**Instruction from Patrick:**

* Place the entire "powerOps" folder at the "model\_w2" level in the CalGUI hierarchy. The scripts use relative referencing to find the DV dss file and HEC jar files. Therefore, if you'd like to place it elsewhere, some re-referencing will be required.
* There are eight py files in this folder. The two main files are ltgen.py and swpgen.py. At the top of both files, you'll find string variables that reference the relative location to the DV file, name of the output DSS file, start date, end date, DSS f-part, and DSS a part. These will need to change to fit with your needs.
* As you scroll down each instantiation of either a powerPlant or pumpPlant object, you'll see pathnames of required input. For example, the first object is a powerPlant object called trinity. The required inputs are S1 (storage), and C1 (release). To port this to CalLite, you'll need to reference the corresponding arc. I've attached the spreadsheet of the mappings between CalLite and CalSim II.
* To run the CVP hydropower calcs, pass "ltgen.py" as an argument to Jython. To run the SWP hydropower calcs, pass "swpgen.py" as an argument to Jython. The script will use the reference DV file and append either an \_LTGen.dss or \_SWPGen.dss to the name. You're welcome to change it.
* We need to pre-process the mapping of CalSim/CalLite corresponding arcs before it'll work.

**Verify your machine have 32 bit Java:**

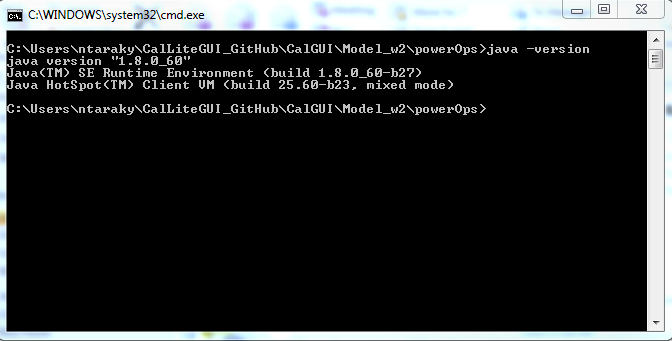
Run “java –version” in command prompt

A 64 bit Version would include “64-Bit” in the output. See example below.

java version "1.x.x\_xx"  
Java(TM) SE Runtime Environment (build 1.x.x\_xx-bxx)  
Java HotSpot(TM) 64-Bit Server VM (build xx.xx-bxx, mixed mode)

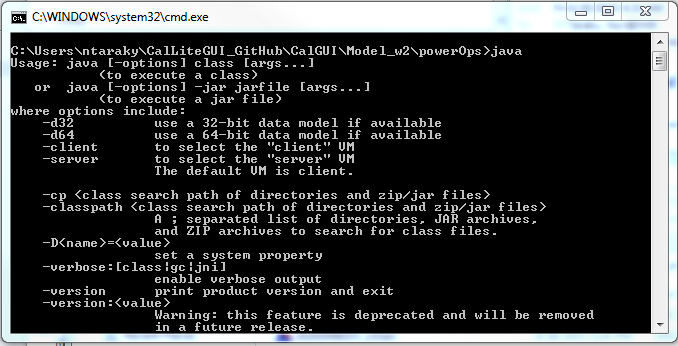
A 32 bit Version would not mention “64-Bit” in the output. See example below.

java version "1.x.x\_xx"  
Java(TM) SE Runtime Environment (build 1.x.x\_xx-bxx)  
Java HotSpot(TM) Client VM (build xx.xx-bxx, mixed mode)



