Python Tool Installation

This page provides references for installing Python tools onto various computer configurations.

Installing Python 3

Python 3 is required to run 699util. Python 2 is not supported. Installation instructions for Python on various operating systems follow. The tools presumably work on Windows, but we do not have specific instructions for using the tools on Windows at this time.

Macintosh Installation

Depending on which version of the Tools you want you run, you will need either Python 3.9.5 or 3.4.3. For MOMA tools version >= 4.00.00, Python 3.9.5 is required. All older versions require Python 3.4.3.

Download Python 3.9.5 directly from the Python Software Foundation's downloads page.

Download Python 3.4.3 directly from the Python Software Foundation's downloads page.

It is **not** recommended that you use the Python distribution packaged with your operating system unless you know what you are doing, because it is easier to download certain third-party pre-compiled Python packages (e.g., Numpy) if you have the python.org distribution. If you are unsure whether your Python is a the python.org version or the Apple version, download and run the installer anyways, it will not break anything. You will need administrative permissions to run the installer.

The installer will infer the location of your shell's configuration file and add a line that will register the new version of Python as an executable program. It will look something like the following:

```
export PATH="/Library/Frameworks/Python.framework/Versions/3.4/bin:${PATH}"
```

Sometimes, particularly on new computers, the installer guesses the configuration file's location incorrectly. (Generally, the most common mistake is putting the line in "~/.profile" rather than "~/.bash_profile" if the latter hasn't been written yet.) **Therefore, it is important to open up your terminal and test the "python3" command yourself.** Type `which python3` at the command line and press enter. If the result looks like `/usr/bin/python3`, you'll probably need to fix your configuration file manually. Open up "~/.bash_profile" in an editor (create it if it isn't present) and copy/paste the line from the beginning of the paragraph into the file.

Warning: A subtle bug can occur when *changing* the version of Python 3 that is installed. If you are installing Python 3 for the first time, then you can ignore this warning. Otherwise, upgrading Python 3 will change your ~/.bashrc file so that

```
export PATH="/Library/Frameworks/Python.framework/Versions/3.4/bin:${PATH}"
```

occurs *after* the line `source ~/py699/bin/activate`. Be sure to edit ~/.bashrc so that `source ~/py699/bin/activate` occurs last. This ensures that the virtual environment's Python installation is what's running when the `python` command is given.

Linux Installation

Linux distributions generally come prepackaged with some version of Python. At this point, the default is usually Python 2, but you may be able to install Python 3 using your system's package manager. Make sure the version is Python 3.4, not Python 3.5.

If your package manager does not have the correct version of Python, then you will have to compile it yourself. It is **very important** that you do not overwrite the system default version of Python! Follow the steps below:

1. Download the bzipped/gzipped/xzipped source code archive for the desired version of Python from the Python Software Foundation's downloads page.
2. Untar the archive (`tar -xvf Python-3.X.X.tar.bz`) and then navigate to the newly-created directory.
3. If desired, read the README file packaged with the source and review any configuration options. Run the configuration script and specify a location where you would like the new Python interpreter to be installed. If you have administrative permissions, "/usr/local" might be a good location. Otherwise, you can create a "local" directory in your home folder. In either case, the command will look like the following: `./configure --prefix=/path/to/installation/directory`. **Do not run ./configure without a --prefix option.** Otherwise, you will install to the "/usr" directory and overwrite the system version of Python, most likely breaking the Linux system administration tools!
4. Run `make` to compile the source. If you are an especially careful person, you could run `make test` afterwards, but to this date I have never seen a compiled version of Python not work.
5. Run `make install`. You may need to use `sudo` if you specified a prefix in a restricted location.
6. Edit your shell's configuration file (most likely ~/.bashrc) so that the directory containing your new Python verison (e.g., /usr/local/bin) is ahead of the directory containing the original system version of Python (e.g., /usr/bin) on the PATH.

Installing Supplementary Files

In order to actually use 699util, you will need some supplementary files (which are also used by the Data View applications), also known as the "GSE folder", as well as data to analyze.

For MOMA, supplementary files can be installed by following the momadatview installation instructions.

For SAM, these supplementary files are included in the SAM Data View Installer. The process for installing the supplementary files for LADEE NMS and MAVEN NGIMS analysis is TBD.

The Python utilities are generally good about detecting these supplementary files if you put them in familiar places (usually your home folder). If you put them in other places, you will have to edit the ~/.699config.INI file and point to the location of these key files.

Installing 699util for Users (i.e., non-developers)

Note that this is also the procedure to follow for upgrading 699util to a later version. The installation and upgrade processes are the same.

