HECTOR
Galaxy Survey

Data Reduction with 2dfdr and the Hector Software

If you find any issues on the hector reduction package or the use of data reduction machine at AAT, please contact Sree Oh (sree.oh@anu.edu.au), Madusha Gunawardhana (madusha.gunawardhana@sydney.edu.au), and cc Julia Bryant (julia.bryant@sydney.edu.au).

Data reduction machine at AAT

Find the machine labelled 'DATA REDUCTION SYSTEM' in the control room. The hector python package runs on aatlxe (preferred) or aatlxh. Log into either machine with the following username and password:

```
Username: hector
Password: RobotisGreat!
```

Daytime Essential: check 2dfdr

The first thing you should check is 2dfDR. Open a shell and try the following:

```
(base) hector@aatlxe:~$ drcontrol
```

It will pop up a window with Saturn. If not, please contact Sree Oh (sree.oh@anu.edu.au) and Madusha Gunawardhana (madusha.gunawardhana@sydney.edu.au), immediately. If you do not get a response in time, you may install 2dfDR following the next step.

Not essential: Installing 2dfDR

Note: skip this step unless it is requested by the DR team

This section explains how to update the 2dfdr data deduction software on the Data Reduction PC in the AAT control room. Note that this process is a slow one, taking of upwards of 30 minutes to configure, make and install the new software. Try and do this at the start of a run!

The 2dfdr software lives in /home/hector/software/2dfdr. Rename this folder with the following format and move it to the 'previous_2dfdr_versions' folder:

```
(base) hector@aatlxe:~$ mv ~/software/2dfdr ~/software/previous_2dfdr_versions/
2dfdr_YYMMDD
# Note: YYMMDD should be the date it has been moved
(base) hector@aatlxe:~$ drcontrol
```

Try running drcontrol from the terminal. This should give an error since the 2dfdr folder has been moved.

Now switch to aatlxh and install 2dfdr.

```
(base) hector@aatlxe:~$ ssh aatlxh
RobotisGreat!
(base) hector@aatlxh:~$ cd ~/software
(base) hector@aatlxh:~/software$ git clone https://dev.aao.org.au/rds/2dfdr/
2dfdr
(base) hector@aatlxh:~/software$ cd 2dfdr
(base) hector@aatlxh:~/software/2dfdr$ git checkout HECTOR2
(base) hector@aatlxh:~/software/2dfdr$ make all
(base) hector@aatlxh:~/software/2dfdr$ make install
(base) hector@aatlxh:~/software/2dfdr$ exit
(base) hector@aatlxe:~/$ drcontrol
```

These will all take ~10 minutes to complete. Now try running drcontrol again to make sure that the new installation completed successfully.

Not essential: Installing required packages for the Hector pipeline

Note: skip this step unless it is requested by the DR team

The Hector package requires numpy, matplotlib, scipy, atropy, ppxf, photutils, bottleneck, ipython

Installing new python packages on the Data Reduction PC is possible ***but is not recommended unless you know what you're doing and have a good reason for doing so***. If you *really* need to install a new package, you can use mamba (a faster version of conda). E.g:

(base) hector@aatlxe:~$ mamba install photutils -c conda-forge

Some niche astronomy packages aren't available on the conda channels (e.g. ppxf). For those, use pip. ***Try mamba first! Only use pip if a package is not available using mamba***. Mixing two package managers (like pip and mamba) causes all kinds of headaches.

At the time of writing (December 2021) installing a python package on the DR PC takes ~10-20 minutes to complete.

Daytime Essential: check the Hector package

This package is used for many purposes while observing, the most relevant being:

- Tramline, focus checking using flat frames
- Quick-look display to simultaneously check centring and rotation etc.
- Reduce frames
- Disable unnecessary frames
- Seeing and transmission measurements for qc check

These functions are described in more detail, with examples, in the rest of this document.

The hector package is version controlled using the online github repository:
https://github.com/Hector-Galaxy-Survey/hector

The hector package has been installed on aatlxe (or aatlxh), and please update that to the latest version:

```
$ cd /home/hector/software/hector/
$ git pull origin master
```

NOTE: Do not modify anything in /home/hector/software/. Restore the change when you accidentally modify any files in /home/hector/software/hector/:

```
$ git status                    #this will show which file has been modified
$ git restore modified_file     #e.g., git restore manager.py
```

If you really messed up the package, you should clone the whole package again:

```
$ rm -rf /home/hector/software/hector
$ cd /home/hector/software/
$ git clone https://github.com/Hector-Galaxy-Survey/hector.git
```

