European Organisation for Astronomical Research in the Southern Hemisphere

Programme: GEN

Project/WP: Science Operation Software Department

ERIS-NIX Pipeline User Manual

Document Number:

Document Version: 1.8.2

Document Type: Manual (MAN)

Released on: 2025-03-12

Document Classification: Public

Prepared by: ERIS Pipeline Team

Validated by:

Approved by:

Name

ESO – Karl-Schwarzschild-Str. 2 – 85748 Garching bei München – Germany
www.eso.org

Authors

Name	Affiliation
J. Lightfoot, A. Modigliani, M. Neeser	

Change Record from previous Version

Release	Affected Section(s)	Changes/Reason/Remarks
1.1.0	All	First public release
1.1.3	None	
1.2.0	Sec 8, 11	typos
1.2.5	Sec 5, 7, 8, 10, 11, 12	Improve document coverage
1.4.1	Sec 1, 5, 7, 8, 11, 12, App A	Improve document
1.4.4	None	
1.5.0	Sec 5, 8, 11, 12, App A	Align to pipeline update
1.6.0	Sec 5	Align to pipeline update
1.7.0	Sec 5	Align to pipeline update
1.8.2	Sec 5,11	Described more parameters controlling skysub and wcs steps, align to pipeline update

Contents

1 Introduction

1.1 Purpose

The ERIS-NIX pipeline is a subsystem of the VLT Data Flow System (DFS). It is used in two operational environments, for the ESO Data Flow Operations (DFO), and the Paranal Science Operations (PSO), in the quick-look assesment of data, in the generation of master calibration data, in the reduction of scientific exposures, and in the data quality control. Additionally, the ERIS-NIX pipeline recipes are made public to the user community, to allow a more personalised processing of the data of the instrument. The purpose of this document is to describe a typical ERIS-NIX data reduction sequence with the ERIS-NIX pipeline. This manual is a complete description of the data reduction recipes implemented by the ERIS-NIX pipeline, reflecting the status of the ERIS-NIX pipeline as of version 1.8.2.

1.2 Scope

This document describes the ERIS-NIX pipeline used at ESO to assess instrument and data quality control.

The examples on running individual pipeline recipes in this manual use the esorex command and manually created list of input files. Several interfaces to automatically organise the data, create the list of input files and execute the pipeline recipes in the proper sequence are available, see the `ESO pipeline page` for details.

Please note that the use of Gasgano as a GUI for processing data is deprecated. Its use is no longer recommended and the related section in this manual, as well as support for Gasgano as a data processing GUI application in general will be dropped entirely in a future release.

Updated versions of the present document may be found on [3]. For general information about the current instrument pipelines status we remind the user of [4] . Additional information on CFITSIO, the Common Pipeline Library (CPL) and EsoRex can be found respectively at [9], [6], [7]. The Reflex front-ends is described in [10] and a description of the instrument can be found in [5]. The ERIS instrument user manual is in [5], the calibration plan in [2]. The ERIS pipeline project in a nutshell is also described in [1].

1.3 Acknowledgements

The ERIS-NIX pipeline was designed, implemented, and verified by John Lightfoot of the Royal Observatory Edinburgh, as part of the ERIS ATC consortium.

Mark Neeser, from the ESO Science Data Quality group, led this pipeline project since its beginning and contributed to the design, algorithms, and overall scientific verification.

Isabelle Percheron, from the ESO Science Data Quality group, is responsible of the integration of the pipeline in the Quality Control environment and, from early commissioning, provided valuable feedback on the data and its quality control.

Lars Lundin and Andrea Modigliani, from the Science Operations Software Pipeline Systems Group, followed the development of the pipeline to ensure that the algorithms are properly implemented and that the code is robust

and easy to maintain. The latter provided support for the commissioning runs, contributed to the documentation and is developing and maintaining the pipeline.

This release and documentation includes improvements to address feedback from scientists and users.

1.4 Stylistic conventions

Throughout this document the following stylistic conventions are used:

bold	in text sections for commands and other user input which has to be typed as shown
italics	in the text and example sections for parts of the user input which have to be replaced with real contents
`teletype`	in the text for FITS keywords, program names, file paths, and terminal output, and as the general style for examples, commands, code, etc

In example sections expected user input is indicated by a leading shell prompt.

In the text **bold** and *italics* may also be used to highlight words.

1.5 Notational Conventions

Hierarchical FITS keyword names, appearing in the document, are given using the dot–notation to improve readability. This means, that the prefix "HIERARCH ESO" is left out, and the spaces separating the keyword name constituents in the actual FITS header are replaced by a single dot.

2 Related Documents

[RD01]	ESO-476500	ERIS User Manual
[RD02]	Vacca, Cushing and Rayner, 2004, PASP, 116, 352	Nonlinearity Corrections and Statistical Uncertainties Associated with Near-Infrared Arrays

[1] E.A. et al. E.A. Wiezorrek. *The ERIS pipeline*. ADS. Proceedings of the SPIE, Volume 13101, id. 131012X 16 pp. (2024). 11

[2] ERIS Consortium. *ERIS Instrument Calibration Plan*. VLT-TRE-???-15860-9303. 11

[3] ESO, *http://www.eso.org/pipelines*. *ERIS-NIX Pipeline Web Page*. . 11

[4] ESO, *http://www.eso.org/observing/dfo/quality/pipeline-status.html*. *ERIS Pipeline Current Status*. 11

[5] ESO, *http://www.eso.org/sci/facilities/paranal/instruments/eris/doc.html*. *ERIS User Manual*. 11

[6] ESO/SDD/DFS, *http://www.eso.org/observing/cpl/download.html*. *Common Pipeline Library User Manual*. VLT-MAN-ESO-19500-2720. 11

[7] ESO/SDD/DFS, *http://www.eso.org/cpl/esorex.html*. *ESOREX home page*. 11, 16

[8] ESO/SDD/DFS, *http://www.eso.org/gasgano/*. *Gasgano User's Manual*. VLT-PRO-ESO-19000-1932. 16

[9] NASA, *http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html*. *CFITSIO home page*. 11

[10] Forchì V. *Reflex User's Manual*. ESO/SDD/DFS, *http://www.eso.org/gasgano/*, 0.7 edition, 2012. VLT-MAN-ESO-19000-5037. 11, 16

3 Definitions, Acronyms and Abbreviations

CalibDB	Calibration Database
CPL	Common Pipeline Library
CCD	Charge Coupled Device
DFS	Data Flow System
DRS	Data Reduction System
ESO	European Southern Observatory
EsoRex	ESO Recipe Execution Tool
FITS	Flexible Image Transport System
FOV	Field Of View
GUI	Graphical User Interface
LSF	Line Spread Function
ERIS	TODO
OB	Observation Block
pixel	picture element (of a raster image)
PSF	Point Spread Function
QC	Quality Control
SDP	Science Data Product
SOF	Set Of Frames
TBD	To be defined
TBC	To be confirmed
VLT	Very Large Telescope
WCS	World Coordinate System

4 Overview

In collaboration with instrument consortia, the Data Flow Systems Department (DFS) of the Data Management and Operation Division is implementing data reduction pipelines for the most commonly used VLT/VLTI instrument modes. These data reduction pipelines have the following three main purposes:

Data quality control: pipelines are used to produce the quantitative information necessary to monitor instrument performance.

Master calibration product creation: pipelines are used to produce master calibration products (*e.g.*, combined bias frames, super-flats, wavelength dispersion solutions).

Science product creation: using pipeline-generated master calibration products, science products are produced for the supported instrument modes. The accuracy of the science products is limited by the quality of the available master calibration products and by the algorithmic implementation of the pipelines themselves. In particular, adopted automatic reduction strategies may not be suitable or optimal for all scientific goals.

Instrument pipelines consist of a set of data processing modules that can be called from the command line, from the automatic data management tools available on Paranal, from EsoReflex, or from Gasgano.

ESO offers three front-end applications for launching pipeline recipes; namely *EsoReflex* [10], *Gasgano* [8] and *EsoRex* [7]. These applications can also be downloaded separately from `www.eso.org/reflex`, `www.eso.org/gasgano`,
`www.eso.org/cpl/esorex.html` respectively.

The ERIS instrument and the different types of ERIS raw frames and auxiliary data are described in Sections 6, 9.

A brief introduction to the usage of the available reduction recipes using EsoRex is presented in Section 7.1. In section 8 we summarise known data reduction problems and solutions, if available.

An overview of the data reduction, what are the input data, and the recipes involved in the calibration cascade is provided in section 11.

More details on what are inputs, products, quality control measured quantities, and controlling parameters of each recipe is given in section 12.

More detailed descriptions of the data reduction algorithms used by the individual pipeline recipes can be found in Section 12.

In Appendix A the installation of the eris pipeline recipes is described and in section 3 a list of used abbreviations and acronyms is given.

5 What's new in pipeline release 1.8.2

This pipeline release includes the following improvements:

- Delivered an EDPS workflow to process ERIS-NIX data.
- Implemented a recipe `eris_nix_pupil` to preprocess PUPIL,LAMP and PUPIL,SKY frames.
- Updated documentation.

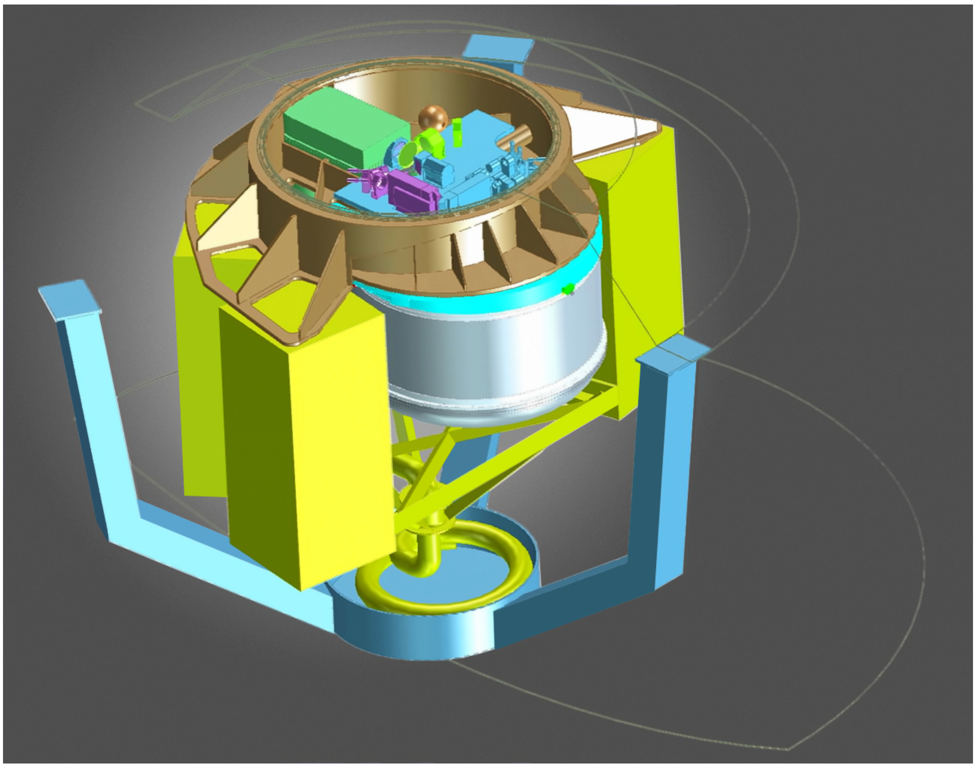


Figure 6.1: Graphical representations of ERIS.

6 The ERIS Instrument

ERIS is part of a new generation of technology for ESO's Very Large Telescope (VLT) located at the Paranal Observatory in Chile. The Enhanced Resolution Imager and Spectrograph (ERIS) instrument combines a general-use infrared imager and integral field spectrograph with the world-class adaptive optics installed on the VLT's Unit Telescope 4 (Yepun).

The versatility of ERIS lends itself to many fields of astronomical research and it aims to take the sharpest images obtained to date using a single 8.2-metre class telescope once operational. ERIS is expected to greatly contribute to probing of distant galaxies, the Galactic Centre, our Solar System and exoplanets.

The new, more sensitive technology of ERIS will succeed the successful NACO and SINFONI instruments, which have been used on the VLT since the early 2000s. ERIS is planned to work for 10 years following first light.

6.1 ERIS-SPIFFER, Spectrograph

The spectrograph of ERIS, named ERIS-SPIFFIER, is a refurbished version of SINFONI's SPIFFI (SPectrometer for Infrared Faint Field Imaging) instrument. The ERIS-SPIFFIER, as an integral field spectrograph, works in much the same way as SPIFFI. Each pixel of the spectrograph is capable of collecting a full spectrum, which is the intensity of each wavelength measured by the instrument. The data collected is multidimensional and contains a lot of information for astronomers to analyse. This type of spectrograph allows astronomers to determine the relative position in space of each part of the target object. Combined with the infrared observing wavelength, this type of spectroscopy can for example allow astronomers to study the spin, structure, and content of high-redshift galaxies. A higher resolution grating on the spectrograph, used to split incoming light to capture a spectrum, works similarly to how a prism splits light into different colours.

6.2 ERIS-NIX, Infrared Imager

The imager on ERIS is the Near Infrared Camera System (NIX), the next generation of infrared imager for the VLT. Astronomers are able to use two main modes for the imager. One mode is direct imaging of astronomical objects. Another mode is coronagraphy, an observing technique that works by suppressing the direct light of a star. This technique on ERIS-NIX allows astronomers to observe exoplanets and discs of gas and dust around young stars.

ERIS-NIX is capable of collecting data in near-infrared between 3000 and 5000 nanometers, which are particularly difficult wavelengths to observe from the ground due to excess light interference from the sky. As such, ERIS is one of the few ground-based instruments with this wavelength coverage on an 8-metre class telescope and observing at these wavelengths allows for a new and unique view of exoplanets and galaxies.

6.3 ERIS and the Adaptive Optics Facility

ERIS is also equipped with an adaptive optics module that works together with the Adaptive Optics Facility (AOF) installed in 2017 on Yepun. Adaptive optics make real-time corrections for atmospheric turbulence that distorts starlight to allow for much sharper image quality, and are an important part of modern ground-based astronomical observations. The AOF capabilities in conjunction with the module allows ERIS to capture the sharpest data possible with the instrument and cover more of the sky than earlier VLT instruments.

7 Quick Start

7.1 The ERIS Pipeline Recipes

The ERIS-NIX pipeline reduction chain uses the following recipes:

```
eris_nix_detmon_ir_lg : Linearity/Gain recipe for the IR domain
eris_nix_dark         : Calculate MASTER_DARK and 'hot-pixel' BPM
eris_nix_flat_lamp    : Calculate a MASTER_FLAT_LAMP_HIFREQ,
eris_nix_flat_twilight: Calculate a MASTER_FLAT_TWILIGHT_LOFREQ
eris_nix_flat_sky     : Calculate a MASTER_FLAT_SKY_HIFREQ/LOFREQ and
eris_nix_cal_det      : Remove detector signature from frames
eris_nix_img_skysub   : Subtract sky background from frames
eris_nix_img_cal_wcs  : Calibrate wcs of ERIS/NIX frames
eris_nix_img_cal_phot : Calibrate photometry of ERIS/NIX frames
eris_nix_img_hdrl_stack  : Stack calibrated ERIS/NIX jitter frames using HDRL
eris_nix_img_scired   : Reduce science data (wrapper of previous five recipes)
eris_nix_lss_skysub   : Subtract sky background from LSS frames.
eris_nix_lss_straighten  : Rectify and wavelength calibrate LSS frames.
eris_nix_lss_stack    : Stack calibrated ERIS/NIX LSS frames using HDRL.
eris_nix_pupil        : Remove detector signature from pupil frames
```

The pipeline fully supports the ERIS/NIX imaging mode (both the 13 & 27 mas/pix scales), but only provides basic (single exposure) image reduction steps for the high contrast modes (dark subtraction, linearity correction, flat-fielding, and bad pixel interpolation).

There is limited support for the ERIS/NIX long-slit spectrum (LSS) mode, sufficient to produce a wavelength calibrated, stacked 2d-spectrum from an observation.

7.2 Running the ERIS Pipeline Recipes

7.2.1 Getting Started with EsoRex

EsoRex is a command-line tool which can be used to execute the recipes of all standard VLT/VLTI instrument pipelines. With *EsoRex* in your path, the general structure of an *EsoRex* command line is

```
1> esorex [esorex options] [recipe [recipe options] [sof [sof]...]]
```

where options appearing before the recipe name are options for *EsoRex* itself, and options given after the recipe name are options which affect the recipe.

All available *EsoRex* options can be listed with the command

```
1> esorex --help
```

and the full list of available parameters of a specific recipe can be obtained with the command

```
1> esorex --help <recipe name>
```

The output of this command shows as parameter values the current setting, i.e. all modifications from a configuration file or the command line are already applied.

The listing of all recipes known to *EsoRex* can be obtained with the command

```
1> esorex --recipes
```

The last arguments of an *EsoRex* command are the so-called *set-of-frames*. A *set-of-frames* is a simple text file which contains a list of input data files for the recipe. Each input file is followed by an unique identifier (frame classification or frame tag), indicating the contents of this file. The input files have to be given as an absolute path, however *EsoRex* allows the use of environment variables so that a common directory prefix can be abreviated. Individual lines may be commented out by putting the hash character (#) in the first column. An example of a *set-of-frames* is shown in the following:

```
1> cat bias.sof
/data/eris/raw/ERIS.2022-03-29T09:48:53.153.fits DARK
$RAW_DATA/ERIS.2022-03-29T09:50:36.645.fits DARK
$RAW_DATA/ERIS.2022-03-29T09:52:16.513.fits DARK
$RAW_DATA/ERIS.2022-03-29T09:53:47.996.fits DARK
#$RAW_DATA/ERIS.2022-03-29T09:55:04.515.fits DARK
```

These *set-of-frames* files will have to be created by the user using a text editor, for instance. Which classification has to be used with which recipe will be shown in section 7.3

Finally, if more than one *set-of-frames* is given on the command-line *EsoRex* concatenates them into a single *set-of-frames*.

7.3 Data Organization

The ERIS-NIX pipeline recipes accept the following input tags and generates frames with the following output tags:

- eris_nix_detmon_ir_lg

```
    Input files:

      DO category:                 Explanation:                 Required:
      ---------------------------  -----------                  ---------
      ON_RAW                       Data (Flat)                  YES
      OFF_RAW                      Data (Dark)                  YES
```

```
Output files:

  DO category:                  Explanation:
  --------------------------    -----------
  BP_MAP_NL                      non-linear pixel map
  COEFFS_CUBE                    cube with pix-to-pix poly coeffs
  GAIN_INFO                      table with information on computed gain
  DET_LIN_INFO                   table with information on linearity check
```

- eris_nix_dark

```
Input files:

  DO category:                  Explanation:                  Required:
  --------------------------    -----------                   ---------
  DARK                          Raw Data                      YES
  BP_MAP_NL                     eris_nix_detmon_ir_lg product  YES
  COEFFS_CUBE                   eris_nix_detmon_ir_lg product  YES
  DET_LIN_INFO                  eris_nix_detmon_ir_lg product  YES
  GAIN_INFO                     eris_nix_detmon_ir_lg product  YES

Output files:

  DO category:                  Explanation:
  --------------------------    -----------
  MASTER_DARK_IMG               master dark
```

- eris_nix_flat_lamp

```
Input files:

  DO category:                  Explanation:                  Required:
  --------------------------    -----------                   ---------
  FLAT_LAMP_ON                  Data                          YES
  FLAT_LAMP_OFF                 Data                          NO
  BP_MAP_NL                     eris_nix_detmon_ir_lg product   YES
  COEFFS_CUBE                   eris_nix_detmon_ir_lg product   YES
  DET_LIN_INFO                  eris_nix_detmon_ir_lg product   YES
  GAIN_INFO                     eris_nix_detmon_ir_lg product   YES
  MASTER_DARK_IMG               master dark                   YES

Output files:

  DO category:                  Explanation:
  --------------------------    -----------
```

```
    MASTER_BPM_LAMP              master bad pixel map
    MASTER_FLAT_LAMP_HIFREQ      master flat lamp (hifreq)
    MASTER_FLAT_LAMP_LOFREQ      master flat lamp (lofreq)
```

- eris_nix_flat_twilight

```
  Input files:

    DO category:                 Explanation:                   Required:
    ---------------------------  ------------                   ---------
    FLAT_TWILIGHT                     Data                           YES
    BP_MAP_NL                    eris_nix_detmon_ir_lg product   YES
    COEFFS_CUBE                  eris_nix_detmon_ir_lg product   YES
    DET_LIN_INFO                 eris_nix_detmon_ir_lg product   YES
    GAIN_INFO                    eris_nix_detmon_ir_lg product   YES
    MASTER_DARK_IMG              master dark                     YES
    MASTER_BPM_LAMP              master bad pixel (lamp)         YES
    MASTER_FLAT_LAMP_HIFREQ      master flat hifreq (lamp)       YES
    MASTER_FLAT_LAMP_LOFREQ      master flat lofreq (lamp)       YES

  Output files:

    DO category:                 Explanation:
    ---------------------------  ------------
    MASTER_FLAT_TWILIGHT_LOFREQ      master flat twilight (lofreq)
```

- eris_nix_flat_sky

```
  Input files:

    DO category:                 Explanation:                   Required:
    ---------------------------  ------------                   ---------
    FLAT_SKY                     Data                           YES
    BP_MAP_NL                    eris_nix_detmon_ir_lg product   YES
    COEFFS_CUBE                  eris_nix_detmon_ir_lg product   YES
    DET_LIN_INFO                 eris_nix_detmon_ir_lg product   YES
    GAIN_INFO                    eris_nix_detmon_ir_lg product   YES
    MASTER_DARK_IMG              master dark                     YES

  Output files:

    DO category:                 Explanation:
    ---------------------------  ------------
    MASTER_FLAT_SKY_HIFREQ      master flat sky (hifreq)
    MASTER_FLAT_SKY_LOFREQ      master flat sky (lofreq)
    MASTER_BPM_SKY              master bad pixel map
```

- eris_nix_cal_det

```
Input files:

  DO category:                  Explanation:                   Required:
  ---------------------------   -----------                    ---------
  OBJECT_JITTER                 Data                           YES
  SKY_JITTER                    Data                           YES
  BP_MAP_NL                     eris_nix_detmon_ir_lg product   YES
  COEFFS_CUBE                   eris_nix_detmon_ir_lg product   YES
  DET_LIN_INFO                  eris_nix_detmon_ir_lg product   YES
  GAIN_INFO                     eris_nix_detmon_ir_lg product   YES
  MASTER_DARK_IMG               master dark                     YES
  MASTER_BPM_LAMP               master bad pixel map            YES
  MASTER_FLAT_LAMP_HIFREQ       master flat lamp (hifreq)       YES
  MASTER_FLAT_LAMP_LOFREQ       master flat lamp (lofreq)       YES
  MASTER_FLAT_TWILIGHT_LOFREQ  master flat twilight (lofreq)   YES
  WCS_REFINE                   static table with coef to      YES
                               correct residual distortions
Output files:

  DO category:                  Explanation:
  ---------------------------   -----------
  CAL_DET_OBJECT_JITTER         instrument signature corrected science frame
  CAL_DET_SKY_JITTER            instrument signature corrected sky frame
```

- eris_nix_img_skysub

```
Input files:

  DO category:                  Explanation:                   Required:
  ---------------------------   -----------                    ---------
  CAL_DET_OBJECT_JITTER         eris_nix_cal_det product        YES
  CAL_DET_SKY_JITTER            eris_nix_cal_det product        YES

Output files:

  DO category:                  Explanation:
  ---------------------------   -----------
  SKYSUB_OBJECT_JITTER          sky corrected science frame
```

- eris_nix_img_cal_wcs

```
Input files:
```

```
   DO category:                  Explanation:                 Required:
   ---------------------------   -----------                  ---------
   SKYSUB_OBJECT_JITTER          eris_nix_img_skysub product   YES
   WCS_MATCHED_CATALOGUE         WCS catalog (static frame)    YES
Output files:

   DO category:                  Explanation:
   ---------------------------   -----------
   CAL_WCS__OBJECT_JITTER         WCS calibrated science frame
   CAL_WCS_CATALOGUE              catalog used objects
   CAL_WCS_MATCHCAT               catalog objects that matches
```

- eris_nix_img_cal_phot

```
Input files:

   DO category:                  Explanation:                 Required:
   ---------------------------   -----------                  ---------
   CAL_WCS_OBJECT_JITTER          eris_nix_img_cal_wcs product    YES
   CAL_WCS_CATALOGUE              catalog used objects           YES
   CAL_WCS_MATCHCAT               catalog objects that matches   YES
   CAL_WCS_REFCAT                 catalog with reference objects YES
   PHOT_DATA                      photometric catalog
Output files:

   DO category:                  Explanation:
   ---------------------------   -----------
   CAL_PHOT__OBJECT_JITTER        photometric calibrated science frame
   CAL_PHOT_CATALOGUE             catalog used objects
   CAL_PHOT_MATCHCAT              catalog objects that matches
   CAL_PHOT_REFCAT                reference catalog objects
```

- eris_nix_img_hdrl_stack

```
Input files:

   DO category:                  Explanation:                 Required:
   ---------------------------   -----------                  ---------
   CAL_PHOT_OBJECT_JITTER         eris_nix_img_cal_wcs product    YES
   CAL_PHOT_CATALOGUE             catalog used objects           YES
   CAL_PHOT_MATCHCAT              catalog objects that matches   YES
   CAL_PHOT_REFCAT                reference catalog objects      YES

Output files:

   DO category:                  Explanation:
```

```
---------------------------   -----------
IMG_OBS_CATALOGUE              catalog used objects
IMG_OBS_COMBINED               combined (stacked) science frame
```

- eris_nix_img_scired

```
Input files:

  DO category:                  Explanation:                   Required:
  ---------------------------   -----------                    ---------
  OBJECT_JITTER                 Data                           YES
  SKY_JITTER                    Data                           YES
  BP_MAP_NL                     eris_nix_detmon_ir_lg product   YES
  COEFFS_CUBE                   eris_nix_detmon_ir_lg product   YES
  DET_LIN_INFO                  eris_nix_detmon_ir_lg product   YES
  GAIN_INFO                     eris_nix_detmon_ir_lg product   YES
  MASTER_DARK_IMG               master dark                     YES
  MASTER_BPM_LAMP               master bad pixel map            YES
  MASTER_FLAT_LAMP_HIFREQ       master flat lamp (hifreq)       YES
  MASTER_FLAT_LAMP_LOFREQ       master flat lamp (lofreq)       YES
  MASTER_FLAT_TWILIGHT_LOFREQ   master flat twilight (lofreq)   YES
  WCS_MATCHED_CATALOGUE         WCS catalog (static frame)      YES
  PHOT_DATA                     photometric catalog             YES
  WCS_REFINE                   static table with coef to        YES
                               correct residual distortions

Output files:

  DO category:                  Explanation:
  ---------------------------   -----------
  CAL_DET_OBJECT_JITTER         instrument signature corrected science frame
  CAL_DET_SKY_JITTER            instrument signature corrected sky frame
```

8 Known Issues

8.1 Use ESO Reflex

We recommend that you reduce the data with ESOReflex. This release provides a workflow for ERIS-NIX IMG data reduction (`eris_nix_img.xml`).

