Python with Qlik Sense AAI

Environment Setup

INTRODUCTION

The intended purpose of this document is to create a virtual Python environment which some/many of our projects can leverage. One of the many benefits of creating a virtual environment is that you can isolate different package distributions for different projects that require different dependencies. For example, one project might leverage BeautifulSoup3 and another might leverage BeautifulSoup4. Without attempting to convert either project to use the same version, we can install both packages in their own isolated environments.

REQUIREMENTS

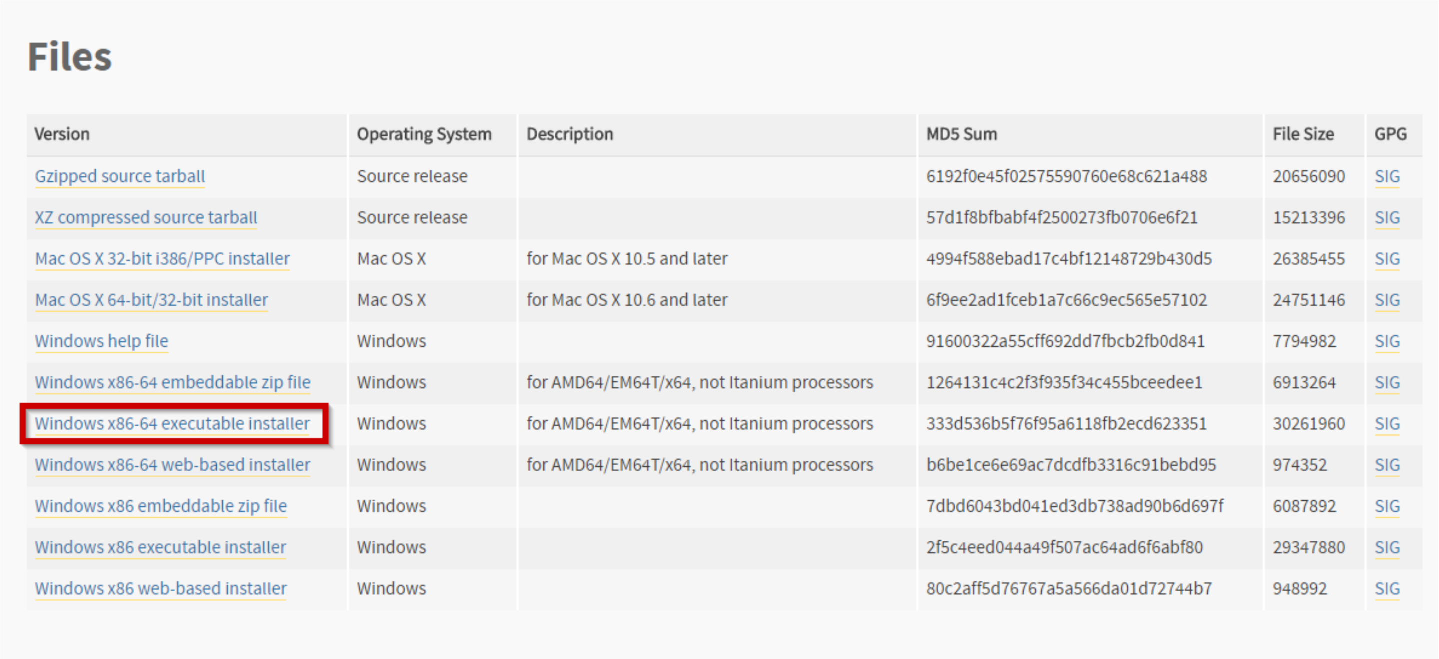
- Qlik Sense June 2017+
- Python 3.5.3 64 bit *(3.4+, however all of my examples are validated against 3.5.3)*
- Python Libraries: grpcio, virtualenv, virtualenvwrapper-win

