Python: Command Line Anaconda Setup for Python2 and Python3

Introduction

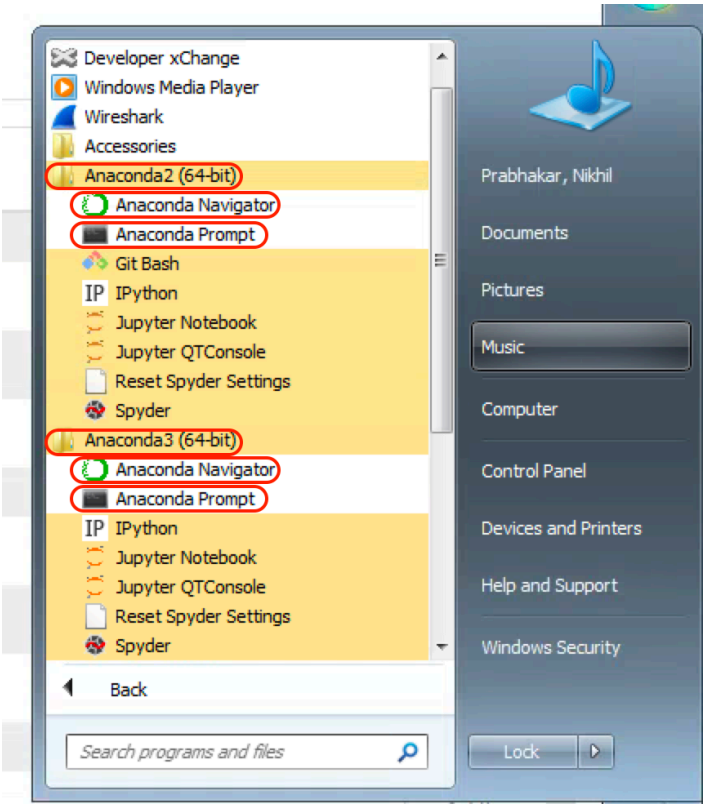
- For the purposes of this guide we are going to use the Anaconda distributions of Python 2 and Python 3
- Both versions will be installed side-by-side
- We will use Python virtual environments
- The command line interface will be our preferred interface

Installing Python on Windows

- Install Anaconda for Python 3.x from Developer Exchange
- Install Anaconda for Python 2.x from Developer Exchange
- You can use py.exe to invoke different versions of python -
    - Use py -2 for 2.x
    - Use py -3 for 3.x

```
py -2 --version
py -3 --version
```

- There are The best way to invoke the version you want though is to launch the appropriate Anaconda Prompt from the version you are interested in. You can use the Windows Start Menu to navigate to 'All Programs' and then find the appropriate version of Anaconda
    - Anaconda2 (Python v2.7.13)
    - Anaconda3 (Python v3.6.0)
- You can then choose to invoke one of the available interfaces
    - Anaconda Navigator (GUI)
    - Anaconda Prompt (command line)



- We will be using the standard Windows command line instead of the Anaconda command line to do our work.
    - This allows us to switch versions and virtual environments with greater ease in one place

```
REM Find the Python 2 version installed
C:\Users\u745163>py -2 --version
Python 2.7.13 :: Anaconda custom (64-bit)

REM Find the Python 3 version installed
C:\Users\u745163>py -3 --version
Python 3.6.0 :: Anaconda 4.3.1 (64-bit)

REM Find the version of pip installed with Python 2
C:\Users\u745163>py -2 -m pip --version
pip 10.0.1 from C:\Users\u745163\AppData\Roaming\Python\Python27\site-
packages\pip (python 2.7)

REM Find the version of pip installed with Python 3
C:\Users\u745163>py -3 -m pip --version
pip 9.0.1 from C:\FAST\anaconda\python36\win64\431\lib\site-packages
(python 3.6)

REM Find the version of virtualenv installed with Python 2
C:\Users\u745163>py -2 -m virtualenv --version
15.1.0

REM Find the version of virtualenv installed with Python 3
C:\Users\u745163>py -3 -m virtualenv --version
15.1.0
```

- The block above shows a few examples of command line options
    - '-m' imports the following module before executing the command