8.2 Astrometric Correction

As is described in detail in section 11.1.3 part 3, the ERIS/NIX pipeline uses GAIA sources in the field of each science exposure to create an astrometric plate solution. However, this will not always be possible. The relatively small fields of ERIS ($57'' \times 57''$ and $27'' \times 27''$), implies that some pointings will have too few GAIA sources to allow an automatic improvement of the frame's absolute world coordinate system (WCS).

An investigation was done to estimate the probability of any given ERIS/NIX field having *N* GAIA sources. For both ERIS field sizes, approximately 4.5×10^6 pointings were randomly distributed in a range of right ascension and declination accessible to Paranal ($0° \leq \alpha < 360°$ and $-90° \leq \delta \leq 30°$). For each such pointing, the DR3 GAIA catalogue was queried and the number of GAIA sources counted.

As is expected, the density of GAIA sources is greatest near the Galactic plane and declines rapidly at high Galactic latitudes. However, because of the small field of ERIS it is possible to have zero GAIA sources at almost any value of Galactic longitude or latitude (see figure 8.1). As can be seen in figure 8.2 and in table 8.1 a full 26% of all random ERIS/NIX pointings will have no GAIA sources in the field. For the $27'' \times 27''$ this is significantly worse and more than one half of the fields will have no GAIA sources (see figure 8.3 and table 8.2). Since the pipeline will require three or more good GAIA catalogue sources to correlate with detections in the ERIS image, the situation can be considerably worse. In cases such as these, the ERIS/NIX pipeline will compute relative offsets from the reference frame in the OB jitter and will use the absolute WCS given by the telescope. In this way, the relative matching between the frames of the OB can be good, however, the absolute astrometry can be off by several arcseconds. If there are insufficient GAIA sources in the field, or if the automatic cross-matching process fails, the User can also manually correct the reference jitter WCS and then reference other jitters onto that. This is described in the **manual specification** description of section 3.

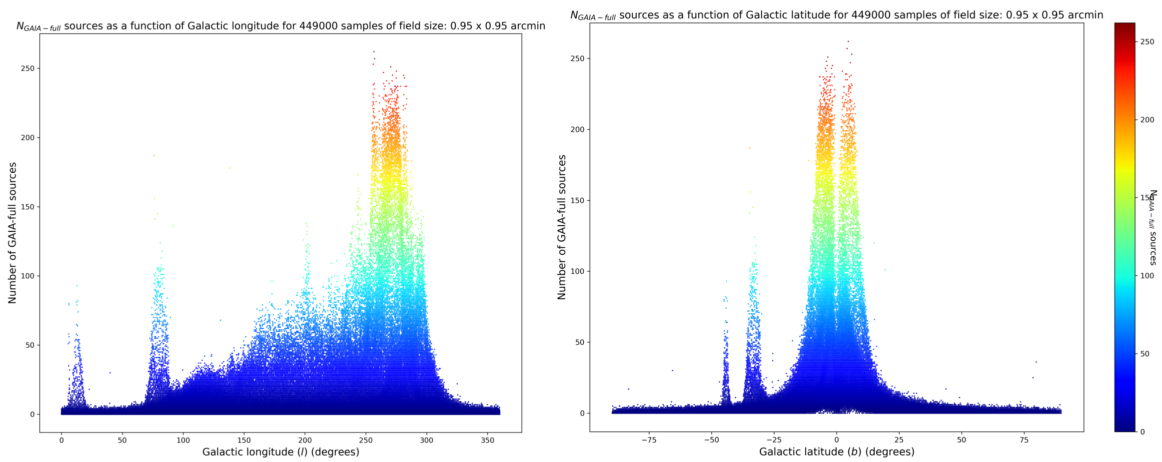


Figure 8.1: The number of GAIA sources in a random distribution of 4.5×10^6 ERIS/NIX pointings with a field-of-view of $57'' \times 57''$ as a function of Galactic longitude (left panel) and latitude (right panel). The density of GAIA sources is obviously greatest near the Galactic plane and declines rapidly at high Galactic latitudes. However, the small field of ERIS implies that it is possible to have zero GAIA sources at any value of (l, b).

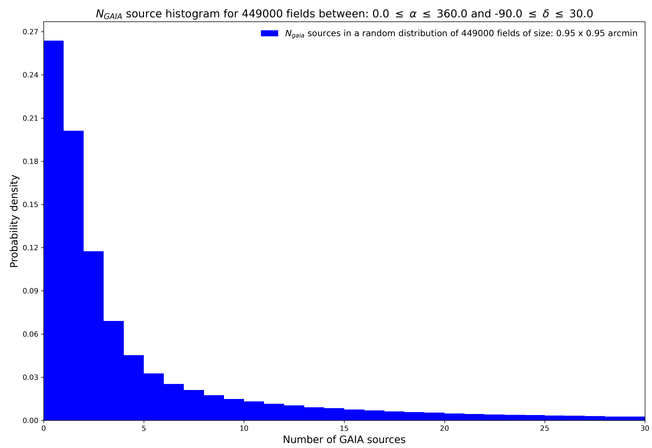


Figure 8.2: The distribution of ERIS/NIX pointings ($57'' \times 57''$ field-of-view) having 0 to 30 GAIA sources.

Table 8.1: Probability of an ERIS/NIX pointing (field-of-view: $57'' \times 57''$) having N GAIA sources

N_{GAIA} sources	$N_{samples}$	Percentage
0	118,437	26.38
1	90,348	20.12
2	52,762	11.75
3	31,035	6.91
4	20,357	4.53
5	14,648	3.26
$5 < N \leq 10$	41,222	9.18
$10 < N \leq 20$	34,173	7.61
$N > 20$	46,018	10.25
sum:	449,000	100.00

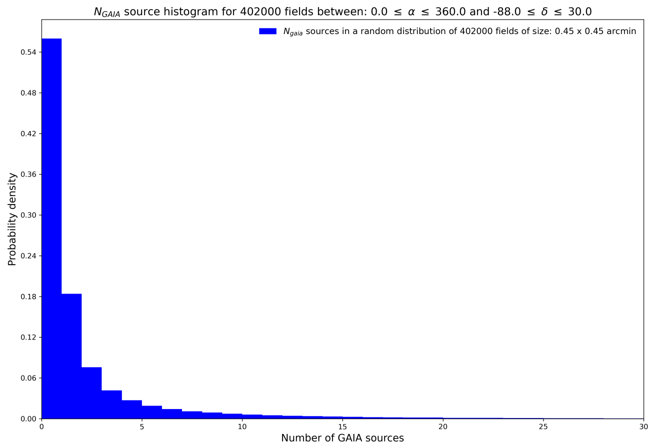


Figure 8.3: The distribution of ERIS/NIX pointings ($27'' \times 27''$ field-of-view) having 0 to 30 GAIA sources.

Table 8.2: Probability of an ERIS/NIX pointing (field-of-view: $27'' \times 27''$) having N GAIA sources

N_{GAIA} sources	$N_{samples}$	Percentage
0	225,084	55.99
1	73,993	18.41
2	30,477	7.58
3	16,734	4.16
4	10,982	2.73
5	7,700	1.92
$5 < N \leq 10$	19,230	4.78
$10 < N \leq 20$	11,436	2.84
$N > 20$	6,364	1.58
sum:	402,000	100.00

8.3 Heavy data processing time for data with DET.FRAM.FORMAT='cube'

The reduction of IMG 'cube' data (DET.FRAM.FORMAT=="cube") may involve processing several frames and with data cubes having 100 planes or more. Users are warned that this is CPU intensive. ERIS-NIX data cubes are processed plane-by-plane and only the detector signature is removed by the eris_nix_cal_det recipe. This release offers a more efficient implementation that was measured to reduce the overall computation time up to a factor 2.8. A further speed-up can be obtained by disabling the computation of the pipeline product checksums. This can be done by running the recipe as: `esorex --nodatamd5=true --no-checksum=true my_recipe` The most demanding data set in terms of RAM and computation time found thus far was one including 84 `OBJECT_JITTER` frames with DET.FRAM.FORMAT='cube', each with 100 planes. This was processed using a 12th Gen Intel(R) Core(TM) i9-12900K with 16 cores and 24 threads, in 2 hours 28 minutes 11sec (including the computation of the pipeline 3D data products checksum), and required 10.35 GB of RAM.

8.4 Observations with an instrumental position angle that is not at zero degrees

When the parameter **cd-matrix-modify** is set to *true* the pipeline pre-corrects a small distortion that is present in the data (pixel size, x- and y-axis scaling, and rotation). The CD matrix transformation parameters have been computed and optimised using observations obtained with an instrumental position angle (PA) set to 0. If, for some reason, the user may want to observe with PA ! = 0, the pipeline may have a less accurate astrometry calibration. In this case the user may want to set the pipeline parameter **cd-matrix-modify** to *false* in the recipe `eris_nix_cal_det`.

8.5 Observations with very few catalogue sources, or close to one another

In a specific User case the ERIS/NIX image had only two sources in the catalogue and one of these was a multiply-lensed quasar. Additionally, the observation was made with PA=180. In this extreme case a good astrometry solution was possible only with **catalogue.bkg.mesh-size** set to 32 (instead than the default 128) in the recipe `eris_nix_img_cal_wcs` (this allowed the recipe to identify each of the four adjacent images within the lensed object), and **cd-matrix-modify** set to false in the recipe `eris_nix_cal_det`.

8.6 Possible issues with the sky subtraction recipe

The sky correction recipe `eris_nix_img_skysub` may occasionally fail. If this occurs, the user should adapt the sky-method to match the object field.

ERIS/NIX object frames are corrected either using dedicated sky observations (when available), or by using the object frame after masking the sources. For each target jitter a pool of sky frames are created from `CAL_DET_OBJECT_JITTER` or `CAL_DET_SKY_JITTER` frames. If sky-method=*collapse-median* the estimate is a median collapse of the sky frame pool. However, this is prone to problems if the field is crowded (see fig. 8.4 for an example of such a field). If sky-method=*median-median* the median of each pool sky frame is obtained and the background frame set to the median of those values. Since the background is forced to be flat, this method can be more robust in crowded fields or in fields where there is nebulosity. However, the *median-median* method will ignore a sloped or subtle structured background. This *first* estimate of the background is subtracted from the frames and then a source detection (`hdrl_catalogue_compute`) is used to find the objects. These sky frame catalogues are used to construct object masks, which are applied to the sky frames to blank out the astronomical sources. A second estimate of the background is then done using the same method as before. Finally, this second background estimate is subtracted from the target frames.

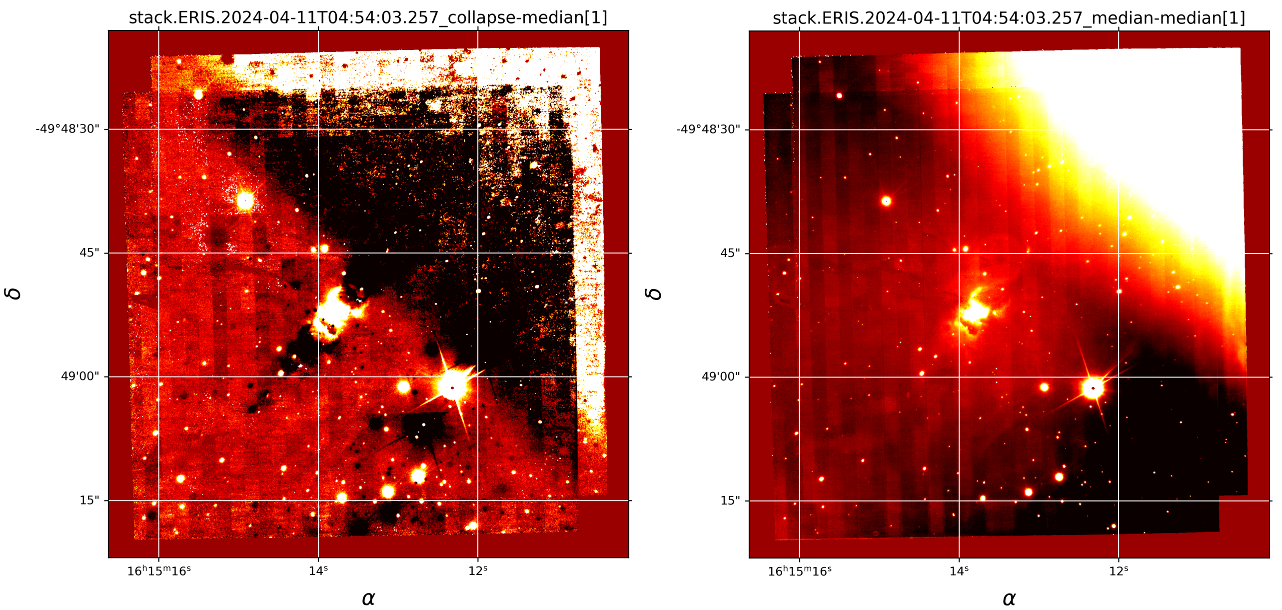


Figure 8.4: A comparison of the `eris_nix_img_skysub` routine when applied to a crowded field. If sky-method=*collapse-median* is used in a crowded field or in a field with extended emission then this can be detected as background and wrongly subtracted (left panel). In this case, using sky-method=*median-median* results in a much improved sky correction (right panel).

9 ERIS Data Description

Each kind of raw frame is typically associated to a single ERIS-NIX pipeline recipe, *i.e.*, the recipe assigned to the reduction of that specific frame type. In the pipeline environment this recipe would be executed automatically. The recipe to compute optical distortions takes as input several kinds of raw frames, including slit-fibre flats, and standard pairs of flat field and arc lamp frames.

In the following sections after a brief description of how most of the raw data look like, all raw ERIS-NIX data frames are listed, together with the keywords used for their classification and correct association. The indicated *DO category* is a label assigned to any data type after it has been classified, which is then used to identify the frames listed in the *Set of Frames*

9.1 Dark

Frame tag: `DARK`
Processed by: `eris_nix_dark`

Classification Keywords	Association Keywords	Remarks
DPR.CATG = CALIB DPR.TYPE = DARK DPR.TECH = IMAGE INSTRUME = ERIS SEQ.ARM = NIX INS1.MODE = nixIMG	DET.DIT TPL.START	

An example is shown in Table 9.1.

9.2 Flat Lamp: ON, OFF

Frame tag: `FLAT_LAMP_ON`
Processed by: `eris_nix_flat_lamp`

Classification Keywords	Association Keywords	Remarks
DPR.CATG = CALIB DPR.TYPE = FLAT,LAMP DPR.TECH = IMAGE INSTRUME = ERIS SEQ.ARM = NIX INS1.MODE = nixIMG	DET.DIT TPL.START	

An example is shown in Table 9.2.

Frame tag: `FLAT_LAMP_OFF`
Processed by: `eris_nix_flat_lamp`

Classification Keywords	Association Keywords	Remarks
DPR.CATG = CALIB DPR.TYPE = FLAT,DARK DPR.TECH = IMAGE INSTRUME = ERIS SEQ.ARM = NIX INS1.MODE = nixIMG	DET.DIT TPL.START	

An example is shown in Table 9.2.

9.3 Twilight Flat

Frame tag: FLAT_TWILIGHT
Processed by: eris_nix_flat_twilight

Classification Keywords	Association Keywords	Remarks
DPR.CATG = CALIB DPR.TYPE = FLAT,TWILIGHT DPR.TECH = IMAGE INSTRUME = ERIS SEQ.ARM = NIX INS1.MODE = nixIMG	DET.DIT TPL.START	

An example is shown in Table 9.3.

9.4 Sky Flat

Frame tag: FLAT_SKY
Processed by: eris_nix_flat_sky

Classification Keywords	Association Keywords	Remarks
DPR.CATG = CALIB DPR.TYPE = FLAT,SKY DPR.TECH = IMAGE INSTRUME = ERIS SEQ.ARM = NIX INS1.MODE = nixIMG	DET.DIT TPL.START	

An example is shown in Table 9.4.

9.5 Object and Sky Jitter

Frame tag: OBJECT_JITTER
Processed by: eris_nix_cal_det, eris_nix_img_skysub, eris_nix_img_cal_wcs, eris_nix_img_cal_phot, eris_nix_img_hdrl_stack, or eris_nix_img_scired

Classification Keywords	Association Keywords	Remarks
DPR.CATG = SCIENCE DPR.TYPE = OBJECT DPR.TECH = IMAGE INSTRUME = ERIS SEQ.ARM = NIX INS1.MODE = nixIMG	DET.DIT TPL.START	

An example is shown in Table 9.5.

Frame tag: SKY_JITTER
Processed by: eris_nix_cal_det, eris_nix_img_skysub, eris_nix_img_cal_wcs, eris_nix_img_cal_phot, eris_nix_img_hdrl_stack, or eris_nix_img_scired

Classification Keywords	Association Keywords	Remarks
DPR.CATG = SCIENCE DPR.TYPE = SKY DPR.TECH = IMAGE INSTRUME = ERIS SEQ.ARM = NIX INS1.MODE = nixIMG	DET.DIT TPL.START	

An example is shown in Table 9.5.

10 Static Calibration Data

10.1 NIX WCS Refinement Data

Frame Tag: `WCS_REFINE`
Classification keywords: `PRO.CATG = WCS_REFINE`

A FITS table for use as a static calibration of the CD matrices. It corrects a slight rotation misalignment in the NIX images.

10.2 Catalog with WCS

Frame Tag: `WCS_MATCHED_CATALOGUE`
Classification keywords: `PRO.CATG = WCS_MATCHED_CATALOGUE`

10.3 Photometric Calibration Table

Frame Tag: `PHOT_DATA`
Classification keywords: `PRO.CATG = PHOT_DATA`

The photometric calibration table file (nix_phot.fits) is used to transform the instrumental magnitudes to a calibrated system. It is a binary FITS table with one extension per system. The name of one extension must match the value of the cdssearch_photom recipe parameter (e.g. '2mass'). A description of the columns for each extension is shown in Table 10.1

10.4 NIX LSS Rectification and Wavelength Calibration Data

Frame Tag: `MASTER_STARTRACE`
Classification keywords: `PRO.CATG = MASTER_STARTRACE`

A FITS table holding polynomials that can be used by `cpl_image_warp_polynomial` to correct a raw 2d LSS spectrum so that slit offset is aligned parallel to the x-axis and dispersion parallel to the y-axis with the wavelength calibrated.

10.5 NIX PUPIL Reference images

Frame Tag: `PUPIL_REF`
Classification keywords: `PRO.CATG = PUPIL_REF`

Column	Name	Type	Description
1	filter_name	char	Name of the filter. It must match the value of the ESO.INS2.NXFW.NAME keyword from the data being calibrated.
2	atm_extcoef	float	The extinction coefficient for airmass of unity for the given filter.
3	mag_offset	float	A scalar value to be added to the instrumental magnitude once the colour equation has been applied.
4	coleq_columns	char	The name(s) of magnitude columns from the photometric standards catalogue to be used in the colour equations, e.g. 'Jmag'. The names must be separated by a comma.
5	coleq_errcols	char	The name(s) of magnitude error columns from the photometric standards catalogue, e.g. 'e_Jmag'. There must be one name for each of the column names given in the coleq_columns field. Any objects with an error in any of the coleq_errcols greater than the value specified by the recipe parameter magerrcut will be excluded from the calibration. The names must be separated by a comma.
6	coleq_coefs	char	The colour equation coefficients. There must be one coefficient for each of the column names given in the coleq_columns field. This should be formatted as a sequence of floating point numbers separated by commas.
7	gal_extcoef	float	Multiply the E(B-V) estimate from the Schlegel maps by gal_extcoef to give the total galactic absorption in filter_name. NOT USED.
8	default_zp	float	If the photometric calibration fails, this default zero point value is used.
9	default_zp_err	float	If photometric calibration fails, this default error on the zero point value is used.

Table 10.1: Structure of the photometric calibration table

These static frames are used in operations to monitor the position of the telescope pupil. They contain the image of the pupil for the following instrument settings: Open1, ND, APP, LYOT, LYON-ND.

10.6 NIX SAM INFO tables

Frame Tag: `SAM_INFO`
Classification keywords: `PRO.CATG = SAM_INFO`
These static tables are used in operations to monitor the position of the telescope pupil mask when in one of thre three possible SAM settings. They contain for each mask hole, the nominal positions of its center and the corresponding radii.

Figure 11.1: The ERIS “Association Map” showing the required and optional input for each recipe.

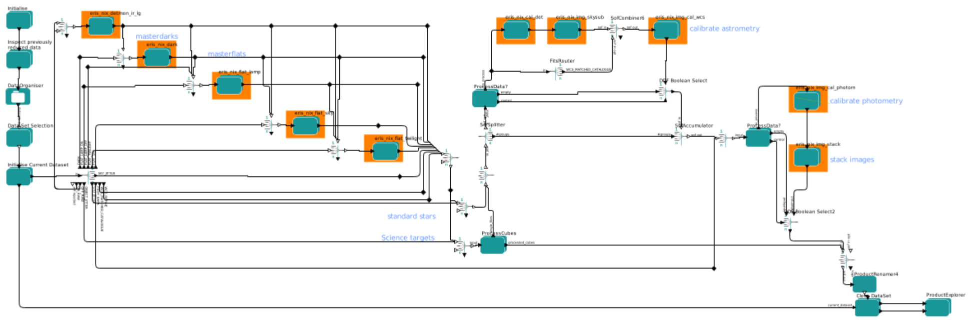


Figure 11.2: The NIX imaging workflow.

11 Data Reduction and algorithm description

11.1 The ERIS Imaging Data Reduction Pipeline

You can use `esoreflex` to assemble a dataset and run it through the recipes or you can assemble the dataset and run the recipes manually. This section describes the manual option in more detail.

Even though a user may not use `esoreflex`, it is instructive to look at the workflow Figure 11.2 to see what needs to be done and in what order.

1. The ‘Data Organisation and Selection’ column chooses the target dataset to reduce and associates it with the calibration data required.

2. In ‘Creation of Master Calibration Products’ recipes are run to reduce the calibration data: linearity correction data products. dark frames and flat-fields.

3. ‘Standard Star and Science Images Reduction’ deals with the reduction of observation data in imaging (’nixIMG’) mode.

11.1.1 Assemble the Dataset

This stage involves identifying and collecting together the raw data needed to reduce the particular observation, including calibration data such as flatfields, darks and linearity information.

A simple way to do this is to run `esoreflex` as far as the `DataChooser` tool. This will collect and show the files required to produce each of your datasets, for example Figure 11.3.

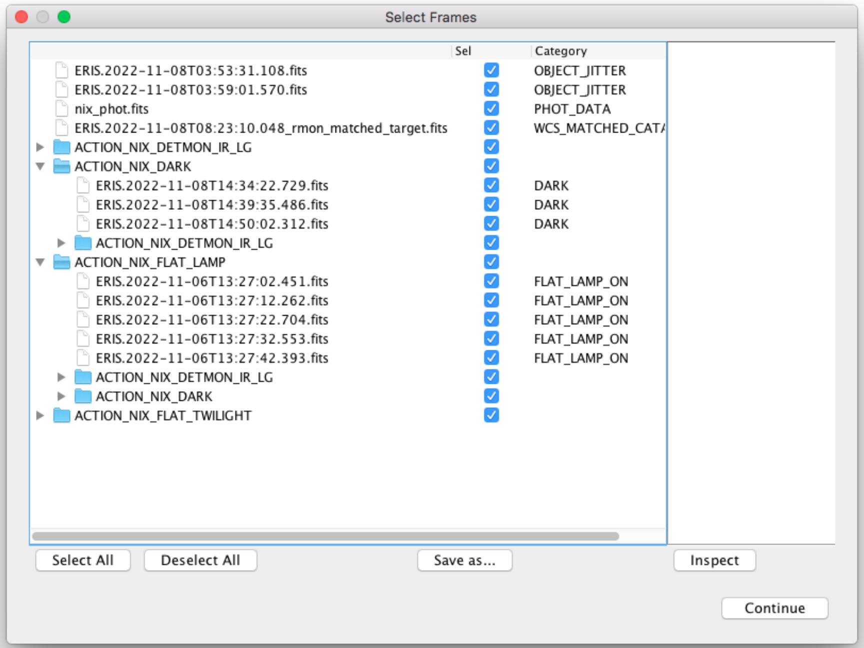


Figure 11.3: Example of a NIX dataset as shown by the esoreflex DataChooser. Not all directories have been expanded.

11.1.2 Reduce the Calibration Data

1. Characterize the linearity and estimate the gain of the detector using recipe `eris_nix_detmon_ir_lg` 12.1. The Linearity reduction part computes the detector response as a function of the pixel intensity, fitting this function to a polynomial whose coefficients help to determine when it becomes non linear. The Gain reduction part computes the Gain of the detector.

2. Calculate the `master_darks` required using recipe `eris_nix_dark`.

   The 'darks' are calculated using `HDRL` routines. They combine the signals due to bias and dark current and are therefore calculated from raw dark frames taken with the same DIT, detector mode and detector window as the measurement to which they are to be applied.