LAYOUT

- Install Python
- Prepare your Virtual Python Environment

INSTALL PYTHON

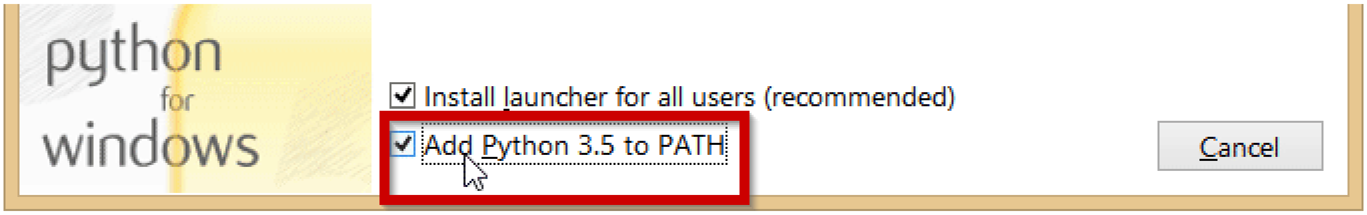
1. Download Python 3.5.3 64 bit
    - Direct link: https://www.python.org/ftp/python/3.5.3/python-3.5.3-amd64.exe
        - https://www.python.org/ → Downloads → All Releases → Python 3.5.3 →



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		6192f0e45f02575590760e68c621a488	20656090	SIG
XZ compressed source tarball	Source release		57d1f8bfbabf4f2500273fb0706e6f21	15213396	SIG
Mac OS X 32-bit i386/PPC installer	Mac OS X	for Mac OS X 10.5 and later	4994f588ebad17c4bf12148729b430d5	26385455	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6f9ee2ad1fceb1a7c66c9ec565e57102	24751146	SIG
Windows help file	Windows		91600322a55cff692dd7fbcb2fb0d841	7794982	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64, not Itanium processors	1264131c4c2f3f935f34c455bceedee1	6913264	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64, not Itanium processors	333d536b5f76f95a6118fb2ecd623351	30261960	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64, not Itanium processors	b6be1ce6e69ac7dcdfb3316c91bebd95	974352	SIG
Windows x86 embeddable zip file	Windows		7dbd6043bd041ed3db738ad90b6d697f	6087892	SIG
Windows x86 executable installer	Windows		2f5c4eed044a49f507ac64ad6f6abf80	29347880	SIG
Windows x86 web-based installer	Windows		80c2aff5d76767a5a566da01d72744b7	948992	SIG

2. Run the installer as Administrator, and select the option to 'Add Python 3.5 to PATH'. This will allow you to simply type 'python' or 'pip' into the command prompt without having to 'cd' to the location where those assets exist
    - *NOTE – if you receive an error saying that a KB couldn't be installed, close the installer and run windows update on your instance until there are no more updates, then try reinstalling.



☑ Install launcher for all users (recommended)