Configuring PIP for internal repositories

- For pip to be able to see the internal Artifactory PyPi repository, you will need to create a pip.ini
- Go to %appdata% (should default to C:\Users\<Your SID>\AppData\Roaming\)
- Create a directory called 'pip'
- Inside the directory create a text file called 'pip.ini' and copy-and-paste the following configuration into it

    - e.g. C:\Users\u745163\AppData\Roaming\pip\pip.ini

```
[global]
index = https://frs-art.jpmchase.net/artifactory/api/pypi/jpmc-public-pypi/
index-url = https://frs-art.jpmchase.net/artifactory/api/pypi/jpmc-public-
pypi/simple
extra-index-url = https://frs-art.jpmchase.net/artifactory/api/pypi/GTI-
pypi-virtual/simple
trusted-host = frs-art.jpmchase.net
disable-pip-version-check = true

[install]
require-virtualenv = true

[uninstall]
require-virtualenv = true
```

- Only add add the 'require-virtualenv = true' entries after you are done setting up the virtual environment otherwise you will not be able to use pip outside of the virtual environment

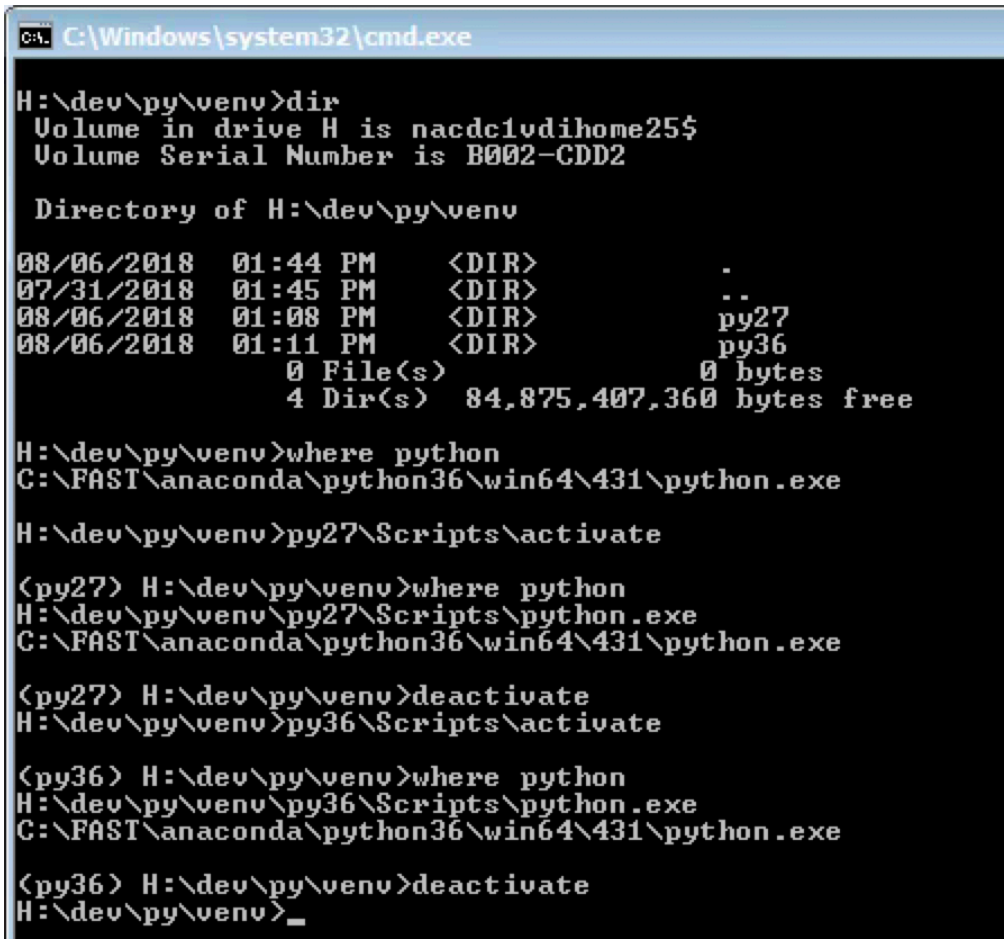
Creating Virtual Environments

- You can create virtual environments from the Windows command line -

```
py -2 -m virtualenv H:\dev\py\venv\py27
py -3 -m virtualenv H:\dev\py\venv\py36
```

