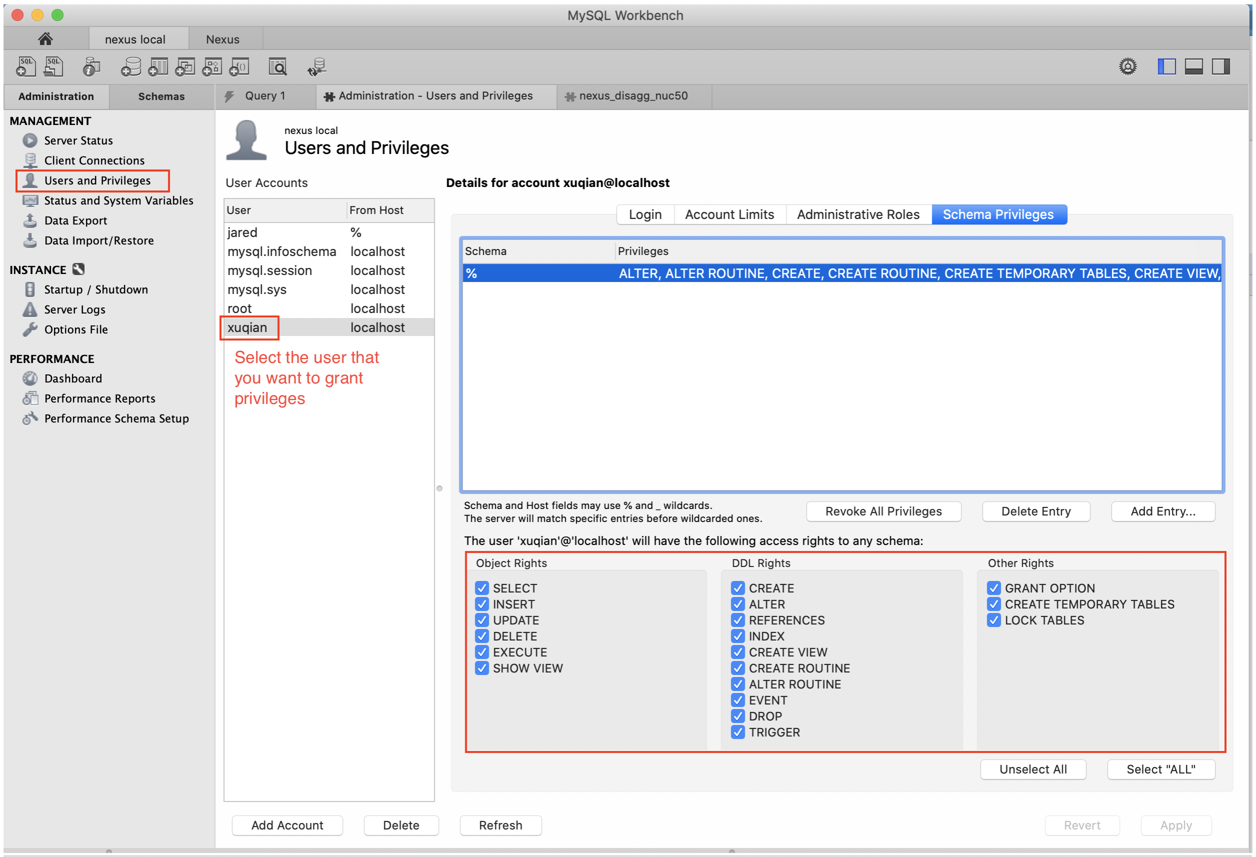
CHAPTER

SEVEN

COMMON ISSUES

7.1 Database

7.1.1 Grant full privileges to a user in MySQL Workbench



User Accounts

User	From Host
jared	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost
xuqian	localhost

