

EXES Redux Pipeline Users Manual

SOF-US-HBK-OP10-2005

|  |  |
| --- | --- |
| **Date: March 30, 2015**  **Revision: -** |  |
|  | USRA-Logo |
| **nasa_meatball** | dlr_logo |
| AFRC  Armstrong Flight Research Center  Edwards, CA 93523 | German Space Agency, DLR  Deutsches Zentrum für Luft und  Raumfahrt |
| ARC  Ames Research Center  Moffett Field, CA 94035 |  |

EXES Redux Pipeline Users Manual

**SOF-US-HBK-OP10-2005**

AUTHOR:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Melanie Clarke, USRA, SOFIA Pipeline Engineer |  | Date |

CONCURRENCE:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| R.Y. Shuping, USRA, SOFIA DPS Lead |  | Date |

APPROVAL:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| William Reach, USRA, SOFIA Associate Director for Science |  | Date |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| William Szolnoki, USRA Business Manager |  | Date |

**REVISION HISTORY**

|  |  |  |
| --- | --- | --- |
| rev | date | DESCRIPTION |
| - | 3/30/15 | Initial Release |
|  |  |  |
|  |  |  |

EXES Redux Users Manual  
SOF-US-HBK-OP10-2005

Table of Contents

1 Introduction 1

2 SI Observing Modes Supported 1

2.1 EXES observing techniques 1

2.2 EXES instrument modes 2

3 algorithm description 2

3.1 Overview of data reduction steps 2

3.2 Reduction algorithms 3

3.2.1 Load Data 3

3.2.2 Flat Field Processing 7

3.2.3 Despike 9

3.2.4 Debounce 9

3.2.5 Subtract Sky 9

3.2.6 Flat Correct 11

3.2.7 Clean Bad Pixels 11

3.2.8 Undistort 11

3.2.9 Coadd Pairs 14

3.2.10 Spatial Profile Construction 15

3.2.11 Aperture Parameters 16

3.2.12 Spectral Extraction 16

3.2.13 Wavelength Calibration 17

3.2.14 Mosaicking spectra 18

3.3 Other Resources 19

4 Flux calibration 19

5 Data products 19

5.1.1 Filenames 19

5.1.2 Pipeline Products 20

5.1.3 Data Format 22

6 Grouping LEVEL\_1 data for processing 22

7 Configuration and execution 24

7.1 Installation 24

7.2 Configuration 24

7.3 Input data 25

7.4 Automatic mode execution 26

7.5 Manual mode execution 27

7.5.1 Basic workflow 27

7.5.2 Display features 31

7.6 EXES Reduction 33

8 Data quality assessment 37

9 Appendix A: Sample configuration files 39

EXES Redux Users Manual  
SOF-US-HBK-OP10-2005

# Introduction

The SI Pipeline Users Manual (OP10) is intended for use by both SOFIA Science Center staff during routine data processing and analysis, and also as a reference for General Investigators (GIs) and archive users to understand how the data in which they are interested was processed. This manual is intended to provide all the needed information to execute the SI Level 2 Pipeline, flux calibrate the results, and assess the data quality of the resulting products. It will also provide a description of the algorithms used by the pipeline and both the final and intermediate data products.

A description of the current pipeline capabilities, testing results, known issues, and installation procedures are documented in the SI Pipeline Software Version Description Document (SVDD, SW06, DOCREF). The overall Verification and Validation (V&V) approach can be found in the Data Processing System V&V Plan (SV01-2232). Both documents can be obtained from the SOFIA document library in Windchill at location: / [Software Management Development or Verification](https://sofiacm.arc.nasa.gov/Windchill/netmarkets/jsp/library/listFiles.jsp?oid=OR%3Awt.folder.Cabinet%3A9440896&u8=1) / Pipelines (DPS).

This manual applies to EXES Redux version 1.0.0 and later.

# SI Observing Modes Supported

## EXES observing techniques

EXES is a mid-infrared instrument, which implies that its observations must be taken with care in order to remove the effects of the spatially and temporally variable sky background. Unlike some other mid-infrared instruments, EXES does not make use of the chopping secondary mirror, but it does use telescope nods to alternate the object observation with a sky observation.

EXES has four basic nod patterns: stare, nod-on-slit, nod-off-slit, and map. The stare mode is simply a continuous observation of a source, with no telescope offsets to sky. This mode is primarily used for testing, rather than for science observations. In either of the nodding modes, the telescope moves in order to take frames at two different positions (the A position and the B position). For nod-along-slit mode, the frames alternate between placing the source at the bottom and top of the slit. This mode can be used for relatively compact sources (FWHM less than a quarter of the slit). For nod-off-slit mode, the frames alternate between the source and a patch of blank sky. This mode is useful for extended sources or crowded fields. Sky background for either nodding mode is removed by subtracting the two adjacent nod beams. Typically, EXES does a BA nod pattern, so that the nod pair is every second frame minus the previous frame. In map mode, the telescope starts in blank sky, then steps the source across the slit, and then returns to blank sky. The sky frames are averaged together to reduce their noise, and then subtracted from each source frame. This mode is used to map extended sources.

Before each observation sequence, EXES takes a series of calibration frames: typically a spectrum of a blackbody and a spectrum of the sky in the same configuration as the science observation. The sky frame is subtracted from the black frame, and the difference is used as a flat field for the 2D object spectrum. The black frame is also used to determine the edges of the orders in cross-dispersed modes.

## EXES instrument modes

EXES is a spectroscopic instrument that uses a slit to provide spatial dimension and gratings to provide a spectral dimension. EXES has two gratings used for science: the echelle grating and the echelon. The echelle grating, used alone, produces single-order longslit spectra. The Medium resolution mode of EXES uses the echelle grating as the primary disperser at angles 30-65 degrees; the Low resolution mode uses the echelle grating at angles less than 30 degrees. When the echelon grating is used as the primary disperser and the echelle grating is used as a secondary disperser, EXES produces multi-order, cross-dispersed spectra. In this mode, the wavelength coverage and the number of orders is determined by the echelle angle. The High-Medium resolution mode uses an echelle angle of 30-65 degrees. The High-Low resolution mode uses an angle of less than 30 degrees.

The slit length and width for all modes can be varied depending on the observational needs, although the length is limited in the cross-dispersed modes by the requirement that echelon orders must not overlap.

EXES does also have an imaging mode, but this mode is used primarily for testing and for acquiring targets for spectroscopic observations. It is not used for science, so it will not be addressed in this document.

# algorithm description

## Overview of data reduction steps

This section will describe, in general terms, the major algorithms used by Redux to reduce an EXES observation. See the figure below for a flow chart showing how these algorithms fit together. This section borrows heavily from the Spextool and TEXES papers; see the Other Resources section, below, for a reference to these papers.

All raw files are first processed as 2D images, with algorithms from the EXES IDL package. Spectroscopy files then undergo spectral extraction and combination of the resulting one-dimensional spectra, using algorithms from the FSpextool reduction package.



Figure : Processing steps for spectral data. Steps in the blue box are carried out by algorithms in the EXES package. Steps in the red box are carried out by algorithms in the FSpextool package.

## Reduction algorithms

The following subsections detail each of the data reduction pipeline steps outlined in the flowchart above.

### Load Data

Raw EXES data is stored in FITS image files, as a 3D data cube in which each plane corresponds to a readout of the array (digitization). Readouts can be destructive, or non-destructive. The readout method can vary for EXES, so the pattern of readout actions used for a particular observation is recorded in the OTPAT header keyword. The value for OTPAT is a combination of the letters S, T, N, D, and C, each with a digit afterwards. The letters have the following meaning:

* S = spin (no reset, no digitization). This action effectively increases the integration time.
* T = trash (reset, no digitization). This action resets the array without storing the previous value.
* N = non-destructive read (no reset, but digitized). This action stores the current value of the array.
* D = destructive read (reset and digitization). This action stores the current value, and resets the array.
* C = coadd (reset and digitization). This action coadds previous reads in hardware and resets the array.

The digit after each letter gives the number of times the action was taken, minus one. For example: S0 is one spin, N3 is four non-destructive reads.

The pattern listed in the OTPAT can then be repeated any number of times (*npat*), so that the final number of frames in the raw file is , where *nnondest* is the number of nondestructive reads and *ndest* is the number of destructive reads per pattern.

Since near-infrared detectors do not have completely linear responses to incoming flux, each frame in the raw file should be corrected for nonlinearity before proceeding. The nonlinearity coefficients for EXES were determined by taking a series of nondestructive flat exposures, followed by a single destructive read. Each readout was subtracted from a bias frame, then the resulting counts in each pixel were fit with a polynomial. The linearity correction algorithm reads in these polynomial coefficients from a 3D array and uses them to calculate a correction factor for each pixel in each raw readout. Pixels with counts below the bias level, or above the maximum value used to calculate the nonlinearity coefficients at that pixel, have no correction applied.

After linearity correction, in order to create a usable estimate of the input flux, the non-destructive and destructive reads must be combined in a way that maximizes the signal-to-noise. The EXES pipeline currently has support for combining frames taken with the patterns listed in the following subsections. Equations for the net signal and the associated variance come from Vacca et. al., 2004 (see Other Resources below). After this initial calculation of the variance for each image pixel, the variance plane is propagated along with the image through all pipeline steps.

After the frames for each pattern are combined according to the algorithms below, the pipeline checks whether multiple patterns were taken at the same nod position. This is recorded in the header keyword NINT. If NINT is greater than 1, then NINT coadded patterns are averaged together before proceeding.

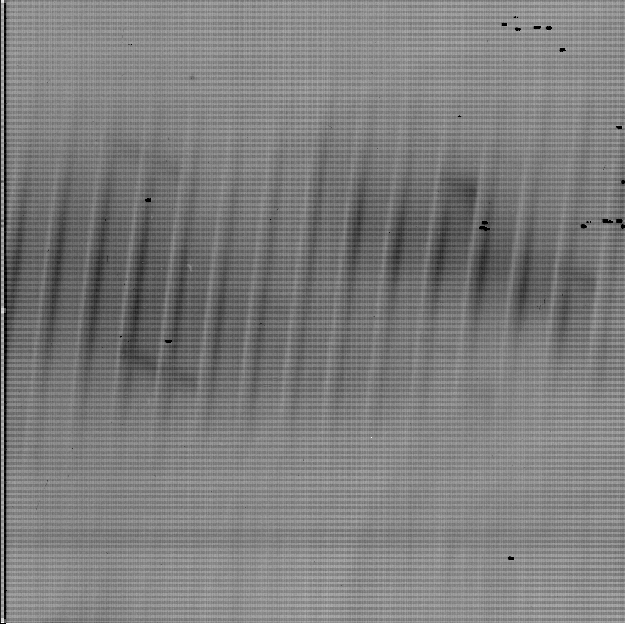
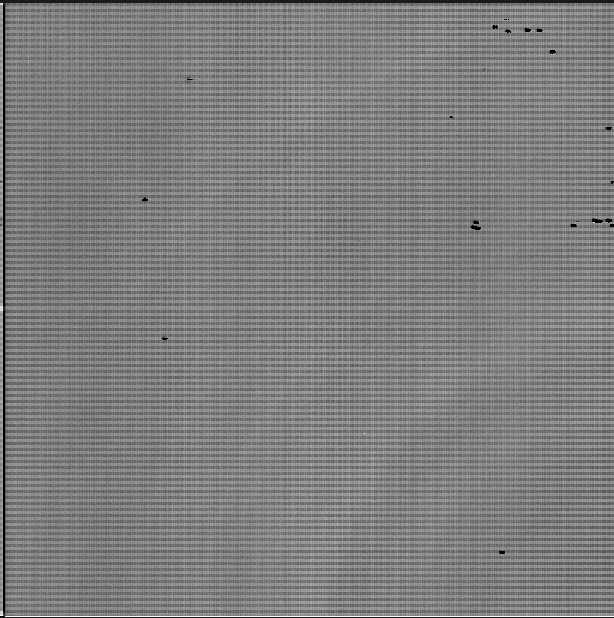
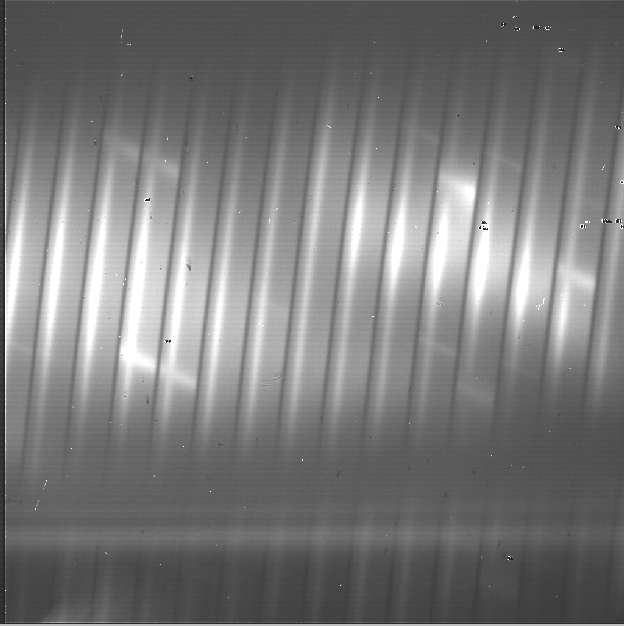


Figure 2: Sample raw frames in nod-off-slit, High-Medium mode. A pedestal image (top left), a final destructive read (top right), and the Fowler coadd (bottom). The bright spectrum is visible as a dark line in the final raw readout, and a bright line in the coadd.