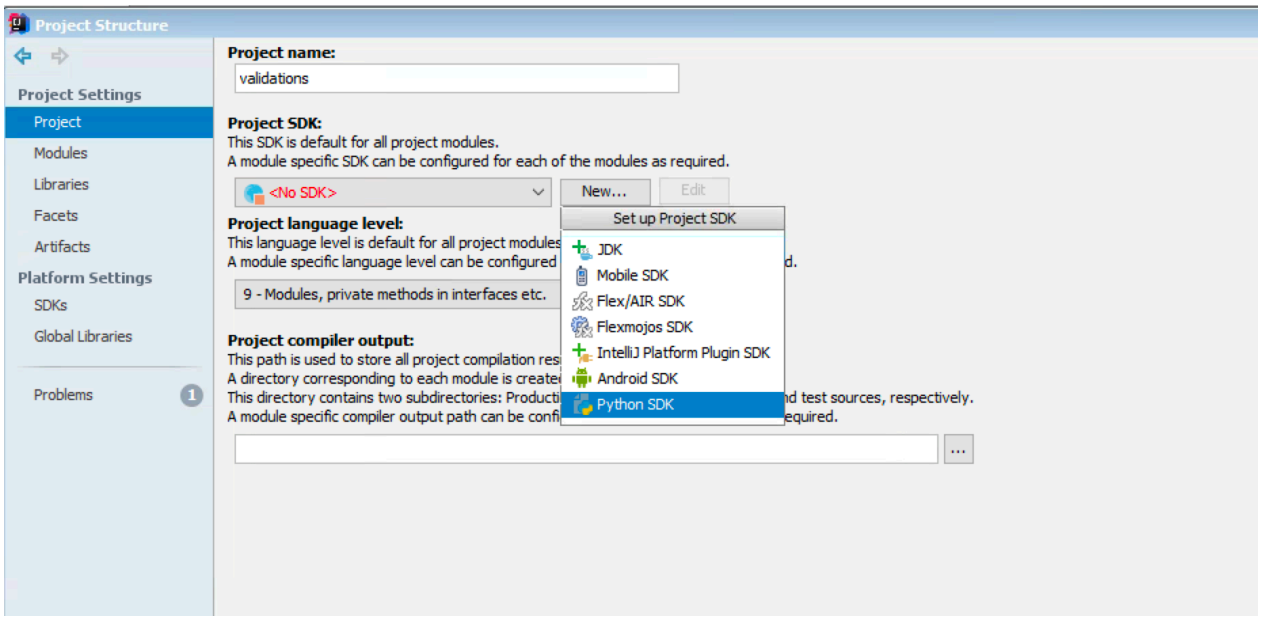
How to Tell Which Virtual Environment is Currently Loaded

- Your command line will display the currently loaded virtual environment to the left of your command prompt
- You can also type 'pip -V' to display the environment where pip is loading from

