pospy - v 1.0

GUI Automation tool for POS simulator.

Table of Content:

1. Overview ----------------------------------------------------p.2

2. Workstation setup ----------------------------------------p.3

3. Running programs ------------------------------------p.4

4. Programs description --------------------------------p.

4. Debugging ----------------------------------p.5

**Overview**:

POSPY is a set of programs - or at least a working attempt - to alleviate the pain that comes with testing SCA commands manually.

POSPY utilizes GUI automation Python module called PyAutoGUI, which used to programmatically control the mouse & keyboard.

The program requires a fixed position of the POS simulator on the screen, as all saved variables are assigned to fixed position on x,y coordinate system. x and y axis start on the upper left corner (0,0) and end at lower right corner (Ex., 1920,1080).

To securely fix POS simulator on the screen, just use “Picture of the Windows logo key + ←" (Windows logo key+left arrow) shortcut. This will allow POS simulator to take exactly half of the left screen of user's monitor.

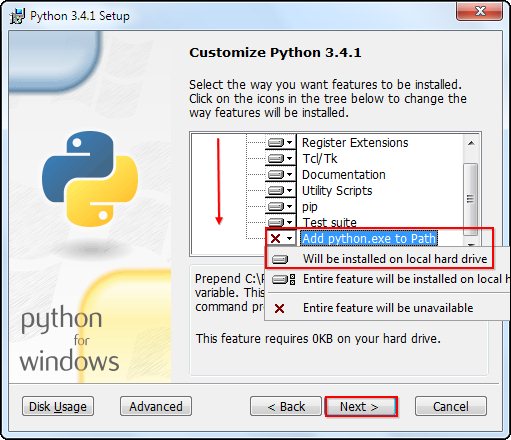
In writing these program 1920x1080 screen resolution was used, but it can be applied to any screen sizes. See “Finding location” section.

**Workstation setup:**

In order to use POS-Automation tool you need to setup your system beforehand. That includes following steps below:

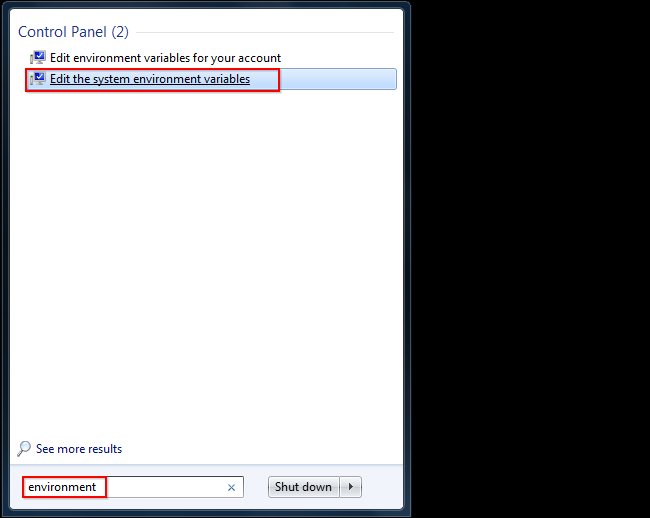
1. **Installing python 2.7 (If you have python on your system, you may skip this section)**
   1. Go to <https://www.python.org/downloads/> and download the latest **2.7** version for Windows. *Do not download Python 3 version!*
   2. Run the installer;

On Customize screen, scroll down in the window and find the “Add Python.exe to Path” and click on the small red “x.” Choose the “Will be installed on local hard drive” option then press “Next.”

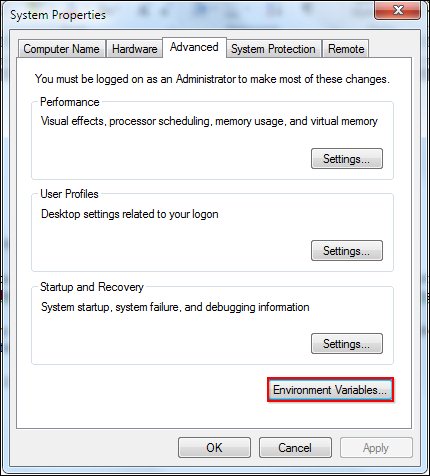


* 1. After python is installed, we need to add Python to System Path Variable. In order to do that, Click on Windows emblem and type ‘environment’ – choose “Edit the system environment variables”