Go to the directory for reductions and start up ipython, load the package, and create an instance of the manager:

```
$ cd /data/hector/reduction/     # ipython should be started in this directory
$ cleanup -k                          # this cleans up crashed previous processes
$ ipython --pylab=auto

# In the ipython shell:
In [1]: import hector             # this imports the *hector* package
In [2]: mngr=hector.manager.Manager("startdate_enddate", fast=True, n_cpu=20)
                                  # start and end dates of the run (e.g. 220817_220904)
                                # start and end dates should include both A and B runs
```

```
                                       # n_cpu=20 allows parallel processing
```

If this gives an error you must troubleshoot it immediately. Contact Sree Oh (sreemarie@gmail.com) and Madusha Gunawardhana (madusha.gunawardhana@sydney.edu.au) and report error messages that you have found.
Ignore the following warning message for now:
/home/hector/software/hector/manager.py:106: UserWarning: molecfit not found. Disabling improved telluric subtraction
  warnings.warn('molecfit not found. Disabling improved telluric subtraction')

Load the observed frames to the data reduction machine

After you get any frames, you start to import the frames to the reduction machine using the manager

```
In [4]: mngr.import_aat()                    # load the data of the day only
or
In [4]: mngr.import_aat(date="YYMMDD")       # load data from a previous date
or
In [4]: mngr.import_dir("path/to/raw/data")
```

At AAT, import_aat() will copy the new observed data of *the day* from the instrument machine to the data reduction machine. If you specify the date, you can copy the data from a previous date. **Every afternoon, observers first run import_aat with the date stamp of the previous night to load day time calibrations.** Now you can find the data have been imported to /data/hector/reduction/startdate_enddate/raw/. You are ready to reduce the data.

```
# Once reduction starts, the following directories will be automatically created:
/data/hector/reduction/
     startdate_enddate/   # Start and end dates of the run
                  raw/    # (Modified) copies of the raw telescope data
              reduced/    # Manager reduced data
                cubes/    # Only appears if you create cubes at the telescope
```

Disable and enable files

Before you start the reduction, it is **ESSENTIAL** to disable files that are not necessary to reduce, such as frames for focusing, acquisition images, saturated (twilight, flat) frames, standard star frames when the star is not well placed in the bundle, and any other faulty frames. If not, the DR team may have to spend days for detecting unnecessary and/or problematic frames as identifying them at a later time is often not obvious. We thank observers for their efforts in disabling these files.

In an ipython shell, you can disable files to prevent them from being used in any later reductions:

```
In [4]: mngr.disable_files(['06mar10003', '06mar40005'])
```

Note that even if you only have one file you want to disable, you still need the square brackets.

If you have many data to disable, there is a way you can disable bulk frames at a time. In a shell, open the following text file then put the file name to be disabled. Then, run disable_files() in an ipython shell.

```
(base) hector@aatlxe:~$ vi /data/hector/reduction/startdate_enddate/disable.txt
22jul30002
22jul40002

In [4]: mngr.disable_files()         #This is to disable files listed in disable.txt
```

Another way to disable lots of files at a time is using the files generator. If you are familiar enough with the files generator, then you may try. If not, do not try this.

```
In [4]: mngr.disable_files(mngr.files(date='230306', plate_id='A3336_002', name='main', exposure_str='300'))
```

The above example will bulk disable acquisition object (name=main) images of A3336_002 field with exposure time of 300 taken on the 6th March 2023.

This allows the keywords as well as:

ccd	'ccd_3'
ndf_class	'MFFFF'
reduced	False
tlm_created	False
flux_calibrated	True
telluric_corrected	True
name	'LTT2179'

For example, specifying the first three of these options as given would disable all fibre flat fields that had not yet been reduced and had not yet had tramline maps created. Specifying the last three would disable all observations of LTT2179 that had already been flux calibrated and telluric corrected. Complicated? Then, please don't try.

Have you accidently disabled files? Don't panic. You can still re-enable them for reduction:

```
In [4]: mngr.enable_files(['06mar10003', '06mar20003', '06mar10047'])
```

Enabling bulk frames at a time is not possible. Sorry. Also, don't forget to remove the file from disable.txt if it is listed in the file.

Please double check the disabled files when the observing of the day is getting to the end.

```
In [4]: exit
$ cd /data/hector/reduction/
$ ipython
In [1]: import hector
In [2]: mngr=hector.manager.Manager("startdate_enddate", fast=True, n_cpu=20)
$ vi /data/hector/reduction/startdate_enddate/filelist.txt
```

The 6th column of the above file now marks the latest disabled files. Please check your disabled files. This is exceedingly important. If it is mislabelled, we will lose the data.