#### All destructive mode

In this mode, each frame is a destructive read. There may be spins in between reads to increase integration time, but there are no non-destructive reads. In this case, the flux is simply the read signal (*s*), minus a standard bias frame (*b*), stored on disk, subtracted from a zero value (*z*), corresponding to a dark current, and divided by the integration time (*Δt*) and the pre-amp gain factor (*gp*):

.

The dark current is typically zero for EXES.

The variance associated with this readout mode is the Poisson noise (first term) plus the readout noise (second term):

,

where *σread* is the read noise, and *g* is the gain factor (to convert from ADU to electrons)

#### Fowler mode

In this mode, a number of non-destructive reads are taken at the beginning of the integration, the array integrates for a while, then a number of non-destructive reads and a single destructive read are taken at the end of the integration. The initial reads are averaged together to estimate the pedestal level; the final reads are averaged together to estimate the total signal. The pedestal is then subtracted from the total signal to get the net signal. We then subtract the negative signal from a zero value (*z*) and divide by the pre-amp gain factor (*gp*) and the integration time (*Δt*, the time between the first pedestal read and the first signal read). The net flux is then:

.

We assume that the number of reads (*nr*) is the same for the pedestal and the signal.

The variance associated with this readout mode is:

.

Here, *δt* is the frame time, *σread* is the read noise, and *g* is the gain factor (to convert from ADU to electrons).

Sample OTPAT values for Fowler mode look like this:

* N0 D0: a single non-destructive/destructive pair. This is the minimal Fowler case: the first read is the pedestal, the second read is the signal. There are 2 frames per pattern.
* N3 S15 N2 D0: a Fowler 4 set. This set comprises 4 non-destructive reads, 16 spins to increase the integration time, 3 non-destructive reads, and a destructive read. There are 8 frames per pattern.

#### Sample-up-the-ramp mode

In this mode, non-destructive reads are taken at even intervals throughout the integration. The count values for all reads are fit with a straight line; the slope of this line gives the net flux. As with the other modes, we then subtract from the zero value and divide by the pre-amp gain and the integration time. The net flux for the evenly-sampled case is calculated as:

,

where *nr* is the total number of reads (non-destructive and destructive).

The variance associated with this readout mode is:

.

Sample OTPAT values for sample-up-the-ramp mode look like this:

* N0 S3 N0 S3 N0 S3 D0: Sample-up-the-ramp with 4 spins in between each non-destructive read, ending with a destructive read. There are 4 frames per pattern.

#### Coadding

If the OT pattern contains a coadd (C), it is treated as a destructive read for which the appropriate combination has already been done. That is, if the pattern indicates Fowler mode, for example, the intensity is simply calculated as for a destructive read:

,

but the variance is calculated from the net intensity as for the Fowler mode, above.

### Flat Field Processing

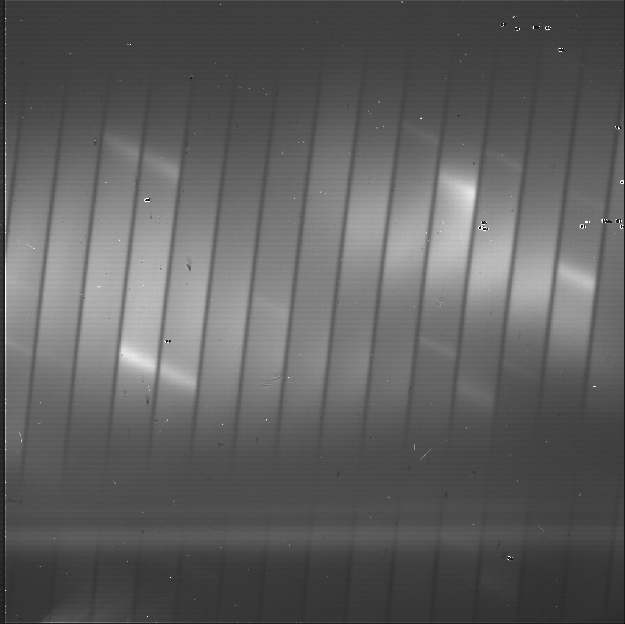
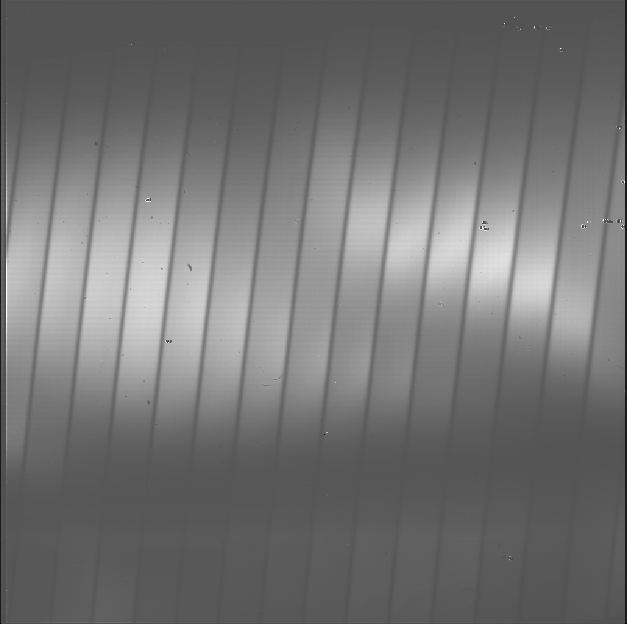
Each EXES reduction requires a flat observation taken in the same mode as the science data. The instrument configuration (eg. HIGH\_MED or MEDIUM), central wavenumber, slit length, slit width, and grating angle must all match. This flat observation is used to correct for spectral and spatial instrumental gain variations, and to calibrate the source intensity of the science observation.

The flat observation consists of an observation of a black-body in EXES’s calibration box (*black*) and a blank sky region (*sky*). If the black-body, the sky, and the telescope are all at the same temperature, then dividing the sky-subtracted science frame by *black-sky*, normalized by the black-body function, gives a calibrated intensity as follows (Lacy et. al., 2002, see Other Resources, below).

where *Sν* is the measured signal, *Bν(T)* is the black-body function at temperature *T*, *ε* is the emissivity of the telescope or sky, *Rν* is the instrumental responsivity, and *t=1-ε* is the transmission of the telescope or sky. The master flat frame produced by the EXES pipeline, therefore is the blackbody function calculated at the temperature recorded in the flat frame’s header (BB\_TEMP), divided by *black-sky*.

In all instrument configurations, the *black* frame is also used to determine an illumination mask that defines the good pixels for extraction. In the cross-dispersed modes, the *black* frame is further used to determine the edges of the orders for extraction and to check the optical distortion parameters. The pipeline cleans and undistorts the *black* frame, takes its derivative, then does a 2D FFT of the resulting image to determine the spacing and orientation of the orders on the array. In particular, the value of *krot*, the angle of the order rotation is adjusted automatically at this step: it is calculated from the characteristics of the FFT, then used to recompute the undistortion of the *black* frame, and redo the FFT, until the value of *krot* converges, or 5 iterations are reached. This process must be closely monitored: if the *black* frame has insufficient signal, or the optical parameters used to calculate the distortion correction are insufficiently accurate, the spacing and rotation angle may be wrongly calculated at this step. These values can be manually overridden in parameters for the pipeline if necessary.

The distortion parameters and order definitions, as determined from the *black* frame, are written to the header of the master flat, to be used in later reduction steps.



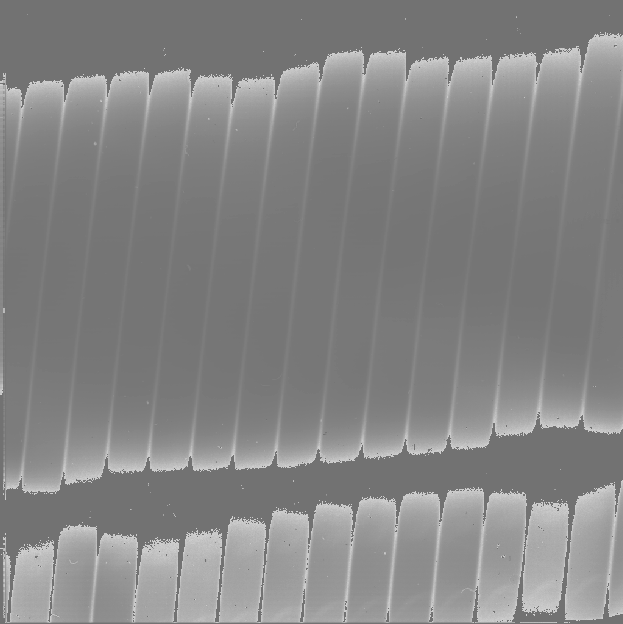


Figure 3: Sample flat in High-Medium mode. Top left is the *black* frame, top right is the *sky* frame. Bottom is the final processed calibration frame. Unilluminated regions are set to zero in the final frame.

### Despike

After coadding the raw frames to make A and B nod pairs, the pipeline attempts to identify and correct spikes (outlier data points) in the individual images. All A frames are compared, and any pixels with values greater than a threshold number of standard deviations from the mean value across the other frames are replaced with that mean value. B frames are similarly compared with each other. The threshold for identifying spikes is a tunable parameter; the default value is 20 sigma. Also in the despike step, frames with significantly different overall background levels (“trashed” frames) may be identified automatically and removed from subsequent reduction. It is common, for example, for the first frame to look significantly different from the rest of the frames. In this case, leaving out this frame may improve the signal-to-noise of the final result.

### Debounce

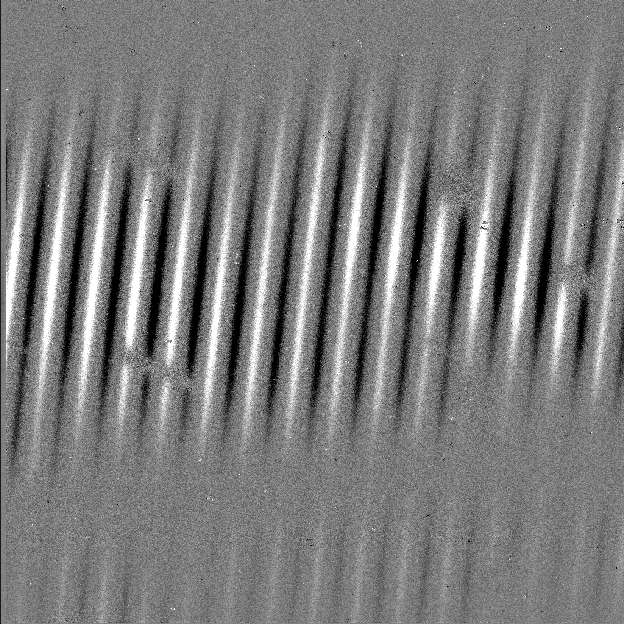
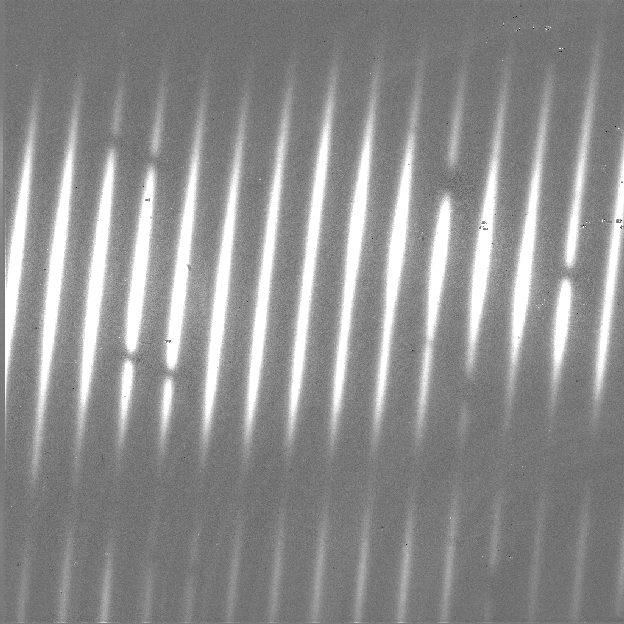
After despiking, there is an optional step called debouncing, which may help alleviate the effects of small optics shifts (“bounces”) between the nod beams. If there is a slight mismatch in the order placement on the array, it can lead to poor background subtraction when the nods are subtracted. In the debouncing algorithm, each nod pair is undistorted, then the B nod is shifted slightly in the spatial direction and differenced with the A nod. The shift direction that results in a minimum squared difference (summed over the spectral direction) is used as the bounce direction. The amplitude of the shift is controlled by the bounce parameter, which should be set to a number whose absolute value is between 0 and 1 (typically 0.1). If the bounce parameter is set to a positive number, only the above shift (the first-derivative bounce) will be corrected. If the bounce parameter is set to a negative value (e.g. -0.1), the debouncing algorithm will also attempt to fix the second-derivative bounce by smoothing the A or B nod slightly; the amount of smoothing is also controlled by the absolute value of the bounce parameter. Note that if the observed source is too near the edge of the order, it may confuse the debouncing algorithm; in this case, it is usually best to turn debouncing off (i.e. set the bounce parameter to 0). It is not yet known whether optics bounces will be common with EXES mounted on SOFIA, so the current default is not to use the debounce algorithm.