**Enable Java Environment:**

Run “java” in command prompt to enable java environment



**Download and Install Jython:**

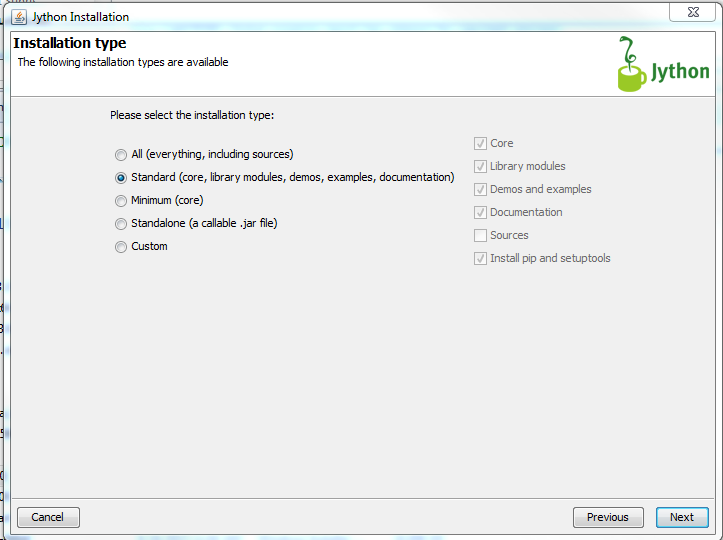
Download Jython from the following link.

<http://www.jython.org/downloads.html>

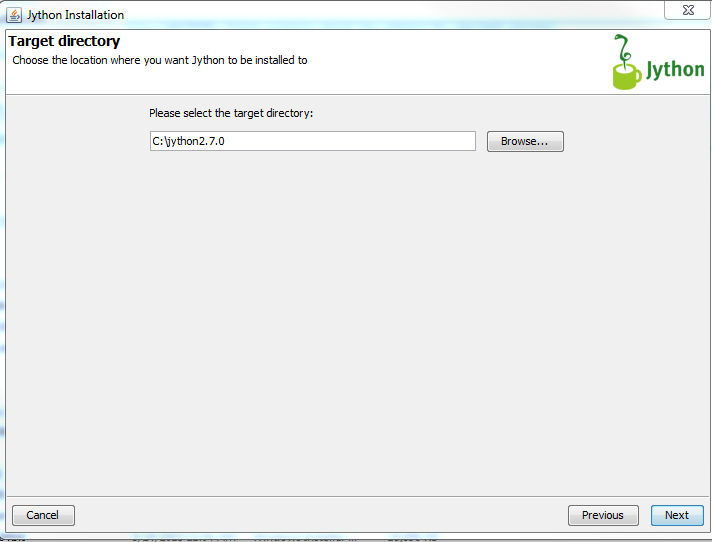
Chose the “installer”, not the “standalone Jar” from the two given options:

* Download Jython 2.7.0 - Installer : Executable jar for installing Jython (Correct)
* Download Jython 2.7.0 - Standalone Jar : For embedding Jython in Java applications (Incorrect)

Run the installer pack. Select “standard” installation.

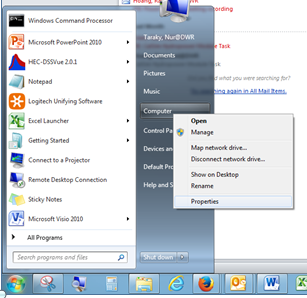


Select “C:/Jython2.7.0” Directory for installation.

