

## Short tutorial on Siril using Sirilic

“Siril” is a reverse acronym for “Iris for Linux”, though it is now also available on Windows and MacOS. I have used it exclusively on Windows and this short tutorial will focus on that version.

Sirilic (Siril Image Converter) is a complementary tool developed by Pascal Burlot that provides a simplified front-end to Siril. It enables you to easily define your image layers, file sets, and processing requirements, and it then copies the files into a work area, generates the corresponding command scripts and passes them to Siril which runs in the background.

Siril does not need to be launched separately when using Sirilic and this tutorial concentrates on using Sirilic. Siril can naturally be used without the front-end – either interactively or by defining scripts and then running them.

Siril can be downloaded from <https://www.siril.org/download/> and sirilic from <https://gitlab.com/free-astro/sirilic/-/releases> using password ‘sirilic’.

Before going into more detail on usage, here is a summary of what Sirilic can perform using Siril (some optional):

- Where necessary, convert source files to FITS format

- Pre-process calibration sub-frames to create masters

- Pre-process light frames with the calibration frames

- Apply cosmetic correction to remove hot/cold pixels

- Register the light frames so they are aligned

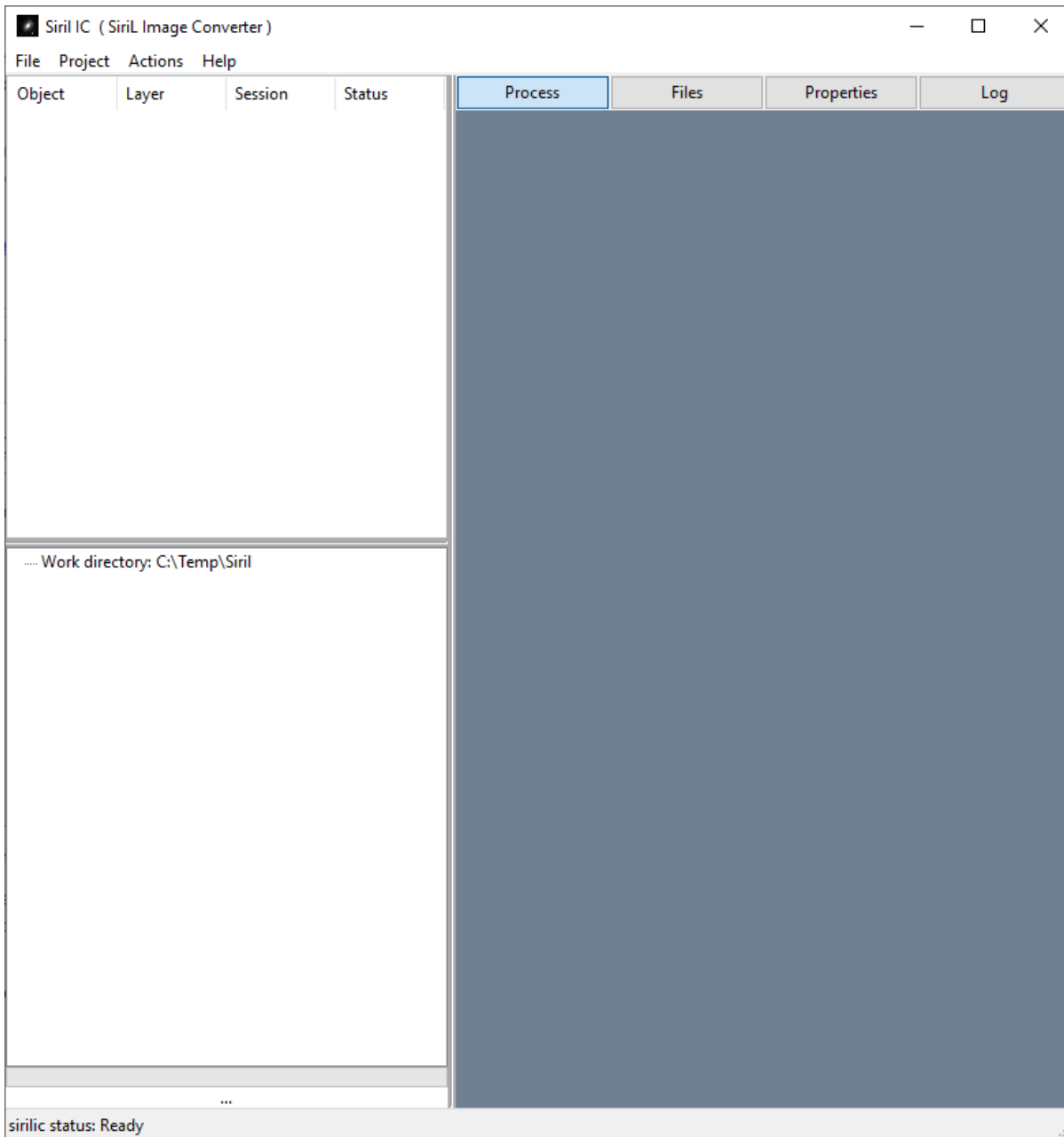
- Stack the light frames according to chosen parameters\*

