

---

# **C/KOSMOS Cookbook Documentation**

*Release 0.9*

**Richard Shaw**

2015-Oct-01



## CONTENTS

<b>1</b>	<b>Contents:</b>	<b>3</b>
1.1	Instrument Overview . . . . .	3
1.2	Getting Started . . . . .	6
1.3	Processing Science Data . . . . .	11
1.4	Useful Resources . . . . .	31
1.5	Glossary . . . . .	33
1.6	About This Guide . . . . .	34
1.7	Background Material . . . . .	35
<b>2</b>	<b>Index and Search</b>	<b>37</b>
	<b>Bibliography</b>	<b>39</b>





Fig. 1: Milky Way over CTIO. Photo credit: Anja von der Linden.

This *Cookbook* is intended as a guide to data reduction and calibration for PIs and Archive users of data from NOAO's twin multi-object imaging spectrographs as used on the Blanco 4-m telescope at CTIO (COSMOS) and on the Mayall 4-m telescope at KPNO (KOSMOS). Throughout this *Cookbook* the instruments will be referred to collectively as C/KOSMOS. Unlike data from the NOAO wide-field imaging instruments, no automated processing pipelines exist at present for these spectrographs.

KOSMOS was the first of these twin spectrographs to be commissioned; it was first offered for community use in semester 2014A. COSMOS was first offered for community use in semester 2015A. Earlier epoch data from the respective commissioning runs exist in the NOAO Science Archive (NSA), but users should be aware that the performance of the instruments was evolving as the systems were tuned during that period, which affects the quality and scientific applicability of these data.

The descriptions, recipes, and scripts offered in this *Cookbook* will provide all the necessary information for deriving scientifically viable spectra and images from C/KOSMOS, but users should be aware that the ultimate utility of the data for any specific scientific goal depends strongly upon a number of external factors, including:

- the environmental conditions that prevailed at the time of the observations,
- the performance of the instrument,
- the observing procedures used to obtain the data,
- the scientific objectives of the original observing program.

The data also suffer from a number of foibles of the instrument and data taking system. Scripts are available in the chapters below to fix or ameliorate these problems.



CONTENTS:

## 1.1 Instrument Overview

The following is a **very** brief description of the C/KOSMOS instruments. Please see the [NOAO Data Handbook](#) (Ch. 5) or the [C/KOSMOS Instrument Manual](#) for more details.

### 1.1.1 Optical Design

The design of the C/KOSMOS imaging spectrographs allows for direct imaging of a large field of view (FoV), as well as spectroscopy of selected targets within the FoV using a slit-mask and grism.

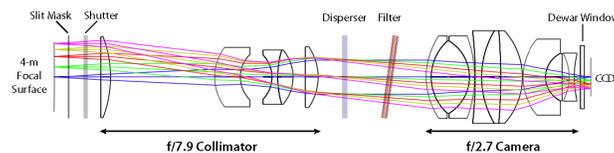


Fig. 1.1: Schematic of the C/KOSMOS optics. Light from the telescope focal surface (*left*) passes through the slit mask and the collimator, the disperser (in spectroscopic mode), and filter (if deployed) before entering the camera and being detected by the CCD (*right*). Click image to enlarge.

### 1.1.2 Imaging and Field Acquisition

Images from C/KOSMOS can be obtained through a filter or in white light. The 11.6-arcmin corrected field of view, as projected onto the detector, is illustrated below. The spatial extent (and the maximum length of a long slit) is limited by the detector format to 10x10 arcmin when the `2kx2k` imaging or `FullFrame` spectroscopic region of interest (*RoI*) is used for read-out. Long-slit or multi-object spectroscopy requires obtaining an exposure with a the slit mask inserted, but with no disperser to ensure that light from all targets in the field passes through the intended slit(s). Subsequent exposures through the disperser place the spectra from the slit(s) onto the detector format.

### 1.1.3 Spectroscopy

There are currently two dispersers offered for each instrument, with attributes summarized in the Table below.

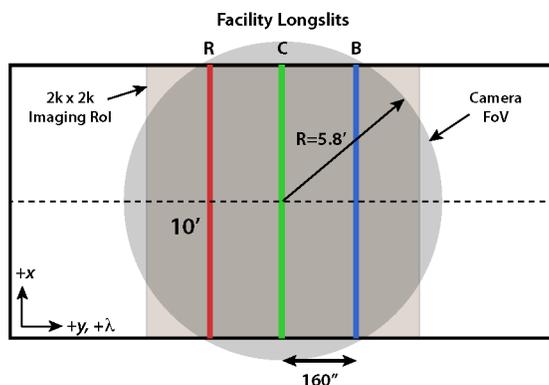


Fig. 1.2: Schematic of the 11.6 arcmin (diameter) corrected *FoV* of the camera (grey circular region) as projected onto the detector geometry (clear rectangular region), the  $2k \times 2k$  imaging region of interest (light brown square), and the boundary separating the two amplifier regions (dashed line). The detector intercepts the dispersed spectra, where wavelength increases left to right along the image *y*-axis. The facility long-slit masks (with 3 positions: R, C, B), subtend 10 arcmin spatially, the slits are offset 160 arcsec from one another. Image orientation in celestial coordinates depends upon the instrument rotator angle. Click image to enlarge.

