RTS2

We are attempting to automate the 61“ using RTS2. This wiki page should give the basics of how to run RTS2

RTS2 source

The version of RTS2 on kuiper is a fork from the main rts2 version. it can be cloned from https://github.com/srswinde/rts2. The only real differences from the main rts2 repo is a few updates to the Azcam driver and some scripts for convenience.

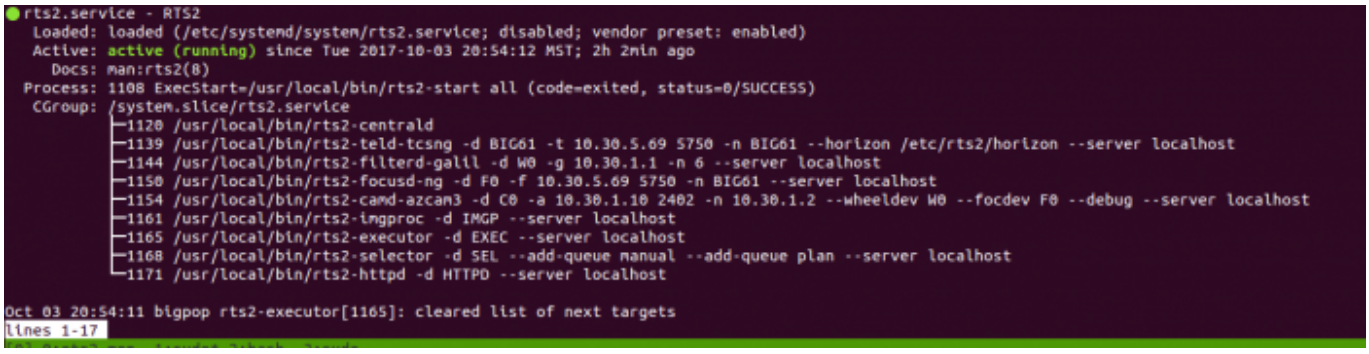
Starting RTS2

Currently RTS2 runs on the bigpop computer. You will need sudo privileges on bigpop to run RTS2. To run it log into bigpop and type