- Where appropriate, register stacked layers so they are aligned

The result is one or more FITS files that can be Opened or Composed in StarTools.

\*One option is “Average Stacking With Rejection” which uses Winsorized Sigma Clipping where an initial step replaces spurious outlier pixel values with best guess local values and then produces a robust mean and a more accurate sigma before the usual sigma clipping is performed.

On running Sirilic you are presented with the following screen.



The left-hand area will hold the image structure you choose (e.g. filter layers, sessions, etc)

The right-hand area is used in 4 different modes, covered in more detail later:

Process: gives a useful visual view of the processing you have chosen

Files: here you drag and drop your sub-frames

Properties: here you define how you want them to be processed

Log: provides the Sirilic and Siril log as the scripts are run

On the first run (only) you need to define a few preferences (File, Preferences):

Preferences

Work directory: C:\Temp\Siril

SiriL executable: C:\Program Files\SiriL\bin\siril.exe

Clean the object directory before to launch the script

use symbolic links ( replace the copy)

return to the workdir at the script end

0 CPU 0 MEM

Debug ( save stdout/stderr in file)

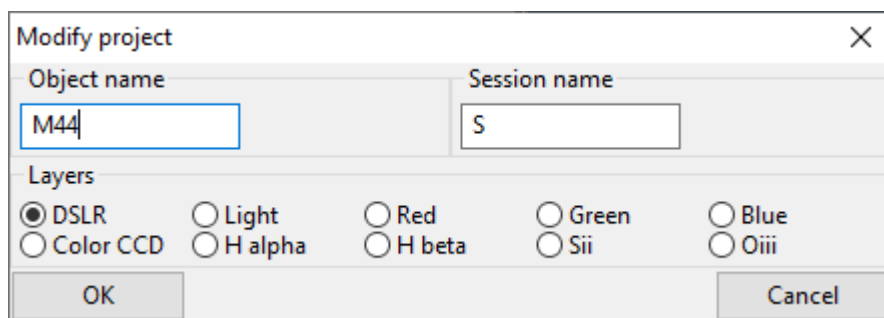
OK Cancel

Clean the object directory... will remove files from previous sessions before running.

Use symbolic links... allows you to reference image sets in their source folder rather than copying them to a work area. However, this requires Windows to be run in Developer mode and I personally prefer not to do so.

CPU and MEM set to 0 will use all available resources.