☑ Add Python 3.5 to PATH

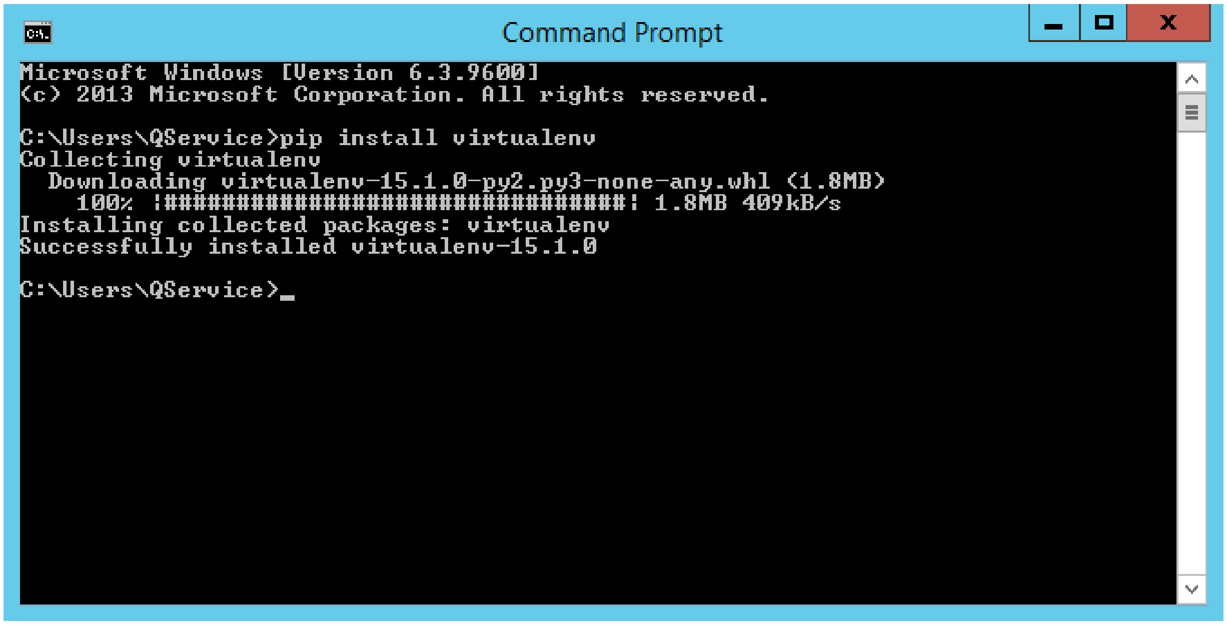
3. Complete and finish the installer

PREPARE YOUR VIRTUAL PYTHON ENVIRONMENT

ALERT

Virtual environments are not necessary, but are frequently considered a best practice when handling multiple Python projects. The ARIMA forecasting example that you might be implementing in another guide requires very specific pacckage distributions that might interfere with other projects, therefore I am suggesting the use of a virtual environment rather than implementing those distributions system-wide.

1. Open a command prompt (⊞+'cmd')
2. First, we will install 'virtualenv' which will allow us to isolate our Python projects, in case multiple projects require different distributions and versions. To do so, execute:
   - pip install virtualenv

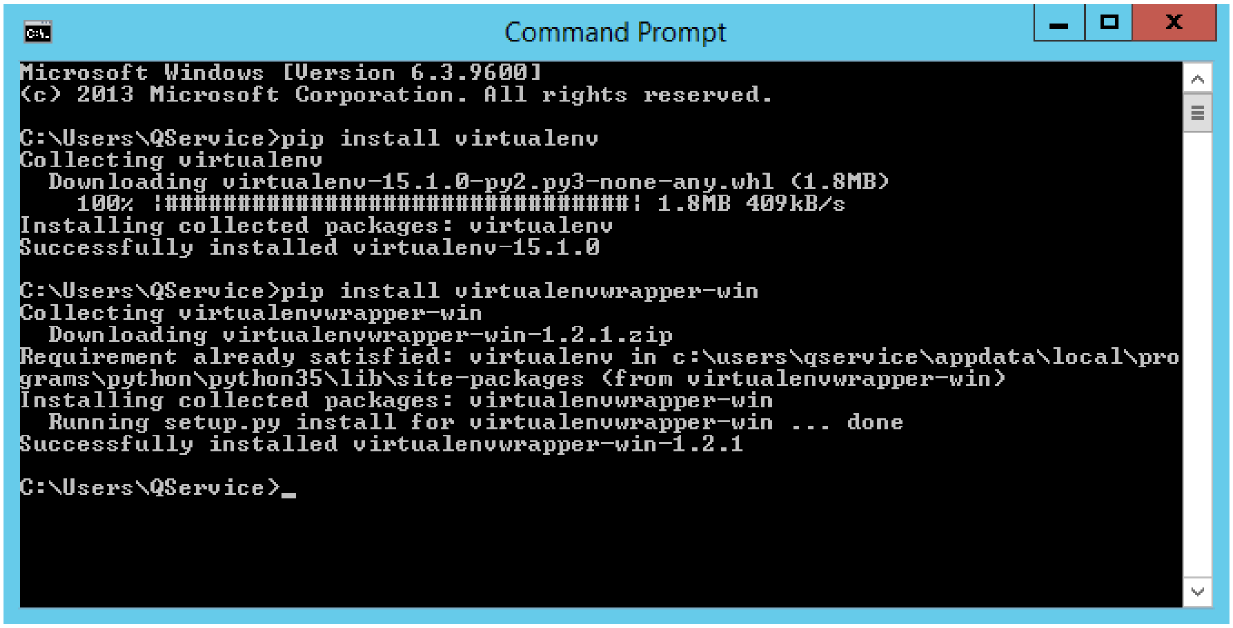


```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\QService>pip install virtualenv
Collecting virtualenv
  Downloading virtualenv-15.1.0-py2.py3-none-any.whl (1.8MB)
    100% |################################| 1.8MB 409kB/s
Installing collected packages: virtualenv
Successfully installed virtualenv-15.1.0

C:\Users\QService>_
```

