

## Installing python and obspy via conda

Last compiled: March 28, 2019

Location of this file: GEOTOOLS/info/GIseis\_conda.tex

To make a pdf, type this: `pdflatex GIseis_conda`

# 1 Installing python and obspy via conda

Most of our software is installed via package managers with the click of a button. However, many package managers do not offer a wide range of software or the latest versions of existing software. To overcome this, we use a utility called conda to install software in a user's home directory (note: *not* on a machine or in a network-wide file space).

Currently, the default python version on the linux network is installed along with Antelope so that python will interact with Antelope scripts and databases. With some effort, Dave Covey installed an obspy version (now old: 0.10.1), network-wide, that worked with our existing python. To see the defaults, try this:

```
[vipul@shavak ~]$ python --version
Python 2.7.13 :: Continuum Analytics, Inc.
```

Most users will want to use the latest version of obspy and the latest version of python. Here is how to install the latest version of obspy with python 3.6.

1. Update the software on your computer:

```
sudo yum update
```

(You may need to ask someone with root permission.)

2. Make sure you have the latest versions of all repositories. Type `git pull` from inside the repositories in `~/REPOSITORIES/`.
3. If you already have an environment installed, then skip to Step 13 to update your existing environment.
4. At the bottom of your `.bashrc` file, add these lines

```
# to over-ride the PYTHONPATH settings from antelope startup
unset PYTHONPATH
# echo $! will show wrong=0 (from the antelope startup script)
# this is a temp fix
set --
```

5. Install miniconda2:

```
bash /usr/local/src/Miniconda2-latest-Linux-x86_64.sh
source .bashrc
```

Answer the prompts. This will append your `.bashrc` file with the path to miniconda.

This will install several packages, including a version of python.

6. Install the conda client and check the conda version:

```
conda install anaconda-client
conda --version
```

This will install the directory `~/miniconda2`. This will take several minutes.

7. Make sure you're in your home dir and that you are *not* inside the python shell.

Create the `sln` environment with python:

```
conda create --name sln python=3.6
source activate sln
python --version
```

8. Install obspy and its dependencies:

```
conda config --add channels conda-forge
conda install obspy jupyter basemap
```

Answer the prompts. This will take several minutes.

9. OPTIONAL: If you want to always have python 3.6 and the latest obspy in your path, then add this line at the **end** of your `.bashrc` file.

```
# for obspy and python
source activate sln
```

10. Open a new terminal (type `source activate sln`) and check that you see the latest versions of python and obspy:

```
(sln) [vipul@shavak ~]$ python --version
Python 3.6.8 :: Anaconda, Inc.
(sln) [vipul@shavak ~]$ which python
~/miniconda2/envs/sln/bin/python

Python 3.6.8 |Anaconda, Inc.| (default, Dec 30 2018, 01:22:34)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import obspy; print(obspy.__version__)
1.1.1
>>> quit()
```

11. Test out the default example in pysep (`cd $PYSEP`):

```
python run_getwaveform.py
```

12. Miscellaneous packages.

```
conda install tqdm      status bar showing progress
conda install pyasdf    asdf data format https://seismicdata.github.io/pyasdf/
```

13. **Updating the sln environment.**

A brute-force update can be achieved with the command

```
conda update --all
```

from inside the current environment.

Alternatively, to be safe, you can save the current environment:

- If you are already in an environment, deactivate it:  
`source deactivate`
- Update conda: `conda update conda`  
Note: Not sure if this is the same result as executing Steps 5-6.
- Rename the environment:

```
conda create --name sln_old --clone sln
```

This will redownload packages. There is a flag `--offline` to disable this option.

- Remove the original environment:

```
conda remove --name sln --all
```

- If you have this line in your `.bashrc` file

```
source activate sln
```

it is probably best to comment it.

- Proceed with Step 7 to create a new `sln` environment.

NOTE: If you install anything else under the same environment it may cause clash of the dependencies. `conda` will upgrade (or downgrade) the dependencies to fulfill the requirements of the later installation. It is probably safest not to install anything else in `sln`; instead, create a new environment.

## 1.1 Optional testing

1. Make sure you have activated the `sln` environment.
2. Test out a set of examples in `pysep`

```
check_getwaveform.bash
```

This will take several minutes as waveforms are fetched from IRIS, then processed locally. The script will compare the lists of output files with a set of pre-saved lists of output files.

3. Test out a LLNL example in `pysep`

- Install `pandas`:

```
conda install pandas
```

- Get the LLNL database client and install it.

```
cd $REPOS
```

```
git clone https://GITHUBUSERNAME@github.com/krischer/llnl_db_client.git
```

```
cd llnl_db_client
```

```
pip install -v -e .
```

- Edit `check_getwaveform.bash` to run the HOYA example only.