

Req 6.3.1

Title:

Pipeline initialisation and image statistics

Objective:

Prepare the data for pipeline processing and provide image statistics and qualifications.

The image part of the Pipeline processing is based on the processing of individual chips. The image data arrives from the telescope in the form of FITS files with the data for each chip stored into a single FITS extension. For the purpose of efficient parallel processing this **seq.-** splits the FITS files (and their extensions) into individual FITS files for individual chips. This functionality is provided generically by the pipeline environment, and is assumed to be used prior to any data processing (including all calibration data processing)

It is assumed that incoming raw data are valid FITS files. Further qualification of the raw data by means of image statistics should determine whether the data can be meaningfully processed by the pipeline.

Separate statistics of pre- and overscan regions are recorded for trend-analysis of detector stability.

Inputs:

Raw science frames.

Outputs:

SeqFile- 631 *Raw science frame*

Estimated time needed:

5 sec/CCD

Recipe:

```
Split -i <raw_unsplit_images> -t image_type [-n number_of_extensions]
```

```
raw_unsplit_images    : raw multi-extension FITS data
image_type            : Type of the supplied raw unsplit images (string)
                        Description of allowed values:
                        bias      - Bias image - see req.541
                        dark      - Dark current image - see req.531
                        dome      - Domeflat image - see req.542
                        science   - Science image - this req.631
                        twilight  - Twilightflat image - see req.543
```

number_of_extensions : Number of extensions (i.e. ccd's) to be found in the

unsplit data (integer)

Range of allowed values: 1 - 64. Default:

32

Needed functionality:

image - statistics

CA:

Process (make):

1. Extract header from given extension and merge with primary header
2. Extract image data from given extension.
3. Determine image statistics in the trim, pre- and overscan regions
4. Write out image with merged header.

Verification (verify):

TBD

Trend Analysis (compare):

None

CAP:

```
main_header = eclipse.header(raw_unsplit_image)
```

```
extension_header = eclipse.header(raw_unsplit_image, extension)
```

```
header = main_header.merge(extension_header)
```

```
# NAXIS1, NAXIS2, PRSCX, PRSCY, OVSCX, OVSCY from header
```

```
# Assumes read-out from left-hand side
```

```
trim = [PRSCX+1, PRSCY+1, NAXIS1-OVSCX, NAXIS2-OVSCY]
```

```
prescan_x = [1, PRSCY+1, PRSCX, NAXIS2-OVSCY]
```

```
overscan_x = [NAXIS1-OVSCX+1, PRSCY+1, NAXIS1, NAXIS2-OVSCY]
```

```
overscan_y = [PRSCX+1, NAXIS2-OVSCY+1, NAXIS1-OVSCX, NAXIS2]
```

```
image = eclipse.image(raw_unsplit_image, extension)
```

```
stat_trim = eclipse.stat(image, region=trim)
```

```
stat_prescan_X = eclipse.stat(image, region=prescan_X)
```

```
stat_overscan_X = eclipse.stat(image, region=overscan_X)
```

```
stat_prescan_Y = eclipse.stat(image, region=prescan_X)
```

```
eclipse.image_save(image, header)
```