Summary: data reduction at AAT

The DR team request observers to reduce the data at the telescope. So, we can immediately check the reduced files. The followings are the essential manager tasks to reduce data. This is the summary, and the detail of each step is described later. Please read the following details at least once!

```
$ cd /data/hector/reduction/
$ cleanup -k
$ ipython --pylab=auto
 In [1]: import hector                       #this imports the *hector* package
 In [2]: mngr=hector.manager.Manager("startdate_enddate", fast=True, n_cpu=20)
      In [2]: mngr.remove_directory_locks() #removes locks from a crashed manager

Essential reduction steps:
 In [3]: mngr.make_tlm()                     #tramline mapping, generating tlm.fits
Quick-look plots to check centring           #it can go parallel with reduce_arc()
 In [4]: mngr.reduce_arc()                   #reduce arc frame, generating red.fits
 In [5]: mngr.reduce_fflat()                  #reduce flat frame, generating ex,
red.fits
 In [6]: mngr.check_tramline()               #check tramline and spectral focus

After running the object dither script (so when you are not busy doing the above)
 In [7]: mngr.reduce_sky()                   #reduce twilight sky frames
 In [8]: mngr.reduce_object()                #reduce object frames
 In [9]: mngr.derive_transfer_function()      #derive transfer function from primary
star
 In [10]: mngr.combine_transfer_function()   #combine transfer functions
 In [11]: mngr.flux_calibrate()            #apply primary flux calibration
 In [12]: mngr.telluric_correct()          #apply telluric correction
 In [13]: mngr.qc_summary()                #to check FWHM of each frame
 In [14]: mngr.fluxcal_secondary()         #apply secondary flux calibration
 In [15]: mngr.scale_frames()              #scale dithered frames
 In [16]: mngr.qc_summary()                #to check FWHM & transmission of each frame

At the end of each day
Double check disabled frames of the day   #check the ‘Disable and enable files’
section
Upload the raw data                       #check the ‘upload raw data ...’ section
      $ rsync -avhPi  -e 'ssh -p 2232' /data/hector/reduction/YYMMDD_YYMMDD/raw
      hector@uploads.datacentral.org.au:/data/Observing/20YYMMDD_20YYMMDD/
YMMDD_YYMMDD/
      $ L3tMe!nPle@se

The below is not essential. If you want to explore cubes, then you may go further.
 In [19]: mngr.measure_offsets()                  #estimate offsets between dithered
frames
 In [20]: mngr.cube()                      #cubing
```

Observers frequently execute the above tasks *in order* when they have time. Do not postpone this until the last minute. If you find any error messages (not warnings), please copy and paste it to /data/hector/reduction/startdate_enddate/reduction_error_YYMMDD.txt.

Tips: reduce subsamples and re-reduce frames

It will happen that, throughout a run, you will need to re-reduce data already reduced. To avoid re-doing it for the entire run, you can force the manager to work on specific files. In most DR commands, keywords can be used to restrict the files that get reduced, e.g.

```
In [4]: mngr.reduce_arc(ccd='ccd_1', date='230306',
plate_id='G12_T001',overwrite=True)
```

The above will only re-reduce arc frames for ccd_1 that were taken on the 6th March 2023 for the field (Tile) ID G12_T001. Allowed keywords and examples are:

date	'230424'
plate_id	‘G12_T001’
ccd	'ccd_3'
exposure_str	'10'
min_exposure	5.0
max_exposure	20.0
reduced_dir	'230417_230430/reduced/230424/G12_T001/G12_T001_F0/calibrators/ccd_3'
overwrite	True

Note that ‘`overwrite=True`’ makes the manager re-reduces frames and overwrites existing reduced frames.

Essential: check tramline mapping and spectral focusing with check_tramline()

On the first night, once the dome is dark and after the spectrograph has been focused, it is ESSENTIAL to take a dome flat (both upper (old) or lower (new) screen are fine) and an arc frame and check the tramline mapping of the file. We are finding frequent failures of tramline mapping, and therefore, it is essential to check the tramline for EVERY SINGLE DOME FLAT in real time. check_tramline() detects tramline failures using the following criteria:

1. failed tlm maps may have active (S, P) fibres allocated where there is no signal
2. failed tlm maps may have inactive (N, U) fibres allocated where there is a signal
3. failed tlm maps sometimes show tlms not allocated in numerical order

Also, it automatically checks the spectral focus based on the contrast between the gap and signal.