   Recipe `eris_nix_dark` calculates the detector `master_dark` with a SoF file looking like this:

```
eris_nix_detmon_ir_lg_bpm.fits               BP_MAP_NL
eris_nix_detmon_ir_lg_coeffs_cube.fits       COEFFS_CUBE
eris_nix_detmon_ir_lg_linearity_table.fits DET_LIN_INFO
eris_nix_detmon_ir_lg_gain_table.fits        GAIN_INFO
<raw data 1>.fits                            DARK
<raw data 2>.fits                            DARK
<raw data 3>.fits                            DARK
...
```

   The default recipe parameters are:

```
eris.eris_nix_dark.collapse.method="MEDIAN"
eris.eris_nix_dark.collapse.mode.bin-size="0.0"
eris.eris_nix_dark.collapse.mode.error-niter="0"
eris.eris_nix_dark.collapse.mode.histo-max="1.0"
eris.eris_nix_dark.collapse.mode.histo-min="10.0"
eris.eris_nix_dark.collapse.mode.method="MEDIAN"
eris.eris_nix_dark.collapse.sigclip.kappa-high="10.0"
eris.eris_nix_dark.collapse.sigclip.kappa-low="10.0"
eris.eris_nix_dark.collapse.sigclip.niter="3"
eris.eris_nix_dark.hotpix.filter.border="NOP"
eris.eris_nix_dark.hotpix.filter.filter="MEDIAN"
eris.eris_nix_dark.hotpix.filter.kappa-high="5.0"
eris.eris_nix_dark.hotpix.filter.kappa-low="5.0"
eris.eris_nix_dark.hotpix.filter.maxiter="3"
eris.eris_nix_dark.hotpix.filter.smooth-x="11"
eris.eris_nix_dark.hotpix.filter.smooth-y="11"
eris.eris_nix_dark.hotpix.legendre.filter-size-x="11"
eris.eris_nix_dark.hotpix.legendre.filter-size-y="11"
eris.eris_nix_dark.hotpix.legendre.kappa-high="5.0"
eris.eris_nix_dark.hotpix.legendre.kappa-low="5.0"
eris.eris_nix_dark.hotpix.legendre.maxiter="3"
```

```
eris.eris_nix_dark.hotpix.legendre.order-x="3"
eris.eris_nix_dark.hotpix.legendre.order-y="3"
eris.eris_nix_dark.hotpix.legendre.steps-x="21"
eris.eris_nix_dark.hotpix.legendre.steps-y="21"
eris.eris_nix_dark.hotpix.method="FILTER"
```

`collapse` parameters control the algorithm used to 'collapse' the input images in `hdrl_imagelist_collapse`.

`hotpix` parameters control the detection of faulty bright pixels by `hdrl_bpm_2d_compute`.

The recipe produces 1 output:

- The master dark stored as a MEF (PRO.CATG = MASTER_DARK_IMG).

The algorithm for calculating the master dark is as follows:

(a) Read from `ESO.QC.GAIN` in the `GAIN_INFO` file the detector gain (`e-/ADU`) computed by `detmon_ir_lg`.

(b) Use routine `cpl_flux_get_noise_window` to estimate the readout-noise (RON) from the difference of the first 2 `DARK` frames. The algorithm assigns typically 100 random 9x9 windows to the difference frame and calculates the standard deviation of the pixels in each; the RON is the median of these.

(c) Apply the gain to the RON to convert from ADU to e-.

(d) Estimate the error plane on each `DARK` according to detector mode.

FAST_UNCORR Estimate the error directly from the median absolute deviation [MAD] of the DARK frames.

SLOW_GR_UTR From Vacca et al. [RD2] equation 44 with I = 0:

$$variance = 12 * \frac{RON^2}{gain^2 * nr * ndit * dit^2}$$

where *nr* is the number of non-destructive samples in the 'read' (keyword ESO.DET.NDSAMPLES).

(e) Use `hdrl_imagelist_collapse` to calculate the `master_dark`.

(f) Use `hdrl_bpm_2d_compute` to calculate the 'hot' pixel bpm.

Calculation of master darks is normally straighforward. The appearance of the dark depends on detector mode: Figure 11.4 shows the workflow output for mode `SLOW_GR_UTR`, Figure 11.5 that for `FAST_UNCORR`. The dark line at the bottom of the `SLOW_GR_UTR` result was a fault in the commissioning data that has been corrected.

3. Calculate the master flats required required using recipes `eris_nix_flat_lamp` and `eris_nix_twilight_flat` at short wavelengths, or `eris_nix_sky_flat` at long wavelengths (L and M).

The flatfields are calculated by `HDRL` routines that produce a `HIFREQ` result describing pixel-to-pixel variation, and a `LOFREQ` result that measures the detector coupling to the sky through the telescope.

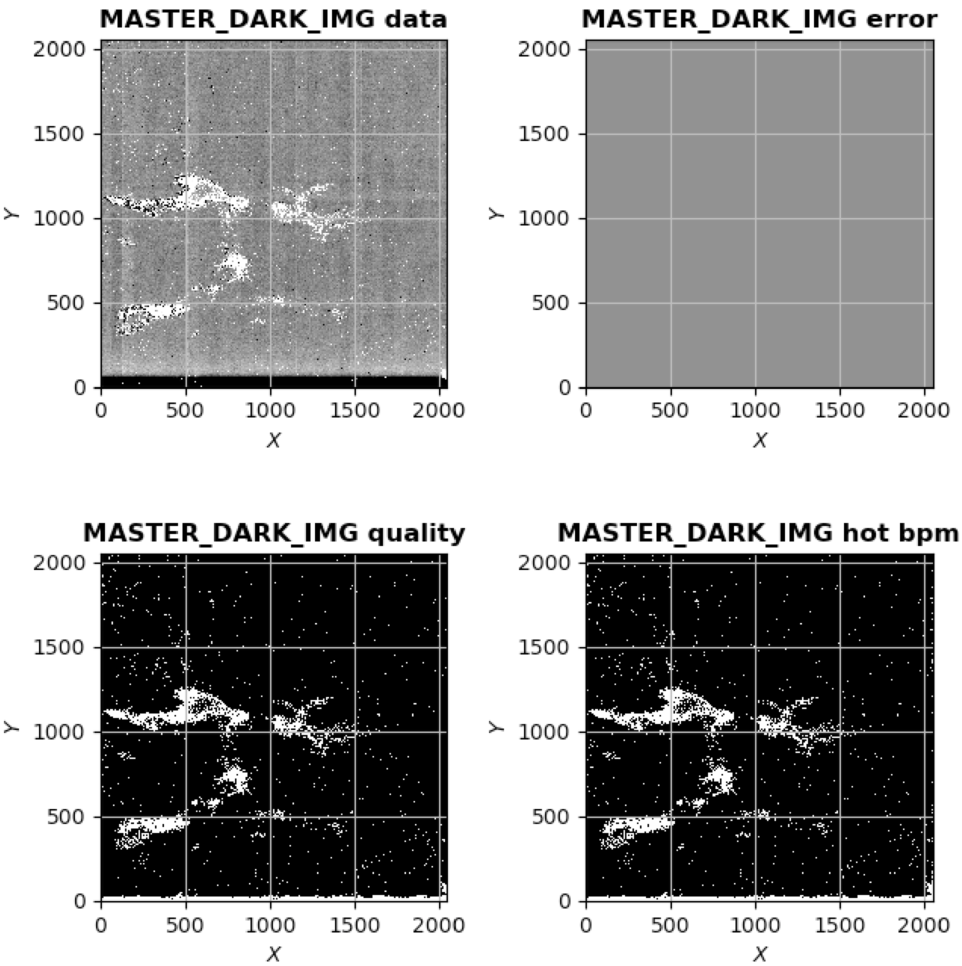


Figure 11.4: Example of a NIX master dark for detector mode SLOW_GR_UTR.

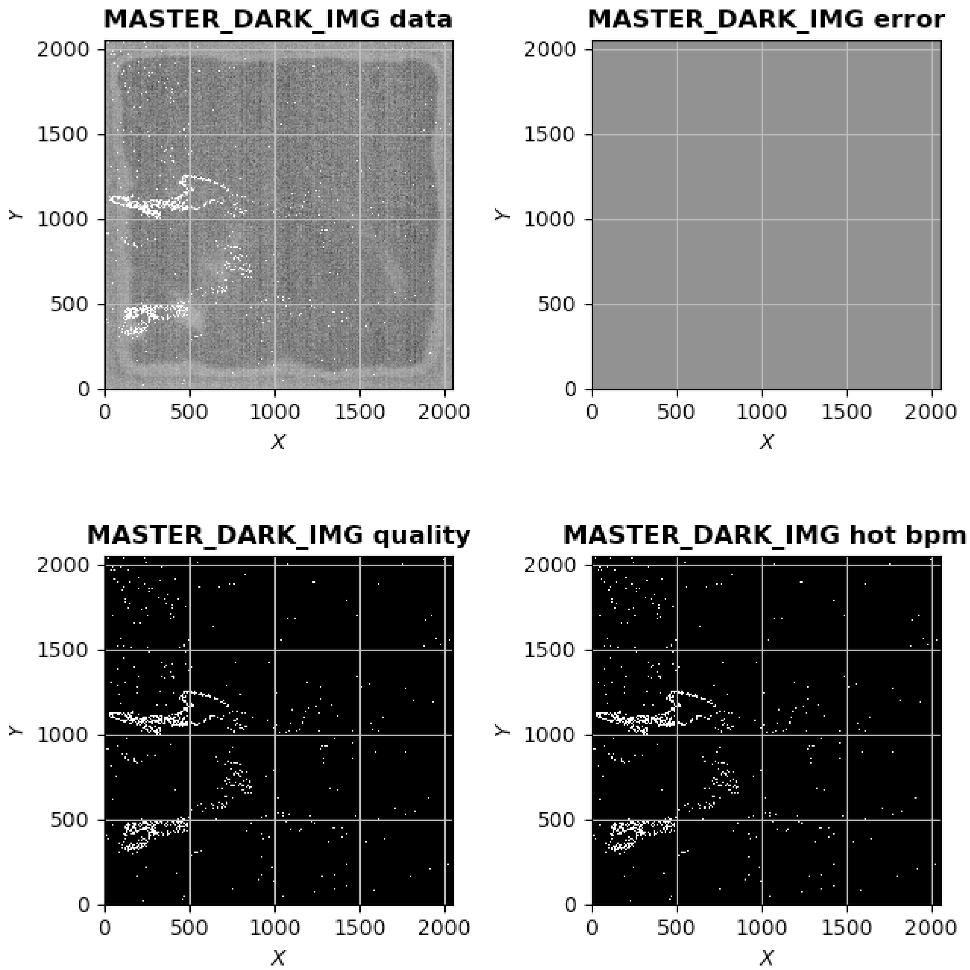


Figure 11.5: Example of a NIX master dark for detector mode FAST_UNCORR.

Lamp and Twilight Flats Recipe `eris_nix_lamp_flat` calculates flatfields from measurements of the calibration lamp, with a SoF file like this:

```
eris_nix_detmon_ir_lg_bpm.fits              BP_MAP_NL
eris_nix_detmon_ir_lg_coeffs_cube.fits      COEFFS_CUBE
eris_nix_detmon_ir_lg_linearity_table.fits  DET_LIN_INFO
eris_nix_detmon_ir_lg_gain_table.fits       GAIN_INFO
master_dark.fits                            MASTER_DARK_IMG
<raw data 1>.fits                           FLAT_LAMP_ON
<raw data 2>.fits                           FLAT_LAMP_ON
<raw data 3>.fits                           FLAT_LAMP_ON
...
<raw off data 1>.fits                       FLAT_LAMP_OFF (optional)
<raw off data 2>.fits                       FLAT_LAMP_OFF (optional)
<raw off data 3>.fits                       FLAT_LAMP_OFF (optional)
...
```

The default recipe parameters are:

```
eris.eris_nix_flat_lamp.coldpix.filter.border="NOP"
eris.eris_nix_flat_lamp.coldpix.filter.filter="MEDIAN"
eris.eris_nix_flat_lamp.coldpix.filter.kappa-high="20.0"
eris.eris_nix_flat_lamp.coldpix.filter.kappa-low="5.0"
eris.eris_nix_flat_lamp.coldpix.filter.maxiter="3"
eris.eris_nix_flat_lamp.coldpix.filter.smooth-x="21"
eris.eris_nix_flat_lamp.coldpix.filter.smooth-y="21"
eris.eris_nix_flat_lamp.coldpix.legendre.filter-size-x="11"
eris.eris_nix_flat_lamp.coldpix.legendre.filter-size-y="12"
eris.eris_nix_flat_lamp.coldpix.legendre.kappa-high="5.0"
eris.eris_nix_flat_lamp.coldpix.legendre.kappa-low="4.0"
eris.eris_nix_flat_lamp.coldpix.legendre.maxiter="6"
eris.eris_nix_flat_lamp.coldpix.legendre.order-x="2"
eris.eris_nix_flat_lamp.coldpix.legendre.order-y="10"
eris.eris_nix_flat_lamp.coldpix.legendre.steps-x="20"
eris.eris_nix_flat_lamp.coldpix.legendre.steps-y="21"
eris.eris_nix_flat_lamp.coldpix.method="FILTER"
eris.eris_nix_flat_lamp.collapse.method="MEDIAN"
eris.eris_nix_flat_lamp.collapse.mode.bin-size="0.0"
eris.eris_nix_flat_lamp.collapse.mode.error-niter="0"
eris.eris_nix_flat_lamp.collapse.mode.histo-max="1.0"
eris.eris_nix_flat_lamp.collapse.mode.histo-min="10.0"
eris.eris_nix_flat_lamp.collapse.mode.method="MEDIAN"
eris.eris_nix_flat_lamp.collapse.sigclip.kappa-high="10.0"
eris.eris_nix_flat_lamp.collapse.sigclip.kappa-low="10.0"
eris.eris_nix_flat_lamp.collapse.sigclip.niter="3"
eris.eris_nix_flat_lamp.flat.filter-size-x="21"
eris.eris_nix_flat_lamp.flat.filter-size-y="21"
```

```
eris.eris_nix_flat_lamp.flat.method="high"
eris.eris_nix_flat_lamp.min_coadds="1"
```

`coldpix` parameters control the detection of unresponsive pixels by `hdrl_bpm_2d_compute`.
`collapse` parameters control the algorithm used to 'collapse' the input images in `hdrl_flat_compute`.
`flat` parameters control the smoothing used by `hdrl_flat_compute` to split the `HIFREQ` and `LOFREQ` results.
`min_coadds` sets the minimum number of lamp images required for the calculation to proceed.

The recipe produces 3 outputs:

- The HIFREQ flatfield MEF (PRO.CATG=`MASTER_FLAT_LAMP_HIFREQ`)
- The LOFREQ flatfield MEF (PRO.CATG=`MASTER_FLAT_LAMP_LOFREQ`)
- The master BPM MEF (PRO,CATG=`MASTER_BPM_LAMP`)

The algorithm for calculating the flatfields is as follows:

(a) Subtract a 'master dark' from each input `FLAT_LAMP_ON`, either by using a matching `FLAT_LAMP_OFF` or the `MASTER_DARK_IMG`.
(b) Use the SoF 'xxx_detmon_xxx' files to linearize the results.
(c) Use `hdrl_flat_compute` to calculate an initial LOFREQ flatfield for the purpose of obtaining an 'unilluminated' bpm; areas around the edge of the NIX detector that are not illuminated by the optics. Apply this bpm to reject pixels in the input data that would otherwise corrupt the LOFREQ result.
(d) Use `hdrl_flat_compute` to calculate the final HIFREQ result.
(e) Use `hdrl_bpm_2d_compute` to calculate the 'cold' pixel bpm.
(f) Create the `MASTER_BPM_LAMP` by combining:
The 'cold' bpm.
The 'hot-pixel' bpm associated with the `MASTER_DARK_IMG`.
The 'unilluminated' bpm.
The 'non-linear' bpm from `BP_MAP_NL`.
(g) Calculate a `CONFIDENCE` array from the `HIFREQ` flatfield. This is essentially a pixel weights array based on the flatfield intensity, normalised to a median value of 100.
(h) Apply `MASTER_BPM_LAMP` to reject bad pixels from the input then use `hdrl_flat_compute` to calculate the `LOFREQ` result.
(i) Calculate a `CONFIDENCE` array to go with the `LOFREQ` result.

Figure 11.6 shows the workflow output for a lamp flatfield. The 'MASTER_FLAT data' pane shows the flatfield result. 'cold_bpm' shows the unresponsive pixels - not including those that were already rejected as 'hot' or 'non-linear'. The 'quality' pane shows all bad pixels; unilluminated areas of the detector can be seen around its edges - the illuminated area looks like a rectangle that has been rotated slightly in the frame.

Recipe `eris_nix_twilight_flat` calculates the NIX `LOFREQ` flatfield from measurements of the brightening or darkening twilight sky, with a SoF that looks like this:

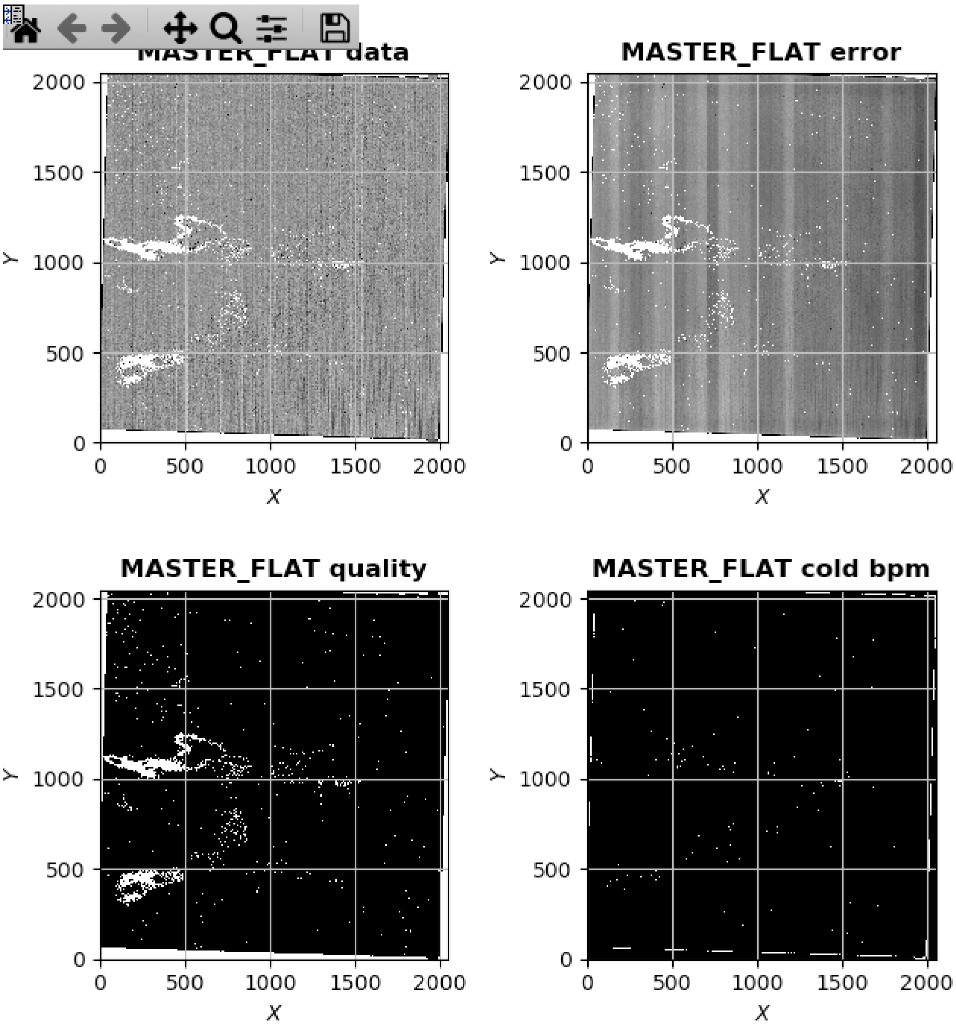


Figure 11.6: Example of a lamp HIFREQ flatfield.

```
eris_nix_detmon_ir_lg_bpm.fits                BP_MAP_NL
eris_nix_detmon_ir_lg_coeffs_cube.fits        COEFFS_CUBE
eris_nix_detmon_ir_lg_linearity_table.fits DET_LIN_INFO
eris_nix_detmon_ir_lg_gain_table.fits         GAIN_INFO
master_dark.fits                              MASTER_DARK_IMG
master_bpm_lamp.fits                          MASTER_BPM_LAMP
<raw data 1 airmass bin 1>.fits               FLAT_TWILIGHT
<raw data 2 airmass bin 1>.fits               FLAT_TWILIGHT
<raw data 3 airmass bin 1>.fits               FLAT_TWILIGHT
...
<raw data 1 airmass bin 2>.fits               FLAT_TWILIGHT
<raw data 2 airmass bin 2>.fits               FLAT_TWILIGHT
<raw data 3 airmass bin 2>.fits               FLAT_TWILIGHT
...
```

The default recipe parameters are:

```
eris.eris_nix_flat_twilight.flat.filter-size-x="21"
eris.eris_nix_flat_twilight.flat.filter-size-y="21"
eris.eris_nix_flat_twilight.flat.method="low"
eris.eris_nix_flat_twilight.min_frames="2"
```

`flat` parameters control the smoothing used by `hdrl_flat_compute` to calculate the `LOFREQ` result.

`min_frame` specifies the minimum number of lamp images needed for the calculation to proceed.

The recipe produces a single output:

- The LOFREQ flatfield MEF (PRO.CATG=`MASTER_FLAT_TWILIGHT_LOFREQ`.

The algorithm for calculating twilight flatfields is as follows:

(a) Subtract the `MASTER_DARK_IMG` from each frame.
(b) Use the 'xxx_detmon_xxx' files to linearize the data.
(c) Apply the `MASTER_BPM_LAMP` to each frame.
(d) Fit straight lines to each pixel's variation through the frames of the observation.
(e) Use `hdrl_flat_compute` on the slopes of the fitted lines to calculate the `LOFREQ` result.
(f) Calculate a `CONFIDENCE` array to go with the `LOFREQ` result.

Figure 11.7 shows the workflow output for a twilight flatfield. The 'MASTER_FLAT data' pane shows the flatfield result.

Sky Flats Recipe `eris_nix_sky_flat` calculates the detector flatfield using measurements of the sky background, with a SoF file like this:

```
eris_nix_detmon_ir_lg_bpm.fits                BP_MAP_NL
eris_nix_detmon_ir_lg_coeffs_cube.fits        COEFFS_CUBE
```

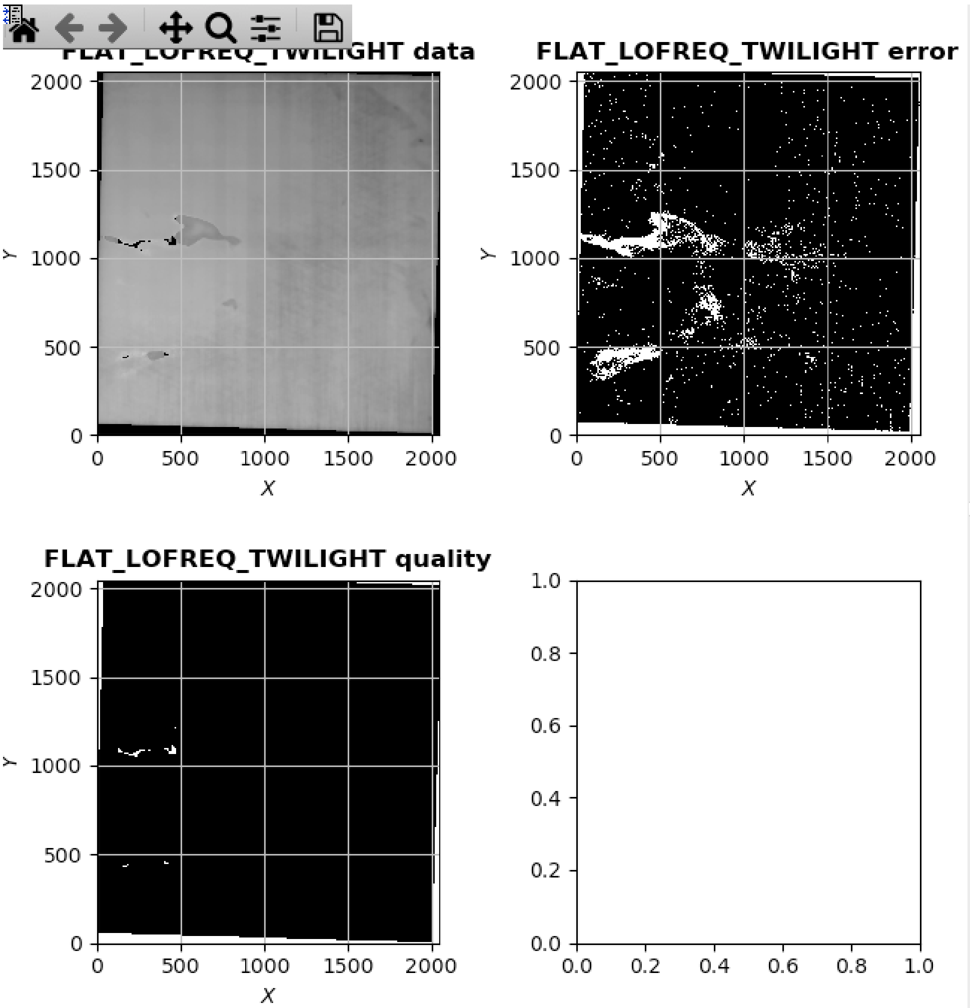


Figure 11.7: Example of a twilight LOFREQ flatfield.

```
eris_nix_detmon_ir_lg_linearity_table.fits DET_LIN_INFO
eris_nix_detmon_ir_lg_gain_table.fits      GAIN_INFO
master_dark.fits                           MASTER_DARK_IMG
<raw data 1>.fits                          FLAT_SKY
<raw data 2>.fits                          FLAT_SKY
<raw data 3>.fits                          FLAT_SKY
...
```

The default recipe parameters are similar to those for recipe `eris_nix_flat_lamp`.

The recipe reduces a set of sky image frames taken at 2 different airmasses to produce the flatfield results. The algorithm is as follows:

(a) Divide the input frames into 2 airmass bins; the observing template should give an equal number in each bin.

(b) Go through the all frames: subtract the `MASTER_DARK_IMG` from each; use the 'xxx_detmon_xxx' files to linearize them.

(c) Pair-wise subtract the low airmass frames from the high airmass.