### Subtract Sky

In either nod-on-slit or nod-off-slit mode, each B nod is subtracted from each adjacent A nod. This step usually removes most of the background emission from the image, but if there were changes in the sky level between the two nod frames, there may still be some residual sky signal. For the nod-off-slit mode, this residual signal can be estimated and corrected for before subtracting the nods, by subtracting a weighted fraction of the B data from the A data. The weighting is chosen to minimize the squared difference between the A and B nods. For the nod-on-slit mode, the mean background at each wavelength may be subtracted after nod subtraction and distortion correction, so that the wavelengths align with the rows or columns. The pipeline performs this step immediately before coadding, if desired.

Figure 4: Background-subtracted frames in nod-off-slit (left) and nod-on-slit (right)

High-Medium modes.



For the mapping mode, each of the steps across the source is treated as an A frame. The three sky frames taken at the end of the map are averaged together and this average is subtracted from each A frame. This mode is usually used for extended sources that fill the entire slit, in which case there is no way to estimate or correct for any residual background remaining after sky subtraction.

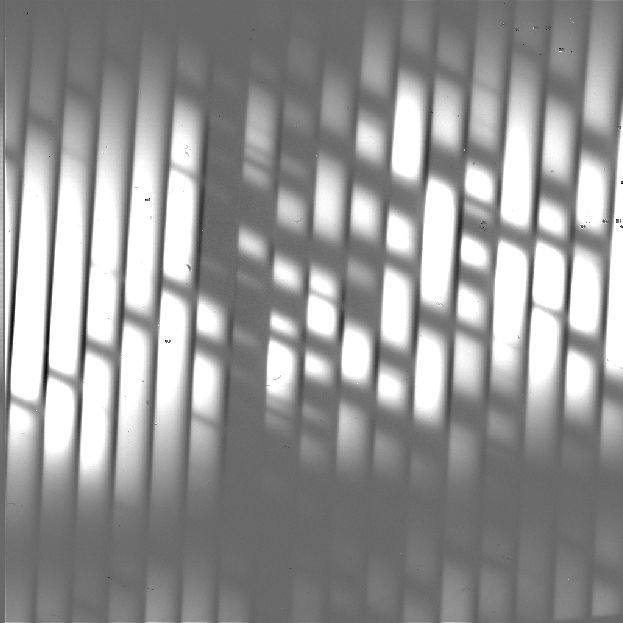


Figure 5: Background-subtracted image in High-Medium map mode. The extended source fills the entire slit.

### Flat Correct

After background subtraction, each science frame is multiplied by the processed flat calibration frame, described above. This has the effect of both correcting the science frame for instrumental response and calibrating it to intensity units.

### Clean Bad Pixels

In this step, bad pixels are identified and corrected. Bad pixels may first be identified in a bad pixel mask, provided by the instrument team. In this FITS image, pixels that are known to be bad are marked with a value of 0; good pixels are marked with a value of 1. Alternately, bad pixels may be identified from their noise characteristics: if the standard deviation associated with a pixel is greater than a threshold value (by default: 20) times the mean standard deviation for the frame, then it is marked as a bad pixel.

Bad pixels are corrected by using neighboring good values to linearly interpolate over the bad ones. The search for good pixels checks first in the y-direction, then in the x-direction. If good pixels cannot be identified within a 10-pixel radius, then the bad pixel will not be corrected. Note that bad pixels are also fixed or ignored in the extraction process, so it is not critical to correct all bad pixels at this stage.

### Undistort

Next, the pipeline corrects the image for optical distortion, resampling the image onto a regular grid of spatial and spectral elements. The EXES distortion correction uses knowledge of the optical design of the instrument, as well as data recorded in the header of the observation, to calculate and correct for the optical distortion in each observation.

The optical parameters used to calculate the distortion correction are listed in the table below. Most of these parameters should not change, or should change rarely, so they have default values listed in configuration tables available to the pipeline (the *.dat* files listed below). Values in the *tortparm.dat* configuration table tend to change over time, as the instrument is adjusted, so their defaults vary by date. Some of the distortion parameters must come from the headers and do not have default values. One parameter is tuned at run-time (*krot*), and one must be manually optimized by the user (*waveno0*).

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Default value** | **Source** | **Comment** |
| *hrfl* | (varies by date) | *tortparm.dat* | High resolution chamber focal length |
| *xdfl* | (varies by date) | *tortparm.dat* | Assumed cross-dispersed focal length |
| *slitrot* | (varies by date) | *tortparm.dat* | Slit rotation |
| *brl* | 0.00 | *tortparm.dat* | Barrel distortion parameter |
| *x0brl* | 0.00 | *tortparm.dat* | X-coordinate for the center of the barrel distortion |
| *y0brl* | 0.00 | *tortparm.dat* | Y-coordinate for the center of the barrel distortion |
| *hrg* | -0.015 | *headerdef.dat* | Gamma (out of plane) angle for the echelon grating (radians) |
| *xdg* | 0.033 | *headerdef.dat* | Gamma angle for cross-dispersion grating (radians) |
| *xdlrdgr* | 0.001328 | *headerdef.dat* | Low resolution groove spacing (cm) |
| *xdmrdgr* | 0.003151 | *headerdef.dat* | Medium resolution groove spacing (cm) |
| *pixelwd* | 0.0025 | *headerdef.dat* | Pixel width |
| *detrot* | 0.10 | *headerdef.dat* | Detector rotation |
| *hrr* | 9.8 | *headerdef.dat /*  Recalculated from header data | R number for echelon grating |
| *waveno0* | (none) | Header keyword WAVENO0 | Planned central wavenumber |
| *echelle* | (none) | Header keyword ECHELLE | Echelle grating angle (deg) |
| *krot* | (varies by date) | *tortparm.dat /*  Optimized at run-time | Rotation between chambers |

Table : Optical distortion parameters. *tortparm.dat* and *headerdef.dat* are configuration tables distributed along with the software code. Values in *tortparm.dat* are expected to change somewhat over time; values in *headerdef.dat* are expected to remain constant.

Several of these parameters are then combined or recalculated before being used in the distortion correction. The cross-dispersed R number (*xdr*) is calculated from the groove spacing (*xdlrdgr* or *xdmrdgr*), the grating angle (*echelle*), and the central wavenumber (*waveno0*). The echelon R number (*hrr*) is recalculated from the above default number, the echelon gamma angle (*hrg*), and the central wavenumber. The expected order spacing for cross-dispersed modes is calculated from the cross-dispersed R number (*xdr*), the cross-dispersed focal length (*xdfl*), the pixel width (*pixelwd*), and the echelon gamma angle (*hrg*). The order spacing and the rotation angle (*krot*) are optimized for cross-dispersed modes when the calibration frame is processed, as described above.

The distortion correction equations calculate undistorted coordinates by correcting for the following effects for the cross-dispersed modes:

* slit skewing within orders due to the echelon “smile” (*hrg, hrr, slitrot)*
* nonlinearity of the echelon spectrum (*hrr, pixelwd, hrfl)*
* skewing by spectrum rotation on the detector due to the angle between the chambers, and the cross-dispersion smile (*xdg, xdr, krot, pixelwd, xdfl, hrfl, hrr)*
* nonlinearity of the cross-dispersion spectrum (*xdr, pixelwd, xdfl)*
* barrel distortion (*brl, x0brl, y0brl*)

For the long-slit modes, the distortion correction accounts for:

* skewing by the cross-dispersion smile (*xdg, xdr, slitrot, pixelwd, xdfl)*
* nonlinearity of the cross-dispersion spectrum (*xdr, pixelwd, xdfl)*
* barrel distortion (*brl, x0brl, y0brl*)

When the undistorted coordinates have been calculated, the 2D science image is interpolated from raw *(x,y)* pixel coordinates onto these coordinates. By default, a cubic interpolation is used that closely approximates a sinc interpolation function; bilinear interpolation is also available.

After distortion correction, the spatial and spectral elements should be aligned with the columns and rows of the image. For the long-slit modes, the x-axis is the dispersion axis; for cross-dispersed modes, the y-axis is the dispersion axis.

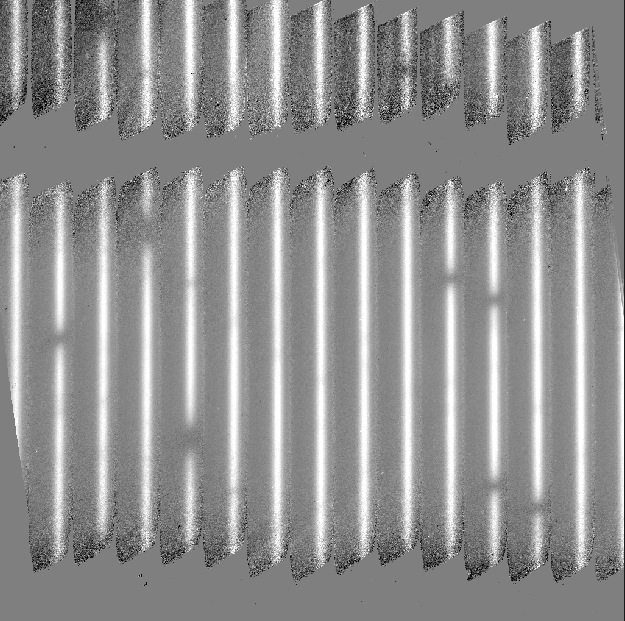
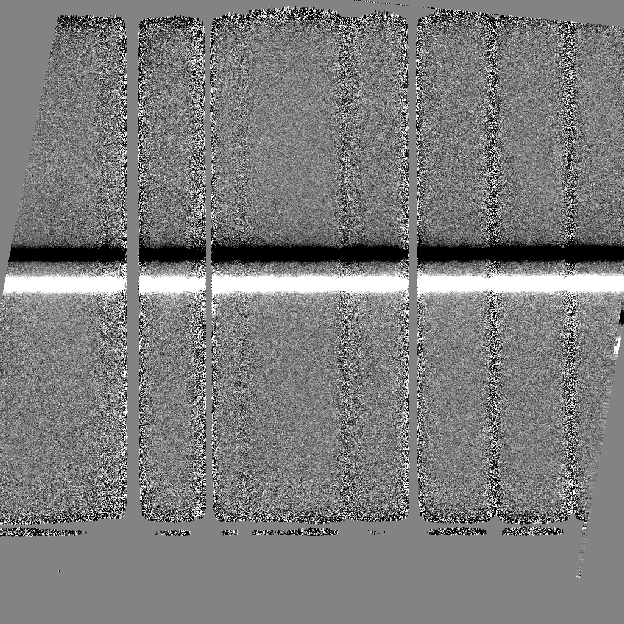


Figure 6: Undistorted images in High-Medium mode (left) and Medium mode (right).

### Coadd Pairs

The final step in the image processing prior to spectral extraction is coadding of the background-subtracted, undistorted frames. If desired, coaddition can be skipped, and individual spectra can be extracted from each frame, to be combined later. For faint sources, however, coaddition of the 2D image is recommended, for more accurate extraction.

Prior to coadding, the spectra can be shifted in the spatial direction to account for slight shifts in the position of the object in the slit between nod pairs. The shift is calculated by creating a template of the spectrum from all frames, averaged in the spectral direction, and cross-correlating each frame with the template.

By default, the pipeline weights all input frames by their correlation with this spectral template, so that if a frame is unusually noisy, or missed a nod, or had some other error, it will not contribute significantly to the coadded frame. However, it is possible to force the pipeline to use an unweighted coadd, if desired, or to specify the weights to use directly, or to explicitly identify frames for exclusion from the coadd.

The final action in the coadd step is to rotate images taken in the cross-dispersed mode in order to align the spectra with the image rows. This enables FSpextool to handle the spectral extraction in a consistent way for both the long-slit and the cross-dispersed data.



Figure 7: Coadded image in High-Medium mode, rotated to align the dispersion axis with the image rows.

### Spatial Profile Construction

In order to aid in spectral extraction, FSpextool constructs a smoothed model of the relative intensity of the target spectrum at each spatial position, for each wavelength. This spatial profile is used to compute the weights in optimal extraction or to fix bad pixels in standard extraction (see below). Also, the user must refer to the median profile, collapsed along the wavelength axis, to define the extraction parameters. Since FSpextool rotates input data so that dispersion is along the x-axis, we will here refer to a wavelength point as a column, and a spatial point as a row.