You should learn how to use check_tramline() in the afternoon on the first day and keep running this function every time you take new dome flat during the night.

```
In [4]: mngr.import_aat()                  #import new files of *the day*
In [5]: mngr.make_tlm()                    #tramline mapping using 2dfdr
In [6]: mngr.reduce_arc()                  #reduce arc frame using 2dfdr
In [7]: mngr.reduce_fflat()                #reduce flat frame
In [8]: mngr.check_tramline()              #check tramline mapping
```

Once you run check_tramline(), it generates a text file /data/hector/reduction/startdate_enddate/tlm_failure.txt where you can find the list of flat field frames checked for tramline mapping. It also gives you directions that you should follow when it detects any failure in tramline. Specifically, the text file may direct you to check whether the frame is saturated or out of focus which can cause a catastrophic failure in tramline mapping. It may also send you to the top end to check the sky fibre position. It also guides you to visually check tramlines. It may also give you a warning of tramline cutoff which may require to adjust spectrograph mounting.

You may find examples from the following pdf file:

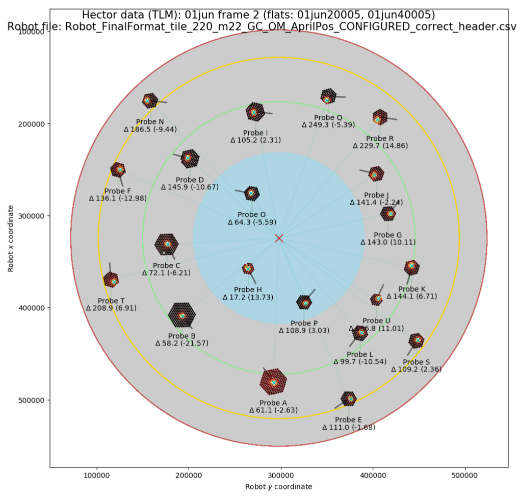
```
In the data reduction machine:
/home/hector/software/hector/observing/check_tramline_example.pdf
```

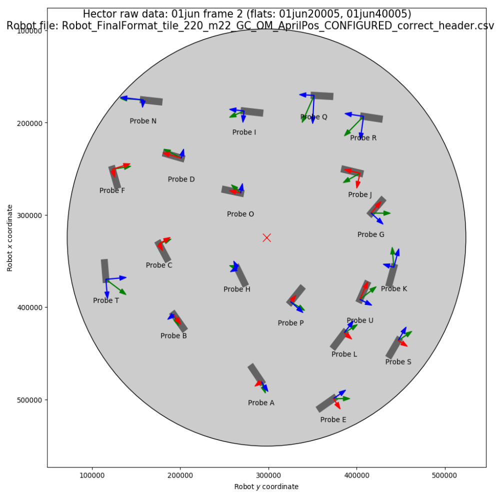
If you confirm any tramline failures that you could not fix, you immediately report this to Sree Oh (sreemario@gmail.com), Madusha Gunawardhana (madusha.gunawardhana@sydney.edu.au), and Scott Croom (scott.croom@sydney.edu.au) and send the raw file (e.g. 29jun40003.fits) and the text file (tlm_failure.txt).

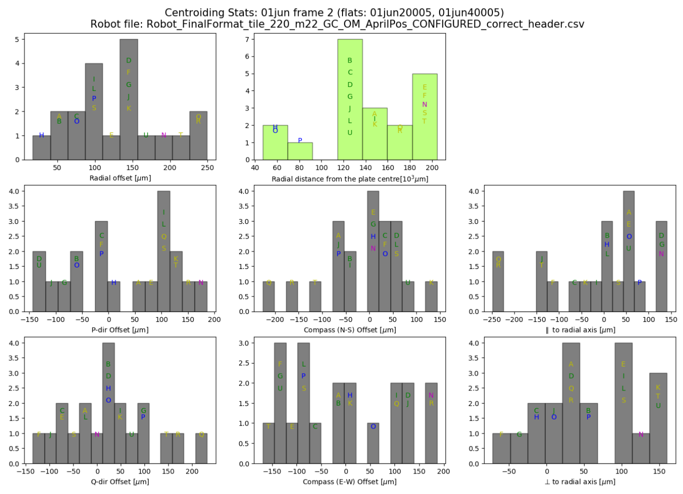
Essential: quick-look plots to check centring

Once the tramline is made (via mngr.make_tlm()) and an object frame is imported (via mngr.import_aat()), you are ready to run the quick-look tool. Note that the quick-look tool and mngr.reduce_arc() can go parallel, and observers do not need to wait until mngr.check_tramline() is done.

The quick-look tool generates the following three plots needed for checking centring.







Instructions on setting up and running the quick-look tool

The quick-look tool access the working data directory you create when you run the Hector manager on the first day of the observing run, i.e. a directory of the name "startdate_enddate" of the current observing run.