(d) Use `hdrl_flat_compute` on the difference images to calculate the HIFREQ result.

(e) Use `hdrl_bpm_2d_compute` to calculate the 'cold' pixel bpm.

(f) Create the `MASTER_BPM_SKY` by combining:
The 'cold' bpm.
The 'hot-pixel' bpm associated with the `MASTER_DARK_IMG`.
The 'non-linear' bpm in `BP_MAP_NL`.

(g) Calculate a `CONFIDENCE` array from the `HIFREQ` flatfield.

(h) Apply `MASTER_BPM_SKY` to reject bad pixels from the input then use `hdrl_flat_compute` to calculate the LOFREQ result.

(i) Calculate a `CONFIDENCE` array to go with the `LOFREQ` result.

The calculation of the flats and confidence arrays is similar to the lamp flat case, except that the unilluminated areas of the detector are not yet explicitly flagged.

Figure 11.8 shows the workflow output for a sky flatfield.

Whatever method is used the `HIFREQ` flatfield should look fairly flat with values fluctuating between roughly 0.95 and 1.05. The `LOFREQ` result should be similar but smoother.

If the flatfields look odd then suspect the raw data. Look through the input frames manually to check that the images are sensible. For 'sky' flats check that the data do indeed fall into 2 clearly separate airmass bins - use the telescope elevation for this; keyword ESO.TEL.ALT.

11.1.3 Reduce the Target Data

1. Remove the detector signature from the data, using recipe `eris_nix_cal_det` with a SoF like this:

```
eris_nix_detmon_ir_lg_bpm.fits             BP_MAP_NL
eris_nix_detmon_ir_lg_coeffs_cube.fits     COEFFS_CUBE
eris_nix_detmon_ir_lg_linearity_table.fits DET_LIN_INFO
```

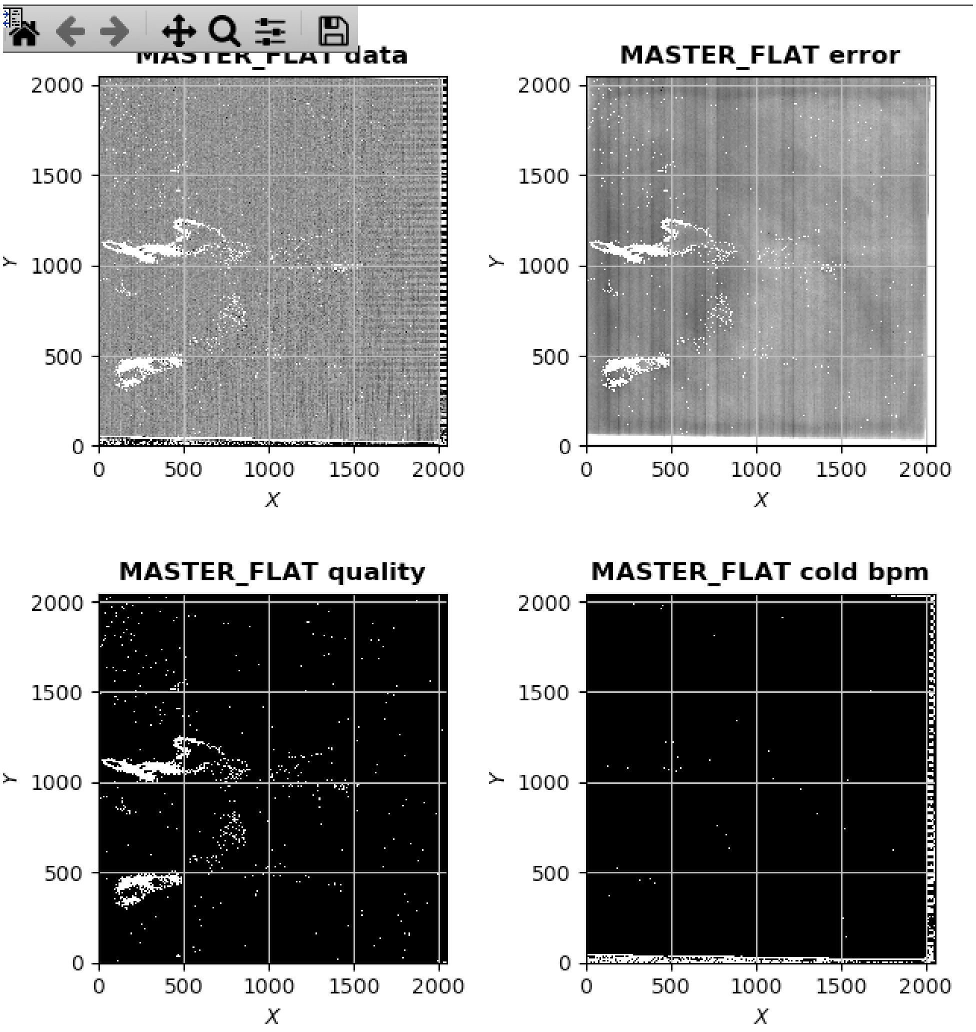


Figure 11.8: Example of a sky HIFREQ flatfield.

```
eris_nix_detmon_ir_lg_gain_table.fits       GAIN_INFO
master_bpm_lamp.fits                        MASTER_BPM_LAMP
master_dark.fits                            MASTER_DARK_IMG
master_flat_lamp_hifreq.fits                MASTER_FLAT_LAMP_HIFREQ
master_flat_lamp_lofreq.fits                MASTER_FLAT_LAMP_LOFREQ
nix_refine_wcs.fits                         WCS_REFINE
<raw data 1>.fits                           OBJECT_JITTER (optional)
<raw data 2>.fits                           OBJECT_JITTER (optional)
...
<raw sky data 1>.fits                       SKY_JITTER (optional)
<raw sky data 2>.fits                       SKY_JITTER (optional)
...
<raw std data 1>.fits                       STD_JITTER (optional)
<raw std data 2>.fits                       STD_JITTER (optional)
...
```

The various calibration files are compulsory and can be applied to a range of raw data files: target objects, standards or sky backgrounds.

Default recipe parameters are:

```
eris.eris_nix_cal_det.cd_matrix_modify="true"
eris.eris_nix_cal_det.fill-rejected="set_value"
eris.eris_nix_cal_det.fill-value="0.0"
```

`cd_matrix_modify` specifies whether to update the CD matrix component of the WCS from the `WCS_REFINE` file. Please note that the coefficients of the CD matrix have been calibrated with observations obtained with Position Angle (PA) set to 0. If the user observes with a different value of PA, it is better to deactivate the corresponding correction by setting the parameter cd-matrix-modify to false.

`fill-rejected` and `fill-value` determine whether pixels that are rejected for any reason are left with their raw value, or set to NaN, or set to some other value in this case 0. Setting them to 0 makes them easier to spot when viewing the data.

The recipe outputs detector-calibrated jitter files:

```
cal_det.<raw data 1>.fits              CAL_DET_OBJECT_JITTER
cal_det.<raw data 2>.fits              CAL_DET_OBJECT_JITTER
cal_det.<raw data 3>.fits              CAL_DET_OBJECT_JITTER
...
```

The algorithm for removing the detector signature is as follows:

(a) If required, update the WCS CD-matrix with the version in the `WCS_REFINE` file.

(b) Correct the WCS to compensate for detector 'windowing'. The FITS WCS keywords have been written assuming the full frame of the detector was used. If it was windowed then `CRPIX1` and `CRPIX2` must be adjusted:

$$CRPIX1 = CRPIX1_{old} - ESO.DET.SEQ1.WIN.STRX + 1$$

$$CRPIX2 = CRPIX2_{old} - ESO.DET.SEQ1.WIN.STRY + 1$$

(c) Subtract read-offsets. Detector read-out is implemented in groups of 64 columns. A low level random offset on each column group appears as broad, faint vertical stripes on images. Here, unilluminated rows at top and bottom are used to measure the offsets for subtraction from the image. If the detector window does not cover any of the unilluminated rows then no correction will be done.

(d) Subtract the `MASTER_DARK_IMG`.

(e) Use the xxx_detmon_xxx files to linearize the data.

DETMON has previously been used to fit a curve to data relating calibrator lamp signal to DIT, of form:

$$s_{obs} = a + bt + ct^2 + dt^3$$

where a should be small because the curve passes near the origin.

A linear detector response would be:

$$s_{lin} = bt$$

We want $s_{lin} = s_{obs} * correction$

$$= s_{obs} * \frac{bt}{a + bt + ct^2 + dt^3}$$

for t from $s_{obs} = a + bt + ct^2 + dt^3$.

A pixel linearization correction is applied if:

- The pixel is not already rejected.
- The linearity curve fit is good.
- The curve linear coefficent is greater than 0.
- The raw pixel is greater than 0.

If the raw pixel value is greater than the saturation associated with the DETMON curve calculation, then it is reset to the saturation level and then linearized. The aim is to keep the image smooth.

(f) Estimate the variance on each pixel. Following Vacca et al. [RD2] this is done by summing estimates of the photon noise and readout noise:

RD2 measures intensity in units of adu/s, so we first divide the raw data by DIT.

For 'continuous' sampling (`ESO.DET.READ.CURNAME= 'SLOW_GR_UTR'`) the appropriate formula is as follows (RD2, equation 56).

$$V_I single = 1.2 \frac{I}{g * nr * dit} \frac{nr^2 + 1}{nr + 1} + \frac{12\sigma_{RON}}{g^2 * nr * dit^2} \frac{nr - 1}{nr + 1}$$

The equation does not account for the integration being repeated NDIT times, so $V_I = V_I single/ndit$.

For `ESO.DET.READ.CURNAME= 'FAST_UNCORR'`, calculate the variance as a special case of RD2's CDS treatment. With $nr = 1$, their equation 44 becomes:

$$V_I = \frac{I}{g * ndit * dit} + \frac{\sigma^2_{RON}}{g^2 * ndit * dit^2}$$

The 2 in the numerator of the readout noise component of RD2 eqn.44 derives from the 2 readouts in CDS ('pedestal' and 'signal') whereas FAST_UNCORR only has 'signal'. Consequently we replace 2 by 1.

There presumably is some error associated with the reset at the start of each integration but that is ignored.

In these calculations g comes from the master dark, $nr = ESO.DET.NSAMPLES$, $dit = ESO.DET.SEQ1.DIT$ and $ndit = ESO.DET.SEQ1.DIT$.

(g) Divide by the `HIFREQ` flatfield. If lamp calibration is being used (signified by the specification of `MASTER_BPM_LAMP`). divide by `MASTER_FLAT_LAMP_HIFREQ`. If sky flats are being used (signified by the specification of `MASTER_BPM_SKY`) divide by `MASTER_FLAT_SKY_HIFREQ`.

(h) Divide by the `LOFREQ` flatfield. If lamp calibration is being used look for a `MASTER_FLAT_TWILIGHT_LOFREQ` first, failing that use the `MASTER_FLAT_LAMP_LOFREQ`. If it's sky flats then use `MASTER_FLAT_SKY_LOFREQ`.

(i) Associate with the bad-pixel-mask, `MASTER_BPM_LAMP` or `MASTER_BPM_SKY` as appropriate.

(j) Set the value of rejected pixels according to `fill-rejected` and `fill-value`.

This is your first chance to look at cleaned up images. Check that they look reasonable, and that they all look similar. For example, you will notice if some measurements were spoiled by clouds or poor seeing.

2. Estimate and subtract the sky background using `eris_nix_img_skysub` with a SoF like this:

```
cal_det.<raw data 1>.fits          CAL_DET_OBJECT_JITTER
cal_det.<raw data 2>.fits          CAL_DET_OBJECT_JITTER
cal_det.<raw data 3>.fits          CAL_DET_OBJECT_JITTER
...
cal_det.<raw data n+1>.fits        CAL_DET_SKY_JITTER (optional)
cal_det.<raw data n+2>.fits        CAL_DET_SKY_JITTER (optional)
cal_det.<raw data n+3>.fits        CAL_DET_SKY_JITTER (optional)
...
```

Default recipe parameters are:

```
eris.eris.eris_nix_img_skysub.catalogue.bkg.mesh-size="64"
eris.eris.eris_nix_img_skysub.catalogue.bkg.smooth-gauss-fwhm="2.0"
eris.eris.eris_nix_img_skysub.catalogue.obj.min-pixels="20"
eris.eris.eris_nix_img_skysub.catalogue.obj.threshold="3.0"
eris.eris.eris_nix_img_skysub.sky-bracket-time="3600.0"
```

```
eris.eris.eris_nix_img_skysub.sky-method="median-median"
eris.eris.eris_nix_img_skysub.sky-selector="bracket"
eris.eris.eris_nix_img_skysub.sky-source="auto"
```

- `catalogue` parameters are passed to `hdrl_catalogue_compute` to help calculate an object mask as part of the background calculation.
- `sky-source` specifies which data to use for calculating the sky background:
  - `target`: use the `CAL_DET_OBJECT_JITTER` frames,
  - `offset`: use the `CAL_DET_SKY_JITTER` frames,
  - `auto`: uses the sky offset frames if available otherwise use the target frames.
- `sky-selector` is set to `bracket` to specify that sky frames used for the sky background should bracket the time of each target frame, `sky-bracket-time` specifies the width of that bracket (period/range of time).

The recipe outputs sky-subtracted jitter files:

```
skysub.<raw data 1>.fits            SKYSUB_OBJECT_JITTER
skysub.<raw data 2>.fits            SKYSUB_OBJECT_JITTER
skysub.<raw data 3>.fits            SKYSUB_OBJECT_JITTER
...
```

The sky subtraction is performed as follows:

(a) For each target jitter construct a pool of ‘sky’ frames from `CAL_DET_OBJECT_JITTER` or `CAL_DET_SKY_JITTER` frames according to `sky-source` and `bracket-time`.

(b) Make a first estimate of the background.

If `sky-method='median-median'` the median of each pool sky frame is obtained and the background frame set at the median of those values. The background is forced to be flat, which is more robust in crowded fields or where there is nebulosity, but which will miss slope or subtle structure in the background.

If instead the user does not have a crowded field and is keen on removing a vertical stripe pattern left in the data after detector calibration, the `sky-method='collapse-median'` is a better choice, as that method preserves the stripes in the background image and removes them when the sky is subtracted. Remember that the width of vertical stripes is 64 columns (point 1.c on page 51) so parameter `bkg.mesh-size` should be 64 or smaller to be able to correct that pattern properly. Figure 11.12 shows this method being used to remove an optical ghost that is present in both target and background images.

(c) Subtract the first-estimate backgrounds from the sky frames, and derive masks that cover any objects detected in the result.

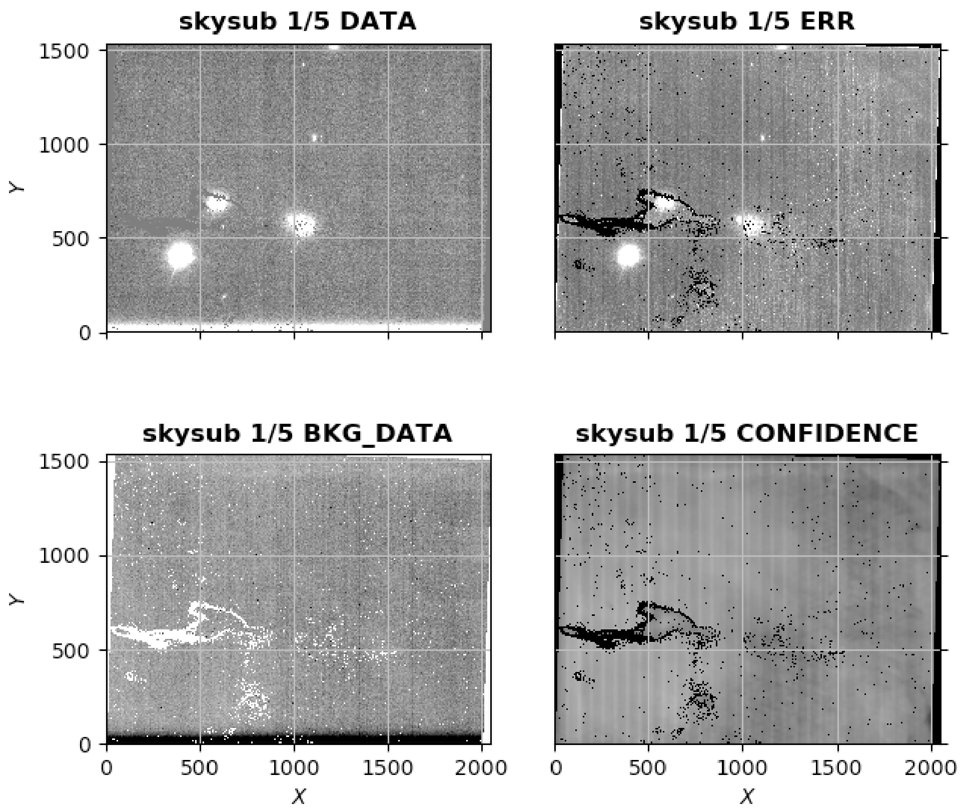


Figure 11.9: An example of a successful sky background subtraction.

The mask calculation depends on parameters `obj.min-pixels`, `obj.threshold`, `bkg.smooth-gauss-fwhm` and `bkg.mesh-size` (whose value also depends on parameter `ao-params`). If the user wants to mask only the largest and brightest objects, the parameters `obj.min-pixels` and `obj.threshold` may need adjustment. Parameter `bkg.smooth-gauss-fwhm` is used to smooth the PSF of each observed object in the definition of the object mask.

(d) Apply the masks to the sky frames to blank out astronomical emission and construct a second estimate of the background using the same method as before.

(e) Subtract the estimated background from the target frames.

Figure 11.9 shows the workflow display for a sky-subtracted jitter image. The DATA frame shows the sky-subtracted result and BKG_DATA the sky background used. Note there is no astronomical structure visible in the background, any such would indicate a failure in the algorithm and a problem.

Figure 11.10 shows a different result illustrating one of the difficulties that can arise. The jitter pattern for this observation had zero throw - that is there was no offset between images. Estimating the sky background from the median of frames cannot work in this case because the stars fall on the same array pixels for all jitters. The effect of this can be seen in BKG_DATA which is now full of stars whereas the sky-subtracted image DATA is empty.

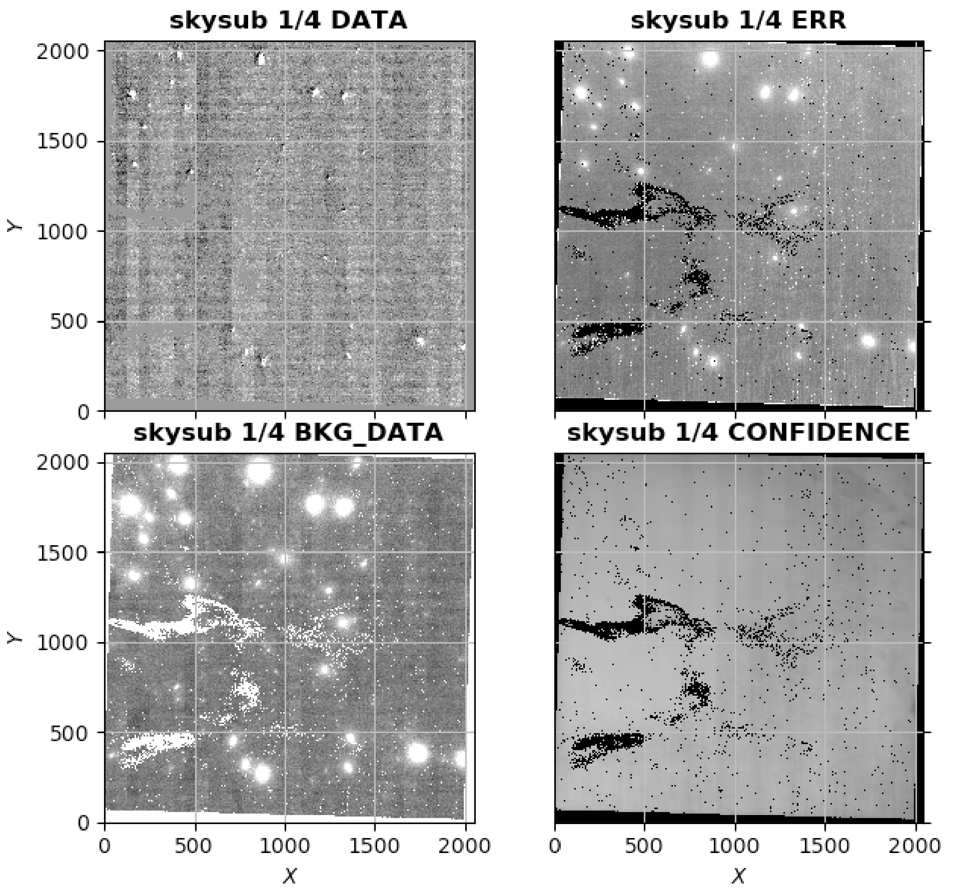


Figure 11.10: An example of a bad sky background where it (BKG_DATA) is full of stars but the sky subtracted image (DATA) is blank.

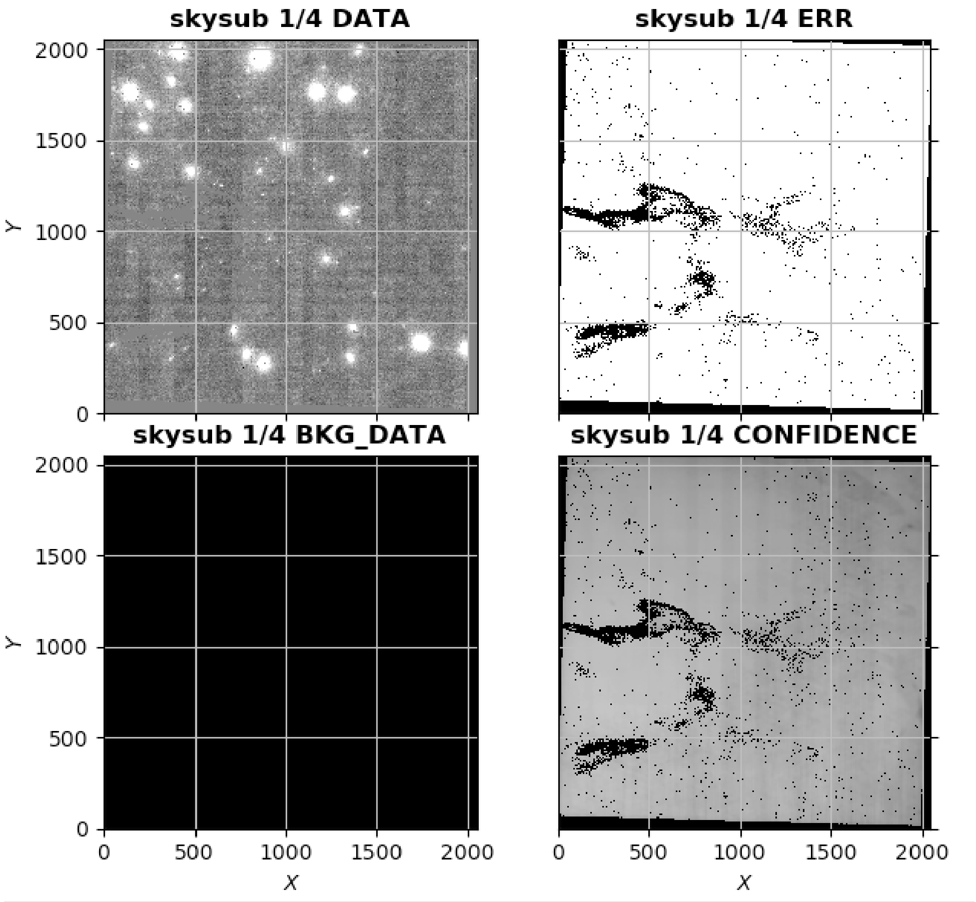


Figure 11.11: Using `sky-method='median-median'` to force a good sky background (the BKG plot appears black because that is how this plot displays a flat image).

Fix this by re-running the recipe with `sky-method='median-median'`, which sets the background to the median of the image medians. Figure 11.11 shows the result; the sky background is flat and stars have reappeared in the image.

Figure 11.12 illustrates the removal of an optical ghost that was present in all images of an observation. Here BKG_DATA shows the optical ghost - the faint circle just above and to the right of the image centre. It is a ghost rather than some seepage of the galaxy image through the sky-background algorithm because it stays in the same place for all jitters. Because it is static it subtracts out well as part of the background.

3. Calibrate the WCS (world coordinate system) of the target images. The raw telescope pointing is not accurate to the resolution of NIX so the WCS of each jitter exposure must be refined. This is essential if sources are to be reliably associated with catalogues for photometric calibration, and if the jitter images are to stack properly. We would like to remind users that the WCS calibration requires to have (in each frame) at least three well observed objects of sources present in the observed catalogue to work. Thus

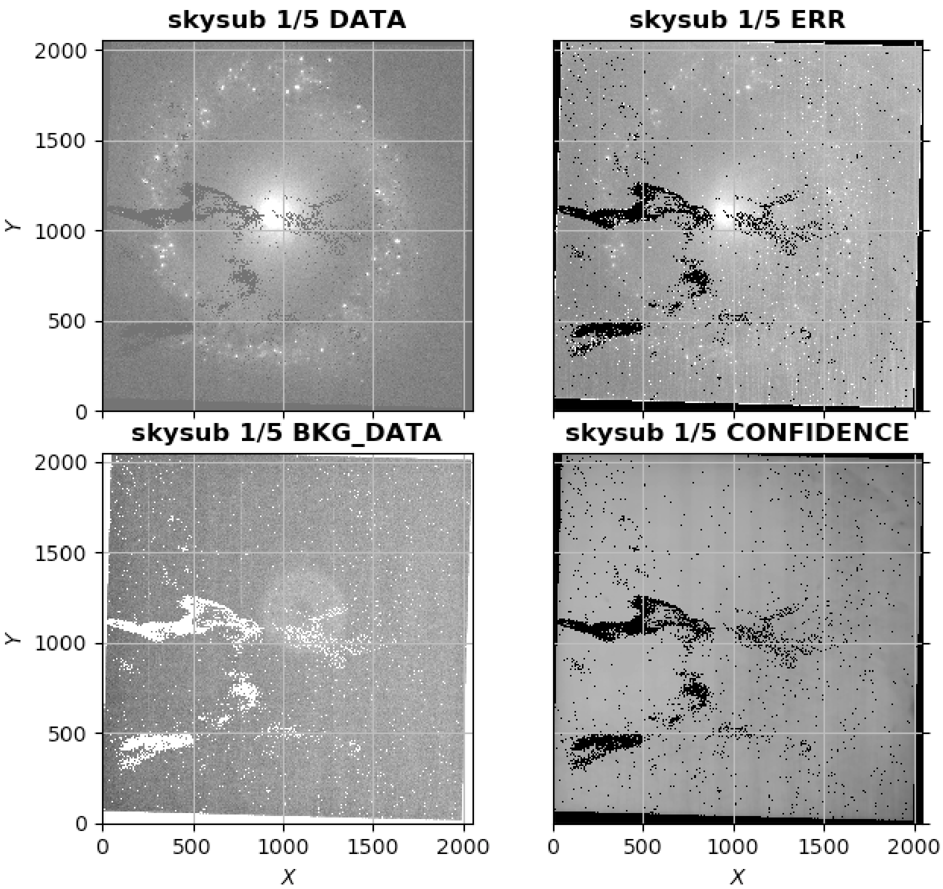


Figure 11.12: An optical ghost in the background.

it is up to the user to define a proper observing sequence. The pipeline parameters have been optimised on cases with several sources in the field of view. In case a user has only three or a few more sources, some parameters may need adjustment in order to obtain a successful and accurate calibration.