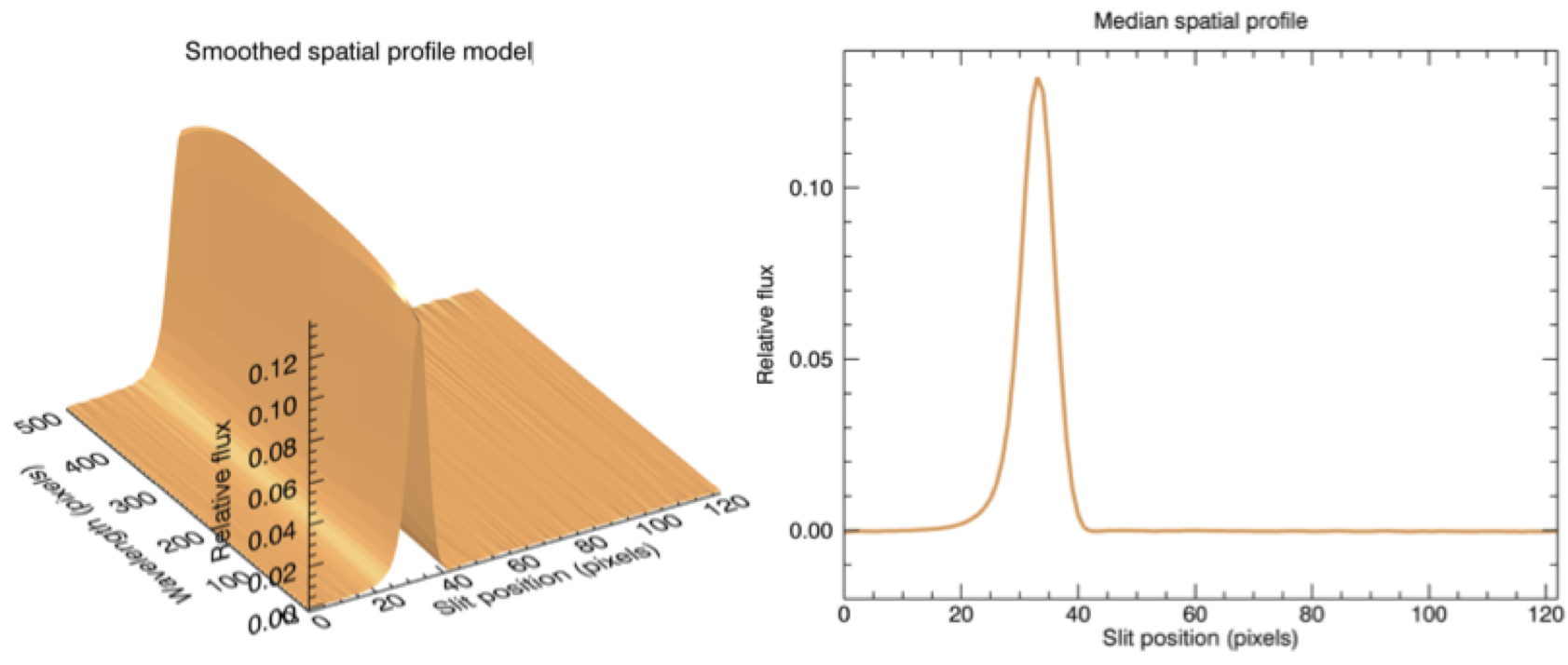
To construct the spatial profile, FSpextool first subtracts the median signal from each column in a spectral order to remove the residual background. It then resamples the order onto a regular grid of wavelength columns and spatial rows; this has the effect of removing any curvature of the order across the detector. The intensity in this resampled image in column i and row s is given by

where *fi* is the total intensity of the spectrum at wavelength *i*, and *Pi,s* is the spatial profile at column *i* and row *s*. To get the spatial profile *Pi,s*, we must approximate the intensity *fi*. To do so, FSpextool computes a median of the columns of the order image to get a first-order approximation of the median spatial profile *Ps*. Assuming that

,

FSpextool uses a linear least-squares algorithm to fit *Ps* to *Oi,s* and thereby determine the coefficients *ci*. These coefficients are then used as the first-order approximation to *fi*: the resampled order image *Oi,s* is divided by *fi* to derive *Pi,s*.  FSpextool then fits a low-order polynomial along the columns at each spatial point *s* in order to smooth the profile and thereby increase its signal-to-noise. The coefficients of these fits can then be used to determine the value of *Pi,s* at any column *i* and spatial point *s* (see Figure 3, left). The median of *Pi,s* along the wavelength axis generates the median spatial profile, *Ps* (see Figure 3, right). Finally, *Pi,s* is resampled back onto the pixel grid (*i*,*j*) to derive a profile *Pi,j* that matches the coordinates of the raw image.

Figure : Spatial profile image as a function of wavelength and slit position (left), and median spatial profile as a function of slit position (right)



### Aperture Parameters

The median spatial profile can now be used to identify extraction apertures for the target. The aperture centers can be identified automatically by iteratively finding local maxima in the absolute value of the spatial profile, or can be specified directly by the user.

Besides the aperture centers, FSpextool also requires that the user specify a few more aperture parameters. For point sources, optimal extraction requires a PSF radius, corresponding to the distance from the center at which the flux from the source falls to zero. For all extractions (point source or extended source, optimal or standard), FSpextool requires an aperture radius, corresponding to the edge of the extraction aperture. It is also possible to specify a background region outside of any extraction apertures, for fitting and removing the residual sky signal. These parameters may also be estimated automatically, from a Gaussian fit to the spatial profile, or may be specified directly by the user.

### Spectral Extraction

Before extracting a spectrum, FSpextool first uses the identified background regions to find the residual sky background level. For each column, it fits a low-order polynomial to the values in the specified regions, as a function of slit position. This polynomial determines the wavelength-dependent sky level (*Ri,j*) to be subtracted from the spectrum (*Di,j*).

FSpextool offers two methods for extracting a spectrum from the sky-subtracted image: optimal and standard extraction. The standard extraction method uses values from the spatial profile image (*Pi,j*) to replace bad pixels and outliers, then sums the flux from all pixels contained within the aperture radius. The flux and variance at column *i* is then:

where *j1* and *j2* are the upper and lower limits of the extraction aperture (in pixels), and *εi,j* is the fraction of the pixel *(i, j)* that is contained within the extraction aperture. This extraction method is the only algorithm available for extended sources.

Point sources may occasionally benefit from using standard extraction, but optimal extraction generally produces higher signal-to-noise ratios for these targets. This method works by weighting each pixel in the extraction aperture by how much of the target’s flux it contains. FSpextool first normalizes the spatial profile by the sum of the spatial profile within the PSF radius defined by the user:

where *k1* and *k2* are the upper and lower limits given by the PSF radius. *P’i,j* now represents the fraction of the total flux from the target that is contained within pixel *(i,j)*, so that *(Di,j –Ri,j)/ P’i,j* is a set of *j* independent estimates of the total flux at column *i*. FSpextool does a weighted average of these estimates, where the weight depends on the pixel’s variance (*VDij + VRij*), as well as the normalized spatial profile (*P’i,j* ). Then the flux and variance at column *i* is:

where is a bad pixel mask and *j1* and *j2* are the upper and lower limits given by the aperture radius. Note that bad pixels are simply ignored, and outliers will have little effect on the average because of the weighting scheme.

### Wavelength Calibration

FSpextool can extract spectra from images where the dispersion axis is well aligned with the image rows. In this case, each column represents a single wavelength bin. The wavelength calibration is one-dimensional, and can be derived from an extracted spectrum of an arc lamp or sky lines. FSpextool can also extract spectra from images where the dispersion axis does not align with the image rows: that is, there may be a significant curvature or tilt to the spectral lines with respect to the columns. In this case, the wavelength calibration must be two-dimensional, and can be derived from a two-dimensional spectral image of an arc or sky lines. FSpextool provides an interactive GUI, called *xwavecal2d*, which allows the user to identify spectral lines at regular intervals across the image, fit a surface to the wavelength and spatial data, and produce a map of the wavelength and spatial coordinates at each pixel.

Two-dimensional wavelength calibration requires an extra step in the extraction process, in order to ensure that each pixel contains only data from a single wavelength bin. FSpextool accomplishes this by first interpolating the x and y image coordinates onto a regularized grid of wavelength and spatial coordinates, as read from the wavelength calibration map. For each wavelength bin in this grid, it steps through the spatial elements, adding up the flux of any pixel or partial pixel contained within the given spatial and wavelength bin (the shaded area in the figure below). The resultant sums construct the flux profile with respect to the spatial coordinate, analogous to a slice along a column in the case of one-dimensional wavelength calibration. This profile is then used to perform either standard or optimal extraction as detailed above.

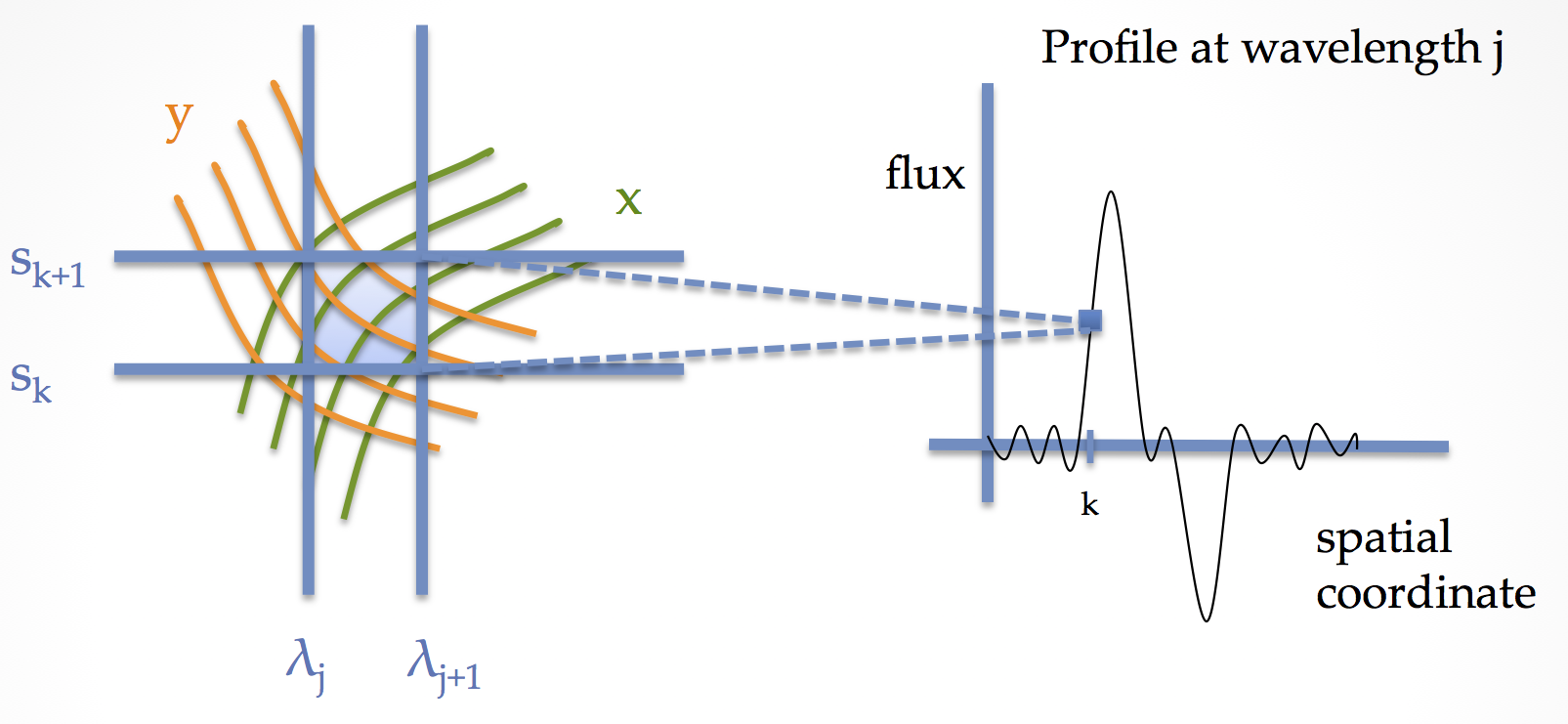


Figure 9: Extraction with two-dimensional wavelength calibration. The flux for the pixel at spatial element k and wavelength j is the weighted sum of the flux from pixels contained, or partially contained, within the grid element.

For EXES, the 2D spectral images are rectified before extraction, so two-dimensional wavelength calibration maps are not required: the dispersion axis should be aligned with the image rows after the distortion correction. Wavelength calibration is calculated from the grating equation, using the assumed central wavelength, and stored as a wavelength calibration map identifying the wavelength for each column in each spectral order.

The wavelength calibration is expected to be good to within one pixel of dispersion, if the central wavelength is accurate. To refine the wavelength solution, the pipeline allows the user to identify a single spectral feature in the extracted spectrum. Using this feature, the central wavelength is recalculated and the wavelength calibration is adjusted to match. However, the distortion correction also depends on the central wavelength, so if the recalculated value is significantly different from the assumed value, the reduction should be repeated with the new value to ensure an accurate distortion correction and dispersion solution.