Select the user that you want to grant privileges

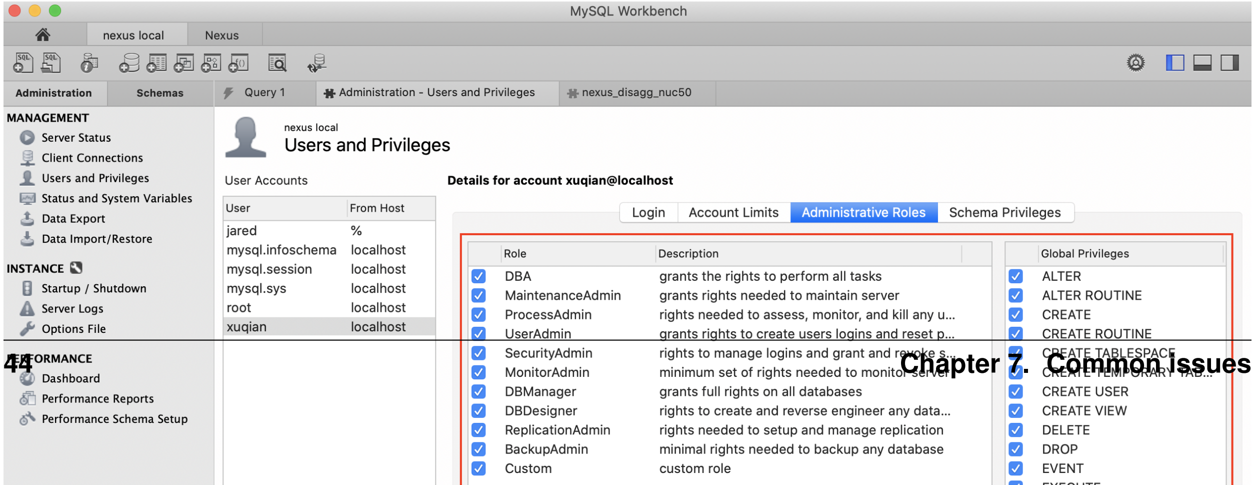
Details for account xuqian@localhost

Schema	Privileges
%	ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE VIEW,

Schema and Host fields may use % and _ wildcards.
The server will match specific entries before wildcarded ones.

The user 'xuqian'@'localhost' will have the following access rights to any schema:

Object Rights	DDL Rights	Other Rights
☑ SELECT	☑ CREATE	☑ GRANT OPTION
☑ INSERT	☑ ALTER	☑ CREATE TEMPORARY TABLES
☑ UPDATE	☑ REFERENCES	☑ LOCK TABLES
☑ DELETE	☑ INDEX	
☑ EXECUTE	☑ CREATE VIEW	
☑ SHOW VIEW	☑ CREATE ROUTINE	
	☑ ALTER ROUTINE	
	☑ EVENT	
	☑ DROP	
	☑ TRIGGER	



User Accounts

User	From Host
jared	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost
xuqian	localhost

Details for account xuqian@localhost

	Role	Description
☑	DBA	grants the rights to perform all tasks
☑	MaintenanceAdmin	grants rights needed to maintain server
☑	ProcessAdmin	rights needed to assess, monitor, and kill any u...
☑	UserAdmin	grants rights to create users logins and reset p...
☑	SecurityAdmin	rights to manage logins and grant and revoke s...
☑	MonitorAdmin	minimum set of rights needed to monitor server
☑	DBManager	grants full rights on all databases
☑	DBDesigner	rights to create and reverse engineer any data...
☑	ReplicationAdmin	rights needed to setup and manage replication
☑	BackupAdmin	minimal rights needed to backup any database
☑	Custom	custom role

	Global Privileges
☑	ALTER
☑	ALTER ROUTINE
☑	CREATE
☑	CREATE ROUTINE
☑	CREATE TABLESPACE
☑	CREATE TEMPORARY TAB...
☑	CREATE USER
☑	CREATE VIEW
☑	DELETE
☑	DROP
☑	EVENT

7.1.2 Access denied to access database

- In Matlab, print out the variables `connInfo.user`, `connInfo.pw`, and `connInfo.host` to make sure that they are exactly what you provided in the your `UserDBInfo` file.
- Make sure you database user account has *full permission to the database*

7.1.3 Database server timezone issue

There are two ways to solve this issue: (1) using command line, but this only takes effect for the duration of the current service uptime, i.e., when the MySQL services restarts, you need to type in these commands again; (2) editing the MySQL configuration file, but the configuration file is platform-dependent, e.g., MacOS doesn't even have a default configuration file.