As part of the correction of the instrumental signature (`eris_nix_cal_det`) an external fits table (`WCS_REFINE`) is used to provide a *starting point* distortion correction to each image WCS. This correction takes into account the linear anamorphic magnification (0.2% stretch along columns and a 0.2% squeeze along rows) and the 0.39 degree rotation of the field. This correction is applied to the 13mas-JHK, 27mas-JHK, and 13mas-LM camera modes. This correction is applied if the parameter `cd-matrix-modify` of recipe `eris_nix_cal_det` is set to true, which is the default. There are a few cases, usually observations where only a few reference catalogue objects are in the field of view were a higher accuracy of (or even a successful) WCS calibration can be found deactivating such correction, by setting `cd-matrix-modify` to false.

The WCS offsets are then calibrated using the recipe `eris_nix_img_cal_wcs` with a SoF file like:

```
skysub.<raw data 1>.fits           SKYSUB_OBJECT_JITTER
skysub.<raw data 2>.fits           SKYSUB_OBJECT_JITTER
skysub.<raw data 3>.fits           SKYSUB_OBJECT_JITTER
...
matched_target.fits                WCS_MATCHED_CATALOGUE (optional)
```

Default recipe parameters are:

```
eris.eris_nix_img_cal_wcs.catalogue.bkg.estimate="true"
eris.eris_nix_img_cal_wcs.catalogue.bkg.mesh-size="64"
eris.eris_nix_img_cal_wcs.catalogue.bkg.smooth-gauss-fwhm="2.0"
eris.eris_nix_img_cal_wcs.catalogue.det.effective-gain="3.0"
eris.eris_nix_img_cal_wcs.catalogue.det.saturation="5000.0"
eris.eris_nix_img_cal_wcs.catalogue.obj.core-radius="5.0"
eris.eris_nix_img_cal_wcs.catalogue.obj.deblending="true"
eris.eris_nix_img_cal_wcs.catalogue.obj.min-pixels="20"
eris.eris_nix_img_cal_wcs.catalogue.obj.threshold="10.0"
eris.eris_nix_img_cal_wcs.cdssearch_astrom="gaiaedr3"
eris.eris_nix_img_cal_wcs.debug-data="false"
eris.eris_nix_img_cal_wcs.pixel_radius="100.0"
eris.eris_nix_img_cal_wcs.strict_classification="true"
```

- If requested, (`bkg.estimate=true`) the possibly-varying sky background is estimated and removed automatically, prior to object detection, using a combination of robust iteratively-clipped estimators. Any variation in sky level over the frame is dealt with by forming a coarsely sampled background map grid (of size specified by `bkg.mesh-size`). Within each background grid pixel, an iteratively k-sigma clipped median value of 'sky' is computed based on the flux values within the grid pixel zone. A robust estimate of sigma is computed by using the Median of the Absolute Deviation (MAD) from the median. This will then be further processed to form the frame background image.

After removing the varying background component, a similar robust estimate of the average sky level and sky noise per pixel is made. This allows to robustly obtain the detection threshold for object analysis.

- Individual objects are detected using a "standard match" filter approach. Since the only sources difficult to locate are those marginally above the sky noise, to assume (after factoring in the confidence map information) the noise as uniform, is a good approximation. The majority of objects detected in this way will have a shape dominated by the point spread function (PSF), which thereby defines the filter to use (controlled by `bkg.smooth-fwhm`. This means that the parameter `bkg.smooth-fwhm` has to be set to a value close to the observed object PSF. This value is set by the instrument spatial resolution on sky, defined by INS2.NXCW.NAME and INS2.NXIS.NAME FITS keywords, and observing seeing conditions (TEL.AMBI.FWHM.START/END), values.
- `catalogue` parameters are used by `hdrl_catalogue_compute` when cataloguing objects in each jitter image.
- `cdssearch_astrom` specifies 'Gaia' as the reference catalogue and should generally be left at that, Gaia being the only catalogue with positions accurate to the resolution of NIX.
- The `deblending` parameter, when set to true, activates a deblending algorithm used to disentangle overlapping objects.
- To be detected an object should consist of at least `min-pixels` above a threshold level, that is controlled by `threshold` parameter multiplied by the rms computed on the sky frame. Thus if an image contains several objects and the user would like to constrain the calibration to the brightest objects with larger area, these two parameters may need to be changed from default values.
- A user may also set the value of the detector effective gain, `effective-gain`, which controls the estimation of the error and thus of the noise and consequently the clipping of outliers. This should have a nominal value corresponding to the usual one valid for the detector.
- Another value that may be adjusted is the one of the `det_saturation` in order to exclude saturated pixels.

The WCS is corrected in one of 2 ways:

Cross-reference with Gaia as follows:

(a) Analyse the jitter pattern and find the jitter closest to pattern centre. This will be referred to as the *reference* jitter; it has the maximum overlap with other jitters in the sequence and the best chance of stitching them together effectively.

(b) Catalogue the sources in the *reference* jitter, using `hdrl_catalogue_compute`.

(c) Cross-match the *reference* jitter catalogue with Gaia sources expected in the field.
The catalogues are matched by trying all possible associations between their objects, and noting for each the number of object pairs found to lie within a small distance of each other. The hope is that the correct trial association will 'click' into place, signalled by having the highest number of object pairs. This trial association is selected.
The catalogues are sorted in order of decreasing brightness using column `'Aper_flux_3'`, if it exists. This means that if so few sources are present that multiple associations are equally plausible then the match between the brightest will be favoured. To guard against crowded fields overwhelming the matching algorithm only the 100 brightest objects in each catalogue are used.

Matched objects are written to an output catalogue with both sets of cartesian coordinates. This is used to calculate the median x and y pixel offsets between image sources and Gaia, which information goes to update the FITS WCS of the image.

(d) The code works through the other jitters in the sequence using a similar process to match their sources with those in the *reference* jitter and correct their WCS accordingly. Because the catalogues being associated are now directly comparable in sensitivity and wavelength this matching process is more robust.

Manual Specification Where there are no Gaia sources in the field or where the automatic matching process fails, you can manually correct the reference jitter WCS, and then reference other jitters onto that automatically as before. The procedure is described here and a worked example of an R Mon observation is described below.

It is recommended that users do this using `esorex` calls to recipes rather than the workflow because multiple runs of the `eris_nix_img_cal_wcs` recipe are required and it's easy to get into a tangle.

(a) Look at the log output of `eris_nix_img_cal_wcs` to find the *reference* jitter, the 'nearest jitter to centre' in the following log snippet.

```
....
[INFO] enu_get_rcore_and_mesh_size: AOMODE: FULL_AO
[INFO] enu_get_rcore_and_mesh_size: catalogue AO-related params: auto 10.000000 64
[INFO] eris_nix_scired_cal_wcs: Jitter pattern centre at RA=99.7912369 Dec= 8.7359762
[INFO] eris_nix_scired_cal_wcs: Nearest jitter to centre is 1
[INFO] eris_nix_scired_cal_wcs: ..filename skysub.ERIS.2022-11-08T08:16:33.160.fits
[WARNING] eris_nix_scired_get_split_param_int: param value: 30,30,30,30
....
```

(b) Display the reference jitter image with a FITS viewer and identify by eye an object with a known RA and Dec. Note the pixel coordinates of the object in the image.

(c) Create an ASCII file that specifies the name of the reference jitter FITS file, the x and y pixel coordinates of the object, and its RA and Dec. Note that if you have more than one reference jitter to calibrate, for a whole set of observations, then you can put information for them all in the one file. The ASCII file should have format:

```
# Comments describing the catalogue (< 80 chars per line)
Filename        RA       Dec     X_coordinate  Y_coordinate  Catalogue
jitter1.fits    99.79131 8.7361  544           599.5         Gaia
jitter2.fits    99.79131 8.7361  561.5         595.2         Gaia
jitter3.fits    99.79131 8.7361  557.2         496.8         Gaia
jitter4.fits    99.79131 8.7361  555.8         494.5         Gaia
........

Each filename will be the name of a reference jitter. The columns
'match' a catalogue RA,Dec to a pixel position in that image.
Using this the recipe can correct the wcs of the reference jitter,
others will be corrected by automatic matching to the first.
```

(d) Convert the ASCII file to FITS using the Python method `python/cat_creator.py`. This will create a FITS file of category `WCS_MATCHED_CATALOGUE` that contains the required information.

```
>>> import cat_creator
>>> cat_creator.go('NIX_rmon_matched_target.lis')
NIX_rmon_matched_target
            Filename                    RA                Dec         X_coordinate Y_coordinate Catalogue
--------------------------------- ----------------- ----------------- ------------ ------------ ---------
ERIS.2022-11-08T08:23:10.048.fits 99.79131666666667 8.736102777777777        999.3       1000.9      Gaia
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / array data type
NAXIS   =                    2 / number of array dimensions
NAXIS1  =                   69 / length of dimension 1
NAXIS2  =                    1 / length of dimension 2
PCOUNT  =                    0 / number of group parameters
GCOUNT  =                    1 / number of groups
TFIELDS =                    6 / number of table fields
TTYPE1  = 'Filename'
TFORM1  = '33A     '
TTYPE2  = 'RA      '
TFORM2  = 'D       '
TTYPE3  = 'Dec     '
TFORM3  = 'D       '
TTYPE4  = 'X_coordinate'
TFORM4  = 'D       '
TTYPE5  = 'Y_coordinate'
TFORM5  = 'D       '
TTYPE6  = 'Catalogue'
TFORM6  = '4A      '
HIERARCH ESO PRO CATG = 'ERIS_NIX_CAL_WCS_MATCHES'
COMMENT This file has positions for R Mon B lifted from Gaia-calibrated HST
COMMENT image u2m7040dt
>>>
```

(e) Re-run the workflow after putting the `WCS_MATCHED_CATALOGUE` file where it can be found or re-run `eris_nix_img_cal_wcs` with the FITS `WCS_MATCHED_CATALOGUE` file in the SoF. The recipe will read the file, locate the entry for the required reference jitter, and correct its FITS WCS to place the specified x,y position at the specified RA,Dec. The recipe will correct the other jitters to match the reference jitter.
As a check you can look at the new log output from recipe `eris_nix_img_cal_wcs`. If you can decipher what is going on you should see mention of the `WCS_MATCHED_CATALOGUE` file being used:

```
...
[INFO] eris_nix_img_cal_wcs: found manual matching list NIX_rmon_matched_target.fits
[INFO] eris_nix_img_cal_wcs: ..reference jitter skysub.ERIS.2022-11-08T08:23:10.048.fits
[INFO] eris_nix_img_cal_wcs: ..found entry for reference jitter
[INFO] eris_nix_img_cal_wcs: ..calculating WCS correction for reference jitter
[INFO] eris_nix_img_cal_wcs: ..reference jitter shift is
          0       1
  0  -189.089 -111.859
[INFO] eris_nix_img_cal_wcs: ..applying WCS correction to reference jitter
...
```

You can also use a FITS viewer to examine the wcs-calibrated reference jitter image and check that the calibrated point is now at the expected RA,Dec.

The recipe outputs wcs-calibrated jitter files, and FITS tables holding the image, reference and matched catalogues:

```
cal_wcs.<raw data 1>.fits           CAL_WCS_OBJECT_JITTER
cal_wcs.<raw data 2>.fits           CAL_WCS_OBJECT_JITTER
cal_wcs.<raw data 3>.fits           CAL_WCS_OBJECT_JITTER
...
cat.cal_wcs.<raw data 1>.fits       CAL_WCS_CATALOGUE
cat.cal_wcs.<raw data 2>.fits       CAL_WCS_CATALOGUE
cat.cal_wcs.<raw data 3>.fits       CAL_WCS_CATALOGUE
...
refcat.cal_wcs.<raw data 1>.fits    CAL_WCS_REFCAT
```

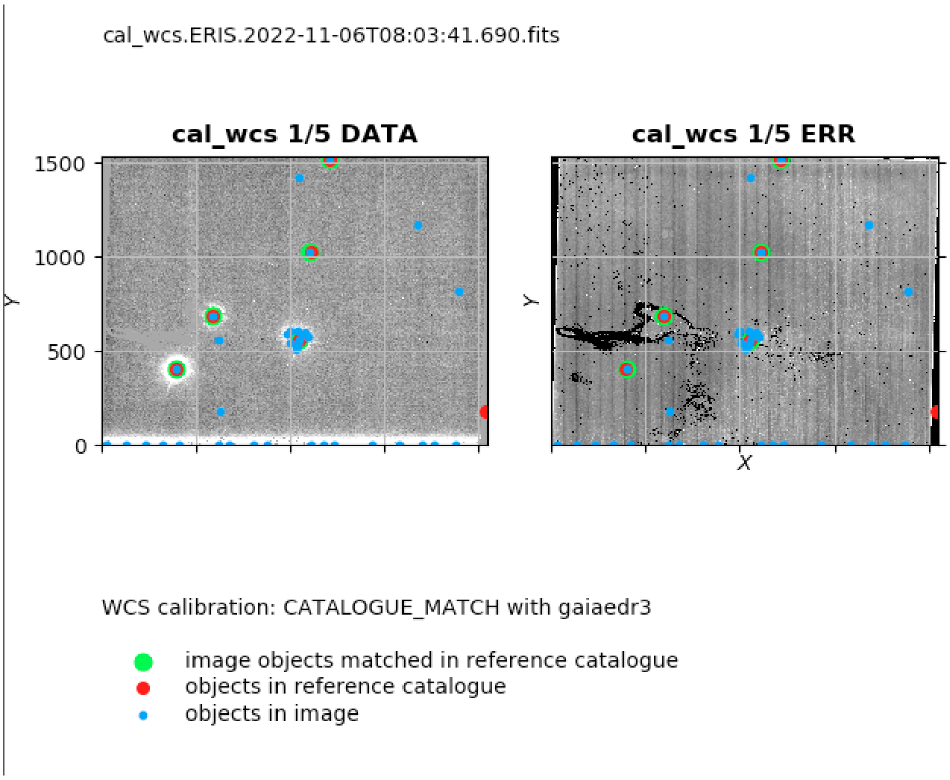


Figure 11.13: An example of a good wcs calibration with several matched Gaia and NIX sources.

```
refcat.cal_wcs.<raw data 2>.fits        CAL_WCS_REFCAT
refcat.cal_wcs.<raw data 3>.fits        CAL_WCS_REFCAT
...
matchcat.cal_wcs.<raw data 1>.fits      CAL_WCS_MATCHCAT
matchcat.cal_wcs.<raw data 2>.fits      CAL_WCS_MATCHCAT
matchcat.cal_wcs.<raw data 3>.fits      CAL_WCS_MATCHCAT
...
```

Figure 11.13 shows the workflow output for a successful WCS calibration of the reference jitter. The figure shows a number of NIX sources (blue), Gaia objects (red), and matched NIX/Gaia objects (circled green). Use common sense here - does the source matching look reasonable?

Figure 11.14 shows the workflow output for a successful WCS calibration between a random jitter in the sequence against the reference jitter.

Figure 11.15 shows a result for the *manual* WCS calibration of a dataset where referencing the Gaia catalogue was not possible. This was an observation of the star R Mon and its associated reflection nebula NGC 2261. R Mon is very bright and not quite stellar. R Mon B nearby is point-like but optically obscured and not catalogued by Gaia, and there are no other stars in the field.

To calibrate the WCS we created an ASCII file to specify the position of R Mon B in the reference jitter, and then matched the other jitters to that.

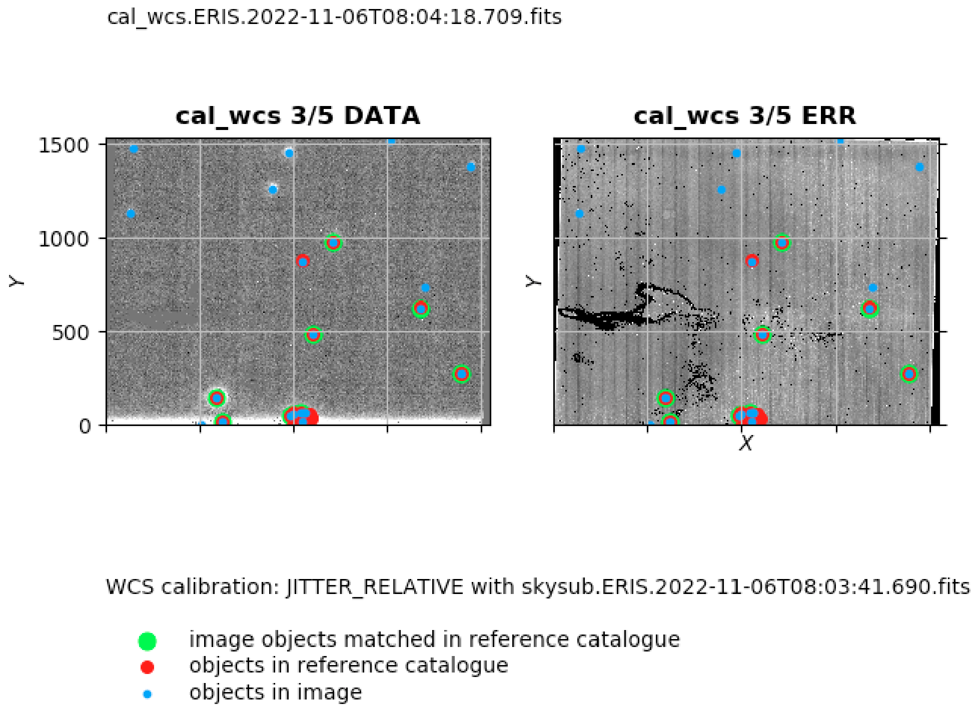


Figure 11.14: Example of a robust WCS calibration between sources in a given jitter and the reference jitter.

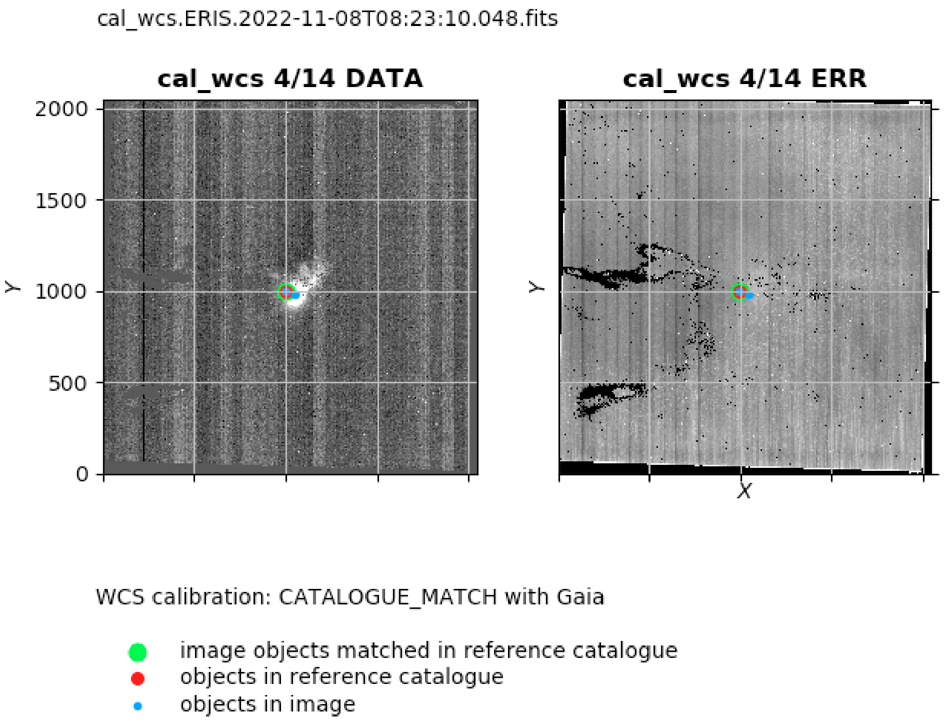


Figure 11.15: Workflow display of the reference jitter image of R Mon. The matched object (green rimmed) is R Mon B to the west or R Mon itself. The 'catalogue' is given as Gaia as specified in the WCS_MATCHED_CATALOGUE 'Catalogue' field.

The ASCII file looked like this:

```
# This file has positions for R Mon B lifted from Gaia-calibrated HST
# image u2m7040dt
Filename                          RA          Dec         X_coordinate Y_coordinate Catalogue
ERIS.2022-11-08T08:23:10.048.fits 99.791316   8.7361027   999.3        1000.9       Gaia
```

Figure 11.15 shows how the catalogued source should look in this case when overplotted on the image; the 'matched' object is R Mon B which is a dimmer point source to the west of R Mon itself at the apex of nebula fan.

4. Calibrate the photometry of the target frames using `eris_nix_img_cal_phot`.

   The calibration result for each jitter image is a photometric zeropoint (*PHOTZP*) that relates the pixel data to magnitudes (*MAG*) according to the equation $MAG = -2.5 * log(data) + PHOTZP$. A variant measure of the zeropoint is *MAGZPT*, which is *PHOTZP* normalised by *EXPTIME* and for atmospheric extinction at the zenith: $PHOTZP = MAGZPT - extinct + 2.5log_{10}EXPTIME$.

   The recipe will be run with a SoF file looking something like this:

```
cal_wcs.<raw data 1>.fits            CAL_WCS_OBJECT_JITTER
```

```
cal_wcs.<raw data 2>.fits               CAL_WCS_OBJECT_JITTER
cal_wcs.<raw data 3>.fits               CAL_WCS_OBJECT_JITTER
...
calib/nix_phot.fits                     PHOT_DATA
cal_wcs.<dataset name>.fits             IMG_STD_COMBINED (optional)
```

PHOT_DATA points to a static calibration file that relates the instrumental magnitudes to a standard system. It is a binary FITS table with one extension per system. The name of one extension must match the value of the cdssearch_photom recipe parameter (e.g. '2mass'). The format of this file is described in Section 10.3.

The default recipe parameters are these:

```
eris.eris_nix_img_cal_phot.catalogue.bkg.estimate="true"
eris.eris_nix_img_cal_phot.catalogue.bkg.mesh-size="64"
eris.eris_nix_img_cal_phot.catalogue.bkg.smooth-gauss-fwhm="2.0"
eris.eris_nix_img_cal_phot.catalogue.det.effective-gain="5.0"
eris.eris_nix_img_cal_phot.catalogue.det.saturation="25000.0"
eris.eris_nix_img_cal_phot.catalogue.obj.core-radius="5.0"
eris.eris_nix_img_cal_phot.catalogue.obj.deblending="true"
eris.eris_nix_img_cal_phot.catalogue.obj.min-pixels="20"
eris.eris_nix_img_cal_phot.catalogue.obj.threshold="10.0"
eris.eris_nix_img_cal_phot.cdssearch_photom="2MASS"
eris.eris_nix_img_cal_phot.debug-data="false"
eris.eris_nix_img_cal_phot.magerrcut="0.5"
eris.eris_nix_img_cal_phot.minphotom="1"
eris.eris_nix_img_cal_phot.pixel_radius="50.0"
```

- The catalogue parameters are passed to hdrl_catalogue_compute for use when the recipe catalogues objects in each jitter image.
- cdssearch_photom specifies 2MASS as the reference catalogue and should generally be left like this.
- If one object in the jitter image catalogue lies closer to a 'standard' than pixel_radius pixels then it will be 'associated' for calibration purposes. If more than one object lies within this range then no association will be made.
- minphotom sets the minimum number of stars required for the photometric calibration solution.
- By default, all objects in the reference catalogue (e.g. 2MASS) are used for photometric calibration. If the reference catalogue is from a magnitude-limited survey that is shallower than the NIX data, a bias will adversely affect this calibration. This bias is due to the large number of faint objects with large errors and/or inaccurate colours. This effect can be mitigated by excluding stars in the reference catalogue that have errors greater than a certain value. This threshold in error is specified by magerrcut.

Photometric calibration is achieved in 1 of 3 ways:

Internally Calibrate by associating objects in the jitter image sequence with a photometric catalogue such as 2MASS, then calculate the implied zeropoint. The process was adapted from that for HAWKI, with details as follows:

(a) Ensure that the jitter image is WCS-calibrated by checking keyword `ESO.WCS_METHOD = CATALOGUE_MATCH`.

(b) Use `hdrl_catalogue_compute` to find objects in the image. This detects the objects, characterises their morphology, and measures their brightness within an aperture (catalogue column `catcor3`) along with an estimate for the correction due to flux falling outside (`apcor3`).

(c) Interrogate the photometric catalogue for objects in the field.

(d) Match image and catalogue objects (this requires the WCS to have been calibrated properly).

(e) Read the NIX colour equations from the `PHOT_DATA` file in the SoF.

(f) Loop through the images in the jitter sequence:

   i. Are there more matched standard stars than specified by `minphotom`?

   ii. Loop through the matched standards, for each calculate the photometric zeropoint from catcor3 via $dm3 = refmag + 2.5 log_{10} catcor3 + apcor3$.

   iii. Calculate the median zeropoint from the matched standards in all images, set the FLUXCAL keyword in the primary header of the output file to "ABSOLUTE". If there are no matched standards then the calibration has failed, set FLUXCAL to "UNCALIBRATED".