sudo service rts2 start

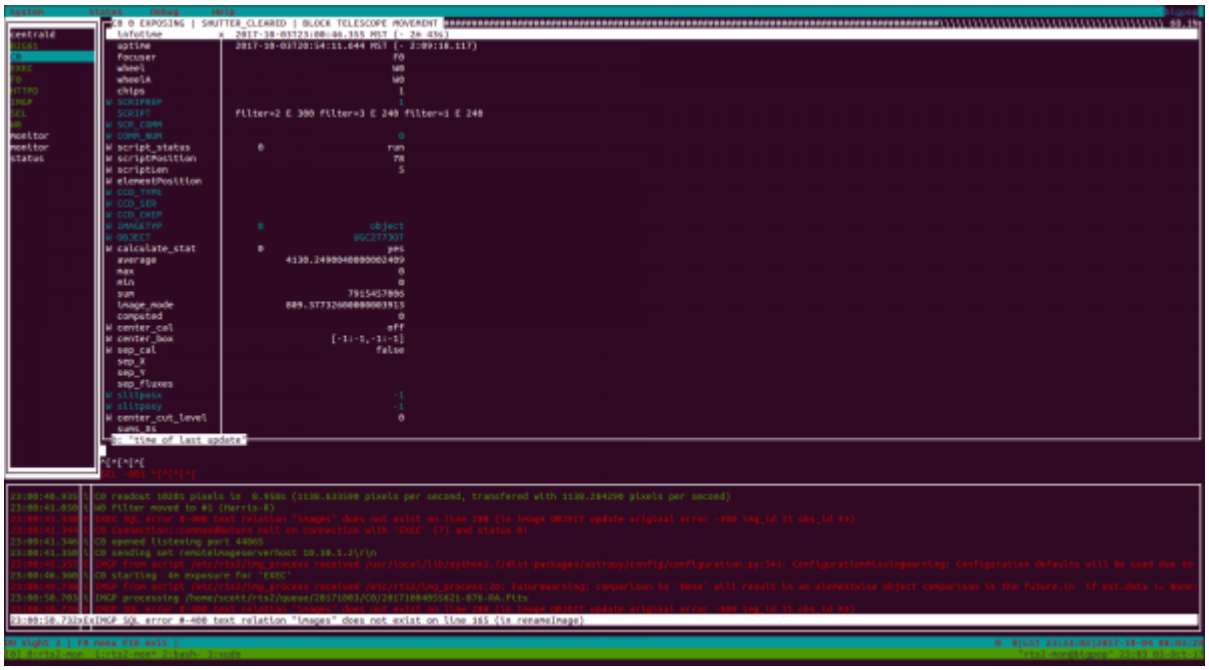
and then sudo service rts2 status

to make sure all the RTS2 processes started. It should look something like this:



Using rts2-mon

The current user interface is a ncurses based tool called rts2-mon. To start it simply type rts2-mon in a shell. It will look



It is separated into three sections and a menubar. To access the menubar use F9. The most important menu items are System → exit to exit rts2-mon and States→on, states→off. RTS2 will not automatically start observing unless you set its state to on.

The three main sections of the rts2-mon are the driver list on the left side the variable list on the right and the logger messages on the bottom. To access each or the main sections use the tab key. Whichever section is active will have highlighted borders.

Driver List

The driver list on the left side gives you a list of the drivers and services currently run by RTS2. You can scroll through them using the up and down arrows.

Variable List

As you scroll through the driver list the variable list on the right side will update. Each variable listed is associated with the highlighted driver or service. Most of the variables are read only but the variables that have a “W” next to the name are editable.

Logger

The logger at the bottom is the real time log of each diver or service.

RTS2 Targets

RTS2 keeps all targets and lots of other information in a postgres database called stars. You should

not have to interact directly with the database for target manipulation. There are several handy tools that will do this for you.

rts2-targetlist

rts2-targetlist command will list all the available targets in the database along with their RA an Dec and weather the object is rising, setting, or transiting. The output will look something like this:

1009 O 02:42:42.960 +61:37:58.80 -09:49 7.19 +07 43 02.73 195 01 25.11 rising NGC 1027 1010 O 01:57:41.040 +37:47:06.00 -09:04 nan -08 43 07.81 213 43 46.11 rising NGC 752 1011 O 02:42:05.040 +42:45:43.20 -09:48 nan -09 00 12.79 203 48 13.53 rising M34 1012 O 20:17:11.320 +58:12:08.00 -03:23 1.36 +47 24 31.91 217 10 02.36 rising SN2017gas 1013 O 20:34:44.240 +60:11:35.90 -03:41 1.42 +44 45 43.40 215 07 40.40 rising SN2017eaw 1014 O 00:03:49.200 +07:28:00.10 -07:10 nan -10 31 39.32 254 03 46.18 rising SN2017gww 1015 O 00:43:33.090 +41:12:10.40 -07:49 nan +03 26 29.63 221 57 57.67 rising M31AfAnd 1016 O 03:32:07.240 +47:47:39.60 -10:38 nan -07 43 54.99 193 39 13.74 rising UGC2773OT 1017 O 02:35:30.150 -09:21:14.90 -09:41 nan -50 39 14.34 241 54 29.52 rising SN2017gmr 1018 O 23:31:53.600 -05:00:43.40 -06:38 nan -10 45 15.31 269 07 30.41 rising SN2017grn

where the number on the far left is the RTS2 target id and the far right is a name for the target supplied by the user.

newtarget.py

To add new targets to the database you can use newtargets.py currently located in /home/scott/git-clones/rts2/scripts on bigpop. There are two ways to add targets with this script. You can use the –create flag or the –search flag

--search

The –search flag is used like this:

newtarget.py –search “M35”

This command will look for m35 in the Simbad database and if it finds it, will add it to the RTS2 target list

--create

You can use the –create option like:

newtarget.py –create 02:35:30.150 ” -09:21:14.90“ SN2017gmr

this will create the target SN2017gmr at 02:35:30.150 -09:21:14.90 . Notice if the declination is negative you have to put it in quotes and add a whitespace before the negative so the script doesn't assume this is a new command line option.

Simple Queues

RTS2 has two basic observing modes a simple queue and a more complicated selector that uses a variety of information to intelligently select targets.

In a simple queue, the users create a list of targets using the rts2-queue command.