

Run_target-2: A Python program for automating SkyX imaging

Ken Sturrock

March 24, 2023

This is a rewrite of my original bash-based automation system. It is more streamlined and simplified compared to the original Bash version. The side effect of this streamlining is that the project does not contain all of the oddball features found in the Bash version. On the good side, though, the nature of the Python code will make it considerably easier for the end-user to modify the code if they so desire. Although the program was intended to run on Macintosh, Raspberry Pi and Linux platforms, it also runs on Windows but does not clean up scratch files left behind from Image Links.

More recently it was updated with a rudimentary graphical user interface. Due to the nature of the program, however, if you are competent to install and start the program then you really don't need a GUI to walk you through the run process. Moreover, the GUI requires that your installation of Python3 contains the TKinter library. You start the GUI by invoking the program without any arguments. On some platforms, this works by double-clicking the program icon. On other platforms, not so much.

Note: In order for this program to work, you must activate the TCP Server under the SkyX Tools menu. You also need to ensure that "TCP Response Closes Socket" is set to "True" under Preferences -> Advanced.

The command line syntax for the program remains essentially the same. To run the program, simply type something like:

```
./run_target-2.py m51 5x300 5x300 5x300 5x300
```

Note that the initial `./` means to run the program from the local directory on a UNIX-style operating system. If you are running Windows, skip the initial `./`.

This will tell the program to image the galaxy M51 and take five images, each, of 300-second duration through the first four filters in your filter wheel. To skip a filter, insert a 0x0 in place of that filter. The program uses the same target syntax as the SkyX's find field but you must enclose the target in quotes if there is a space in the name. You can, similarly, specify a set of J2k coordinates:

```
./run_target-2 "13h 29m 52s, 47d 11m 44s" 5x300 5x300 5x300 5x300
```

As before, you can image multiple targets in a night by using semi-colons (ampersand "&" for windows) between separate commands for each target. If you need to abort a program run, you can usually do so by aborting the current operation (e.g. Take an Image, @Focus2, etc) with the SkyX's GUI abort button. This is generally cleaner and safer than typing "control-c".

As noted, if you type `./run_target-2` all by itself, it should launch the GUI which will ask you for the same information as the command line, but will more easily allow you to specify which focus routine (@F2 or @F3) that you want in addition to the guider exposure & delay.

The program will wait to begin a run until the sky is dark and the target is above 30 degrees (See below). The software will also handle guiding and dithering. By default, it will refocus using @Focus2 if the temperature has changed by about a half a degree (See below) or 45 minutes has elapsed since the previous focus. The `Run_target-2` program will flip the mount when the target crosses the meridian, although it may get confused if you have configured your mount to flip at a custom hour angle and not adjusted the appropriate variable (see below). The program also has a limited ability to detect and pause in the case of clouds. At the conclusion of the run, the automation software will disconnect your cameras, slew your mount towards the appropriate pole and turn off the tracking motor. If you have a defined park position, the program will attempt to use that. It will also stop imaging when it becomes light in the morning or the target has sunk below 35 degrees.

In order to accomplish this magic, your system must be: Polar & GoTo aligned, able to guide (if applicable), able to Image Link & use Closed Loop Slew, able to Focus with @Focus2 or @Focus3 and have any necessary filter offsets entered in the filter name table. Your system must also have the OTA focal lengths noted in the camera settings along with your "observer's name".

There are two other special invocation modes:

1.) You can duplicate images in a given image set with an extra "x" option:

```
./run_target-2.py m51 5x300x2 5x300 5x300 5x300
```

Assuming that your filters are mapped LRGB, this will execute an LLRGB. While:

```
./run_target-2.py m51 5x300x2 5x300x3 5x300 5x300x4
```

will execute an LLRRRGBBBB < dither> LLRRRGBBBB pattern. Be advised that there will be no scheduled dithers between these "multiple" images within a set. It's a specialized function for people with specific needs.

2.) Another special mode is to use a second (remote) imaging camera. This requires that you have mounted a second OTA & camera on the mount, which is under the control of a second version of SkyX running on a (preferably) second computer or as a second instance on the same computer.

```
./run_target-2.py m51 5x300 5x300 5x300 5x300 -r 10.0.1.7:3040 5x240  
5x240 5x240 5x240
```

will take five 5-minute exposures through the first four filters on the main system and five four-minute exposures through the second camera under the control of a copy of SkyX listening to port 3040 on a machine with the IP address of 10.0.1.7.

Please ensure that the second imaging camera is configured with the appropriate temperature, auto-save and binning. The images will be synchronized between the two cameras along with focusing and dithers. This option is also available through the GUI if the command-line parameters sound confusing.

In addition to being connected to the second camera & focuser (if applicable) the second instance of SkyX must also be connected to the “Telescope Simulator” in order to ensure proper automatic imaging naming and focusing exposure selection for @Focus2. If you use @Focus2 then you must calibrate @Focus2 to work on a star of the same magnitude as the primary (local) camera & focuser.

If you wish to “go unguided” then set the variable for the guider’s exposure duration to zero at the top of the file (or don’t have a guide camera selected in SkyX).

If you don’t want run_target-2 to automatically focus then you can either unselect a focuser or set the focuser method variable at the top of the program file to zero.

In order to run the program, you must have Python 3 (not just Python 2) installed on your system. Python 3 is included out-of-the-box as part of Raspbian, most versions of Linux, as well as the Windows 10 Subsystem for Linux. If you are running a Macintosh, you’ll probably need to install Python 3 yourself. Please download the installer here:

<https://www.python.org/ftp/python/3.6.4/python-3.6.4-macosx10.6.pkg>

or select a newer version. There is also a Windows-native (non-WSL) version of Python3 as well.

Variables to set (appropriate section is found near the top of the run_target-2 file):

altLimit – The altitude at which a target is considered OK for imaging. The default of 30 degrees is reasonable due to atmospheric thickness, but there are always some who are greedy.

defaultFilter – Sets the default filter for use in focusing and Closed Loop Slew. Luminance is a good choice. If you don’t have a filter wheel selected, it is ignored.

guiderDelay – The default delay for the guider. If you don’t have a guider, it is ignored.

guiderExposure – the default guide exposure, which will get adjusted if you have an assigned calibration library. If you do not have a guide camera selected, it will be ignored. Moreover, if you wish to be trendy and go guiderless, set this to “0” (zero)

focusStyle – Set as “Two” or “Three” for @Focus2 or @Focus3 respectively. If you don’t have a guider selected, it will be ignored (You can also specify “Zero” if you want it to ignore a selected focuser).

tempChangeLimit – What is the temperature change required to trigger a refocus. The default of 0.5 degrees is a good value for my steeply curved Asian-sourced Stellarvue Quad. Your rig may need less frequent focusing.

flipAngle – This is the hour angle for which the program tries to flip. If you have a mount with a custom flip angle (such as a Paramount) then make sure that this value agrees with the value set in the mount.

While the older Bash version of the automation program will remain available on GitHub, I will no longer update or change it. I will also not create customized versions of this program but will happily try to help you modify it yourself. Thanks to everyone who helped me write & test the original Bash version of this program. I also want to thank Anat Ruangrassamee, Colin McGill and Rick McAlister for encouraging me to learn new tricks with Python as well as everyone at Software Bisque for their continued development of SkyX and their patience with me when I pester them with questions and unsolicited advice.

Clear Skies.

-Ken