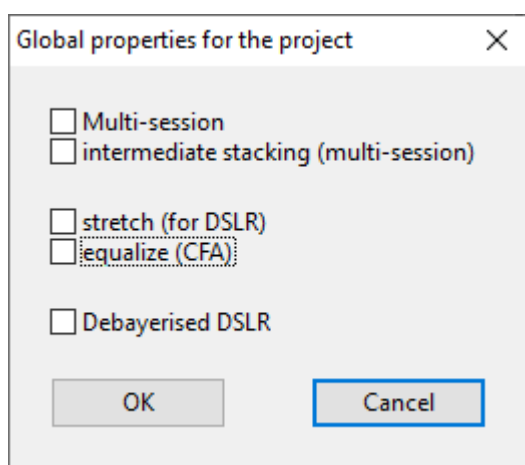
Each processing of a target is referred to as a project, and the first step is File, New to bring up the following dialog box:



Here you define the target name (used for folder naming), session name (also), and the layers you wish to process. Color CCD is One Shot Colour. "Light" is for luminosity in LRGB.

I'll use DSLR in the following example but add a note about using filters afterwards.

Once defined you can then set the project properties using Project, Edit Properties as follows:



For a single session I leave all these unchecked, but they can have the following effect:

The multi-session mode is to add two or more sessions. First, Sirilic launches a script to preprocess each session (offset, dark, ...) but does not align and stack. Then, Sirilic groups the files for each session into one folder. At the end, it launches a script that aligns and stacks all the files. The options are as follows:

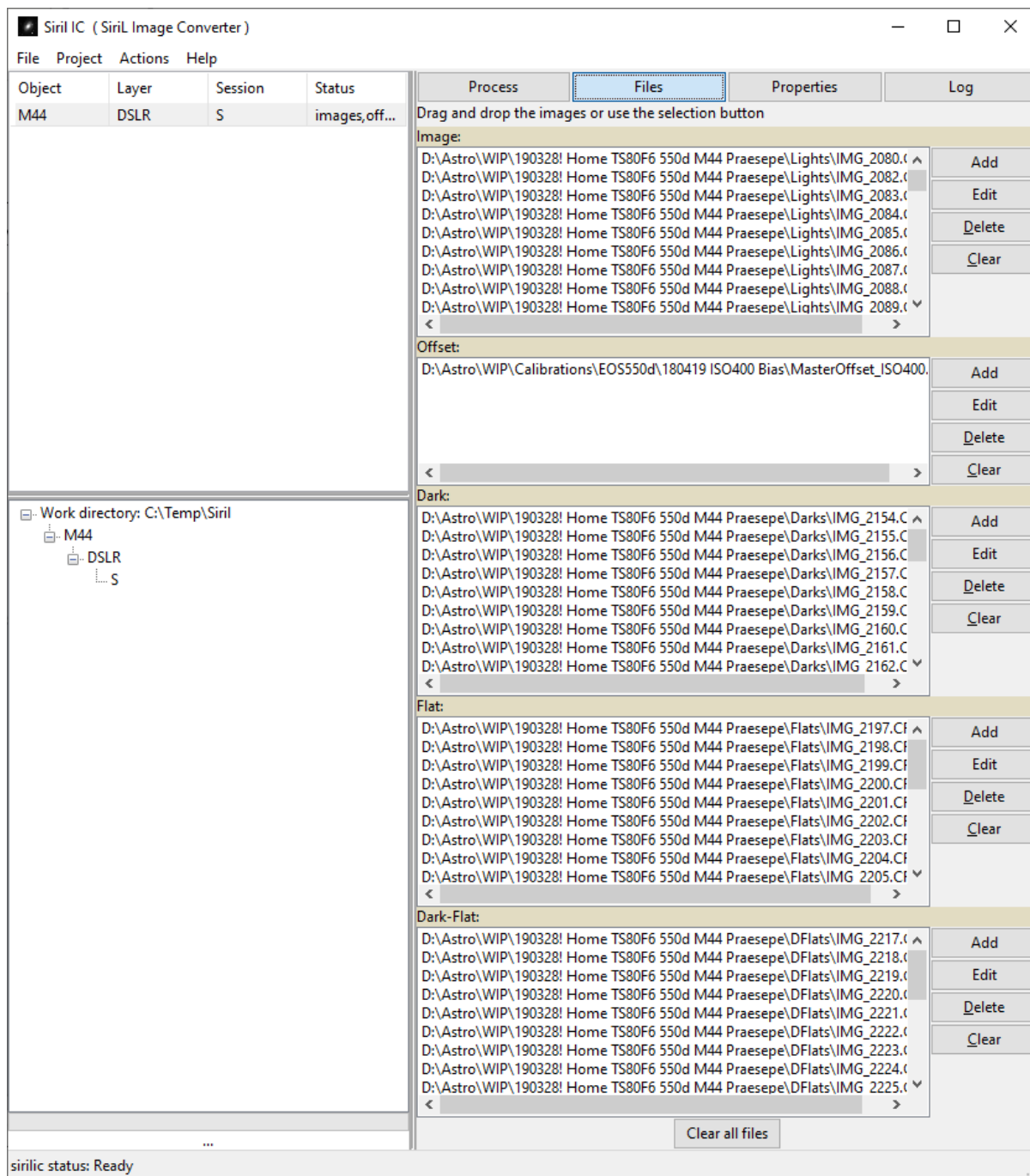
"Multi-session"	"intermediate stacking"	Description
checked	unchecked	The layer with multiple sessions are grouped, aligned and stacked but no stacking on the sessions of this layer.
checked	checked	In the layer with multiple sessions, Sirilic aligns, stacks each session of this layer, and saves intermediate results. Then it groups the sessions into one and then aligns and stacks.
unchecked	unchecked, checked	In the layer with multiple sessions, Sirilic aligns, stacks each session of this layer, and saves the results. It does not group sessions.