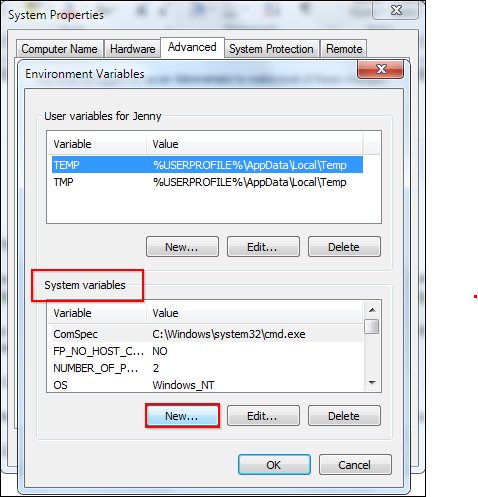
When the “System Properties” window appears, click on “Environment Variables…”



When the “System Properties” window appears, click on “Environment Variables…”:

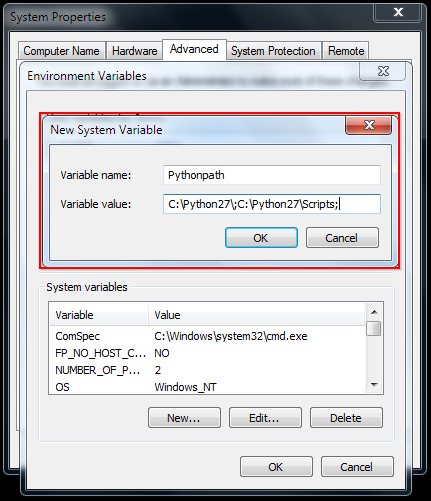


Once you have the “Environment Variables” window open, direct your focus to the bottom half. You will notice that it controls all the “System Variables” rather than just this associated with your user. Click on “New…” to create a new variable for Python.



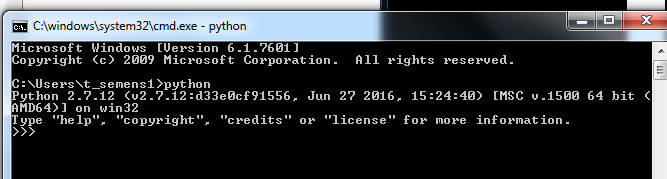
Simply enter a name for your Path and the code shown below. For the purposes of this example we have installed Python 2.7.3, so we will call the path: “Pythonpath.”

The string that you will need to enter is: “C:\Python27\;C:\Python27\Scripts;”



Press “OK,” then “OK,” then “OK,” then the red “X” to accept all changes and exit the “System Properties” window.

1.2 Check that you’ve installed python successfully. Open ‘cmd’ or Command Prompt and type *python*. You should see something similar to this. If you getting error message, try repeating step 1.2



To exit from python scripting mode in cmd, type *quit()* and press enter:

**2. Installing pyautogui and its dependencies:**

**2.1 Install Pillow**

In Command Prompt, navigate to C:\Python27\Scripts and type easy\_install pillow

**2.3 Install PyAutoGUI**

Once pillow installed, type *pip install pyautogui*

**2.4 Check installation**

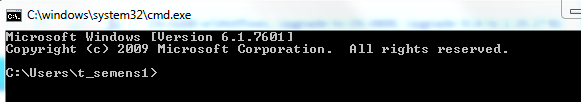
Once pyautogui installed, type pip list (or pip freeze) to check whether all needed packages are installed:

Seems like everything is installed, so we can move to the next step now.