Figure 11.16 shows the workflow display for one jitter in a photometrically calibrated sequence. The 'matched' objects in this jitter are shown with green rims. `FLUXCAL=ABSOLUTE` says that the calibration has succeeded. `MAGZPT` gives the calibrated zeropoint. The table below gives information on the 'matched' objects: their reference magnitudes, observed fluxes, etc. `APCOR3` gives an estimate from `hdrl_catalogue_compute` of the fraction of flux missed by the photometry.

Externally If the SoF contains an `IMG_STD_COMBINED` frame then that is the calibrated and stacked result of the observation of a standard star. The calibration process for the target dataset consists simply of transferring the keywords with the calibration from that result to these jitter images:

(a) Read FILTER from the calibrated frame and ensure that it is the same as that of the first jitter image.

(b) Read the calibration information of the calibrated frame: ESO.QC.MAGZPT, ESO.QC.MAGZERR, PHOTZP, ESO.DRS.EXTCOEF and FLUXCAL.

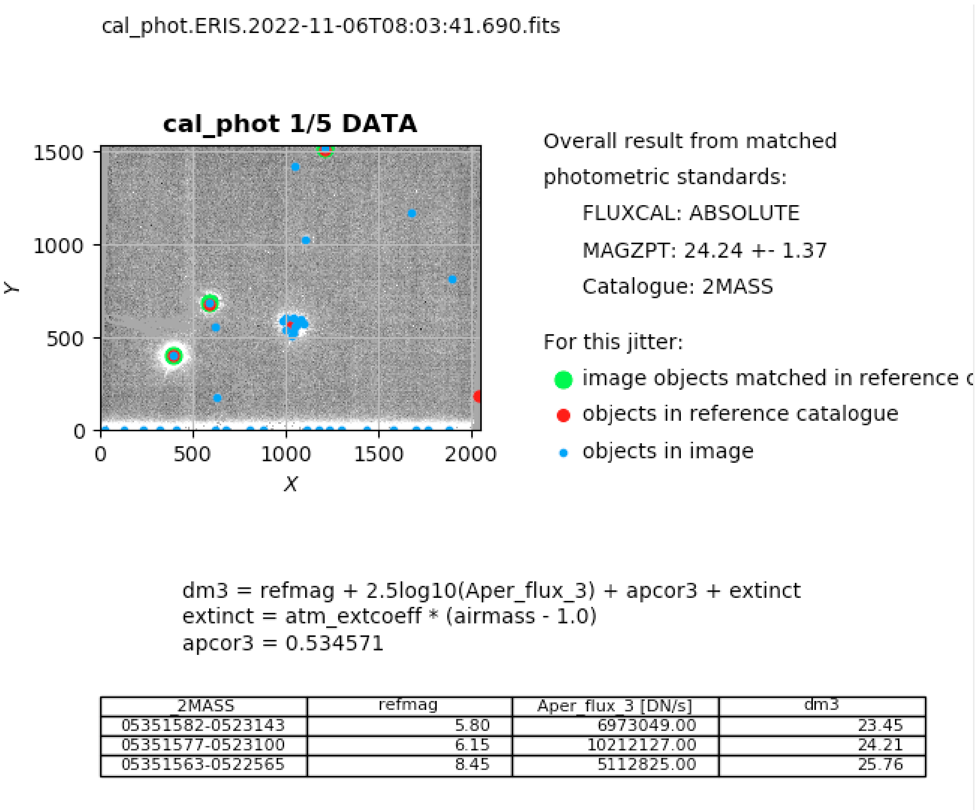
(c) For each jitter image calculate the zeropoint for the observation airmass: $photzp = magzpt - extinct + 2.5 log_{10} exptime$. All images that have been through the linearization process are in units of DN/sec so that $exptime = 1$ second and drops out.

(d) Write the updated calibration information to the jitter image.

Default If no calibrated frame is supplied and if internal calibration fails, then transfer default zeropoint values for this filter and camera combination from the `PHOT_DATA` file.

The recipe outputs calibrated jitter files:

```
cal_phot.<raw data 1>.fits          CAL_PHOT_OBJECT_JITTER
cal_phot.<raw data 2>.fits          CAL_PHOT_OBJECT_JITTER
cal_phot.<raw data 3>.fits          CAL_PHOT_OBJECT_JITTER
...
```



2MASS	refmag	Aper_flux_3 [DN/s]	dm3
05351582-0523143	5.80	6973049.00	23.45
05351577-0523100	6.15	10212127.00	24.21
05351563-0522565	8.45	5112825.00	25.76

Figure 11.16: Workflow display of a jitter image of Theta Ori that contains objects ‘matched’ (green rims) with the photometric reference catalogue.

For an ‘internal’ calibration the recipe will also output FITS tables holding the image, reference and matched catalogues:

```
cat.cal_phot.<raw data 1>.fits        CAL_PHOT_CATALOGUE
cat.cal_phot.<raw data 2>.fits        CAL_PHOT_CATALOGUE
cat.cal_phot.<raw data 3>.fits        CAL_PHOT_CATALOGUE
...
refcat.cal_phot.<raw data 1>.fits     CAL_PHOT_REFCAT
refcat.cal_phot.<raw data 2>.fits     CAL_PHOT_REFCAT
refcat.cal_phot.<raw data 3>.fits     CAL_PHOT_REFCAT
...
matchcat.cal_phot.<raw data 1>.fits   CAL_PHOT_MATCHCAT
matchcat.cal_phot.<raw data 2>.fits   CAL_PHOT_MATCHCAT
matchcat.cal_phot.<raw data 3>.fits   CAL_PHOT_MATCHCAT
...
```

For the J, H, and Ks filters, the source catalogue extracted for each science exposure is matched to the 2MASS source catalogue and an in situ photometric calibration is done when an adequate number of unsaturated 2MASS sources are available.

Currently, we are assuming a one-to-one match between the ERIS/NIX detector and filter band-passes and those of 2MASS. A project is on-going to map the colour-colour terms for each ERIS filter, by regular observations of 2MASS touchstone fields. When completed, the ERIS/NIX pipeline and archive data photometric calibration will be updated. Until this is done, we are assuming an error of roughly $\pm$ 0.5 magnitudes in the J, H, and Ks filters. For the other ERIS/NIX filters not covered by 2MASS, this error will be larger.

If an inadequate number of 2MASS stars are available in the ERIS/NIX science field, and a photometric standard star observation in the same filter and instrument mode is made during the same night, then this is used to determine the image zeropoint, following a correction of the differences in airmass and apertures between the two images. If neither 2MASS stars or a suitable photometric standard star observation are available, then a default zeropoint for the given filter is used. When this is done, the header keyword ZPMETHOD = DEFAULT.

ERIS-NIX images are background subtracted and normalised to the exposure time. Thus, the flux is in units of ADU/second. Converting the source catalogue fluxes (here, using any of the 13 aperture flux values: `Aper_flux_1` to `Aper_flux_13`) to magnitudes can be done with the following relation:

$$magnitude = PHOTZP - 2.5 * log10(Aper\_flux\_i) - APCORi \qquad \text{(for i = 1 ... 13)}$$

where uppercase parameters indicate header keywords:

PHOTZP:the photometric zeropoint [magnitude]
APCORi:the stellar aperture correction for ith aperture flux [magnitude] (this keyword can be found in the source catalogue file)

The recommended catalogue value of flux and aperture correction (written in the header of the catalogue) is `Aper_flux_3` and APCOR3, respectively.

5. Stack the target frames to produce the final 'combined' image using `eris_nix_img_hdrl_stack`.

   This recipe stacks the jitter images onto a common pixel grid to give a final observaion result.

   Its SoF file will look like this (files with tags other than `CL_PHOT_OBJECT_JITTER` will be ignored):

```
cal_phot.<raw data 1>.fits            CAL_PHOT_OBJECT_JITTER
cal_phot.<raw data 2>.fits            CAL_PHOT_OBJECT_JITTER
cal_phot.<raw data 3>.fits            CAL_PHOT_OBJECT_JITTER
...
```

   The default recipe parameters are:

```
eris.eris_nix_img_hdrl_stack.catalogue.bkg.estimate="true"
eris.eris_nix_img_hdrl_stack.catalogue.bkg.mesh-size="64"
eris.eris_nix_img_hdrl_stack.catalogue.bkg.smooth-gauss-fwhm="2.0"
eris.eris_nix_img_hdrl_stack.catalogue.det.effective-gain="3.0"
eris.eris_nix_img_hdrl_stack.catalogue.det.saturation="5000.0"
eris.eris_nix_img_hdrl_stack.catalogue.obj.core-radius="5.0"
eris.eris_nix_img_hdrl_stack.catalogue.obj.deblending="true"
eris.eris_nix_img_hdrl_stack.catalogue.obj.min-pixels="20"
eris.eris_nix_img_hdrl_stack.catalogue.obj.threshold="3.0"
eris.eris_nix_img_hdrl_stack.critical_radius="5.0"
eris.eris_nix_img_hdrl_stack.interpolation_method="lanczos"
eris.eris_nix_img_hdrl_stack.kernel_size="2"
eris.eris_nix_img_hdrl_stack.loop_distance="1"
```

   - The `catalogue` parameters are passed to `hdrl_catalogue_compute` for use in cataloguing objects in the combined image.
   - `interpolation_method`, `critical_radius`, `kernel_size` and `loop_distance` are parameters passed to `hdrl_resample_compute` which does the pixel resampling and stacking.

   The stacking process is as follows:

   (a) Adopt for the photometric calibration of the stacked result `PHOTZP` and `MAGZPT` that of the first jitter.
   (b) Loop through jitters.
   (c) Check that the atmsopheric extinction coefficient used in the photometric calibration is consistent - from `ESO.DRS.DRS.EXTCOEF`, issue warning message if not.
   (d) Correct the pixel values of the jitter to the adopted photometric calibration.
   (e) Call `hdrl_resample_image_to_table` to add the jitter to the result to be stacked.

   The recipe outputs one file containing the stacked result:

```
stack .<raw data 1>.fits              IMG_OBS_COMBINED or IMG_STD_COMBINED
```

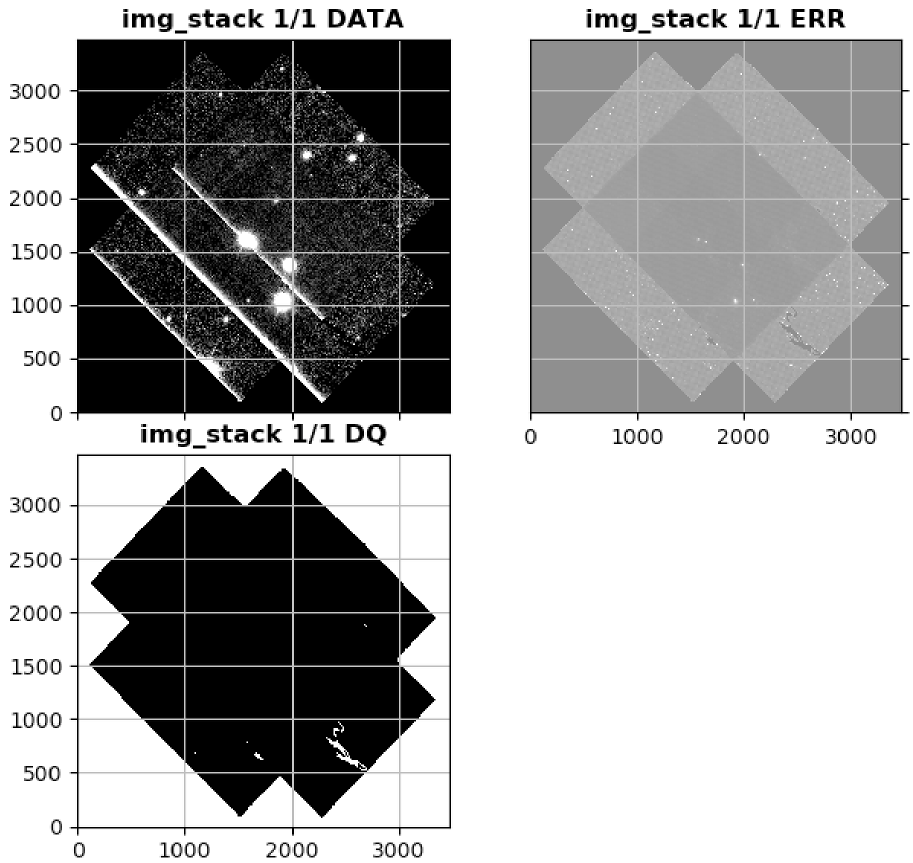


Figure 11.17: An example of a successful stack.

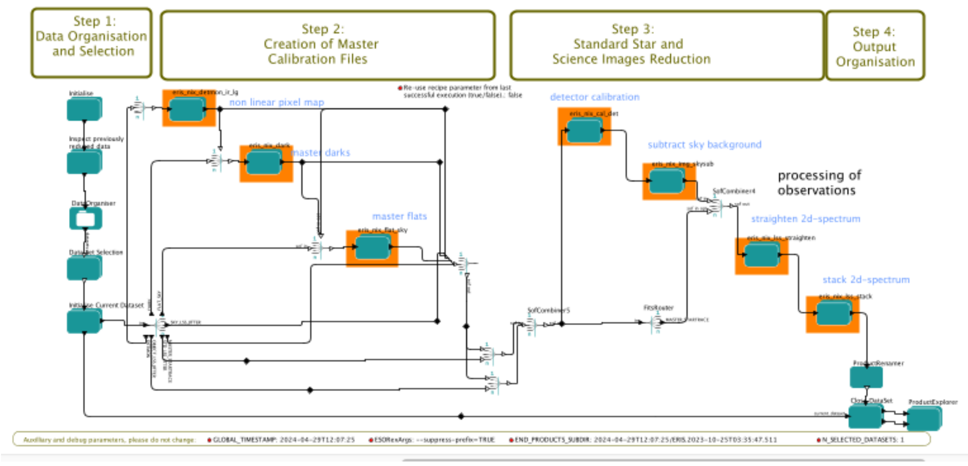


Figure 11.18: The NIX LSS workflow.

Figure 11.17 shows the workflow display for the stacked result of an observation.

If problems do occur then they are generally not caused by the stack process but by problems with some of the component jitter images. Chickens come home to roost at this final stage and any faults in the reduction of individual frames will reveal themselves. For example, one jitter frame might have a higher background level then the rest. If this happens the observer should search the input frames of the stack to identify the culprit. That done, the fault in the frame reduction can be investigated or the frame simply left out of the stack.

Another failure that might show up in the stack is a frame for which the wcs calibration has not worked. If this happens it is likely that sources covered by that frame will appear 'doubled' in the stack as the same source is projected onto 2 different points.

Please note that the parameters catalogue.bkg.mesh-size and catalogue.obj.core-radius defaults (affecting recipes `eris_nix_img_skysub`, `eris_nix_img_cal_wcs`, `eris_nix_img_cal_phot`, `eris_nix_img_hdrl_stack`) are set to optimally handle the cases JHK-13mas (no AO and FULL AO) and JHK-27mas (no AO and FULL AO).

if HIERARCH.ESO.OBS.AOMODE = `NO_AO`: default Rcore = 25 pixels and default mesh-size = 128 pixels

if HIERARCH.ESO.OBS.AOMODE = `FULL_AO`: default Rcore = 10 pixels and default mesh-size = 64 pixels

11.2 The ERIS Long-Slit Data Reduction Pipeline

As for the imaging pipeline you can use `esoreflex` to assemble a dataset and run it through the recipes or you can assemble the dataset and run the recipes manually. This section describes the manual process in more detail.

Even though a user may not use `esoreflex`, it is instructive to look at the workflow Figure 11.18 to see what needs to be done and in what order.

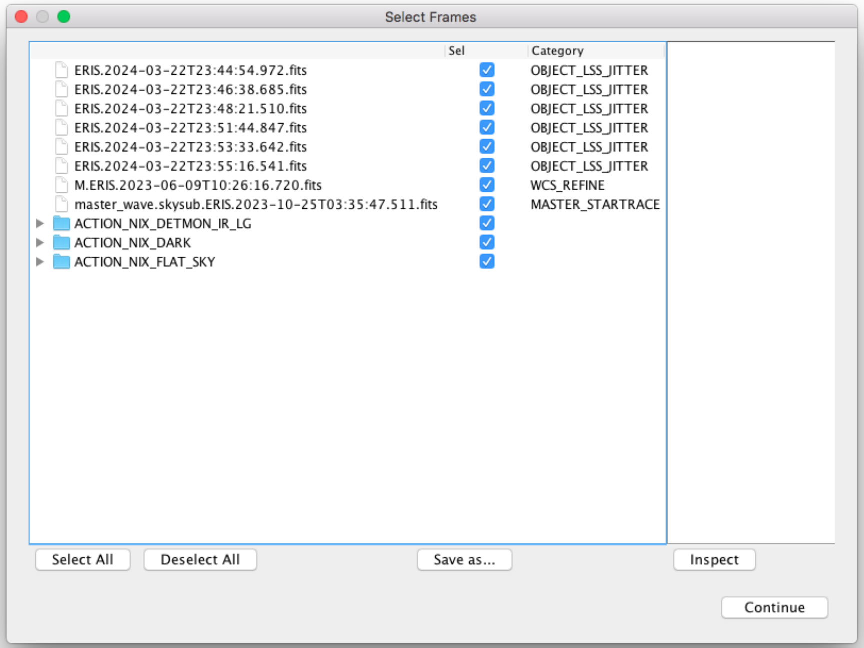


Figure 11.19: Example of a NIX long-slit dataset as shown by the esoreflex DataChooser. Not all directories have been expanded.

11.2.1 Assemble the Dataset

This stage identifies and collects together the raw data needed to reduce the particular observation, including calibration data such as flatfields, darks, linearity and wavelength calibration information.

A simple way to do this is to run `esoreflex` as far as the `DataChooser` tool. This will collect and show the files required to produce each of your datasets, for example Figure 11.19.

You will see from Figure 11.19 that the calibration files are the same as required as for imaging, with the addition of a `MASTER_STARTRACE` file that holds information to correct the optical distortion of the 2d spectrum and wavelength calibrate it.

11.2.2 Reduce the Calibration Data

The calibration data required is the same as that for the imaging pipeline: linearity, darks and flatfields, and their reduction will not be covered again here. See section 11.1.2 for details.

The `MASTER_STARTRACE` file calibration file is produced outside the reduction workflow and will usually be available as a static file.

11.2.3 Reduce the Target LSS Data

1. Remove the detector signature from the data, using recipe `eris_nix_cal_det`. The process is identical to that described for the imaging pipeline in section 11.1.2.

2. Estimate and subtract the sky background using `eris_nix_lss_skysub`. This recipe is very similar to `eris_nix_img_skysub` as described in section 11.1.2 but there are some differences.

   The categories of input file are different so that the SoF will look like this:

```
cal_det.<raw data 1>.fits             CAL_DET_OBJECT_LSS_JITTER
cal_det.<raw data 2>.fits             CAL_DET_OBJECT_LSS_JITTER
cal_det.<raw data 3>.fits             CAL_DET_OBJECT_LSS_JITTER
...
cal_det.<raw data n+1>.fits           CAL_DET_SKY_LSS_JITTER (optional)
cal_det.<raw data n+2>.fits           CAL_DET_SKY_LSS_JITTER (optional)
cal_det.<raw data n+3>.fits           CAL_DET_SKY_LSS_JITTER (optional)
...
```

   The recipe outputs sky-subtracted jitter files:

```
skysub.<raw data 1>.fits              SKYSUB_OBJECT_LSS_JITTER
skysub.<raw data 2>.fits              SKYSUB_OBJECT_LSS_JITTER
skysub.<raw data 3>.fits              SKYSUB_OBJECT_LSS_JITTER
...
```

   The sky background is calculated in the same way as for the imaging case. The main additional step is that the sky-subtracted result is afterwards divided by the slit-response. The response is calculated by collapsing the background spectrum in the dispersion direction and then normalising it – all slit offsets should see the same sky background spectrum.

   Figure 11.9 shows the workflow display for a sky-subtracted jitter image. The DATA frame shows the sky-subtracted result and BKG_DATA the sky background used.

3. Calibrate the WCS (world coordinate system) of the target LSS images. This involves: correcting the distortion of the raw 2d spectrum images to align the slit direction exactly parallel to the x-axis and dispersion exactly parallel to the y-axis: setting the WCS keywords to describe the slit position and wavelengths.

   Image rectification and WCS calibration are performed by recipe `eris_nix_lss_straighten` with a SoF file like:

```
skysub.<raw data 1>.fits              SKYSUB_OBJECT_LSS_JITTER
skysub.<raw data 2>.fits              SKYSUB_OBJECT_LSS_JITTER
skysub.<raw data 3>.fits              SKYSUB_OBJECT_LSS_JITTER
...
master_wave.fits                      MASTER_STARTRACE
```

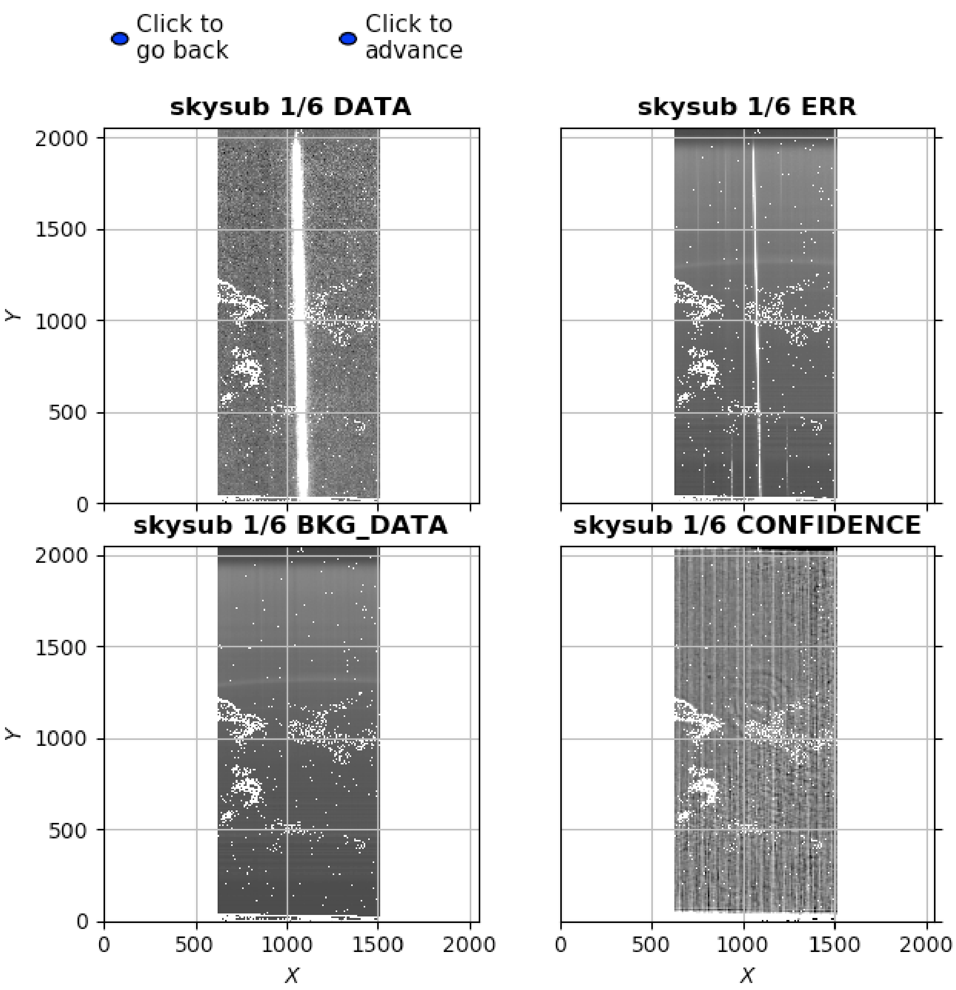


Figure 11.20: An example of a successful LSS sky background subtraction.

Default recipe parameters are:

```
eris.eris_nix_img_cal_wcs.debug-data="false"
```

- `debug_data` signals if debugging information is to be output by the recipe. It should be set `"false"` during normal use.

The recipe works in the following way:

(a) It is assumed that the jitter pattern will have moved the object up and down the line of a notional slit on the sky. The 'notional slit' reference position and the offset of each jitter from it are calculated as follows:

  i. The reference is taken from jitter 0. Often this jitter lies near the middle of the pattern and has had the object manually positioned near the middle of the slit.
  ii. Derive the PA of the notional slit (0 = N, increasing E) from the extrema of the jitter centres.
  iii. Report how far each jitter pointing lies from the line of the notional slit. This lets the user check that the jitter pattern behaves as expected.
  iv. Estimate the offset of each jitter along the notional slit relative to the reference position.

(b) Resample the spectra to: align the slit parallel to the y-axis, the dispersion parallel to the x-axis, and calibrate the wavelengths. In addition, register each spectrum so that an object in the slit appears in the same column it did in the reference jitter. Registration is done at the this stage so that no further resampling need be done later when the spectra are stacked.
The detailed steps for each jitter are:

  i. Estimate the shift required to place the object in the same column as in the reference jitter:
  ..Take a copy of the 2d spectrum.
  ..Apply the `MASTER_WAVE` correction to straighten the copy.
  ..Collapse the copy along the dispersion axis.
  ..Locate the star/object.
  ..Calculate the shift required to move the star to the position it had in the reference jitter.
  This assumes there is only one object and that the jitter throw is not so large that the object moves right off the jitter 0 slit.
  ii. Modify the `MASTER_WAVE` warp polynomials to add the required shift to the straightening and wavelength calibration when applied.
  iii. Apply the shifted warp polynomials to:
  ..the data plane
  ..the error plane
  ..the background spectrum
  ..the confidence array

(c) Set the confidence to 0 outside the illuminated range of the shifted spectrum – to reduce edge effects when stacking them later.

(d) Save the straightened spectra.