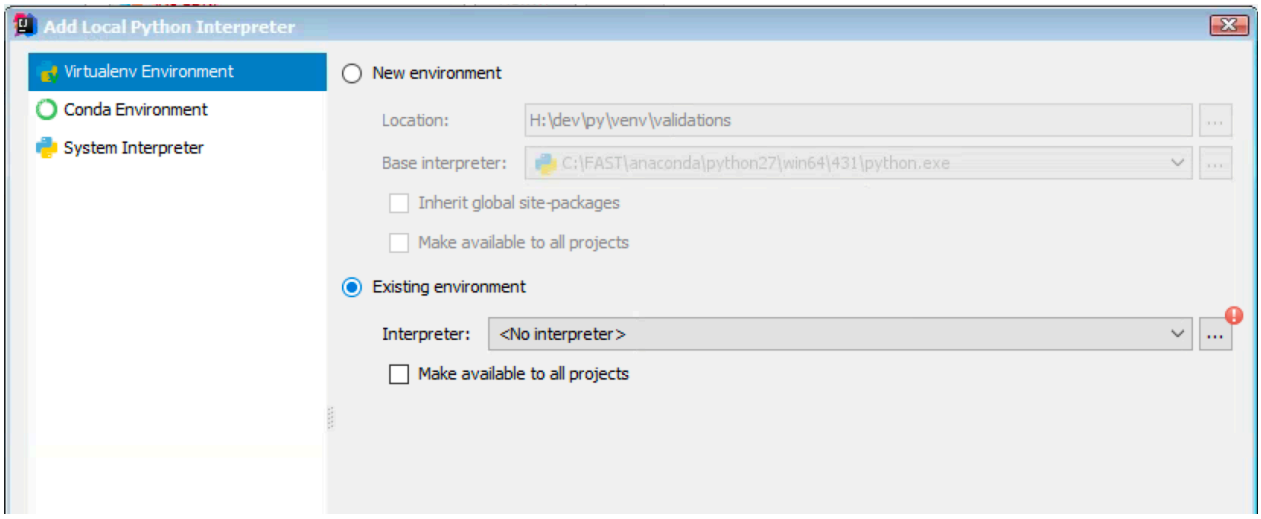


Configuring IntelliJ with an Existing Virtual Environment

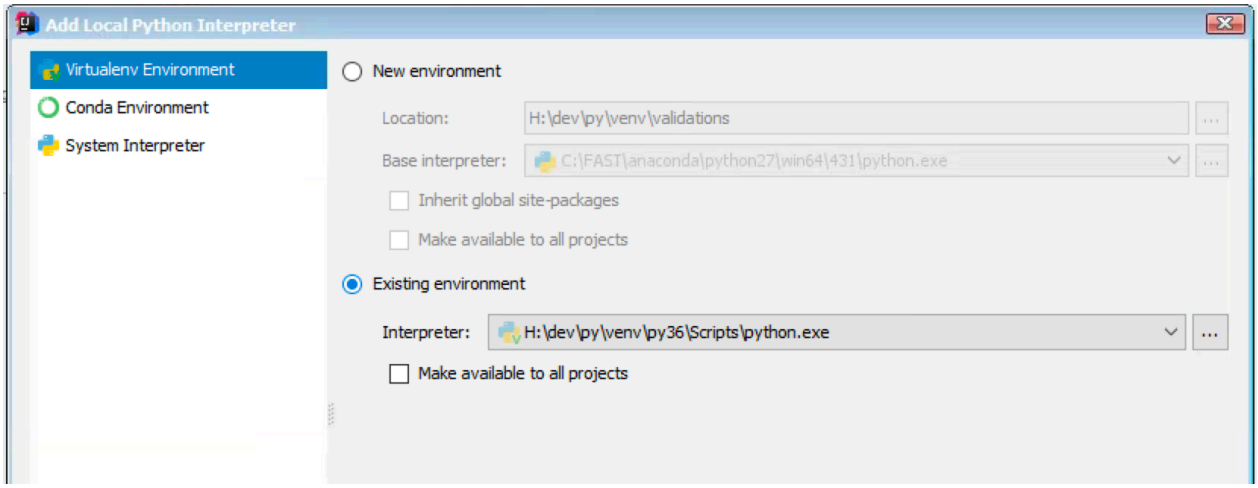
- Git clone the repo you want
- Launch IntelliJ and open the newly cloned project
  - You should get a bubble or alert telling you that IntelliJ detected a Python framework
- Go to 'File' 'Project Structure...' and click on 'Project' on the left



- Under 'Project SDK' click the 'New' button next to where it says '<No SDK>' and select 'Python SDK'
- Select 'Add Local'
- On the 'Add Local Python Interpreter' screen select 'Existing Environment' and click the button to the right of where it says <No Interpreter>