Stretch for DSLR... stretches the dynamic range to 16 bits.

Equalize... equalizes the mean intensity of RGB layers for the flats master (CFA = Bayer matrix)

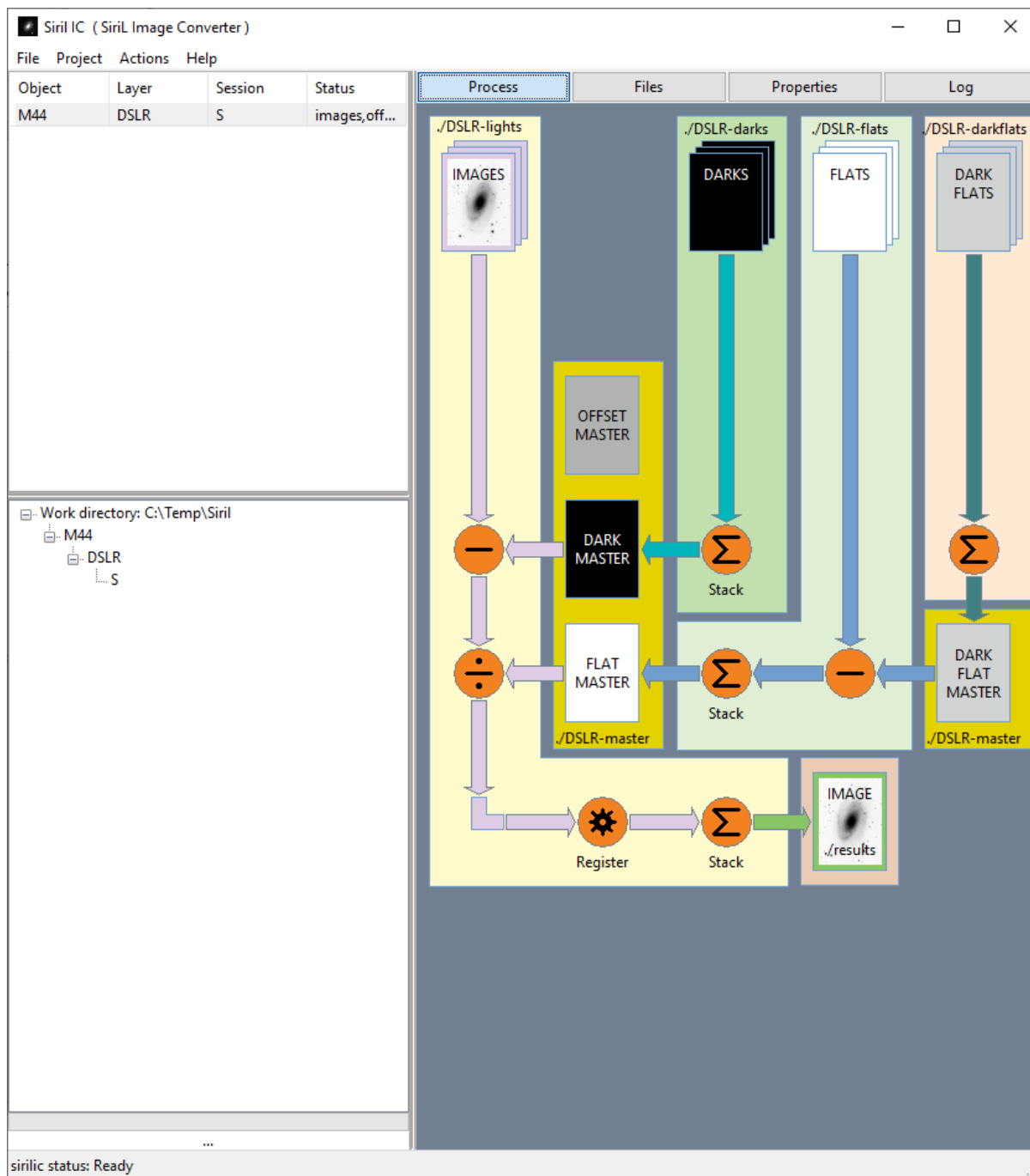
Additional objects or layers can be added or edited via the Project menu.