1. Download the installation package for the latest version of the 699util tools from the MOMA Python Package or (for all other missions) 699util page. Mac users should download the disk image (.dmg) file (but can use the compressed tar archive if they want), and Linux users should download the compressed tar archive (.tar.gz).
2. If you downloaded the disk image, mount (open) the disk image. Using your terminal, change directories into the installation disk image, which is located under "Volumes". The command will probably be: `cd /Volumes/699util_Installer`. If you downloaded the compressed tar archive, uncompress it and cd into the resulting folder. If you downloaded the tar archive to your "Downloads" folder, then the commands will look something like this:

```
cd ~/Downloads
```

```
tar -xvf 699util-X.YY.Z-installer.tar.gz
```

```
cd 699util-X.YY.Z-installer
```

3. There is an executable script in the folder called "install699util.sh". Execute it by typing `./install699util.sh`.
4. After the installation program is done (it will print out a message saying so when it is finished), close your terminal and start it up again.
5. Note that installing 699util does not give you the supplementary files you need to interpret telemetry data, nor does it give you any data to analyze. See the Installing Supplementary Files section, above.

If you wish to use plotting scripts such as tmplot.py and momascan.py, skip ahead to the Installing Plotting Tools section.

Installation for Developers

This is only for people who are actively developing Python scripts. To complete this installation, you must be connected to the internet

MOMA Developer Prerequisites

MOMA Developers should install Xcode from the App Store, if they have not already done so.

Installing config699

1. Open up a terminal tab and type the following.

```
ssh AUID@repos699.gsfc.nasa.gov -L 6990:localhost:3690 # AUID is your launchpad username
```

2. Switch to a new tab. If SVN prompts you for login credentials, your username is your first initial followed by your last name (e.g., mlefavor) and your password is "user4svn" followed by your first and last initial (e.g., user4svnml). Type:

```
svn co svn://localhost:6990/labcode/config699 ~/config699
```

3. Add the line `source ~/config699/bash_config.sh` to your bash configuration file (.bashrc or .bash_profile).

4. Close the tab you were working with and start a new tab. (You can leave the tunnel open if you are going to be following the steps below).

Installing 699util

1. Make sure you have an ssh tunnel open (see step 1 in the Installing config699 section, above).
2. Update your config699 folder: `svn up ~/config699`.
3. Check out the python code: `699checkout.sh 699util`.
4. If you do not have it already, download virtualenv here: http://www.virtualenv.org/en/latest/. Follow the instructions on the website to install the utility.
5. Navigate to your home directory and create a virtual environment, like so: `virtualenv –p /[path to python install]/bin/python3 ~/py699`. If you installed Python 3 through the Python website as suggested in this article, then the command will be `virtualenv -p /Library/Frameworks/Python.framework/Versions/3.4/bin/python3 ~/py699`.
6. Add the following line to your bash configuration file (.bash_profile or .bashrc): `source ~/py699/bin/activate`
7. Restart your terminal (close your tabs and open anew).
8. Navigate to your 699util directory (probably under ~/labcode/699util)
9. Type `pip install -r requirements.txt`
10. You will now have all the dependencies you need.

Installing Plotting Tools

1. Download the following dmg and tar files:

Gnuplot

XQuartz 2.7.7 for OSX 10.10 (Yosemite)

XQuartz 2.7.4 for OSX < 10.10

2. Install xquartz first, then install gnuplot.
3. run gnuplot and use the command "set term x11".
4. You should now be able to use python plotting tools (c_tmplot.py, c_momascan.py, etc.).

Distributing 699util

These instructions assume you have a developer's setup for the Python tools.

1. Navigate to the 699util directory (probably under ~/labcode/699util).
2. Add any new scripts or modules to the setup.py script and update VERSION_STRING.
3. Change the version number in the setup.py script.
4. From the 699util directory, run `./package.sh`. This will create a "dist" folder, inside of which you find a DMG file (699util-) and a tar archive (699util-X.YY.ZZ-installer.tar.gz).
5. Put the DMG file and tar archive on the Amazon server. Put them under 699_bin/699util. Right-click on both files and make them public.
6. Update the Version History on the 699util page, and add links to the DMG and tar file.

7. Make sure to update the pointers in the Installing 699util section, above.

Other Resources

- How to install Python modules
  - The basic, officially-sanctioned Python installation process: http://docs.python.org/3.3/install/
  - The python community as a whole, and I as well, heartily recommend using the "pip" python package manager. You can use pip to search PyPI for useful packages, download them, and install them with only a few simple commands. To get pip, follow the instructions here: http://www.pip-installer.org/en/latest/installing.html. (Note that you will have to download and install setuptools, another Python package, as part of the process. The website gives instructions.) Once you have pip, all you need to do to install a package is type `pip install name-of-package`.

Revision #2
Created 23 March 2023 14:32:35 by Nick Dobson
Updated 22 April 2026 20:28:49 by Bradley Tse