On the first day:
- open an ipython shell, and type:
run update_quick_look_config.py --observing_run_dates [the name of the working directory]

Note that while the quicklook tool is within the hector package, to run this you need to be in the /home/hector/software/hector/observing/Quick_Look_Tools/ folder. Its probably better to have a separate tab or terminal running this, separately from the pipeline reduction.

Example:
```
In [2]: run update_quick_look_config.py --observing_run_dates 220627_220706
```

This sets up the directory structure within the quick-look tool, and you only need to run this command once.

HECTOR
Galaxy Survey

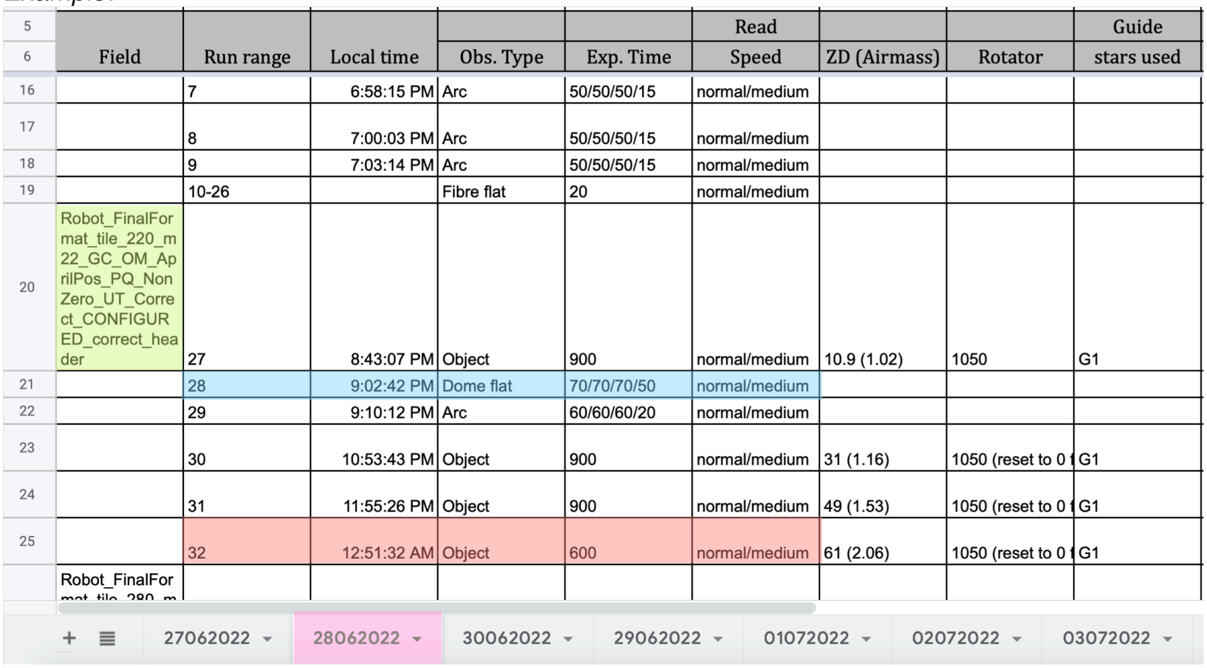
How to run the quick-look tool:
Once you set up the directory structure, you are ready run the quick-look script. To run this script, you need to provide the following:

1. Object frame run number
2. Partially reduced flat frame run number (it is the *tlm.fits file that the code looks for)
3. File prefix (e.g. 28jun, 14mar, etc.)
4. Robot file name (only need to provide the robot file name, but **both the robot and tile files should be in the raw data directory under the current run date**)

In Ipyton shell type:

```
run display_probes.py [object run number] [flat run number] --file_prefix  [file prefix] --robot_file_name [robot file name]
```

Example:



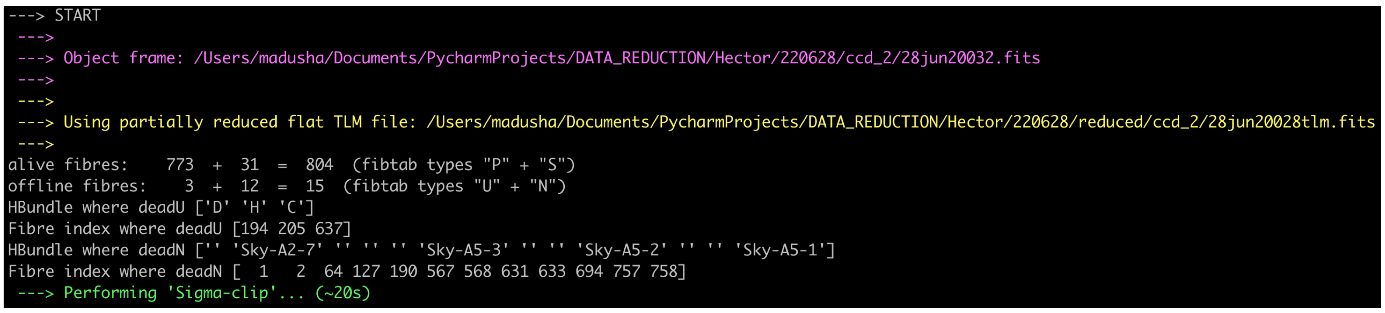
5						Read			Guide
6	Field	Run range	Local time	Obs. Type	Exp. Time	Speed	ZD (Airmass)	Rotator	stars used
16		7	6:58:15 PM	Arc	50/50/50/15	normal/medium			
17		8	7:00:03 PM	Arc	50/50/50/15	normal/medium			
18		9	7:03:14 PM	Arc	50/50/50/15	normal/medium			
19		10-26		Fibre flat	20	normal/medium			
20	Robot_FinalFormat_tile_220_m22_GC_OM_AprilPos_PQ_NonZero_UT_Correct_CONFIGURED_correct_header	27	8:43:07 PM	Object	900	normal/medium	10.9 (1.02)	1050	G1
21		28	9:02:42 PM	Dome flat	70/70/70/50	normal/medium			
22		29	9:10:12 PM	Arc	60/60/60/20	normal/medium			
23		30	10:53:43 PM	Object	900	normal/medium	31 (1.16)	1050 (reset to 0	G1
24		31	11:55:26 PM	Object	900	normal/medium	49 (1.53)	1050 (reset to 0	G1
25		32	12:51:32 AM	Object	600	normal/medium	61 (2.06)	1050 (reset to 0	G1
	Robot_FinalFor								

Above is a screenshot of the observations taken on 28 June 2022 (highlighted in magenta). To generate the quick-look plots for the object run number 32 (highlighted in red) using its flat (run 28; highlighted blue) and the robot file (highlighted in light green), you need to run the following command.

In Ipython:

```
run display_probes.py 32 28 --file_prefix 28jun --robot_file_name Robot_FinalFormat_tile_220_m22_GC_OM_AprilPos_PQ_NonZero_UT_Correct_CONFIGURED_correct_header.csv
```

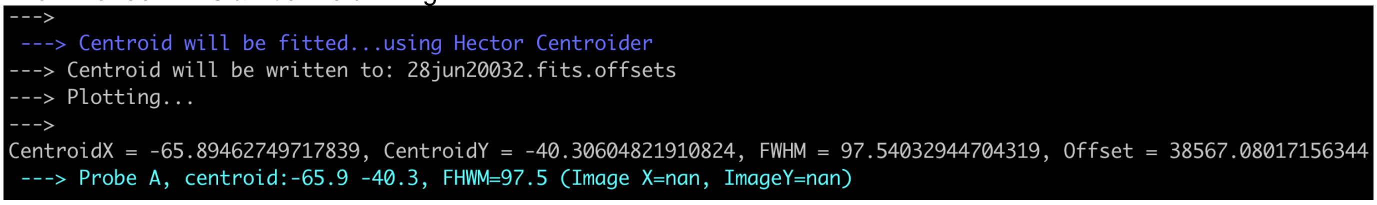
Once the quick-look tool starts to run, the terminal will display some information relevant to the data/flat frames given. This is your chance to check whether the quick-look tool uses the correct data and flat frames across all CCDs. The terminal output will look like:



```
---> START
 --->
 ---> Object frame: /Users/madusha/Documents/PycharmProjects/DATA_REDUCTION/Hector/220628/ccd_2/28jun20032.fits
 --->
 --->
 ---> Using partially reduced flat TLM file: /Users/madusha/Documents/PycharmProjects/DATA_REDUCTION/Hector/220628/reduced/ccd_2/28jun20028tlm.fits
 --->
alive fibres:    773  +  31  =  804  (fibtab types "P" + "S")
offline fibres:    3  +  12  =  15  (fibtab types "U" + "N")
HBundle where deadU ['D' 'H' 'C']
Fibre index where deadU [194 205 637]
HBundle where deadN ['' 'Sky-A2-7' '' '' '' 'Sky-A5-3' '' '' 'Sky-A5-2' '' '' 'Sky-A5-1']
Fibre index where deadN [  1   2  64 127 190 567 568 631 633 694 757 758]
 ---> Performing 'Sigma-clip'... (~20s)
```

The colour-coded lines provide information on the inputted data/flat frames.