The recipe outputs straightened and wcs-calibrated jitter files:

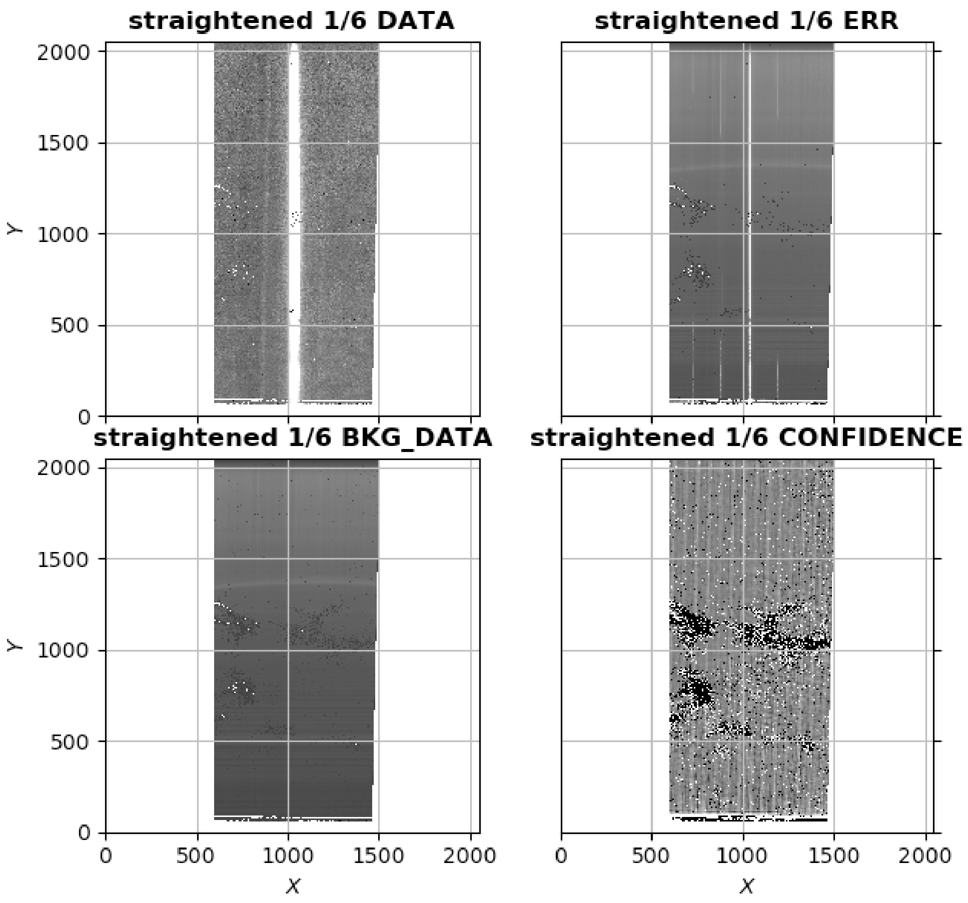


Figure 11.21: An example of a well ‘straightened’ LSS spectrum.

```
straightened.<raw data 1>.fits          CORRECTED_OBJECT_LSS_JITTER
straightened.<raw data 2>.fits          CORRECTED_OBJECT_LSS_JITTER
straightened.<raw data 3>.fits          CORRECTED_OBJECT_LSS_JITTER
...
```

Figure 11.21 shows the workflow output from a successful run of `eris_nix_lss_straighten`. The spectrum of the star in the slit now lies directly parallel to the y-axis.

4. Stack the jitter spectra to produce the final 2d result using `eris_nix_lss_stack`. The SoF file will look like this:

```
straightened.<raw data 1>.fits         CORRECTED_OBJECT_LSS_JITTER
straightened.<raw data 2>.fits         CORRECTED_OBJECT_LSS_JITTER
straightened.<raw data 3>.fits         CORRECTED_OBJECT_LSS_JITTER
...
```

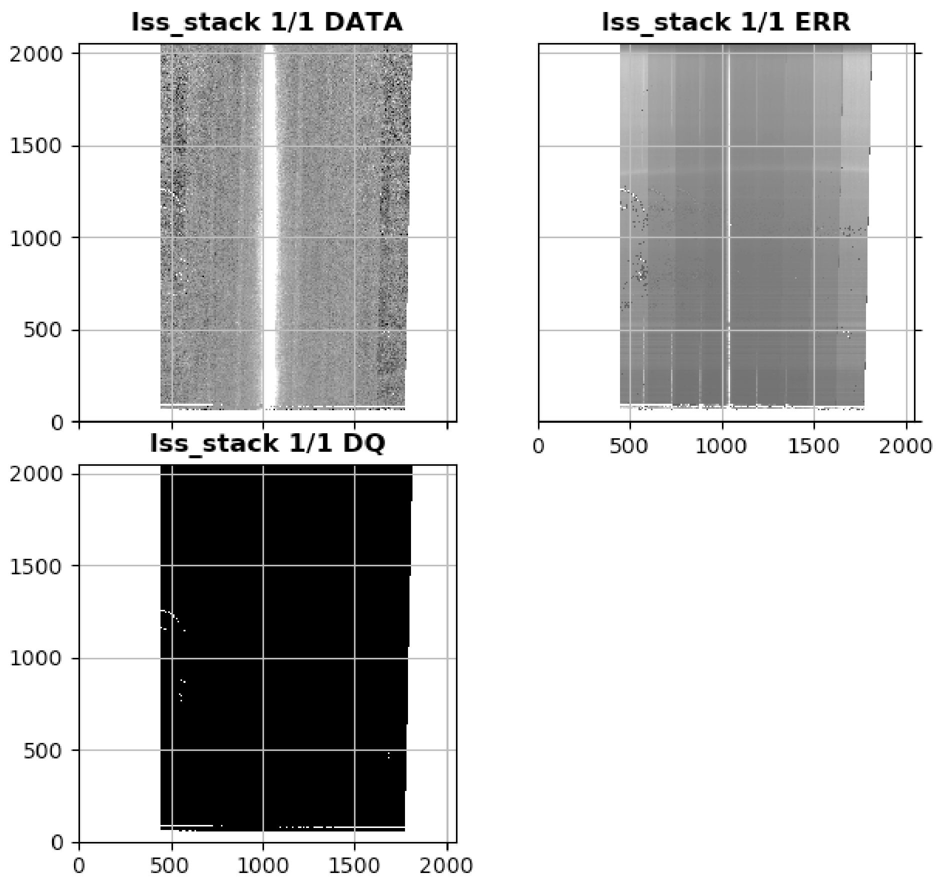


Figure 11.22: An example of a successful LSS stack.

The default recipe parameters are:

```
eris.eris_nix_lss_stack.x_probe="-1"
eris.eris_nix_lss_stack.y_probe="-1"
```

- The `x_probe` and `y_probe`parameters are for diagnostic use and should normally be left set to -1.

This recipe uses an HDRL collapse operation to stack a set of LSS jitter frames that have already been registered at the ‘straightening’ stage. Unlike the stacking of image data no resampling or correction for photometric calibration is involved.

The recipe outputs one file with the stacked result:

```
stack.<raw data 1>.fits                    LSS_OBS_COMBINED or LSS_STD_COMBINED
```

Figure 11.22 shows the workflow display for the stacked result of an LSS observation.

5. Further recipes for the extraction of 1d-spectra from the 2d results are under development.

12 Recipe Reference

12.1 eris_nix_detmon_ir_lg

This recipe determines a map of non linear pixels and the detector gain.

12.1.1 Description

Please refer to section 11.1.2, point 1, at page 39.

12.1.2 Input Frames

Frame Tag	Type	Count	Description
OFF_RAW	raw	42 (min)	Raw linearity off frame
ON_RAW	raw	42 (min)	Raw linearity on frame

12.1.3 Product Frames

Default File Name	Frame Tag	Description
eris_nix_detmon_ir_lg_linearity_table.fits	DET_LIN_INFO	Table with linearity information
eris_nix_detmon_ir_lg_gain_table.fits	GAIN_INFO	Gain table
eris_nix_detmon_ir_lg_coeffs_cube.fits	COEFFS_CUBE	Cube with fit coefficients
eris_nix_detmon_ir_lg_bpm.fits	BP_MAP_NL	Bad pixel mask

12.1.4 Quality Control Parameters

`QC.NUM.BPM`	Number of bad pixels detected
`QC.LIN.COEFi`	order-i linearity coefficient value
`QC.LIN.COEFi.ERR`	error on order-i linearity coefficient value
`QC.ERRFIT`	Error of fit
`QC.METHOD`	Method applied to compute GAIN
`QC.CONAD`	[ADU/e-] Conversion from e- to ADUs
`QC.GAIN`	[e-/ADU] GAIN (see QC.METHOD)
`QC.COUNTS.MIN`	[ADU] Minimum median value used in inearity test (in a user defined region)
`QC.COUNTS.MAX`	[ADU] Maximum median value used in inearity test (in a user defined region)
`QC.LIN.EFF`	Effective non-linearity correction
`QC.LIN.EFF.FLUX`	[ADU] FLux level at which effective non-linearity correction is computed

QC.FPN	[ADU] Fixed Pattern Noise
QC.GAIN.ERR	[e-/ADU] Error associated to gain

12.1.5 Recipe Parameters

Parameter	Type	Values	Description
product_depth	int	0,1, ...	set what kind of products are created (0 minimal, 1 more, ...)
method	string	**PTC**,MED	Method to be used when computing GAIN. Methods appliable: <PTC | MED>. By default PTC method will be applied. [PTC]
order	int	**3**	Polynomial order for the fit (Linearity). [3]
kappa	double	**3.0**	Kappa value for the kappa-sigma clipping (Gain). [3.0]
niter	int	**5**	Number of iterations to compute rms (Gain). [5]
llx	int	**-1**	x coordinate of the lower-left point of the region of interest. If not modified, default value will be 1. [-1]
lly	int	**-1**	y coordinate of the lower-left point of the region of interest. If not modified, default value will be 1. [-1]
urx	int	**-1**	x coordinate of the upper-right point of the region of interest. If not modified, default value will be X dimension of the input image. [-1]
ury	int	**-1**	y coordinate of the upper-right point of the region of interest. If not modified, default value will be Y dimension of the input image. [-1]
ref_level	int	**10000**	User reference level. [10000]
intermediate	boolean	**FALSE**	De-/Activate intermediate products. [FALSE]
autocorr	boolean	**FALSE**	De-/Activate the autocorr option. [FALSE]
collapse	boolean	**FALSE**	De-/Activate the collapse option. [FALSE]
rescale	boolean	**TRUE**	De-/Activate the image rescale option. [TRUE]
pix2pix	boolean	**TRUE**	De-/Activate the computation with pixel to pixel accuracy. [TRUE]

continued on next page

continued from previous page			
Parameter	**Type**	**Values**	**Description**
bpmbin	boolean	**FALSE**	De-/Activate the binary bpm option. [FALSE]
m	int	**26**	Maximum x-shift for the autocorr. [26]
filter	int	**-1**	Upper limit of Median flux to be filtered. [-1]
n	int	**26**	Maximum y-shift for the autocorr. [26]
tolerance	double	**0.001**	Tolerance for pair discrimination. [0.001]
pafgen	boolean	**FALSE**	Generate PAF file. [FALSE]
pafname	string	**eris_nix_detmon_ir_lg**	Specific name for PAF file. [eris_nix_detmon_ir_lg]
exts	int	**0**	Activate the multi-exts option. Choose -1 to process all extensions. Choose an extension number to process the appropriate extension. [0]
fpn_method	string	**HISTOGRAM**	Method for computing Fixed Pattern Noise (SMOOTH or HISTOGRAM). [HISTOGRAM]
fpn_smooth	int	**13**	template size in pixels for smoothing during FPN computation (only for SMOOTH method). [13]
saturation_limit	double	**65535.0**	all frames with mean saturation above the limit would not be used in linearity calculation. [65535.0]
gain_threshold	double	**65535.0**	all frames with mean flux above the threshold would not be used in gain calculation. [65535.0]
coeffs_cube_split	boolean	**FALSE**	if TRUE, the recipe writes as many COEFFS_CUBE_Pi (i=0..order) as the value of the order parameter in a separate file. [FALSE]

Please note that to get proper results for the fast aand slow NIX read out modes the user should set quite different values of the parameters **saturation_limit** and **gain_threshold**. This is automatically done by ESOReflex. If a user prefers to reduce the data with esorex the values we use with ESOReflex are:

Parameter	RO mode	value
saturation_limit	fast	49000
gain_threshold	fast	35000
saturation_limit	slow	15000
gain_threshold	slow	15000

12.2 eris_nix_dark

This recipe determines a master dark frame stacking with kappa-sigma clipping of outliers a set of (at least five) raw dark frames, and computes for quality control the detector read out noise and the associated variance; the statistics (mean, median, rms) on the master dark; the number of hot pixels and their fraction to the total number of pixels; the detector gain and the fixed pattern noise.

12.2.1 Description

Please refer to section 11.1.2, point 2, at page 39.

12.2.2 Input Frames

Frame Tag	Type	Count	Description
DARK	raw	5 (min)	Raw dark
DET_LIN_INFO	cdb	1	det lin info
GAIN_INFO	cdb	1	gain info

12.2.3 Product Frames

Default File Name	Frame Tag	Description
master_dark.fits	MASTER_DARK_IMG	Master dark frame

12.2.4 Quality Control Parameters

QC.READ.NOISE	[e-] Detector read out noise
QC.READ.NOISE.VAR	[e-] variance on Detector read out noise
QC.DARK.MED	Dark median
QC.DARK.MEAN	Dark mean
QC.DARK.RMS	Dark RMS
QC.NUMBER.HOT.PIXEL	Number of detected hot pixels
QC.HOT.PIXEL.FRACTION	fraction of detected hot pixel to total
QC.PARTICLE_RATE	Particle rate
QC.GAIN	detector gain
QC.DARKFPN	fixed pattern noise

12.2.5 Recipe Parameters

Parameter	Type	Values	Description
collapse.method	string	**MEDIAN**, MEAN, WEIGHTED_MEAN, MEDIAN, SIGCLIP, MINMAX, MODE	Method used for collapsing the data.
collapse.sigclip.kappa-low	double	**10.0**	Low kappa factor for kappa-sigma clipping algorithm. [10.0]
collapse.sigclip.kappa-high	double	**10.0**	High kappa factor for kappa-sigma clipping algorithm. [10.0]
collapse.sigclip.niter	int	**3**	Maximum number of clipping iterations for kappa-sigma clipping. [3]
collapse.mode.histo-min	double	**10.0**	Minimum pixel value to accept for mode computation. [10.0]
collapse.mode.histo-max	double	**1.0**	Maximum pixel value to accept for mode computation. [1.0]
collapse.mode.bin-size	double	**0.0**	Binsize of the histogram. [0.0]
collapse.mode.method	string	**MEDIAN**, MEDIAN, WEIGHTED, FIT	Mode method (algorithm) to use.
collapse.mode.error-niter	int	**0**	Iterations to compute the mode error. [0]
hotpix.method	string	**FILTER**, LEGENDRE	Method used.
hotpix.legendre.kappa-low	double	**4.0**	Low RMS scaling factor for image thresholding. [4.0]
hotpix.legendre.kappa-high	double	**5.0**	High RMS scaling factor for image thresholding. [5.0]
hotpix.legendre.maxiter	int	**6**	Maximum number of algorithm iterations. [6]
hotpix.legendre.steps-x	int	**20**	Number of image sampling points in x-dir for fitting. [20]
hotpix.legendre.steps-y	int	**21**	Number of image sampling points in y-dir for fitting. [21]
hotpix.legendre.filter-size-x	int	**11**	X size of the median box around sampling points. [11]
hotpix.legendre.filter-size-y	int	**12**	Y size of the median box around sampling points. [12]
hotpix.legendre.order-xy	int	**2**	Order of x polynomial for the fit. [2]
hotpix.legendre.order-y	int	**10**	Order of y polynomial for the fit. [10]
hotpix.filter.kappa-low	double	**10.0**	Low RMS scaling factor for image thresholding. [10.0]
hotpix.filter.kappa-high	double	**10.0**	High RMS scaling factor for image thresholding. [10.0]

continued on next page

continued from previous page			
Parameter	**Type**	**Values**	**Description**
hotpix.filter.maxiter	int	**3**	Maximum number of algorithm iterations. [3]
hotpix.filter.filter	string	**MEDIAN**, AVERAGE, AVERAGE_FAST, MEDIAN	Filter mode for image smooting.
hotpix.filter.border	string	**FILTER**, CROP, NOP, COPY	Border mode to use for the image smooting filter (only for MEDIAN filter).
hotpix.filter.smooth-x	int	**21**	Kernel y size of the smoothing filter. [21]
hotpix.filter.smooth-y	int	**21**	Kernel y size of the image smoothing filter. [21]

12.3 eris_nix_flat_lamp

This recipe uses a set of raw flat ON and flat OFF frames to generate a hi-frequence, a low-frequence master flat and a bad pixel map.

12.3.1 Description

Please refer to section 11.1.2, point 3, at page 40.

12.3.2 Input Frames

Frame Tag	Type	Count	Description
FLAT_LAMP_OFF	raw	5 (min)	Raw flat lamp off frames
FLAT_LAMP_ON	raw	5 (min)	Raw flat lamp on frames
MASTER_DARK_IMG	cdb	1	master dark image
GAIN_INFO	cdb	1	gain info table
BP_MAP_NL	cdb	1	not linear bad pixel map
COEFFS_CUBE	cdb	1	non-linear fit coefficients
DET_LIN_INFO	cdb	1	non-linearity information table

12.3.3 Product Frames

Default File Name	Frame Tag	Description
master_bpm_lamp.fits	MASTER_BPM_LAMP	master bad pixel map
		continued on next page

continued from previous page

Default File Name	Frame Tag	Description
master_flat_lamp_hifreq.fits	MASTER_FLAT_LAMP_HIFREQ	master flat (hi frequency)
master_flat_lamp_lofreq.fits	MASTER_FLAT_LAMP_LOFREQ	master flat (low frequency)

12.3.4 Quality Control Parameters

QC.NUMBER.BAD.PIXELS	number of detected bad pixels
QC.FRACTION.BAD.PIXELS	fraction of bad pixels to total
QC.FLAT.MED	[ADU] median flat value
QC.FLAT.MEAN	[ADU] mean flat value
QC.FLAT.RMS	[ADU] RMS of flat value
QC.FLAT.NORM	flat normalization value
QC.NUMBER.COLD.PIXELS	number of detected cold pixels
QC.FLAT_ONi.NPOSSAT	Number of flat pixels above threshold on FLAT ON frame i
QC.FLAT_ONi.NNEGSAT	Number of flat pixels below neg threshold on FLAT ON frame i
QC.FLAT_ONi.MEAN	[ADU] Mean of flat on FLAT ON frame i
QC.FLAT_ONi.MEDIAN	[ADU] Median of flat on FLAT ON frame i
QC.FLAT_ONi.STDEV	[ADU] Stdev of flat on FLAT ON frame i
QC.FLAT_ON.NPOSSAT.MAX	Max Number of flat pixels above threshold on FLAT ON
QC.FLAT_ON.NNEGSAT.MAX	Max Number of flat pixels below neg threshold on FLAT ON
QC.FLAT_ON.MEAN.MIN	[ADU] Minimum Means of flat on FLAT ON
QC.FLAT_ON.MEDIAN.MIN	[ADU] Minimum Medians of flat on FLAT ON
QC.FLAT_ON.MEAN.MAX	[ADU] Max Means of flat on FLAT ON
QC.FLAT_ON.MEDIAN.MAX	[ADU] Max Medians of flat on FLAT ON
QC.FLAT_ON.MEAN.FRAC	[ADU] Min/Max Means of flat on FLAT ON
QC.FLAT_ON.MEDIAN.FRAC	[ADU] Min/Max Medians of flat on FLAT ON
QC.FLAT_ON.STDEV	[ADU] Stdev of flat on FLAT ON frame i

12.3.5 Recipe Parameters

Parameter	Type	Values	Description
collapse.method	string	**MEDIAN**, MEAN, WEIGHTED_MEAN, MEDIAN, SIGCLIP, MINMAX, MODE	Method used for collapsing the data.
collapse.sigclip.kappa-low	double	**10.0**	Low kappa factor for kappa-sigma clipping algorithm. [10.0]
collapse.sigclip.kappa-high	double	**10.0**	High kappa factor for kappa-sigma clipping algorithm. [10.0]

continued on next page

continued from previous page

Parameter	Type	Values	Description
collapse.sigclip.niter	int	**3**	Maximum number of clipping iterations for kappa-sigma clipping. [3]
collapse.mode.histo-min	double	**10.0**	Minimum pixel value to accept for mode computation. [10.0]
collapse.mode.histo-max	double	**1.0**	Maximum pixel value to accept for mode computation. [1.0]
collapse.mode.bin-size	double	**0.0**	Binsize of the histogram. [0.0]
collapse.mode.method	string	**MEDIAN**, MEDIAN, WEIGHTED, FIT	Mode method (algorithm) to use.
collapse.mode.error-niter	int	**0**	Iterations to compute the mode error. [0]
flat.filter-size-x	int	**21**	Smoothing filter size in x-direction. [21]
flat.filter-size-y	int	**21**	Smoothing filter size in y-direction. [21]
flat.method	string	**high**, low	Method to use for the master flat-field calculation.
coldpix.method	string	**FILTER**, LEGENDRE	Method used.
coldpix.legendre.kappa-low	double	**4.0**	Low RMS scaling factor for image thresholding. [4.0]
coldpix.legendre.kappa-high	double	**5.0**	High RMS scaling factor for image thresholding. [5.0]
coldpix.legendre.maxiter	int	**6**	Maximum number of algorithm iterations. [6]
coldpix.legendre.steps-x	int	**20**	Number of image sampling points in x-dir for fitting. [20]
coldpix.legendre.steps-y	int	**21**	Number of image sampling points in y-dir for fitting. [21]
coldpix.legendre.filter-size-x	int	**11**	X size of the median box around sampling points. [11]
coldpix.legendre.filter-size-y	int	**12**	Y size of the median box around sampling points. [12]
coldpix.legendre.order-x	int	**2**	Order of x polynomial for the fit. [2]
coldpix.legendre.order-y	int	**10**	Order of y polynomial for the fit. [10]
coldpix.filter.kappa-low	double	**5.0**	Low RMS scaling factor for image thresholding. [5.0]
coldpix.filter.kappa-high	double	**20.0**	High RMS scaling factor for image thresholding. [20.0]

continued on next page

continued from previous page

Parameter	Type	Values	Description
coldpix.filter.maxiter	int	**3**	Maximum number of algorithm iterations. [3]
coldpix.filter.filter	string	**MEDIAN**, AVERAGE, AVERAGE_FAST, MEDIAN	Filter mode for image smooting.
coldpix.filter.border	string	**NOP**, FILTER, CROP, NOP, COPY	Border mode to use for the image smooting filter (only for MEDIAN filter).
coldpix.filter.smooth-x	int	**21**	Kernel y size of the smoothing filter. [21]
coldpix.filter.smooth-y	int	**21**	Kernel y size of the image smoothing filter. [21]
min-coadds	int	**1**	minimum acceptable number of (lamp_on - lamp_off) images. [1]
x-probe	int	**-1**	x coord of diagnostic pixel. [-1]
y-probe	int	**-1**	y coord of diagnostic pixel. [-1]
collapse-cube	boolean	**FALSE**	Mean collapse cube if DET.FRAM.FORMAT is cube. [FALSE]

12.4 eris_nix_flat_twilight

From a set of input twilight flat frames this recipe generates a low-frequence master flat.

12.4.1 Description

Please refer to section 11.1.2, point 3, at page 40.

12.4.2 Input Frames

Frame Tag	Type	Count	Description
FLAT_TWILIGHT	raw	5 (min)	Raw flat twilight frames
BP_MAP_NL	cdb	1	not linear bad pixel map
COEFFS_CUBE	cdb	1	non-linear fit coefficients
DET_LIN_INFO	cdb	1	non-linearity information table

12.4.3 Product Frames

Default File Name	Frame Tag	Description
master_flat_twilight_lofreq.fits	MASTER_FLAT_TWILIGHT_LOFREQ	master twilight flat (low frequency)

12.4.4 Quality Control Parameters

QC.FLAT.MED	[ADU] median flat value
QC.FLAT.MEAN	[ADU] mean flat value
QC.FLAT.RMS	[ADU] RMS of flat value
QC.FLAT.NORM	flat normalization value
QC.FLAT_ON.MEAN.MIN	[ADU] Minimum Means of flat on FLAT ON
QC.FLAT_ON.MEDIAN.MIN	[ADU] Minimum Medians of flat on FLAT ON
QC.FLAT_ON.MEAN.MAX	[ADU] Max Means of flat on FLAT ON
QC.FLAT_ON.MEDIAN.MAX	[ADU] Max Medians of flat on FLAT ON
QC.FLAT_ON.MEAN.FRAC	[ADU] Min/Max Means of flat on FLAT ON
QC.FLAT_ON.MEDIAN.FRAC	[ADU] Min/Max Medians of flat on FLAT ON
QC.FLAT_ON.NPOSSAT.MAX	Max Number of flat pixels above threshold on FLAT ON
QC.FLAT_ON.NNEGSAT.MAX	Max Number of flat pixels below neg threshold on FLAT ON

12.4.5 Recipe Parameters

Parameter	Type	Values	Description
flat.filter-size-x	int	**21**	Smoothing filter size in x-direction. [21]
flat.filter-size-y	int	**21**	Smoothing filter size in y-direction. [21]
min-frames	int	**2**	minimum acceptable number of twilight images. [2]
x-probe	int	**-1**	x coord of diagnostic pixel. [-1]
y-probe	int	**-1**	y coord of diagnostic pixel. [-1]

12.5 eris_nix_flat_sky

From a set of input sky flat frames, initially separated in two sub-groups of low and high airmass data, this recipe generates a hi-frequence and a low-frequence master flat and a bad pixel map.

12.5.1 Description

Please refer to section 11.1.2, point 3, at page 40.