You have now defined the initial structure and if you click on Files in the right-hand pane and drag and drop your sub-frames or masters (single files are assumed to be masters) you get the following:

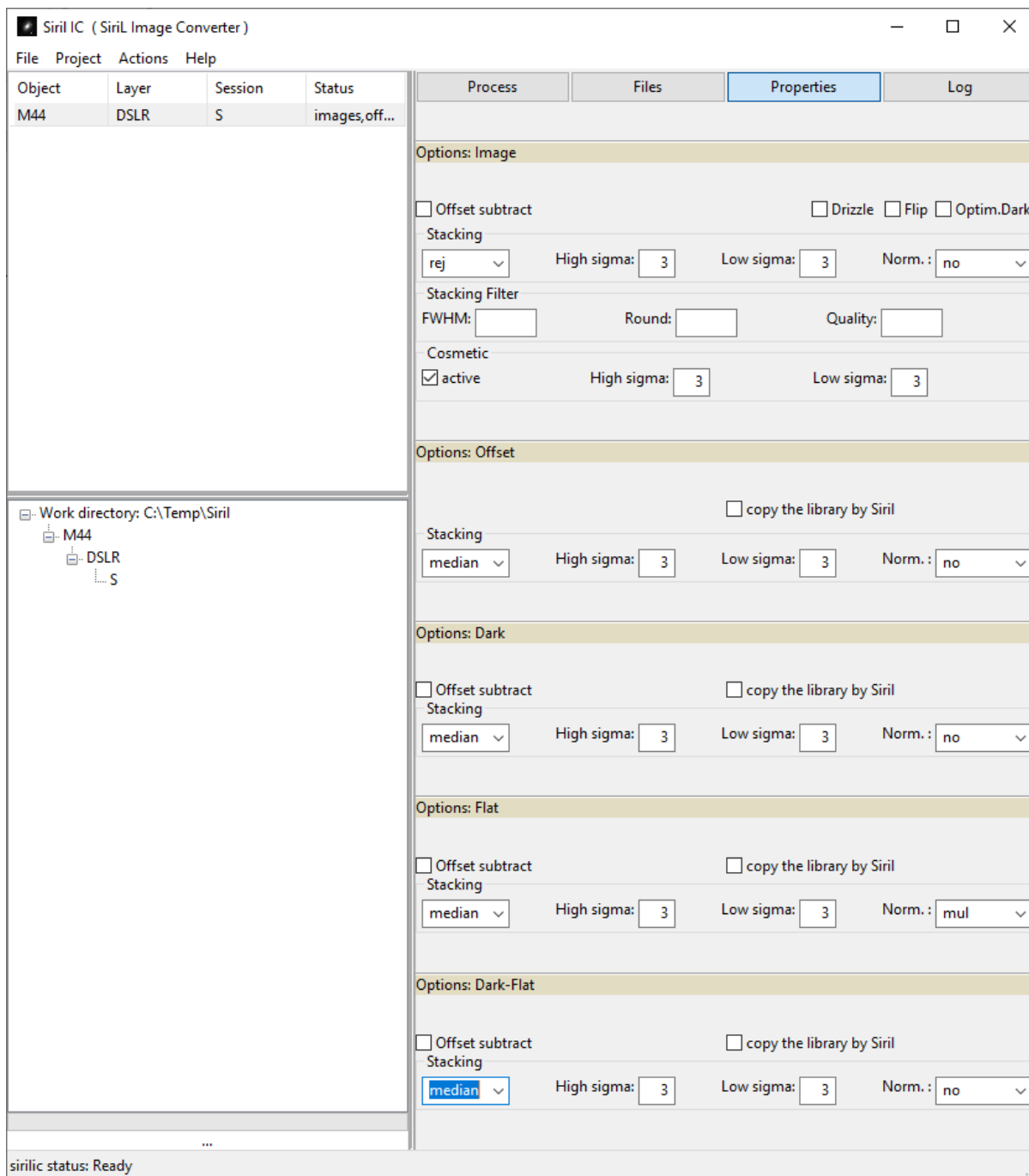


N.B. If you have multiple layers/sessions you select the desired row in the top left box and populate the files for each.

If you now select the Process tab you can see a useful visualisation of the steps you have chosen:



The next step is to further define each step of the process using the Properties tab:



This may look a little daunting at first sight but is quite straightforward and repetitive once familiar. Most of it is self-explanatory but here are a few notes:

Offset subtract depends on how you built the master-dark - if you have removed the offset signal from your master-dark then you also need to subtract the offset from the lights; if your master-dark still includes the offset signal then you should not do so. Similarly, with the Flats / Dark-Flats.

Stacking choices are average stacking with Rej(ection) which uses Winsorized sigma clipping (other options are available in Siril itself), Sum, Max, Min, Med(ian); I generally use Rej for lights where I have enough (>6), and Median for the rest. With Rej you can defined the high and low sigma levels which I tend to leave at 3. Normalisation can be enabled (probably not wise for use with StarTools) with choices for additive and multiplicative with optional scaling.

Copy the library... normally unchecked but should be checked if the master library does not exist at launch of the script but the master created in the previous layer should be used for this layer.

You can choose quality thresholds (values or percentages) and ignore frames outside these.

You can choose Cosmetic rejection sigma levels for hot and cold pixels.

N.B I found that needed to check Flip for the lights from my QHY OSC camera in order to read the Bayer matrix in the correct order.

That's it – you're good to go. There are two steps under Actions – 1) Build folder structure and copy/link to files; 2) Run script(s).

In the example above with 60 lights, and 20 each of darks, flats and dark flats this went through in less than 13 minutes on my i7 desktop with SSD and 16GB memory. This included converting all 120 RAW files of c. 20Mb each to FITS – if you are dealing with FITS files in the first place no conversion is necessary.

With another setup I take fewer longer exposure narrowband images on a mono camera. I select Ha and Oiii layers, drag and drop the relevant files for each, and it produces stacks for each filter and then aligns them with each other. These can then be loaded into StarTools using the Compose function.

I hope you have found this short tutorial useful; please contact me with any suggestions on how to improve it.