### Mosaicking spectra

It may sometimes be useful to stitch together separate spectra to view the entire wavelength coverage of an observation in a single plot. FSpextool provides a tool called *xmergeorders* to perform this task. This tool can, for example, combine all orders of a cross-dispersed spectrum into a single spectrum, using the overlapping regions to scale the spectra to each other. It is also possible to combine spectra extracted from separate observations with different central wavenumbers, although it will be more successful if there is some overlap in the wavelength coverage.

As a final step for cross-dispersed spectra, the pipeline performs an automatic merge of the spectra from all orders. This algorithm uses the signal-to-noise ratio in overlapping regions to determine which pixels to exclude, and performs a weighted mean of any remaining overlapping regions. No scaling is performed. This merged product is intended as a quick reference spectrum and may contain artifacts like discontinuous jumps between orders. A more accurate merge may be performed manually with the *xmergeorders* tool.

## Other Resources

For more information on the reduction algorithms used in the EXES package, adapted from the TEXES pipeline, see the TEXES paper:

[TEXES: A Sensitive High-Resolution Grating Spectrograph for the Mid-Infrared](http://arxiv.org/pdf/astro-ph/0110521v1.pdf), J.H. Lacy, M.J. Richter, T.K. Greathouse, D.T. Jaffe and Q. Zhu (2002, PASP 114, 153)

For more information on the reduction algorithms used in FSpextool, see the Spextool papers:

[Spextool: A Spectral Extraction Package for SpeX, a 0.8-5.5 Micron Cross-Dispersed Spectrograph](http://irtfweb.ifa.hawaii.edu/~spex/Spextool.pdf), Michael C. Cushing, William D. Vacca and John T. Rayner  (2004, PASP 116, 362).

[A Method of Correcting Near-Infrared Spectra for Telluric Absorption](http://irtfweb.ifa.hawaii.edu/~spex/Telluric.pdf), William D. Vacca, Michael C. Cushing and John T. Rayner (2003, PASP 115, 389).

[Nonlinearity Corrections and Statistical Uncertainties Associated with Near-Infrared Arrays](http://irtfweb.ifa.hawaii.edu/~spex/Nonlinearity.pdf), William D. Vacca, Michael C. Cushing and John T. Rayner  (2004, PASP 116, 352).

For more information on how to run the FSpextool interactive tools (*xmergeorders, ximgtool*, *xvspec*, *xwavecal2d, xcombspec*, *xtellcor\_general*, and *xcleanspec*), see the help files distributed with the FSpextool code, under *fspextool/Helpfiles*.

# Flux calibration

EXES spectra are calibrated to physical intensity units (*erg/s/cm2/sr/cm-1*) when they are multiplied by the blackbody-normalized calibration frame. This process also partially corrects for atmospheric absorption, but there are frequently residual atmospheric effects that must be corrected for. These should be possible to remove by dividing by a telluric standard spectrum. FSpextool provides a general-purpose GUI called *xtellcor\_general*, which can be used to perform this task.

# Data products

### Filenames

EXES output files from Redux are named according to the convention:

*FILENAME = Flight\_EX\_SPE\_AOR-ID\_SPECTEL1SPECTEL2\_Type\_FN1[-FN2][\_SN].fits*,

where Flight is the SOFIA flight number, EX is the instrument identifier, SPE specifies that it is a spectral file, AOR-ID is the 8 digit AOR identifier for the observation, SPECTEL1SPECTEL2 are the keywords specifying the spectral elements used, Type is three letters identifying the product type (listed in the table below), and FN1 is the file number corresponding to the input file. FN1-FN2 is used if there are multiple input files for a single output file, where FN1 is the file number of the first input file and FN2 is the file number of the last input file.  If the output file corresponds to a single frame of the input file, there may also be a trailing serial number (SN) that identifies the input frame associated with the output file.

### Pipeline Products

The following table lists all intermediate products generated by Redux for EXES spectral modes, in the order in which they are produced. The product type is stored in the FITS headers under the keyword PRODTYPE. By default, the *raw*, *undistorted, coadded, spec, combspec, mrgordspec, flat*, *mask,* and *wavemap* products are saved. Specifying the appropriate option in either the automatic or interactive modes will save all products. If a subarray readout was used, the second dimension for the image products may be smaller than 1024.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Product Type** | **File code** | **Saved by default** | **Size** |
| Load Data | Raw frames coadded according to OTPAT; variance planes appended at end of all raw frames | *raw* | RAW | Y | 1024×1024×(2 Npat),  where Npat is the number of readout patterns |
| Make Flat | Calibration frame created from *black-sky* | *inverseflat* | IFT | N | 1024×1024×2 |
| Despike | Outlier pixels identified and corrected. Comparison is across similar frames | *despiked* | DSP | N | 1024×1024×(2 Npat) |
| Debounce | Corrected for optics shifts between nods | *debounced* | DBC | N | 1024×1024×(2 Npat) |
| Subtract Nods | B nods or sky frames subtracted from A nods or map steps | *nodsubtracted* | NSB | N | 1024×1024×(2 NA),  where NA is the number of A nods or map steps |
| Flat Correct | Multiplied by flat | *flatted* | FTD | N | 1024×1024×(2 NA) |
| Clean Bad Pixels | Bad pixels identified from variance or bad pixel mask | *cleaned* | CLN | N | 1024×1024×(2 NA) |
| Undistort | Corrected for optical distortion | *undistorted* | UND | Y | 1024×1024×(2 NA) |
| Undistort | Undistorted flat frame (1/inverseflat) with order definition keywords | *flat* | FLT | Y | 1024×1024×2 |
| Coadd Pairs | All frames averaged together, if desired | *coadded* | COA | Y | 1024×1024×(2 Nc),  where Nc is the number of coadded frames  (typically, Nc=1) |
| Coadd Pairs | All non-coadded frames separated into different files | *image* | IMG | N | 1024×1024×2 |
| Extract Spectra | 1D extracted spectra from all orders and apertures. | *spec* | SPC | Y | Nw×4×(Norder Nap),  where Nw is the number of wavelength bins, Norder is the number of orders, Nap is the number of apertures |
| Extract Spectra | Bad pixel mask, indicating pixels ignored during extraction | *mask* | MSK | Y | 1024x1024 |
| Combine Spectra | 1D spectra from separate files or apertures combined together | *combspec* | CMB | Y | Nw×4×Norder |
| Merge Orders | 1D spectrum from all cross-dispersed orders merged together | *mrgordspec* | MRD | Y | Nmw×4,  where Nmw is the number of merged wavelength bins |

Table : Final and intermediate data products

### Data Format

All files produced by the pipeline are FITS single-extension image files. All image products are 3-D arrays of data, where the first half of the planes are the image frames and the second half of the planes are the variance associated with each pixel in each frame. The extracted spectral products are one-dimensional spectra, stored in four rows of data. The first row is the wavelength, the second is the flux, the third is the error (standard deviation), and the fourth is a reference modeled atmospheric transmission spectrum. If there were multiple orders in the spectrum (e.g. the High-Medium mode) or multiple apertures selected (e.g. for the nod-on-slit mode), then the spectrum for each aperture and order is stacked into a different plane. The length of the row varies depending on the data, but is typically around 1000 pixels before merging.

# Grouping LEVEL\_1 data for processing

For EXES spectroscopy modes, there are only two kinds of data: flats and sources. Flat frames are required for all reductions, as they are used to calibrate the data, correct for instrumental response, and identify the edges of the slit. Flat frames should have the same instrument configuration, slit length, slit width, and grating angle as the science observation. They should also be taken near the same altitude and zenith angle and in the same mission in order for the sky frame to match the science sky conditions. In order for separate science files to be grouped together, these same conditions apply, and they should additionally share a common target and be taken with the same nodding mode. These requirements translate into a set of FITS header keywords that must match in order for a set of data to be usefully reduced together: see the figure and table below for a listing of these keywords.

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Datatype** | **Match Criterion** |
| **DATATYPE** | STR | Exact |
| **OBSTYPE** | STR | Exact |
| **OBJECT** | STR | Exact |
| **INSTMODE** | STR | Exact |
| **INSTCFG** | STR | Exact |
| **SPECTEL1** | STR | Exact |
| **SPECTEL2** | STR | Exact |
| **SLIT** | STR | Exact |
| **ECHELLE** | FLT | Exact |
| **SDEG** | FLT | Exact |
| **WAVENO0** | FLT | Exact |
| **MISSN-ID** | STR | Exact |
| **ALTI\_STA** | FLT | +/- 500 |
| **ALTI\_END** | FLT | +/- 500 |
| **ZA\_START** | FLT | +/- 2.5 |
| **ZA\_END** | FLT | +/- 2.5 |
| **PLANID** | STR | Exact |
| **AOR-ID (optional)** | STR | Exact |
| **LASTREW (optional)** | STR (DATE/TIME) | Exact |

Table : Grouping criteria

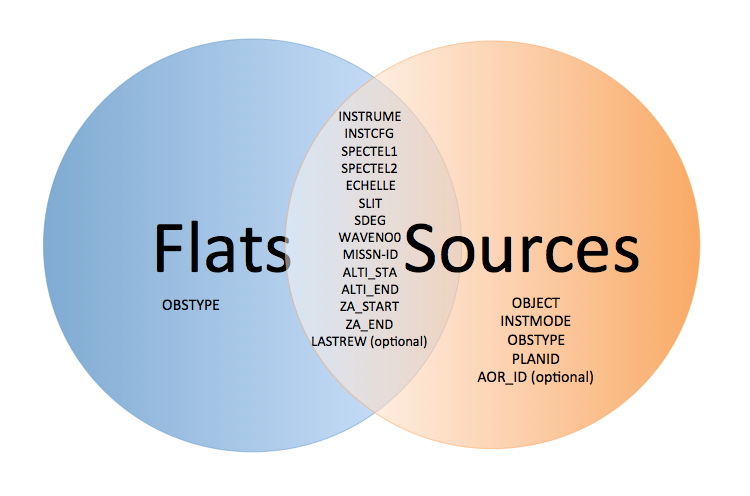


Figure : Keywords required for grouping. Keywords in the Flats region must match across all flat files; keywords in the Sources region must match across all source files. Keywords in the overlap region must match across all files (flats and sources).

# Configuration and execution

## Installation

Redux is a software package written in IDL that is designed to be a framework for executing any number or combination of data reduction algorithms. For EXES, it has been developed to support seamlessly running image processing algorithms from the EXES IDL package, translated from the FORTRAN pipeline developed for the TEXES instrument, alongside spectral extraction algorithms from the FSpextool package originally developed for the SpeX instrument. Redux can run in an automatic batch mode, integrated with the SOFIA Data Pipeline System (DPS), or it can run with a graphical front end as a quick-look data viewer in flight or during manual data reduction and analysis. Redux with EXES and FSpextool was developed under Linux and MacOS X operating systems. Other operating systems may also work, but have not been tested.

Running Redux requires IDL 8.1 or later, as well as the latest version of the IDL Astronomy User’s Library, the Coyote graphics library, the EXES package, the FSpextool package, and the Redux code. EXES, FSpextool, and Redux are under SOFIA DPS revision control and can be obtained directly from git repositories there. The IDL Astronomy User’s Library (astrolib) is publicly available, and can be downloaded from the website at the following URL:

<http://idlastro.gsfc.nasa.gov/homepage.html>.

The Coyote graphics library (coyote) is also publicly available and can be downloaded from:

<http://www.idlcoyote.com/documents/programs.php>.

When these packages have been installed, their locations should be added to the IDL\_PATH environment variable, so that their procedures are accessible to Redux.

SOFIA may distribute the Redux, EXES, and FSpextool codes as gzipped tar files. If so, unpack them, as, for example:

tar xvzf redux.tar.gz

tar xvzf exes.tar.gz

tar xvzf fspextool.tar.gz

This will create directories called *redux*, *exes*, and *fspextool*, which will contain a number of subdirectories. Each of these package directories should be added to the IDL\_PATH as well.

## Configuration

For EXES algorithms, default parameter values are specified in a configuration table called *headerdef.dat*, located in the *exes* package directory. This table lists the keyword name, whether it is a value required to be present in the input headers, its default value, the data type of the value, and any requirements on the value range (minimum value, maximum value, or enumerated value. A sample of this configuration file is given in Appendix A. All keywords in the table will be written to output files produced by the EXES package.

For FSpextool algorithms, default options are specified in a configuration file called *EXES.dat*, located in the *fspextool* package directory. This file contains keyword-value pairs, in the format *parameter=value*. The parameters must all be present and in the correct order, but can have any number of spaces or comments between them. Comment lines begin with the *%* or *#* character. See Appendix A for a sample of this configuration file as well.

In automatic pipeline mode, parameters set in the configuration files are the values actually used by the pipeline, unless overridden by an input parameter file. In interactive mode, the configuration files set the default values, but the parameter values used can be modified at run-time.