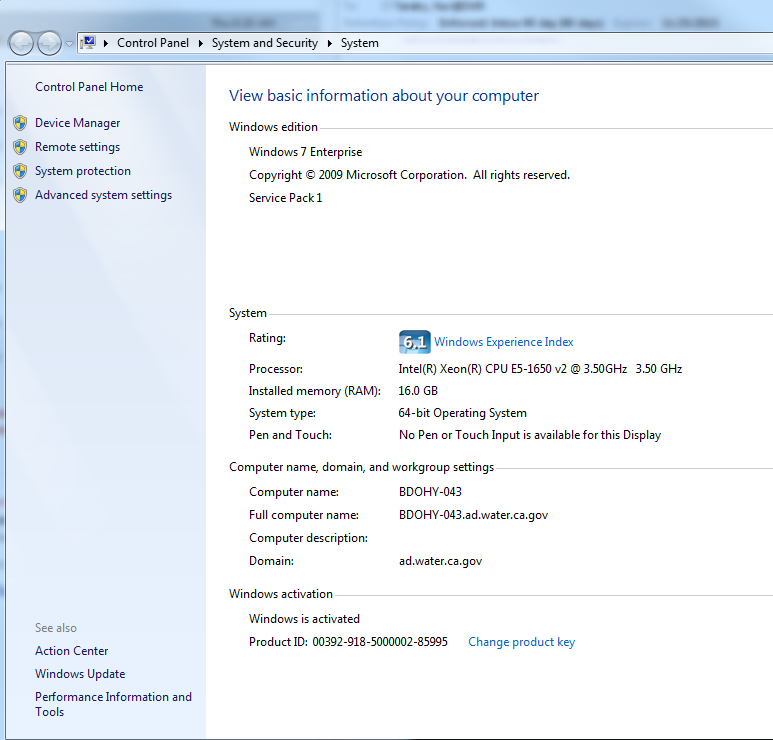


**Add Path Names:**

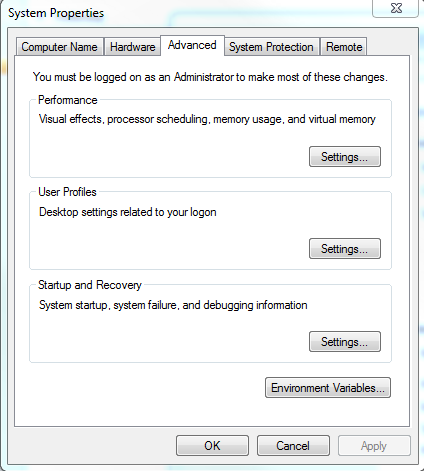
From “start menu” right click on “Computer” and select “properties”.



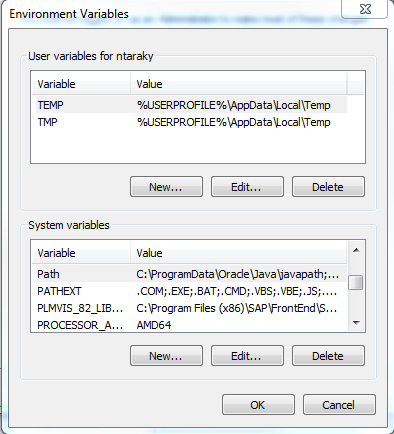
Select “Advanced System Settings” from the “System” window



Select “Environment Variables”



Scroll down to “Path” and select “Edit” from “System variables” window.

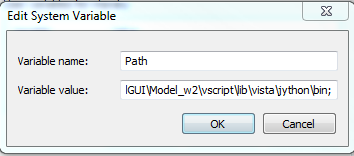


Add following path name in “variable value” window

“C:/Jython2.7.0/”

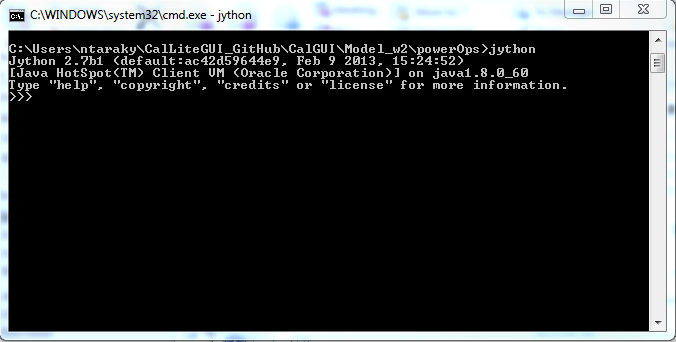
Search for “Jython.bat” file and add the path name in the “variable value” window as well. In my case I had the following path.

“C:\Users\ntaraky\CalLiteGUI\_GitHub\CalGUI\Model\_w2\vscript\lib\vista\jython\bin”



**Enable Jython Environment:**

Run “jython” in command prompt to enable Jython environment.



**Run Jython Script:**

Run “Jython ltgen.py” and “jython swpgen.py” in command prompt from the Power Ops folder to generate DSS file for power generation.