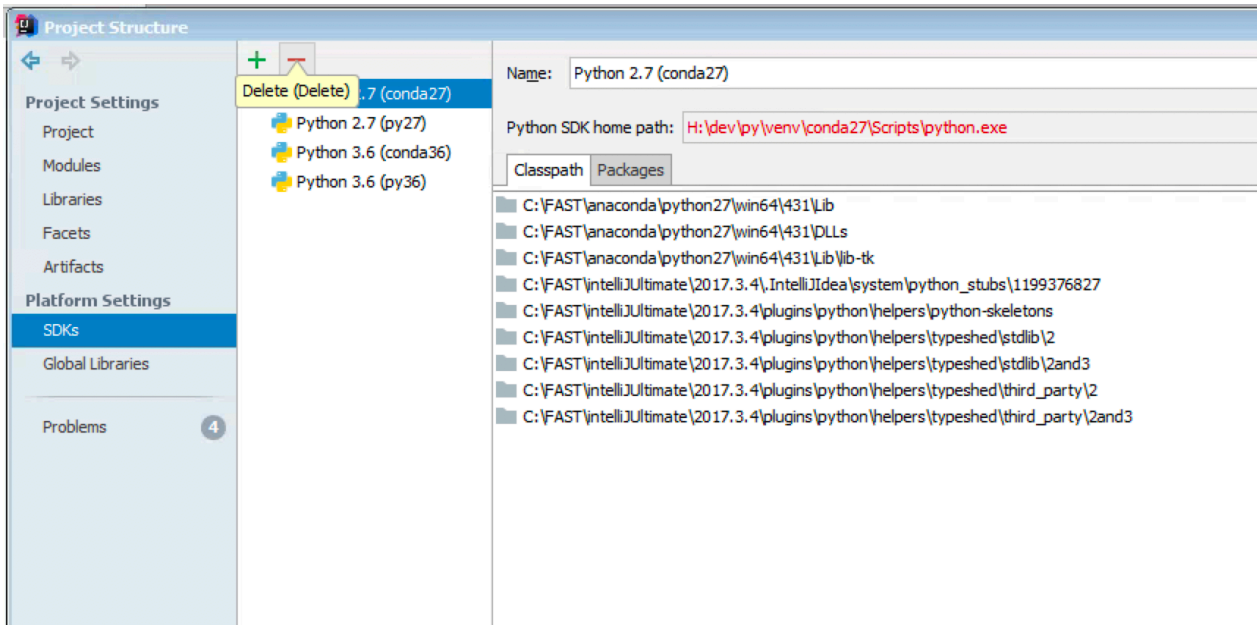
- Browse to the the virtual environment you created above and find the 'Scripts' sub-directory and select 'python.exe'



- Click 'OK' and then 'OK' again
- It will take a few minutes for the new environment to be indexed by IntelliJ after which it will automatically configure the virtual environment for the project

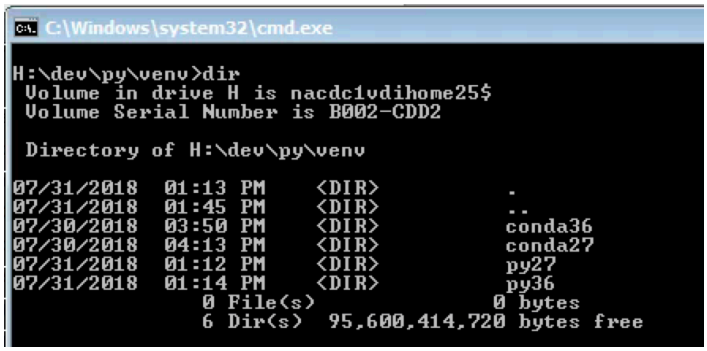
Removing Old Virtual Environments From IntelliJ Projects

- Go to 'File' 'Project Structure...' and click on 'SDKs' on the left
- Select the SDK you want to delete by clicking on it and use the minus sign above to delete it from the list