Then the tool will start centroid-fitting...



```
--->
 ---> Centroid will be fitted...using Hector Centroider
---> Centroid will be written to: 28jun20032.fits.offsets
---> Plotting...
--->
CentroidX = -65.89462749717839, CentroidY = -40.30604821910824, FWHM = 97.54032944704319, Offset = 38567.08017156344
 ---> Probe A, centroid:-65.9 -40.3, FHWM=97.5 (Image X=nan, ImageY=nan)
```

The quick-look script doesn't run to completion without user input. You will be asked to provide some feedback on the centroid fits.

At this stage, examine the figure that the quick-look tool has produced and exclude any hexabundles with incorrect centroid fits.

All outputs from the quick-look tool are saved in the "Outputs" folder under the working data directory.

Changing the default running mode of the quick-look tool

There are several keywords that you can use to change the default running mode of the quick-look tool. Some keywords can be invoked directly when running the quick-look tool, and some via the "update_quick_look_config.py" script.

IMPORTANT: keywords invoked via the "update_quick_look_config.py" script will be added permantly to the quick-look configuration file, whereas the keywords invoked when running the quick-look tool applies only to that instance.

Table 1: A list of available keywords

Keyword	Can be invoked when running the quick-look tool*	Can be invoked prior to running the quick-look tool (i.e., in the “"update_quick_look_config” script)**	Example
--file_prefix	yes	yes	--file_prefix 28jun
--robot_file_name	yes	yes	--robot_file_name Robot_FinalFormat_etc
--spectrograph_not_used	yes	yes	--spectrograph_not_used AAOmega
--red_or_blue	yes	yes	--red_or_blue red
--sigma_clip	No	Yes	--sigma_clip True
--centroid	No	Yes	--centroid True

*** the change applies only to that instance of the quick-look tool**
**** permanently adds the keyword to the quick-look configuration file.**

Table 2: Descriptions of the keywords

Keyword	Explanation
--file_prefix	the data prefix appended to the raw data frames
--robot_file_name	the name of the robot file
--spectrograph_not_used	Specify which spectrograph to remove from the analysis (options: AAOmega, Hector, None). By default set to None
--red_or_blue	Specify which arm to use for the analysis (options: red, blue, both), By default set to 'both'
--sigma_clip	Specify whether to perform sigma clipping on the data (options: True, False). By default set to True
--centroid	Specify whether to perform centroiding (options: True, False). By default set to True

Essential: QC summary

***** July 2023; Sree: the transmission calculation is on test. So, observers may only consider FWHM for qc.

At any time you can print out a summary of the frames observed, including some basic quality control metrics:

```
In [4]: mngr.qc_summary()                    #this will print out summary of the frames
```

```
QC summary table
===========================================================================
Use space bar and cursor keys to move up and down; q to quit.
If one file in a pair is disabled it is marked with a +
If both are disabled it is marked with a *

Summary of shared calibrations
---------------------------------------------------------------------------
BIAS Frames:
  230419: 50 frames
  230424: 12 frames
  230425: 62 frames
  230426: 1 frames
  230428: 2 frames
  TOTAL BIASs: 127 frames
DARK Frames:
  230419: 4 frames
  230420: 11 frames
  230421: 6 frames
  230422: 6 frames
  TOTAL DARKs: 27 frames
LFLAT Frames:
  230420: 32 frames
  TOTAL LFLATs: 32 frames
Flux standards
  230421:
    LTT7987: 9 frames
    Total: 9 frames
```

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
G15_T025_F0
---------------------------------------------------------------------------
File        Exposure  FWHM (")  Transmission  Sky residual
24apr 0045      1800      2.68             -         0.012
24apr 0046      1800      2.73             -         0.012
24apr 0047      1800      2.43             -         0.007
24apr 0048      1800      2.57             -         0.006
25apr 0090+     1800      1.81             -         0.008
25apr 0091+     1800      2.27             -         0.005
25apr 0092+     1800      2.75             -         0.001
25apr 0093*     1800         -             -             -
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

The QC values are not filled in until certain steps of the data reduction have been done. After the mngr.telluric_correct(), it starts to show seeing (FWHM “). You need to get as far as mngr.scale_frames() to see transmission. While observing, keep an eye on the seeing and the transmission. **If the seeing is above 3" or the transmission is below about 0.7, the data are unlikely to be of much use. If so, plan to re-take the frame.**

Essential: upload raw data to the data central cloud

It is extremely important to upload the raw data to the data central cloud. Note that the data at AAT machines are cleaned up regularly. Do not wait until the last day, and observers upload data daily basis.