## Input data

Redux takes as input raw EXES FITS data files, which are image cubes composed of 1024x1024 pixel image arrays. The number of frames per raw data cube depends on the readout mode used to acquire the data. The FITS headers contain data acquisition and observation parameters and, combined with the pipeline configuration files, comprise the information necessary to complete all steps of the data reduction process. Some critical keywords are required to be present in the raw data in order to perform a successful grouping, reduction, and ingestion into the SOFIA archive (see the table below, and a full description in the EXES configuration table in Appendix A).

|  |  |
| --- | --- |
| **Keyword** | **Required for:** |
| ALTI\_END | Grouping |
| ALTI\_STA | Grouping |
| AOR\_ID | Grouping |
| BB\_TEMP | Data reduction |
| DATASRC | Archiving |
| DATE-OBS | Archiving |
| ECHELLE | Grouping, data reduction |
| EPERADU | Data reduction |
| ECTPAT | Data reduction |
| FILENAME | Archiving |
| FRAMETIM | Data reduction |
| INSTCFG | Grouping, data reduction |
| INSTMODE | Grouping, data reduction |
| INSTRUME | Archiving, grouping |
| MISSN-ID | Archiving, grouping |
| NINT | Data reduction |
| OBJECT | Grouping |
| OBS\_ID | Archiving |
| OBSTYPE | Grouping, data reduction |
| OTPAT | Data reduction |
| PAGAIN | Data reduction |
| PLANID | Archiving |
| PROCSTAT | Archiving |
| SDEG | Grouping, data reduction |
| SLIT | Grouping |
| SPECTEL1 | Archiving |
| SPECTEL2 | Archiving |
| SRCTYPE | Data reduction |
| WAVENO0 | Grouping, data reduction |
| ZA\_END | Grouping |
| ZA\_START | Grouping |

Table : Required FITS header keywords

It is assumed that the input data have been successfully grouped before beginning reduction: Redux considers all input files in a reduction to be part of a single homogeneous reduction group, to be reduced together with the same parameters. A single input flat frame is required to complete a reduction past the Clean Bad Pixels step; Redux will abort at the Undistort step if a flat frame is not provided along with the science data.

## Automatic mode execution

Redux is an object-oriented program whose basic unit is a reduction object (*exes\_reduction\_\_define.pro*). To run the pipeline from the IDL command line as a DCS black box pipeline, run the pipeline wrapper (*redux\_pipe.pro*). This wrapper takes as input the path to an input manifest file. This text file should contain a line specifying the number of input files, then the relative path to each input file, one per line. The script then reads these input files, instantiates the appropriate reduction object according to the mode specified in the input FITS headers, then calls the object’s reduce method. This method calls each processing step in order, as appropriate for the given mode. Finally, the wrapper script will write an output manifest called *outfiles.txt* containing the names of the produced data files, and will save the as-run parameters to a file called *redux\_param.json*.

This wrapper can be invoked from the IDL prompt, as

IDL> redux\_pipe, ‘infiles.txt’

or directly from a terminal as

$ echo “redux\_pipe, ‘infiles.txt’” | idl

The wrapper accepts a single input parameter on the command line, which allows the user to give it a parameter file that specifies any desired reduction parameters. This option is given as, for example:

IDL> redux\_pipe,‘infiles.txt’,PARAM\_FILE=‘exes\_param.json’

where *exes\_param.json* is a file that lists parameter keywords and values in JSON format for any step in the pipeline reduction. Parameter files can be generated interactively with the GUI (see below), then saved and fed to the automatic pipeline for batch reduction of a large number of files.

## Manual mode execution

It is also possible to run the pipeline interactively, using a graphical user interface. The IDL command redux, called without arguments, will launch the Redux GUI.[[1]](#footnote-2)

### Basic workflow

To start an interactive reduction, open a set of EXES files, using the File menu (**File->Open New Reduction**). This will bring up a file dialog window (see figure, below). All files selected will be reduced together as a single reduction set. For EXES, a flat file is required to complete a science reduction, so an appropriate flat should be selected along with the science data.

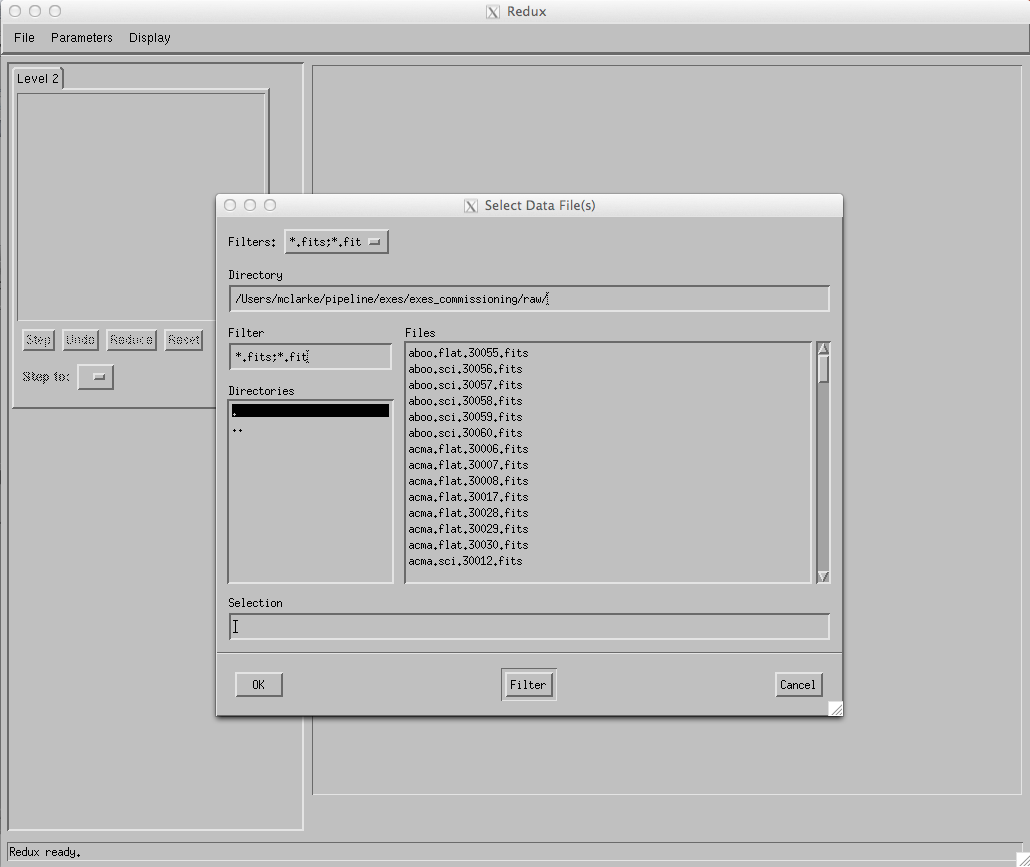


Figure : Open New Reduction

Redux will decide the appropriate reduction steps from the input files, and load them into the GUI, as in the figure below.



Figure : Sample reduction in High-Medium mode.

Each reduction step has a number of parameters that can be edited before running the step. To examine or edit these parameters, click the **Edit Param** button next to the step name to bring up the parameter editor for that step. Within the parameter editor, all values may be edited, but will not be used unless **Save** or **Done** is selected. Clicking **Save** will leave the parameter editor window open; clicking **Done** will save values and close the window. Clicking **Reset** will restore any edited values to their defaults; clicking **Cancel** will discard all changes to the parameters.

The current set of parameters can be displayed, saved to a file, or reset all at once using the **Parameters** menu. A previously saved set of parameters can also be restored for use with the current reduction (**Parameters->Load Parameters**). Note that edited parameters will retain their values for future reductions unless they are manually reset.

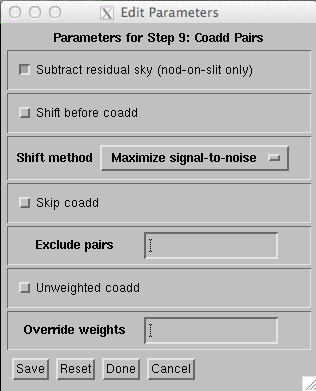


Figure : Sample parameter editor (for Coadd Pairs step)

After all parameters for a step have been examined and set to the user’s satisfaction, a processing step can be run on all loaded files either by clicking **Step**, or the **Run** button next to the step name. Each processing step must be run in order, but if a processing step is selected in the **Step to:** widget, then clicking **Step** will treat all steps up through the selected step as a single step. When a step has been completed, its buttons will be grayed out and inaccessible. It is possible to undo one previous step by clicking **Undo**. All remaining steps can be run at once by clicking **Reduce**. After each step, the results of the processing will be displayed in the display window. Clicking **Reset** will restore the reduction to the initial state, without resetting parameter values.

Files can be added to the reduction step (**File->Add Files**) or removed from the reduction set (**File->Remove Files**), but either action will reset the reduction for all loaded files. Selecting **Display->Display File Information**, or the **More Info** button, will pull up a table of information about the currently loaded files (Figure 14). The table rows displayed can be filtered by entering a search string into the **Filter** text box.

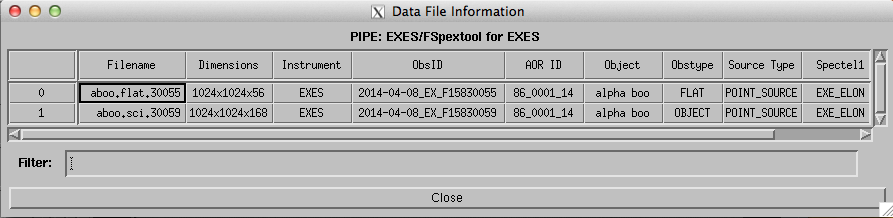


Figure : File information table

### Display features

Redux displays images using ximgtool, a full-featured display tool distributed with FSpextool. For more information, see the ximgtool help file, available from Redux via the **Help** button just above the display. See below for a quick listing of the most useful ximgtool features.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Menu button** | **Keyboard shortcut** |
| Load new file | **File->Load FITS** | -- |
| Load file into new frame | **File->New Frame** | -- |
| View FITS header | **File->View Header** | -- |
| Zoom | **Zoom->Zoom In**, **Zoom Out**, **Zoom To Fit** | Press **z** to enter zoom mode, then **i** to zoom in, **o** to zoom out, or **t** to zoom to fit |
| Color stretch | **Cursor->Stretch** | Press **s** to enter stretch mode, click and drag to change brightness and contrast |
| Set display range | **Cursor->Range** | Press **r** to enter range mode, click and drag to select the box that sets the display range |
| Display distance | -- | Press **d** to enter distance mode, then click and drag to identify start and end points |
| Line cut | -- | Press **l** to enter line cut mode, then click and drag to identify start and end points |
| Display image statistics | -- | Press **m** to enter moments mode, then click and drag to identify box for which the statistics should be calculated |
| Photometry | -- | Press **a** over a star to do basic photometry. |
| Clear current mode | -- | Press **c** |
| Buffer select | **Buffer->Buffer 1, Buffer 2…** | Press **f** to move to the next buffer, **b** to move to the previous buffer. |
| Buffer math | **Buffer->Buffer Math**, then select buffers and arithmetic operation to perform | -- |
| Blink buffers | **Buffer->Blink Buffers** | -- |
| Cycle frames | **Buffer->Cycle frames** | Press **n** to move to the next available frame, **p** to move to the previous frame |

Table : Useful ximgtool features

Ximgtool has five buffers available for simultaneous display of images. If there are more than five files loaded into Redux, they can only be viewed by selecting **Display->Quick Look** from the Redux menu. This will cycle through the data in its current processing state, allowing interaction and analysis with each image in turn. To move between images, click the **Next File** or **Previous File** buttons, below the image. Click **Cancel** to quit the quick look display.

Median spatial profiles generated from the 2D spectra are displayed in an independent plot window, with a separate plot for each order. When the aperture parameters are set, they will be overplot on the spatial profile in this window.

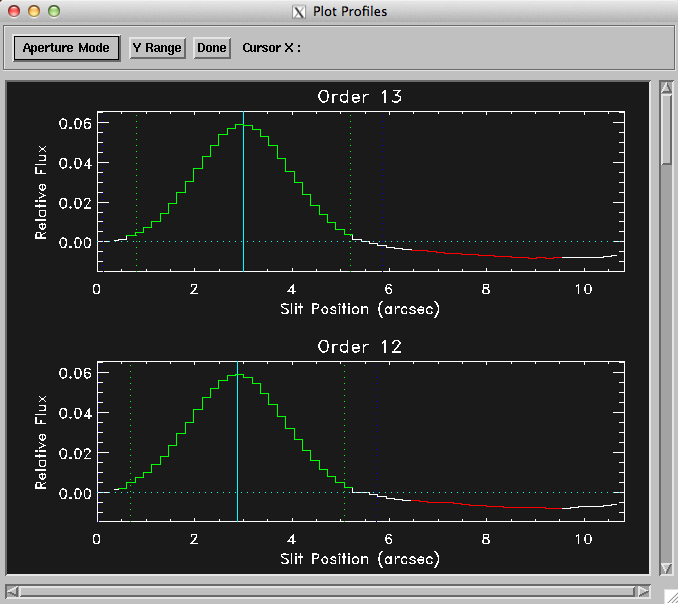


Figure : Aperture locations identified and overplotted on the spatial profile for each order. Aperture centers are identified with light blue lines, aperture radius with green lines, PSF radius with dark blue lines, and background regions with red lines.