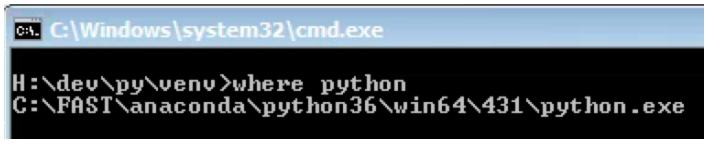


Managing Virtual Environments

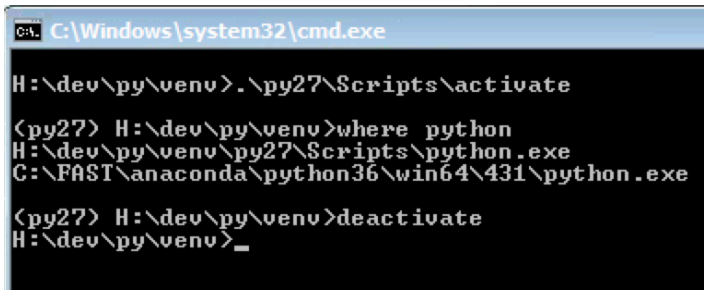
- By default virtual environments are created in your profile directory on the C:\ drive. So if your VM were to crash or if you received a new VM, these settings would be lost
- You can use virtualenvwrapper to manage your virtual environments if you like (this is just an FYI - I do not use it and will not be documenting it)
- I create all my virtual environments in a specific directory - H:\dev\py\venv\
  - So if I list the contents of that directory I can see all my virtual environments
  - In the example below I have four virtual environments - conda36, conda27, py27 and py36



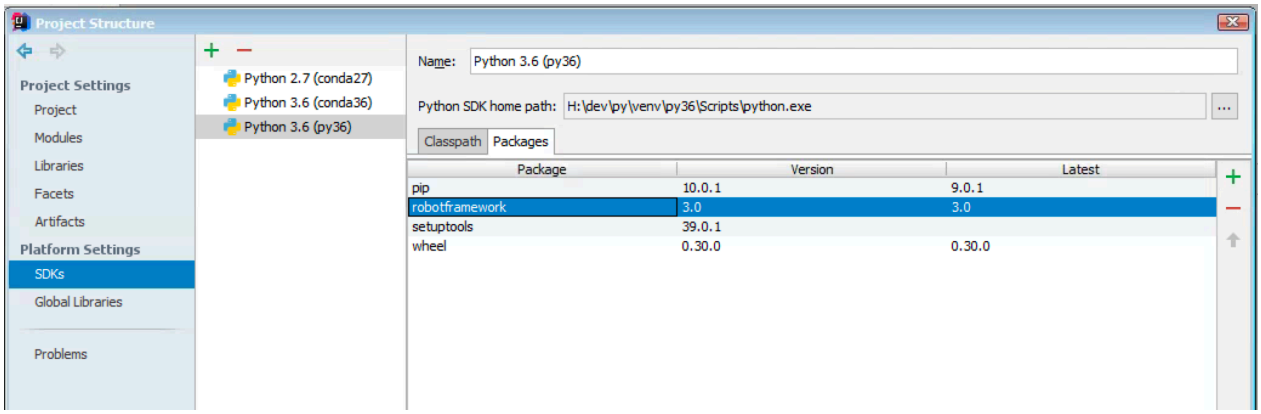
- I can check the first python executable in my current path by using 'where python' ('which python' on Linux, MacOS or UNIX)
  - My default python installation is under 'C:\FAST\anaconda\python36\win64\431\python.exe'



- I can load any virtual environment by invoking the 'activate' script in the virtual environment I want -
  - Once the environment is loaded you will notice the environment name is displayed to the left of the command prompt
  - If I check for the first python executable in my path you will notice that the virtual environment now takes precedence
  - The 'deactivate' command deactivates the current virtual environment



- If you install packages while a virtual environment is loaded (e.g. pip install robotframework) it will be added to that virtual environment
    - If you happen to be using that virtual environment in IntelliJ, the package will also become available in IntelliJ



Links

- Conda cheat sheet
- Firmwide Repository Service
- How To Configure Python For FRS Artifactory
- How To Upload a Pypl Package To FRS Artifactory
- Installing packages using pip and virtualenv
- Machine Learning in Python (with pip.ini config)
- Getting started with conda
- JPMC pip config
- Python Core Home
- Managing Virtual Environments
- conda - managing virtual environments
- Python Development Environment on macOS High Sierra