First night

```
(base) hector@aatlxe:~$ ssh -p 2232 hector@uploads.datacentral.org.au “mkdir -p /data/Observing/20YYMMDD_20YYMMDD/YYMMDD_YYMMDD/”
L3tMe!nPle@se
```

The above will make a directory in the data central cloud. Do not need to repeat this once the directory has been made.

Every night, when the observing is getting to the end, observers upload raw data from **the reduction machine**.

```
(base) hector@aatlxe:~$ rsync -avhPin -e 'ssh -p 2232' /data/hector/reduction/YYMMDD_YYMMDD/raw hector@uploads.datacentral.org.au:/data/Observing/20YYMMDD_20YYMMDD/YYMMDD_YYMMDD/
```

The above with the ‘n’ flag prints the list of frames to be uploaded, but it does not actually transfer any frames.

```
(base) hector@aatlxe:~$ rsync -avhPi -e 'ssh -p 2232' /data/hector/reduction/YYMMDD_YYMMDD/raw hector@uploads.datacentral.org.au:/data/Observing/20YYMMDD_20YYMMDD/YYMMDD_YYMMDD/
```

Once the list has been confirmed, upload data to the data central, without using the ‘n’ flag.

Last night special

After taking the final frames of the run, observers should double check that all the frames have been loaded to the reduction machine, before uploading.

```
In [4]: mngr.import_aat(date=”220702”)      # date is in YYMMDD
```

Repeat the above command for every single date of the run. Then, upload the raw frames to the data central cloud using rsync.

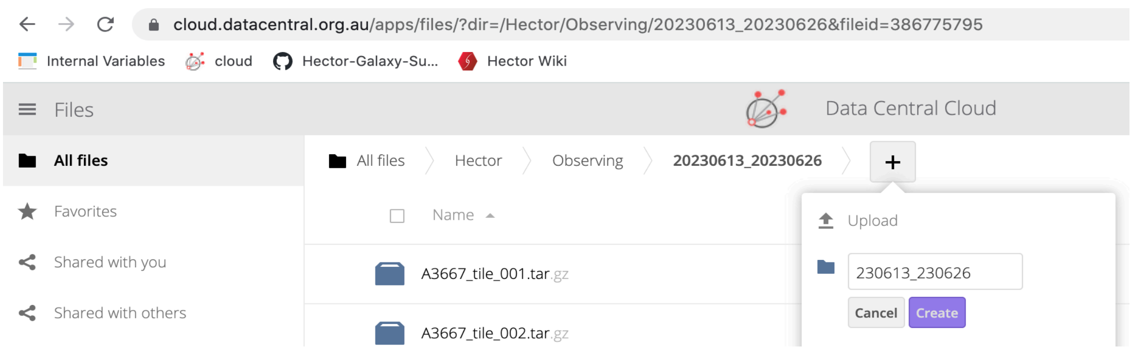
```
(base) hector@aatlxe:~$ scp -P 2232 *.txt hector@uploads.datacentral.org.au: /data/Observing/20YYMMDD_20YYMMDD/YYMMDD_YYMMDD/raw/
```

Also, don’t forget to upload txt files that are very useful while reduction.

HECTOR
Galaxy Survey

The rsync (or scp) is recommended but if it doesn't work, observers may drag and drop the 'raw' folder from the reduction machine to the data central cloud. Navigate to Hector/Observing/20YYMMDD_20YYMMDD/ and generate a folder named "`YYMMDD_YYMMDD`" using the + button. Then, come into the folder and drag and drop /data/hector/reduction/`YYMMDD_YYMMDD`/raw there.

Also, don't forget to upload all txt files from `YYMMDD_YYMMDD`



It is possible (or extremely likely!) that the browser will give weird errors. If so, reload it and consider copying one folder at the time. This should not be done on the last day, but at the end of every night given how temperamental can be the system. You may double check the size of uploaded and local data to make sure you uploaded all the raw data. Don't forget to upload daytime calibration frames.

Essential: upload log on the Hector wiki

At the end of the run, please upload the log to the Hector wiki at https://hector.survey.org.au/wiki/observing/observing-summary. This is important for other team members to understand what happened during observing. Download the log Excel file via File-Download- Excel file. At the end-left of the wiki page you need to click the "Edit" bottom, you need to create a new Header specifying the number of the run, the dates, the observers. Then, add the downloaded Excel file as a "File" block. Have a look at this video on how to edit wiki pages https://hector.survey.org.au/wiki/wiki-help. If you have any issue, please contact Stefania Barsanti (stefania.barsanti@anu.edu.au)

If you take any calibration frames, please list them at this page: https://hector.survey.org.au/wiki/calibration-frames.