Here we introduce the 1st approach using command line. For the 2nd approace of editing the MySQL configuration file, see here.

- Open command prompt and login to mysql using the command:`mysql -u root -p`
- Enter your MySQl password for the root user, this should start mysql and you should see the prompt `mysql>`
- Type the following two command to reset the MySQL time zone:`SET @@global.time_zone = 'Europe/Paris';SET @@session.time_zone = 'Europe/Paris';`
  - If you get the error "Unknown or incorrect time zone: 'XXX'", use the command (ref):`mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -u root mysql -p`Then rerun the `SET @@...` commands.
- You can check that these were properly set using the command:`SELECT @@global.time_zone, @@session.time_zone;`
- After confirming these are now correct you can exit mysql using the command:`exit`

7.1.4 Database definer missing

Add a dummy user (same name as the missing definer - you can see it from the error message) to the database and *grant all previleges* .

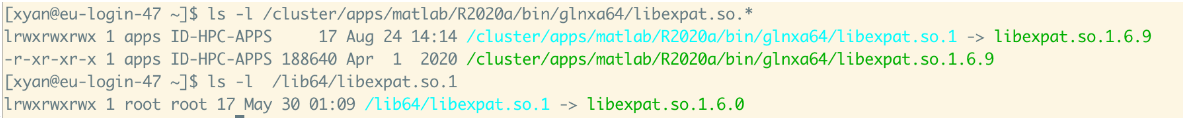
7.2 Software compatibility

7.2.1 Matlab & python

On Euler, sometimes you get a similar error `ImportError: ...pyexpat...: undefined symbol: XML_SetHashSalt`. This might be because of conflice between Matlab's and Python's `libexpat.so.1` library. If you encounter this, it is better to consult IT. Or, you could try different combinations of Matlab and python by yourself.Here is an example to compare the versions of the `libexpat.so.1` library of Python 3.7.1 and Matlab R2020a:

- Input command `ldd /cluster/apps/python/3.7.1/x86_64/lib64/python3.7/lib-dynload/pyexpat.cpython-37m-x86_64-linux-gnu.so`. It gives `libexpat.so.1 => /lib64/libexpat.so.1 (0x00002b00348bd000)`. This means the Python 3.7.1's library `pyexpat.cpython-37m-x86_64-linux-gnu.so` is linked against the operating system's `libexpat.so.1` library and not the one from Matlab (`/cluster/apps/matlab/R2020a/bin/glnxa64/libexpat.so.1`).

- The screenshot below shows that the operating system's library links to the version 1.6.0; while Matlab R2020a's library links to the version 1.6.9. Experiences show that when if Matlab has a newer version than Python (namely, newer than the operating system), it should work.



```
[xyan@eu-login-47 ~]$ ls -l /cluster/apps/matlab/R2020a/bin/glnxa64/libexpat.so.*
lrwxrwxrwx 1 apps ID-HPC-APPS     17 Aug 24 14:14 /cluster/apps/matlab/R2020a/bin/glnxa64/libexpat.so.1 -> libexpat.so.1.6.9
-r-xr-xr-x 1 apps ID-HPC-APPS 188640 Apr  1  2020 /cluster/apps/matlab/R2020a/bin/glnxa64/libexpat.so.1.6.9
[xyan@eu-login-47 ~]$ ls -l  /lib64/libexpat.so.1
lrwxrwxrwx 1 root root 17 May 30 01:09 /lib64/libexpat.so.1 -> libexpat.so.1.6.0
```