Extracted spectra are displayed using xvspec (Figure 16), a display tool packaged with FSpextool. By default, xvspec displays one spectrum at a time, but each loaded spectrum can be examined by using the **Quick Look** feature, or else additional spectra may be directly loaded for display by clicking **Add Spectra**. For more information on xvspec features, use the Help button in the xvspec window. EXES spectra contain a reference atmospheric transmission spectrum; this may be accessed by clicking **Transmission**, or overplotted on the flux spectrum by selecting **Plot->Overplot->Transmission**.



Figure : Final extracted spectrum for each order, displayed in xvspec.

## EXES Reduction

EXES data reduction with Redux follows the data reduction flowchart given in section 3. At each step, Redux attempts to determine automatically the correct action, given the input data and default parameters, but each step can be customized as needed.

Some key parameters to note are listed below.

* **Load Data**:
  + *Check headers*: By default, Redux will abort reduction if the input header keywords do not meet requirements. Uncheck this box to attempt the reduction anyway.
  + *Save all intermediate files*: Check this box to save a file after each reduction step.
  + *Central wavenumber*: If the wavenumber is known more precisely than the value in the WAVENO0 keyword, enter it here to use it in the distortion correction calculations and wavelength calibration. This value will be filled in automatically at the end of a reduction, if the Refine Wavecal step modifies the central wavenumber, so that the reduction may be reset and re-run with the new parameter.
  + *HR focal length*: This distortion parameter may be adjusted to tune the dispersion solution for cross-dispersed spectra. Reducing HRFL decreases the wavenumber of the left end of the order.
  + *XD focal length*: This distortion parameter may be adjusted to tune the dispersion solution for long-slit spectra. Reducing XDFL decreases the wavenumber of the left end of the order for a correctly identified central wavenumber. This parameter may also be adjusted to tune the predicted spacing between orders in cross-dispersed mode. In this case, increasing XDFL increases the predicted spacing.
  + *Slit rotation*: This distortion parameter may be adjusted to tune the slit skewing, to correct for spectral features that appear tilted across an order. For cross-dispersed modes, increasing the slit rotation angle rotates the lines clockwise in the undistorted image. For long-slit modes, decreasing the slit rotation angle rotates the lines clockwise in the undistorted image.
* **Make Flat**:
  + *Threshold factor*: This value defines the illumination threshold for a flat, which determines where the edges of cross-dispersed orders are placed. Enter a number between 0 and 1. If the pipeline reports that it can not determine order edges, increasing the threshold slightly, to widen the gaps between orders, may help.
  + *Optimize rotation angle*: Uncheck this box to prevent the pipeline from using the 2D FFT of the edge-enhanced flat to try to determine the best rotation angle (*krot*).
  + *Edge enhancement method*: Depending on the noise and illumination characteristics of the input flat, it may work better to take the 2D FFT of the squared derivative, or the Sobel filtered image, rather than the derivative, of the flat. Use the drop-list to select an alternate algorithm.
  + *Starting rotation angle*: Enter a number here to use as the starting value for *krot.* Otherwise, the starting value will be taken from the default parameters in configuration tables. After the Make Flat step is run, this value will be automatically filled in with the best calculated value.
  + *Predicted spacing*: Enter a value here, in pixels, to use as the predicted spacing for cross-dispersed orders. This number will be used as a first guess for the spacing parameter in the fit to the 2D FFT, overriding the value calculated from header parameters.
  + *Debug*: Check this box to view diagnostic images to determine whether the distortion parameters are being set correctly. If checked, Redux will display the undistorted flat, the derivative of the flat, the 2D FFT of the derivative, the fit region for the fundamental and second harmonic of the 2D FFT (this should show two round peaks, if all went well), and the illumination mask resulting from the order determination. If *Debug* is checked, Redux will raise an error and refuse to continue reduction: uncheck *Debug* and run the step again to continue with processing.
* **Despike**:
  + *Spike factor*: Enter a value for the threshold for a pixel to be considered a spike. This value is multiplied by the standard deviation of the pixel across all frames: e.g. a value of 20 marks pixels with values over 20 sigma as bad pixels.
  + *Mark trashed frames for exclusion*: If checked, Redux will identify frames that have significantly more noise compared to the other frames and exclude them from future processing. These frames will not be spike-corrected.
* **Debounce**:
  + *Bounce factor*: Enter a number to use as the amplitude of the bounce correction. A value of 0 means no bounce correction will be applied. Values greater than 0 apply the first derivative bounce (shifting) only. Values less than 0 also apply the second derivative bounce (smoothing). Typical nonzero values are 0.1 or -0.1.
* **Subtract Nods**:
  + *Skip nod subtraction*: If checked, this option will cause the pipeline to continue without performing nod-subtraction. This option may be used to extract reference sky spectra.
  + *Subtract residual sky*: This option applies only to nod-off-slit mode. If checked, it will attempt to remove any residual sky signal before subtracting pairs. This option will remove any average continuum value along the slit; if this is not desired, background regions should be set at the time of extraction.
* **Clean Bad Pixels**:
  + *Bad pixel threshold*: Enter a value for the threshold for a pixel to be considered a bad pixel. This value is multiplied by the standard deviation of all good pixels in the frame.
* **Coadd Pairs**:
  + *Subtract residual sky*: This option applies only to nod-on-slit mode. If checked, the mean value at each wavelength bin will be subtracted to remove any residual sky noise.
  + *Shift before coadd*: Check to attempt to shift spectra in the spatial direction to align with each other before coadding.
  + *Shift method*: Select whether to maximize the signal or the signal-to-noise when determining the spatial shift.
  + *Skip coadd*: Check to skip coadding images and instead extract separate spectra from each frame. If selected, each frame will be treated as a separate file after this step. For map mode, coadding is always skipped.
  + *Exclude pairs*: Enter comma-separated numbers to identify specific nod-subtracted frames to exclude from the coadd. For example, entering *1,4,7* will exclude the first, fourth, and seventh frame from the coadd. If multiple files are loaded, lists for each file may be entered, separated by a semi-colon. For example, *1,4;;3,10* will exclude the first and fourth frames from the coadd for the first file and the third and tenth from the coadd for the third file.
  + *Unweighted coadd*: Check to force the pipeline to use equal weights for all input frames.
  + *Override weights*: To override the default calculated weights, enter a fractional weight for each input frame, separated by commas. If multiple files are loaded, lists for each file may be entered, separated by a semi-colon. Weights must add to 1.0 for each file.
* **Make Profiles**:
  + *Subtract median background*: In nod-on-slit mode or in long-slit observations of non-extended sources, the median level of the spatial profile is subtracted to account for a residual background level. If this is not desired (eg. it results in negative values in the spatial profile), uncheck this box.
* **Find Objects**:
  + *Number of auto apertures*: Enter the number of apertures to try to find automatically. The default is to look for two apertures for nod-on-slit mode, and one otherwise. If the observation is marked as SRCTYPE=EXTENDED\_SOURCE, or the slit is very short, the default is to mark a single aperture at the center of the order; to override this behavior, enter a value for this parameter, or for *Guess* or *Fix*.
  + *Guess*: Enter a guess value for the aperture to use as a starting point. Values are in arcseconds across the slit (refer to the spatial profile). Separate multiple apertures by commas; separate values for multiple files by semi-colons. For example, *3,8;2,7* will look for apertures near 3” and 8” in the first image and 2” and 7” in the second image. If there are multiple files loaded, but only one aperture list is given, the aperture parameters will be used for all images.
  + *Fix*: Enter a value to use as the aperture center. The format of this parameter is the same as for the *Guess* parameter, but no fit will be done to attempt to refine the position: it will be used as entered.
  + *Exclude orders*: Enter comma-separated numbers to identify specific orders to exclude from spectral extraction.
* **Set Apertures**:
  + *Auto*: If checked, the pipeline will use the header parameters to determine how to set the apertures automatically. If SRCTYPE=EXTENDED\_SOURCE, or the slit is very short, it will extract the full slit. Otherwise, it will set a PSF radius to use in optimal extraction. If the data is cross-dispersed, it will not attempt to set background regions, since the slit is typically too short to determine them robustly. Uncheck *auto* to use either of the next two parameters.
  + *Subtract background*: If checked, and *auto* is not checked, then the pipeline will attempt to determine suitable background regions from the median spatial profile.
  + *Set PSF radius for optimal extraction*: If checked, and *auto* is not checked, then the pipeline will attempt to determine a PSF radius from the median spatial profile. If a PSF radius is not set at this step, standard extraction will be used as the extraction method.
  + *Override* parameters: If numbers are entered for any of the following parameters, they will be used as the aperture values. No automatic fit will be done for the specified parameter. For each one, values should be given in arcseconds across the slit (refer to the spatial profile). Separate multiple apertures by commas; separate values for multiple files by semi-colons.
    - *Override aperture radius*: Enter a value to use as the aperture radius.
    - *Override PSF radius*: Enter a value to use as the PSF radius. This number should be **larger** than the aperture radius.
    - *Override background regions*: Enter a range to use as the background region. For example, *0-1,8-10* will use the regions between 0” and 1” and between 8” and 10” to determine the background level to subtract in extraction.
  + *Override aperture signs*: enter either 1 or -1 to override the automatic determination of the aperture sign from the spatial profile. If the value is -1, the spectrum will be multiplied by -1. Separate multiple apertures by commas; separate values for multiple files by semi-colons.
* **Extract Spectra**:
  + *Extraction algorithm*: If set to *auto*, the pipeline will use standard extraction for SRCTYPE=EXTENDED\_SOURCE and optimal extraction otherwise. To override this, select either *optimal* or *standard*.
  + *Use median profile*: By default, the pipeline uses a wavelength-dependent spatial profile for extraction, but this method may occasionally give poor results, if the signal-to-noise in the profile is low. Check this option to use the median spatial profile across all wavelengths instead.
  + *Background fit order*: Set to a number greater than or equal to zero for the polynomial order of the fit to the background regions. The default is zero for EXES.
  + *Fix bad pixels*: The pipeline usually uses the spatial profile to attempt to fix bad pixels during extraction. Occasionally, this results in a failed extraction. Uncheck this box to extract the spectra without bad pixel correction.
  + *Bad pixel threshold:* Enter a value for the threshold for a pixel to be considered a bad pixel. This value is multiplied by the standard deviation of all good pixels in the aperture at each wavelength bin.
* **Combine Spectra**:
  + *Combine apertures*: If unchecked, spectra from separate files will be combined, but separate apertures will remain separate in the output file.
  + *Scale to median*: If checked, each spectrum will be scaled to the median across all spectra before combination.
  + *Correct spectral shape*: If checked, each spectrum’s shape will be corrected to the shape of the first spectrum before combination.
  + *Combination statistic*: Select the combination method. The default is a robust weighted mean.
* **Refine Wavecal**:
  + *Select line interactively?*: If unchecked, no wavelength calibration refinement will be done. If checked, a GUI window will pop up, allowing selection and editing of a spectral feature. To select the feature, click on the spectrum, then in the new window, type *f* and click on either side of the desired feature. From the GUI window, click **Add from Plot**, then edit the wavelength to the known value. Click**Done** when finished.
* **Merge Orders**:
  + *Selection threshold*: By default, pixels in overlapping regions are used if their signal-to-noise ratio is greater than 0.1 times the maximum signal-to-noise in the order. Increasing or decreasing this value may reduce artifacts like discontinuous jumps or excessively noisy overlap regions.

# Data quality assessment

After the pipeline has been run on a set of input data, the output products should be checked to ensure that the data has been properly reduced:

* Check the output to the terminal (or the log, in the case where the pipeline has been run by the automatic DPS system) for warnings or errors. Non-fatal warnings will be prepended with the string *WARNING*. Fatal errors will be prepended with the string *ERROR*.
* Check that the expected files were written to disk: there should, at a minimum, be a coadded 2D image (*COA*) and an extracted spectrum (*SPC* or *CMB)*.
* Look at the reduced image. Check that the distortion correction appears to have been calculated and applied correctly: order edges and spectral lines should appear to be aligned with the image columns and rows. If they are not, there may be a problem with the flat or with the optical parameters used.
* Look at the intermediate extracted spectra in *xvspec*. The spectra from positive and negative apertures should look similar. Spectra from separate observations of the same source in the same mode should look similar.
* Compare the signal-to-noise in the extracted spectrum to the reduced image. If the target seems bright in the reduced image, but the spectrum looks noisy, it may be that the aperture centers were not defined correctly.
* Check the aperture parameters recorded in the pipeline output and in the output file headers. The calculated aperture and PSF radii should have similar values for all apertures. The aperture centers for each order should also be similar in cross-dispersed spectra.
* Check the extracted spectra for excessive outliers. Most bad pixels should be removed in the extraction process. If they are not, it may be helpful to use the median spatial profile, or set the bad pixel threshold lower.

# Appendix A: Sample configuration files

Sample EXES configuration file, typically located in *exes/data/headerdef.dat*. Values marked with a Y in the *reqd?* column are keywords required to be present in input data. They must meet the type and range requirements listed for grouping and data reduction to be successful.