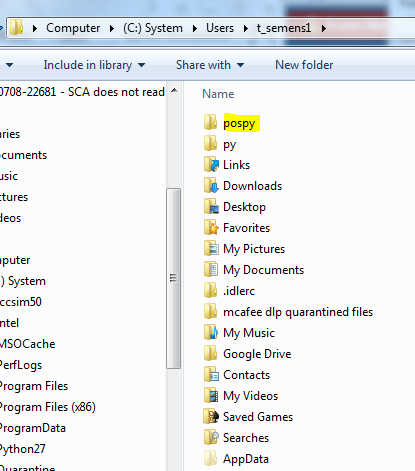
**3. Locating your programs.**

**3.1 Placing *pospy* folder**

As you noticed, whenever you open a Command line, the path is always pointing to your user directory:

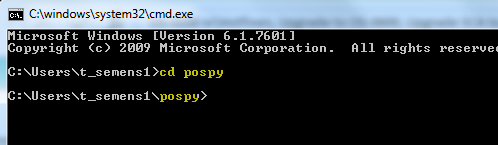


Place your extracted *pospy* folder inside your user directory (it will save you from endless navigation):



**3.2 Navigate to pospy folder in cmd:**

Type ***cd pospy*** to go inside the folder:

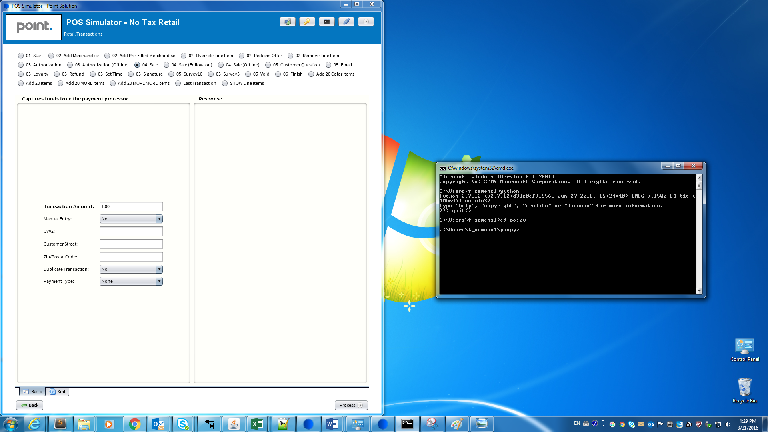


Great! You’re in the right working directory! You can now launch the programs.

**Running programs**

In order to run your programs, you need:

1. Pair POS-simulator to your terminal and start a session. *(For error-free results please use the pos-simulator provided along with pospy. If you for any reason need to use different version of POS-simulator, and pospy programs don’t seem to work properly, please refer to Debugging section)*
2. While POS-simulator window is active, press “Picture of the Windows logo key + ←" (Windows logo key+left arrow) shortcut. This will allow POS simulator to take exactly half of the left screen of user's monitor.



1. Navigate to *pospy* folder in cmd (See section 3.2)
2. In cmd, type *python sale.py* to run *sales.py* program. Same goes for any other program – type ‘*python program\_name’*

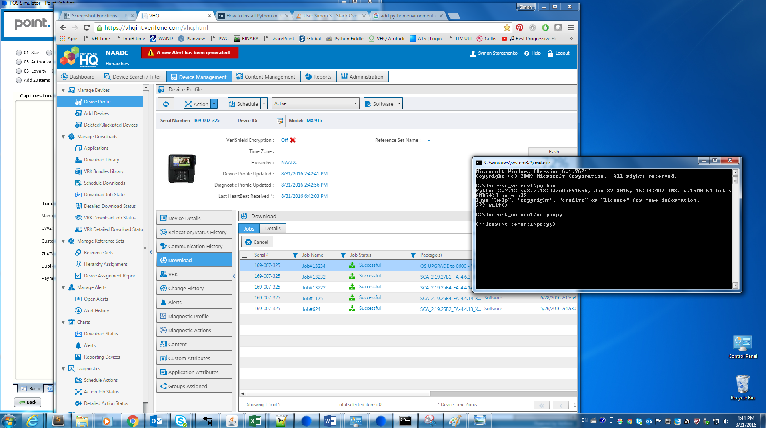
**Debugging:**

1. **POSPy not running**

One of the first things to look at is whether your POS-simulator obstructed by other programs/applications or not.

Make sure your POS-simulator is not FULLY obstructed by other programs/applications when you are to run your programs. POSPy simulates mouse and You need to allow at least a small visible part of POS-simulator on the uttermost left side of the screen:

Below example is acceptable:



1. **Finding Screen Location**

If your mouse clicks are off the target or you’ve decided to use right side of the screen to place POS-simulator, you can easily overcome this obstacle by figuring out your mouse/cursor location on the screen and overwriting variables inside each program with new values.