12.5.2 Input Frames

Frame Tag	Type	Count	Description
FLAT_SKY	raw	5 (min)	Raw flat sky frames
GAIN_INFO	cdb	1	gain information table
BP_MAP_NL	cdb	1	not linear bad pixel map
COEFFS_CUBE	cdb	1	non-linear fit coefficients
MASTER_DARK_IMG	cdb	1	master dark

12.5.3 Product Frames

Default File Name	Frame Tag	Description
master_bpm.fits	MASTER_BPM_SKY	master bad pixel map
master_flat_hifreq.fits	MASTER_FLAT_SKY_HIFREQ	master sky flat (high frequency)
master_flat_lofreq.fits	MASTER_FLAT_SKY_LOFREQ	master sky flat (low frequency)

12.5.4 Quality Control Parameters

`QC.NUMBER.BAD.PIXELS`	number of bad pixels
`QC.NUMBER.COLD.PIXELS`	number of cold pixels
`QC.FRACTION.BAD.PIXELS`	fraction of bad pixels to total
`QC.FLAT.MED`	[ADU] median flat value
`QC.FLAT.MEAN`	[ADU] mean flat value
`QC.FLAT.RMS`	[ADU] RMS of flat value
`QC.FLAT.NORM`	flat normalization value
`QC.FLAT_ON.MEAN.MIN`	[ADU] Minimum Means of flat on FLAT ON
`QC.FLAT_ON.MEDIAN.MIN`	[ADU] Minimum Medians of flat on FLAT ON
`QC.FLAT_ON.MEAN.MAX`	[ADU] Max Means of flat on FLAT ON
`QC.FLAT_ON.MEDIAN.MAX`	[ADU] Max Medians of flat on FLAT ON
`QC.FLAT_ON.MEAN.FRAC`	Min/Max Means of flat on FLAT ON
`QC.FLAT_ON.MEDIAN.FRAC`	Min/Max Medians of flat on FLAT ON
`QC.FLAT_ON.NPOSSAT.MAX`	Max Number of flat pixels above threshold on FLAT ON
`QC.FLAT_ON.NNEGSAT.MAX`	Max Number of flat pixels below neg threshold on FLAT ON

12.5.5 Recipe Parameters

Parameter	Type	Values	Description
collapse.method	string	**MEDIAN**, MEAN, WEIGHTED_MEAN, MEDIAN, SIGCLIP, MINMAX, MODE	Method used for collapsing the data.
collapse.sigclip.kappa-low	double	**10.0**	Low kappa factor for kappa-sigma clipping algorithm. [10.0]
collapse.sigclip.kappa-high	double	**10.0**	High kappa factor for kappa-sigma clipping algorithm. [10.0]
collapse.sigclip.niter	int	**3**	Maximum number of clipping iterations for kappa-sigma clipping. [3]
collapse.mode.histo-min	double	**10.0**	Minimum pixel value to accept for mode computation. [10.0]
collapse.mode.histo-max	double	**1.0**	Maximum pixel value to accept for mode computation. [1.0]
collapse.mode.bin-size	double	**0.0**	Binsize of the histogram. [0.0]
collapse.mode.method	string	**MEDIAN**, MEDIAN, WEIGHTED, FIT	Mode method (algorithm) to use.
collapse.mode.error-niter	int	**0**	Iterations to compute the mode error. [0]
flat.filter-size-x	int	**21**	Smoothing filter size in x-direction. [21]
flat.filter-size-y	int	**21**	Smoothing filter size in y-direction. [21]
flat.method	string	**high**, low	Method to use for the master flatfield calculation.
coldpix.method	string	**FILTER**, LEGENDRE	Method used.
coldpix.legendre.kappa-low	double	**4.0**	Low RMS scaling factor for image thresholding. [4.0]
coldpix.legendre.kappa-high	double	**5.0**	High RMS scaling factor for image thresholding. [5.0]
coldpix.legendre.maxiter	int	**6**	Maximum number of algorithm iterations. [6]
coldpix.legendre.steps-x	int	**20**	Number of image sampling points in x-dir for fitting. [20]
coldpix.legendre.steps-y	int	**21**	Number of image sampling points in y-dir for fitting. [21]
coldpix.legendre.filter-size-x	int	**11**	X size of the median box around sampling points. [11]
coldpix.legendre.filter-size-y	int	**12**	Y size of the median box around sampling points. [12]
coldpix.legendre.order-x	int	**2**	Order of x polynomial for the fit. [2]
coldpix.legendre.order-y	int	**10**	Order of y polynomial for the fit. [10]
coldpix.filter.kappa-low	double	**5.0**	Low RMS scaling factor for image thresholding. [5.0]

continued on next page

continued from previous page

Parameter	Type	Values	Description
coldpix.filter.kappa-high	double	**20.0**	High RMS scaling factor for image thresholding. [20.0]
coldpix.filter.maxiter	int	**3**	Maximum number of algorithm iterations. [3]
coldpix.filter.filter	string	**MEDIAN**, AVERAGE, AVERAGE_FAST, MEDIAN	Filter mode for image smooting.
coldpix.filter.border	string	**NOP**, FILTER, CROP, NOP, COPY	Border mode to use for the image smooting filter (only for MEDIAN filter).
coldpix.filter.smooth-x	int	**21**	Kernel y size of the smoothing filter. [21]
coldpix.filter.smooth-y	int	**21**	Kernel y size of the image smoothing filter. [21]
min-coadds	int	**1**	minimum acceptable number of (lamp_on - lamp_off) images. [1]
x-probe	int	**-1**	x coord of diagnostic pixel. [-1]
y-probe	int	**-1**	y coord of diagnostic pixel. [-1]
threshold	int	**6e+04**	positive saturation level (for QC). [6e+04]
saturation_neg	double		**-4.5e+07** negative saturation level (for QC). [-4.5e+07]

12.6 eris_nix_cal_det

This recipe performs on the input observations the basic calibrations and the distortions are corrected.

12.6.1 Description

Each input frame is processed as follows: the master dark is subtracted, each pixel is linearized, the error associated to the data is determined, the result is divided by the flat-field, a bad pixel map is created using the input bad pixel map. Finally if `cd-matrix-modify` is set to true, the distortions are corrected by applying the CD-matrix parameters specified in the input `WCS_REFINE`.

For more details, please refer to section 11.1.3, point 1, at page 48.

12.6.2 Input Frames

Frame Tag	Type	Count	Description
OBJECT_JITTER	raw	1 (min)	Raw science frames

continued on next page

continued from previous page

Frame Tag	Type	Count	Description
BP_MAP_NL	cdb	1	not linear bad pixel map
COEFFS_CUBE	cdb	1	non-linear fit coefficients
DET_LIN_INFO	cdb	1	non-linearity information table
GAIN_INFO	cdb	1	non-linearity gain information table
MASTER_BPM_LAMP	cdb	1	master bad pixel map
MASTER_DARK_IMG	cdb	1	master dark image
MASTER_FLAT_LAMP_HIFREQ	cdb	1	master lamp flat (high frequency) image
MASTER_FLAT_LAMP_LOFREQ	cdb	1	master lamp flat (low frequency) image
MASTER_FLAT_TWILIGHT_LOFREQ	cdb	1	master lamp twilight (low frequency) image
WCS_REFINE	cdb	1	NIX wcs refinement file with updated CD-matrix

12.6.3 Product Frames

Default File Name	Frame Tag	Description
cal_det.<input_raw_frame_filename>	CAL_DET_OBJECT_JITTER	instrument signature corrected science frame

12.6.4 Quality Control Parameters

No quality control parameter are currently determined.

12.6.5 Recipe Parameters

Parameter	Type	Values	Description
fill-rejected	string	**set_value**	how to treat reject pixel. <set_value | set_NaN | noop>
fill-value	double	**0.0**	value to use in 'set_value' case.
cd-matrix-modify	boolean	**TRUE**	True to insert corrected CD-matrix.
x-probe	int	**-1**	x coord of diagnostic pixel.
y-probe	int	**-1**	y coord of diagnostic pixel.

The parameter cd-matrix-modify, when set to true, enable a correction of small distortions introduced by optics on the frames. The CD matrix coefficients have been determined on observations taken with Position Angle (PA) set to 0 degrees. If the user has observations taken with a different PA value, a more accurate astrometry calibration may be obtained if cd-matrix-modify is set to false.

12.7 eris_nix_img_skysub

The object frame(s) are corrected by the sky frame(s), using sky observations, if available, or using the object frame data after masking the object(s).

12.7.1 Description

For more details, please refer to section 11.1.3, point 2, at page 52.

12.7.2 Input Frames

Frame Tag	Type	Count	Description
CAL_DET_OBJECT_JITTER	chain	1 (min)	product of eris_nix_cal_det recipe

12.7.3 Product Frames

Default File Name	Frame Tag	Description
skysub.<input_raw_frame_filename>	SKYSUB_OBJECT_JITTER	sky corrected science frame

12.7.4 Quality Control Parameters

No quality control parameter are currently determined.

12.7.5 Recipe Parameters

Parameter	Type	Values	Description
catalogue.obj.min-pixels	int	**20**	Minimum pixel area for each detected object. [20]
catalogue.obj.threshold	double	**3.0**	Detection threshold in sigma above sky. [3.0]
catalogue.bkg.mesh-size	int	**64**	Background smoothing box size. [64]
catalogue.bkg.smooth-gauss-fwhm	double	**2.0**	The FWHM of the Gaussian kernel used in convolution for object detection. [2.0]
sky-source	string	**auto**, target, offset	data to be used for calculation of sky background.
sky-selector	string	**bracket**	method for selecting sky frames.

continued on next page

continued from previous page

Parameter	Type	Values	Description
sky-method	string	**collapse-median**, median-median	method for combining sky frames.
sky-bracket-time	double	**1.8e+03**	2 * max.time between target and sky measurement. [1.8e+03]
debug-data	boolean	**FALSE**	true to save interim results. [FALSE]
esoformat	boolean	**TRUE**	: TRUE to output MEF file conforming to ESO standard. [TRUE]

Please note that the parameters catalogue.bkg.mesh-size and catalogue.obj.core-radius defaults are set to optimally handle the cases JHK-13mas (no AO and FULL AO) and JHK-27mas (no AO and FULL AO).

if HIERARCH.ESO.OBS.AOMODE = `NO_AO`: default mesh-size = 128 pixels

if HIERARCH.ESO.OBS.AOMODE = `FULL_AO`: default mesh-size = 64 pixels

12.8 eris_nix_img_cal_wcs

This recipe performs the astrometry calibration.

12.8.1 Description

For more details, please refer to section 11.1.3, point 3, at page 56.

12.8.2 Input Frames

Frame Tag	Type	Count	Description
SKYSUB_OBJECT_JITTER	chain	1 (min)	product of eris_nix_cal_img_skysub recipe
WCS_MATCHED_CATALOGUE	ref	1	WCS catalogue

12.8.3 Product Frames

Default File Name	Frame Tag	Description
cal_wcs.<input_raw_frame_filename>	CAL_WCS_OBJECT_JITTER	wcs calibrated science frame
cat.cal_wcs.<input_raw_frame_filename>	CAL_WCS_CATALOGUE	wcs catalogue frame
matchcat.cal_wcs.<input_raw_frame_filename>	CAL_WCS_MATCHCAT	objects matching with wcs catalogue frame
refcat.cal_wcs.<input_raw_frame_filename>	CAL_WCS_REFCAT	reference WCS catalogue frame

12.8.4 Quality Control Parameters

No quality control parameter are currently determined.

12.8.5 Recipe Parameters

Parameter	Type	Values	Description
catalogue.obj.min-pixels	int	**20**	Minimum pixel area for each detected object. [20]
catalogue.obj.threshold	double	**3.0**	Detection threshold in sigma above sky. [3.0]
catalogue.obj.deblending	boolean	**TRUE**	Use deblending?. [TRUE]
catalogue.obj.core-radius	double	**5.0**	Value of Rcore in pixels. [5.0]
catalogue.bkg.estimate	boolean	**TRUE**	Estimate background from input, if false it is assumed input is already background corrected with median 0. [TRUE]
catalogue.bkg.mesh-size	int	**64**	Background smoothing box size. [64]
catalogue.bkg.smooth-gauss-fwhm	double	**2.0**	The FWHM of the Gaussian kernel used in convolution for object detection. [2.0]
catalogue.det.effective-gain	double	**3.0**	Detector gain value to rescale convert intensity to electrons. [3.0]
catalogue.det.saturation	double	**5e+03**	Detector saturation value. [5e+03]
cdssearch-astrom	string	**none**, 2mass, gaiaedr3	CDS astrometric catalogue.
pixel-radius	double	**5.0**	Max. distance between object and catalogue entry for association (pixels). [5.0]
strict-classification	boolean	**TRUE**	TRUE if objects to match must be classified stellar and round. [TRUE]
debug-data	boolean	**FALSE**	true to save interim results. [FALSE]

Please note that the parameters catalogue.bkg.mesh-size and catalogue.obj.core-radius defaults are set to optimally handle the cases JHK-13mas (no AO and FULL AO) and JHK-27mas (no AO and FULL AO).

if HIERARCH.ESO.OBS.AOMODE = `NO_AO`: default core-radius = 25 pixels and default mesh-size = 128 pixels

if HIERARCH.ESO.OBS.AOMODE = `FULL_AO`: default core-radius = 10 pixels and default mesh-size = 64 pixels

In certain cases, where the input frame may have a few sources adjacent one to another, reducing the value of the parameter catalogue.bkg.mesh-size may improve accuracy, in particular enable to distinguish between adjacent sources.

12.9 eris_nix_img_cal_phot

This recipe performs the photometry calibration.

12.9.1 Description

For more details, please refer to section 11.1.3, point 3, at page 64.

12.9.2 Input Frames

Frame Tag	Type	Count	Description
CAL_OBJ_OBJECT_JITTER	chain	1 (min)	product of eris_nix_cal_img_cal_wcs recipe
CAL_WCS_CATALOGUE	cdb	1	WCS catalogue product of eris_nix_cal_img_cal_wcs recipe
CAL_WCS_MATCHCAT	cdb	1	WCS match catalogue product of eris_nix_cal_img_cal_wcs recipe
CAL_WCS_REFCAT	cdb	1	WCS ref catalogueproduct of eris_nix_cal_img_cal_wcs recipe

12.9.3 Product Frames

Default File Name	Frame Tag	Description
cal_phot.<input_raw_frame_filename>	CAL_PHOT_OBJECT_JITTER	photometric calibrated science frame
cat.cal_phot.<input_raw_frame_filename>	CAL_PHOT_CATALOGUE	photometric catalogue frame
matchcat.cal_phot.<input_raw_frame_filename>	CAL_PHOT_MATCHCAT	objects matching with photometric catalogue frame
refcat.cal_phot.<input_raw_frame_filename>	CAL_PHOT_REFCAT	reference photometric catalogue frame

12.9.4 Quality Control Parameters

No quality control parameter are currently determined.

QC.MAGZPT	[mag] photometric zeropoint
QC.MAGZERR	[mag] photometric zeropoint

12.9.5 Recipe Parameters

Parameter	Type	Values	Description
cdssearch_photom	int	**2MASS**, none	CDS photometric catalogue.
catalogue.obj.min-pixels	int	**20**	Minimum pixel area for each detected object. [20]
catalogue.obj.threshold	double	**3.0**	Detection threshold in sigma above sky. [3.0]
catalogue.obj.deblending	boolean	**TRUE**	Use deblending?. [TRUE]
catalogue.obj.core-radius	double	**5.0**	Value of Rcore in pixels. [5.0]
catalogue.bkg.estimate	boolean	**TRUE**	Estimate background from input, if false it is assumed input is already background corrected with median 0. [TRUE]
catalogue.bkg.mesh-size	int	**64**	Background smoothing box size. [64]
catalogue.bkg.smooth-gauss-fwhm	double	**2.0**	The FWHM of the Gaussian kernel used in convolution for object detection. [2.0]
catalogue.det.effective-gain	double	**3.0**	Detector gain value to rescale convert intensity to electrons. [3.0]
catalogue.det.saturation	double	**5e+03**	Detector saturation value. [5e+03]
pixel-radius	double	**5.0**	Max. distance between object and catalogue entry for association (pixels). [5.0]
minphotom	int	**1**	Min number of matched stars for photometric calibration. [1]
magerrcut	double	**0.5**	Matched stars with magnitude error above this cutoff will not be used. [0.5]
debug-data	boolean	**FALSE**	true to save interim results. [FALSE]

Please note that the parameters catalogue.bkg.mesh-size and catalogue.obj.core-radius defaults are set to optimally handle the cases JHK-13mas (no AO and FULL AO) and JHK-27mas (no AO and FULL AO).

if HIERARCH.ESO.OBS.AOMODE = `NO_AO`: default core-radius = 25 pixels and default mesh-size = 128 pixels

if HIERARCH.ESO.OBS.AOMODE = `FULL_AO`: default core-radius = 10 pixels and default mesh-size = 64 pixels

12.10 eris_nix_img_hdrl_stack

This recipe stacks fully calibrated observations to maximise their Signal to Noise Ratio.

12.10.1 Description

For more details, please refer to section 11.1.3, point 3, at page 69.

12.10.2 Input Frames

Frame Tag	Type	Count	Description
CAL_PHOT_OBJECT_JITTER	chain	1 (min)	product of eris_nix_cal_img_cal_hot recipe
CAL_PHOT_CATALOGUE	cdb	1	photometric catalogue product of eris_nix_cal_img_cal_phot recipe
CAL_PHOT_MATCHCAT	cdb	1	photometric match catalogue product of eris_nix_cal_img_cal_phot recipe
CAL_PHOT_REFCAT	cdb	1	photometric ref catalogue product of eris_nix_cal_img_cal_phot recipe

12.10.3 Product Frames

Default File Name	Frame Tag	Description
cal_stack.<input_raw_frame_filename>	IMG_OBS_CATALOGUE	photometric calibrated science frame
stack.<input_raw_frame_filename>	IMG_OBS_COMBINED	stacked combined resampled frame

12.10.4 Quality Control Parameters

No quality control parameter are currently determined.

`QC.MAGZPT`	[mag] photometric zeropoint
`QC.MAGZERR`	[mag] photometric zeropoint

12.10.5 Recipe Parameters

Parameter	Type	Values	Description
interpolation-method	string	**lanczos**, nearest, linear, quadratic, renka, drizzle	The interpolation method.
loop-distance	int	**1**	maximum pixel offset taken into account. [1]
kernel-size	int	**2**	(Lanczos method) size of kernel in pixels. [2]

continued on next page

continued from previous page			
Parameter	**Type**	**Values**	**Description**
critical-radius	double	**5.0**	(Renka method) distance beyond which weights set to 0. [5.0]
pix-frac-x	double	**50.0**	(Drizzle method) percentage of flux to drizzle from original to target pixel. [50.0]
pix-frac-y	double	**50.0**	(Drizzle method) percentage of flux to drizzle from original to target pixel. [50.0]
catalogue.obj.min-pixels	int	**20**	Minimum pixel area for each detected object. [20]
catalogue.obj.threshold	double	**3.0**	Detection threshold in sigma above sky. [3.0]
catalogue.obj.deblending	boolean	**TRUE**	Use deblending?. [TRUE]
catalogue.obj.core-radius	double	**5.0**	Value of Rcore in pixels. [5.0]
catalogue.bkg.estimate	boolean	**TRUE**	Estimate background from input, if false it is assumed input is already background corrected with median 0. [TRUE]
catalogue.bkg.mesh-size	int	**64**	Background smoothing box size. [64]
catalogue.bkg.smooth-gauss-fwhm	double	**2.0**	The FWHM of the Gaussian kernel used in convolution for object detection. [2.0]
catalogue.det.effective-gain	double	**3.0**	Detector gain value to rescale convert intensity to electrons. [3.0]
catalogue.det.saturation	double	**5e+03**	Detector saturation value. [5e+03]
debug-data	boolean	**FALSE**	true to save interim results. [FALSE]

Please note that the parameters catalogue.bkg.mesh-size and catalogue.obj.core-radius defaults are set to optimally handle the cases JHK-13mas (no AO and FULL AO) and JHK-27mas (no AO and FULL AO).

if HIERARCH.ESO.OBS.AOMODE = `NO_AO`: default core-radius = 25 pixels and default mesh-size = 128 pixels

if HIERARCH.ESO.OBS.AOMODE = `FULL_AO`: default core-radius = 10 pixels and default mesh-size = 64 pixels

12.11 eris_nix_lss_skysub

12.11.1 Description

12.11.2 Input Frames

Frame Tag	Type	Count	Description
CAL_DET_OBJECT_LSS_JITTER	chain	1 (min)	product of eris_nix_cal_det recipe

12.11.3 Product Frames

Default File Name	Frame Tag	Description
skysub.<input_raw_frame_filename>	SKYSUB_OBJECT_LSS_JITTER	sky corrected science frame

12.11.4 Quality Control Parameters

No quality control parameter are currently determined.

12.11.5 Recipe Parameters

Parameter	Type	Values	Description
sky-source	string	**auto**, target, offset	data to be used for calculation of sky background.
sky-selector	string	**bracket**	method for selecting sky frames.
sky-method	string	**collapse-median**, median-median	method for combining sky frames.
sky-bracket-time	double	**1.8e+03**	2 * max.time between target and sky measurement. [1.8e+03]
debug-data	boolean	**FALSE**	true to save interim results. [FALSE]

12.12 eris_nix_lss_straighten

12.12.1 Description

12.12.2 Input Frames

Frame Tag	Type	Count	Description
SKYSUB_OBJECT_LSS_JITTER	chain	1 (min)	product of eris_nix_lss_skysub recipe
MASTER_STARTRACE	cdb	1	2d rectification and wavelength calibration static file

12.12.3 Product Frames

Default File Name	Frame Tag	Description
straightened.<input_raw_frame_filename>	CORRECTED_OBJECT_LSS_JITTER	rectified/wavelength-calibrated science frame

12.12.4 Quality Control Parameters

No quality control parameter are currently determined.

12.12.5 Recipe Parameters

Parameter	Type	Values	Description
debug-data	boolean	**FALSE**	true to save interim results. [FALSE]

12.13 eris_nix_lss_stack

12.13.1 Description

12.13.2 Input Frames

Frame Tag	Type	Count	Description
CORRECTED_OBJECT_LSS_JITTER	chain	1 (min)	product of eris_nix_lss_straighten recipe

12.13.3 Product Frames

Default File Name	Frame Tag	Description
stack.<input_raw_frame_filename>	LSS_OBJECT_COMBINED	stacked 2d spectrum

12.13.4 Quality Control Parameters

No quality control parameter are currently determined.

12.13.5 Recipe Parameters

Parameter	Type	Values	Description
x-probe	int	**-1**	x coord of diagnostic pixel. [-1]
y-probe	int	**-1**	y coord of diagnostic pixel. [-1]

12.14 eris_nix_pupil

This recipe is used only by Operations to remove the instrument signature from frames later used to perform quality control on NIX pupil images. It performs on the input observations the basic calibrations and the distortions are corrected, as the recipe `eris_nix_cal_det` (see Sec.), with the only difference that the input raw frames are different and have an appropriate frame tag, and the products have also peculiar tags.

12.14.1 Input Frames

Frame Tag	Type	Count	Description
PUPIL_LAMP or PUPIL_SKY or PUPIL_BKG	raw	1	Raw pupil frames
BP_MAP_NL	cdb	1	not linear bad pixel map
COEFFS_CUBE	cdb	1	non-linear fit coefficients
DET_LIN_INFO	cdb	1	non-linearity information table
GAIN_INFO	cdb	1	non-linearity gain information table
MASTER_BPM_LAMP	cdb	1	master bad pixel map
MASTER_DARK_IMG	cdb	1	master dark image
MASTER_FLAT_LAMP_HIFREQ	cdb	1	master lamp flat (high frequency) image
MASTER_FLAT_LAMP_LOFREQ	cdb	1	master lamp flat (low frequency) image
MASTER_FLAT_TWILIGHT_LOFREQ	cdb	1	master lamp twilight (low frequency) image
WCS_REFINE	cdb	1	NIX wcs refinement file with updated CD-matrix

12.14.2 Product Frames

Default File Name	Frame Tag	Description
cal_det.<input_raw_frame_filename>	DO_CATG_SPEC_OPEN	instrument signature corrected pupil frame DO_CATG is the input raw frame tag, SPEC is OPEN, CROSS, JHK or SAM

12.14.3 Quality Control Parameters

No quality control parameter are currently determined by this recipe.

12.14.4 Recipe Parameters

Parameter	Type	Values	Description
fill-rejected	string	**set_value**	how to treat reject pixel. <set_value | set_NaN | noop>
fill-value	double	**0.0**	value to use in 'set_value' case.
cd-matrix-modify	boolean	**FALSE**	True to insert corrected CD-matrix.
x-probe	int	**-1**	x coord of diagnostic pixel.
y-probe	int	**-1**	y coord of diagnostic pixel.

The parameter cd-matrix-modify, when set to true, enable a correction of small distortions introduced by optics on the frames. The CD matrix coefficients have been determined on observations taken with Position Angle (PA) set to 0 degrees. If the user has observations taken with a different PA value, a more accurate astrometry calibration may be obtained if cd-matrix-modify is set to false.

A Installation

ESO pipelines can be installed via several methods, depending on your OS, most of which facilitate easy installation, upgrade and removal. Please see the "ESO Data Reduction Pipelines and Workflow Systems" page (https://www.eso.org/pipelines).

A.1 System Requirements

The ERIS-NIX pipeline implements recipes and algorithms in a very efficient manner. The only system recommendation we would like to give to the user, as the implementation employes OpenMP instructions, is to run the recipe on a system with multi core architecture and enough RAM and disk space. For example the following requirements are recommended:

- 11 (minimum)/16 (better) GB of memory
- 4 CPU cores (physical cores)
- 1 (minimum)/(4 better) TB of free disk space
- GCC 8.3.1 (or newer)

We note that there was a NIX observation involving 88 frames, that required 38 GB to perform the sky subtraction, 31 GB to perform the HDRL stack, and 24 GB for the other recipes involved in the science processing. For most usual observations the RAM requirements are less demanding.

A.2 Installing the ERIS Pipeline

A.3 Building the ERIS Pipeline

Installation via RPM or MacPorts is recommended. In case the user platform is not one for which an RPM or MacPorts are provided the user may use the `install_esoreflex script.`

ESOReflex can be installed as:

```
./install_esoreflex
```

Then follow the instructions on the screen (selecting ERIS). Once the script finishes successfully and the path variables have been set, the installation of the ERIS pipeline ESOReflex workflow and the pipeline are complete.

A.3.1 Build Requirements

There are no particular build requirements for the ERIS pipeline. Installation based on the `install_esoreflex` script, requires the user has installed on her/his desktop certain softwre as describe at:

https://www.eso.org/sci/software/pipelines/installation/software_prerequisites.html.