3. Now we will install a helpful package which will help us maintain our virtual environments. Execute:
   - pip install virtualenvwrapper-win



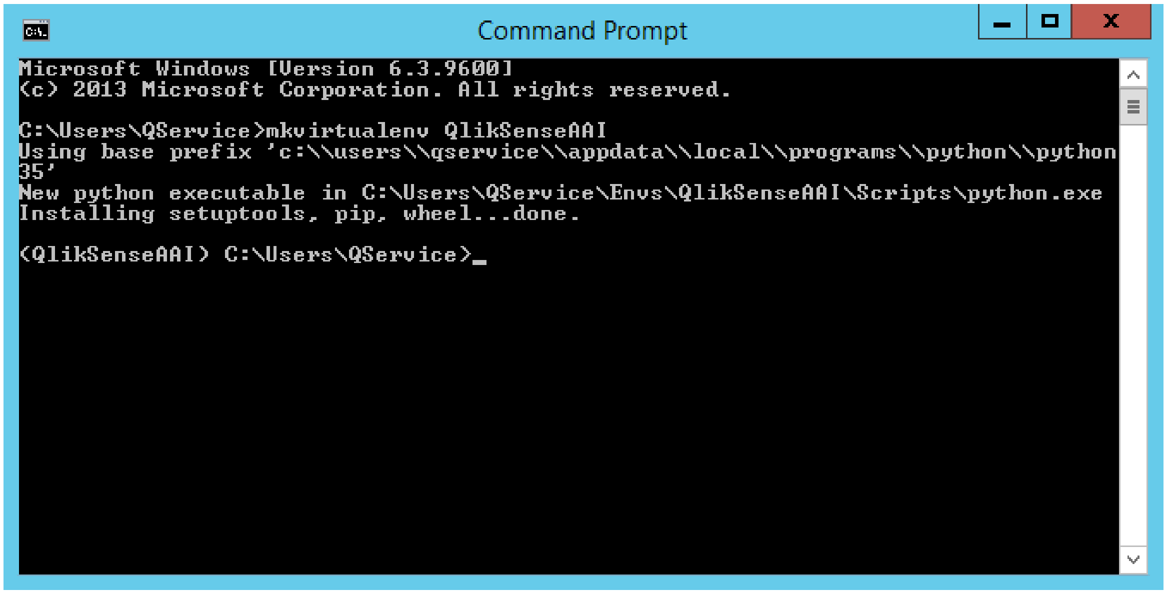
```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\QService>pip install virtualenv
Collecting virtualenv
  Downloading virtualenv-15.1.0-py2.py3-none-any.whl (1.8MB)
    100% |################################| 1.8MB 409kB/s
Installing collected packages: virtualenv
Successfully installed virtualenv-15.1.0

C:\Users\QService>pip install virtualenvwrapper-win
Collecting virtualenvwrapper-win
  Downloading virtualenvwrapper-win-1.2.1.zip
Requirement already satisfied: virtualenv in c:\users\qservice\appdata\local\pro
grams\python\python35\lib\site-packages (from virtualenvwrapper-win)
Installing collected packages: virtualenvwrapper-win
  Running setup.py install for virtualenvwrapper-win ... done
Successfully installed virtualenvwrapper-win-1.2.1

C:\Users\QService>_
```

4. Once installed, lets create a virtual environment. In this case, I'm going to name it 'QlikSenseAAI'. This will create a new virtual environment under my user account with the path 'C:\Users\QService\Envs\QlikSenseAAI'. Execute:
   - mkvirtualenv QlikSenseAAI



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\QService>mkvirtualenv QlikSenseAAI
Using base prefix 'c:\\users\\qservice\\appdata\\local\\programs\\python\\python
35'
New python executable in C:\Users\QService\Envs\QlikSenseAAI\Scripts\python.exe
Installing setuptools, pip, wheel...done.

(QlikSenseAAI) C:\Users\QService>_
```

Note the (QlikSenseAAI) preceding the directory now, meaning that we are in the new virtual environment. Anything we install now will be specific to this project and available to the projects we connect to this environment.

We have now created a virtual Python environment where we can install specific dependencies that projects require. In the other module documents, we will either leverage this virtual environment, or create a new environment if necessary.