#

# This table lists keywords, allowed values, and defaults for

# all keywords to be written to output files. Keywords

# required to be present in input files for correct data

# processing are marked with a 'Y'. A '.' in min, max, or enum

# means no requirement. Some output keywords have defaults

# defined in other configuration files

#

# keyword reqd? default type min max enum

#-----------------------------------------------------------------

ADDTIME N -9999. float 0. . .

AIRSPEED N -9999. float . . .

ALTI\_END Y -9999. float 0. 60000. .

ALTI\_STA Y -9999. float 0. 60000. .

AOR\_ID Y UNKNOWN string . . .

AOT\_ID N UNKNOWN string . . .

BB\_TEMP Y 273.16 float 1 9999. .

BEAMTIME N 0.0 float . . .

BOUNCE N 0.0 float . . .

BPM N UNKNOWN string . . .

BRL N -9999. float . . .

CARDMODE N BLKSKY string . . NONE|BLK|SKY|SHINY|BLKSKY|BLKSHINY|BLKOBJ|OBJ

CHOPPING N F bool . . .

CLOUD N 0.0 float . . .

CREATOR N UNKNOWN string . . .

CUMTIME N -9999. float 0. . .

DARKVAL N 0. float 0. . .

DATAPATH N UNKNOWN string . . .

DATASRC Y UNKNOWN string . . ASTRO|CALIBRATION|LAB|TEST|OTHER|FIRSTPOINT

DATATYPE N UNKNOWN string . . IMAGE|SPECTRAL|OTHER

DATE N UNKNOWN string . . .

DATE-OBS Y UNKNOWN string . . .

DEPLOY N UNKNOWN string . . .

DETECTOR N UNKNOWN string . . .

DETROT N 0.10 float . . .

DETSEC N UNKNOWN string . . .

DETSIZE N UNKNOWN string . . .

DITHER N F bool . . .

DRKFILE N UNKNOWN string . . .

DTHINDEX N -9999 int . . .

DTHNPOS N -9999 int . . .

DTHOFFS N -9999. float . . .

DTHPATT N UNKNOWN string . . 3-POINT|5-POINT|9-POINT|CUSTOM

ECHELLE Y 60.0 float -35. 70. .

ECTPAT Y UNKNOWN string . . .

EFL N -9999. float . . .

EFL0 N -9999. float . . .

EPERADU Y 75.0 float 1. 9999. .

EQUINOX N -9999. float . . .

EXCLUDE N None string . . .

EXPTIME N -9999. float 0. . .

FBC-STAT N UNKNOWN string . . FBC\_OFF|FBC\_QS|FBC\_DY|FBC\_ON

FDATE N 0.0 float 0. 9999. .

FILENAME Y UNKNOWN string . . .

FILEREV N UNKNOWN string . . .

FLIGHTLG N UNKNOWN string . . .

FOCUS\_EN N -9999. float -5000 5000 .

FOCUS\_ST N -9999. float -5000 5000 .

FRAMETIM Y 1.0 float 0. 9999. .

GRATANGL N 0.0 float -360. 360. .

GRDSPEED N -9999. float . . .

HEADING N -9999. float . . .

HEADSTAT N UNKNOWN string . . ORIGINAL|UNKNOWN|CORRECTED|ERROR

HRDGR N 0.7588 float . . .

HRFL N -9999. float . . .

HRFL0 N -9999. float . . .

HRG N -0.015 float . . .

HRR N 9.8 float 0. 9999. .

IMAGEID N -9999 int 0 . .

INSTCFG Y UNKNOWN string . . HI-MED|MED|HI-LO|LO|CAM|PUP|HIMED|HILOW|LOW|CAMERA|HIGH\_MED|HIGH\_LOW|MEDIUM|HIGH-MED|HIGH-LOW

INSTMODE Y UNKNOWN string . . STARE|NOD\_ON\_SLIT|NOD\_OFF\_SLIT|MAP

INSTRUME Y EXES string . . EXES|TEXES

KROT N -9999. float . . .

KWDICT N UNKNOWN string . . .

LASTREW N UNKNOWN string . . .

LAT\_END N -9999. float . . .

LAT\_STA N -9999. float . . .

LINFILE N UNKNOWN string . . .

LO\_LIM N 0.0 float . . .

LON\_END N -9999. float . . .

LON\_STA N -9999. float . . .

MAPCRSYS N UNKNOWN string . . EQUATORIAL|GALACTIC|ECLIPTIC|USER

MAPINTX N -9999. float . . .

MAPINTY N -9999. float . . .

MAPNXPOS N -9999 int . . .

MAPNYPOS N -9999 int . . .

MAPPING N F bool . . .

MCCSMODE N UNKNOWN string . . .

MISSN-ID Y UNKNOWN string . . .

NBELOW N -9999 int 0 . .

NEXP N -9999 int 1 . .

NFRAME N -9999 int 1 . .

NINT Y 1 int 1 . .

NODAMP N -9999. float . . .

NODANGLE N -9999. float -180 180 .

NODBEAM N UNKNOWN string . . .

NODCRSYS N UNKNOWN string . . ERF|ECRF|GALRF|TARF|FPIRF|FFIRF|WFIRF|SIRF|USER

NODDING N F bool . . .

NODN N -9999 int . . .

NODPATT N UNKNOWN string . . ABBA|AB|CUSTOM

NODSETL N -9999. float . . .

NODSTYLE N UNKNOWN string . . NMC|NPC|C2NC2|CUSTOM

NODTIME N -9999. float . . .

NORDERS N -9999 int 0 9999 .

NSPAT N 1024 int 0 1025 .

NSPEC N 1024 int 0 1025 .

NSUM N 1 int 0 9999 .

NT N 1024 int 0 1024 .

OBJECT Y UNKNOWN string . . .

OBS\_ID Y UNKNOWN string . . .

OBSDEC N UNKNOWN string . . .

OBSERVER N UNKNOWN string . . .

OBSRA N UNKNOWN string . . .

OBSTYPE Y UNKNOWN string . . OBJECT|STANDARD\_FLUX|STANDARD\_TELLURIC|LAMP|FLAT|DARK|BIAS|SKY

OMEGAP N -9999. float . . .

OPERATOR N UNKNOWN string . . .

ORDR\_B N UNKNOWN string . . .

ORDR\_E N UNKNOWN string . . .

ORDR\_S N UNKNOWN string . . .

ORDR\_T N UNKNOWN string . . .

ORIGIN N UNKNOWN string . . .

OTPAT Y UNKNOWN string . . .

PAGAIN Y 1.0 float 0. 9999. .

PINHOLE N 0 int -1 2 .

PIPELINE N UNKNOWN string . . .

PIPEVERS N UNKNOWN string . . .

PIXELWD N 0.0025 float 0. 9999. .

PIXSCAL N -9999. float . . .

PKGPATH N UNKNOWN string . . .

PLANID Y UNKNOWN string . . .

PLTSCALE N -9999. float . . .

PROCSTAT Y UNKNOWN string . . .

PRODTYPE N UNKNOWN string . . .

READNOIS N 30. float 0. 9999. .

RESOLUN N -9999. float . . .

SATVAL N 16000. float 0. 65536 .

SCANNING N F bool . . .

SDEG Y 180. float -45. 405. .

SIBS\_X N -9999 int . . .

SIBS\_Y N -9999 int . . .

SLIT Y UNKNOWN string . . .

SLITOFF N -0.3 float . . .

SLITROT N -9999. float . . .

SLITVAL N -9999. float . . .

SLITWID N -9999. float . . .

SPACING N -9999. float . . .

SPECTEL1 Y UNKNOWN string . . .

SPECTEL2 Y UNKNOWN string . . .

SPIKEFAC N 20.0 float 0. 9999. .

SRCTYPE Y UNKNOWN string . . POINT\_SOURCE|EXTENDED\_SOURCE|OTHER|UNKNOWN

STDFAC N 20.0 float 0. 9999. .

SUBARRNO N -9999 int . . .

TELCONF N UNKNOWN string . . .

TELEL N -9999. float 0. 90. .

TELEQUI N UNKNOWN string . . .

TELESCOP N UNKNOWN string . . .

TELDEC N UNKNOWN string . . .

TELLOS N -9999. float -180 180 .

TELRA N UNKNOWN string . . .

TELVPA N -9999. float 0. 360. .

TELXEL N -9999. float -90. 90. .

TEMP\_OUT N -9999. float . . .

TEMPPRI1 N -9999. float -273. . .

TEMPPRI2 N -9999. float -273. . .

TEMPPRI3 N -9999. float -273. . .

TEMPSEC1 N -9999. float . . .

THRFAC N 0.5 float 0. 1.001 .

TORTMODE N CARD string . . NONE|OBJ|CARD

TRACERR N F bool . . .

TRACKANG N -9999. float . . .

TRACMODE N UNKNOWN string . . OFF|CENTROID|ROF|LIMB|OFFSET|ROF+OFFSET|CENTROID+INERTIAL|ROF+INERTIAL|OFFSET+INERTIAL|ROF+OFFSET+INERTIAL

TRASH N 100.0 float . . .

TSC-STAT N UNKNOWN string . . .

UP\_LIM N 11000.0 float . . .

UTCSTART N UNKNOWN string . . .

UTCEND N UNKNOWN string . . .

WAVENO0 Y 0.0 double 0. . .

WNO0 N -9999. double 0. . .

WVZ\_END N -9999. float 0. . .

WVZ\_STA N -9999. float 0. . .

X0BRL N -9999. float . . .

XD0 N 0.0 float . . .

XDDELTA N 0.0235 float . . .

XDDGR N -9999. float . . .

XDFL N -9999. float . . .

XDFL0 N -9999. float . . .

XDG N 0.033 float . . .

XDLRDGR N 0.001328 float . . .

XDMRDGR N 0.003151 float . . .

XDR N -9999. float . . .

XORDER1 N 0.0 float 0. 1024 .

Y0BRL N -9999. float . . .

ZA\_END Y -9999. float 0. 90. .

ZA\_START Y -9999. float 0. 90. .

Sample FSpextool instrument configuration file, usually located at *fspextool/Instruments/EXES/Data/EXES.dat*.

#

# Generated by M. Clarke, Jan. 2014

#

# This is the calibration file for the EXES spectrograph on SOFIA.

# Note the values must be in the correct order, but can have any number of

# spaces/comments between them.

#

#============================================================================

#

INSTRUMENT=EXES

NCOLS=1024

NROWS=1024

STDIMAGE=1024

PLOTWINSIZE=700 512

FILENAME=FILENAME

EXPTIME=EXPTIME

TIME=TIME\_OBS

POSANGLE=None

HA=None

AIRMASS=None

NINT=4

BADPIXMASK=bpm.fits

%

%CAL BASE

%

CALMODULE=mc\_exescals1d

%

%FILE READ MODE

%

FILEREADMODE=Filename

IPREFIX=F

OPREFIX=r

SUFFIX=\*.fits\*

FITSREADPROGRAM=mc\_readexesfits

HEADCOMBPROGRAM=mc\_exesdcshdr

YUNITS=erg/s/cm2/sr/cm-1

YTITLE=I (erg/s/cm!u2!n/sr/cm!u-1!n)

XUNITS=cm-1

XTITLE=Wavenumber (cm!u-1!n)

%

% Reduction Mode

%

REDUCTIONMODE=A

%

% Combine Base Information

%

COMBMODE=A

COMBSTAT=Median (Median Error)

COMBTHRESH=8.0

COMBODIR=proc/

%

% Sky Base Information

%

SKYSTAT=Robust Weighted Mean

SKYTHRESH=8.0

%

% Profile Parameters

%

YBUFFER=2

OVERSAMP=1

ATMOSTHRESH=0

%

% Point Source Base

%

PSNAPS=2

PSPSFRAD=9.0

PSAPRAD=3.0

PSBGSUB=1

PSBGSTART=9.0

PSBGWIDTH=10

PSBGDEG=0

PSBGMULT=2.0

%

% Extended Source Base

%

XSBGSUB=1

XSBG=0-9,24-36,54-60

XSBGDEG=0

%

% Additional processing base

%

ADDLMODULE=None

%

%Other Base Parameters

%

TRACEDEG=2

TRACESTEP=7

TRACESUMAP=7

TRACESIGTHRESH=1

TRACEWINTHRESH=5

BADPIXELTHRESH=7

PLOTSATURATEDPIXELS=0

SATURATION=3000

CHECKSEEING=0

SEEINGTHRESH=1.0

LINCORRECT=0

ERRORPROPAGATION=1

FLATFIELD=0

FIXBADPIXELS=1

OPTIMALEXTRACTION=1

%

% FITS Header Keywords to Grab

% ----------------------------

%

% Keywords required for processing, may also be required

% by SOFIA DCS

%

%

[…]

Here follows a list of all keywords required to appear in FSpextool output files. This list is the same set as in the above configuration file for the EXES package.

1. The *redux* GUI has a single, rarely-used parameter: if called with the parameter */SMALL*, it will produce an initial window that is sized for small laptop screens. [↑](#footnote-ref-2)