Table 1.1: C/KOSMOS Spectral Configurations

Attribute	Blue Grism	Red Grism
Fringe frequency	1172 lines/mm	842 lines/mm
Resolution	2000–2600	2000–2600
Spectral scale	0.53–0.75 Ang/pix	0.84–1.06 Ang/pix
Coating optimization	3500–7000 Ang	5000–10000 Ang
Spectral coverage:		
Blue slit:	3500–6200 Ang	5010–9060 Ang
Center slit:	3800–6630 Ang	5540–9660 Ang
Red slit:	4200–7100 Ang	6100–10280 Ang

There are a variety of facility long-slits that are used on C/KOSMOS, all of which are 10 arcmin in length, with widths specified in the table below. The suffix “R” or “B” refers to the “Red” or “Blue” slit position, respectively, in the FoV; no suffix implies the “C” or “Center” slit.

Table 1.2: Facility Longslits

Slit Name	Width	Width
2px [R B]	2 pix	0.6 arcsec
3px [R B]	3 pix	0.9 arcsec
4px [R B]	4 pix	1.2 arcsec
5px [R B]	5 pix	1.5 arcsec
10px [R B]	10 pix	3.0 arcsec

In practice the `SLIT` keyword value has a number of aliases, such as `3pixblue` and so on, because the keyword value is taken from a configuration file in the observing environment that is updated manually (and, inconsistently).

## Instrument Foibles

### Unstable Bias Structure

Zero-second “bias” exposures, which are used to characterize patterns in the bias structure after overscan correction, show low-level structure that is not temporally stable, even in sequences of consecutive bias exposures. The average structure along each axis is somewhat more stable, and can be characterized and removed from the instrumental signature. See *Bias Residual* for details.

### Anomalous Spectral Features

In the early days of operation (i.e., commissioning and shortly thereafter) the C/KOSMOS spectrographs allowed zero-order light to enter from the slit and pass through the VPH grating (i.e., without being dispersed). One of the features is located at approximately  $y=2050$  pix, and is similar in appearance to emission lines from the comparison arc lamp, except that it is wider and does not match the geometric distortion (i.e., curvature) along the slit of the other emission lines. An example is shown below. An anomalous “bright emission” feature will appear in science target spectra (including MOS spectra) for the same reason. See *Anomalous Spectral Features* for more details.

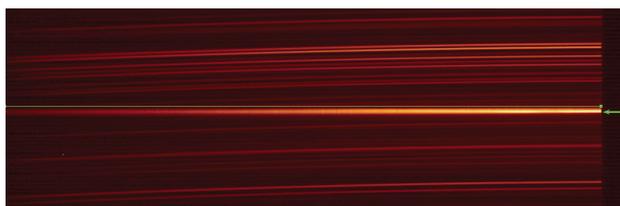


Fig. 1.3: Enlargement of a b2k 3pxB He-Ne-Ar spectrum from Amp 1 near  $y=2050$  pix, with non-dispersed light indicated (*green arrow*). Emission lines from the arc lamp show a slight curvature from the horizontal (*green line*). Click image to enlarge.

### 1.1.4 Arc Lamp

There is only one option for a wavelength comparison, the He-Ne-Ar arc lamp. The spectrum is shown below; see *Wavelength Calibration* for determining and applying a spectral WCS.

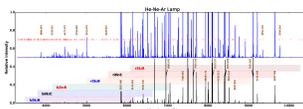


Fig. 1.4: He-Ne-Ar spectrum at full scale (*black*) and at 10x (*blue*, offset vertically for clarity). Identifiable lines are marked (*red ticks*), and some of the brighter or more isolated lines are labelled. Approximate wavelength coverage for combinations of the VPH grisms and the facility longslits is shown (*colored bars*). Click image to enlarge.

At present the data taking system writes a “trial” WCS solution into the headers of raw dispersed exposures, where the central wavelength is given as 5216 Ang at the center of the spectrum. This WCS is incomplete and incorrect; the spatial scale is populated with the wrong units (arcsec/pix rather than deg/pix). See the chapter on *World Coordinate Systems* for details.

## Spectral Resolution

The resolution of C/KOSMOS depends upon a number of factors, including the optics, the dispersers, and the size of the entrance slit. The 3px (3-pixel, or 0.9 arcsec) facility longslits have a resolution of about 2500, with a somewhat nonlinear dispersion, as shown below.

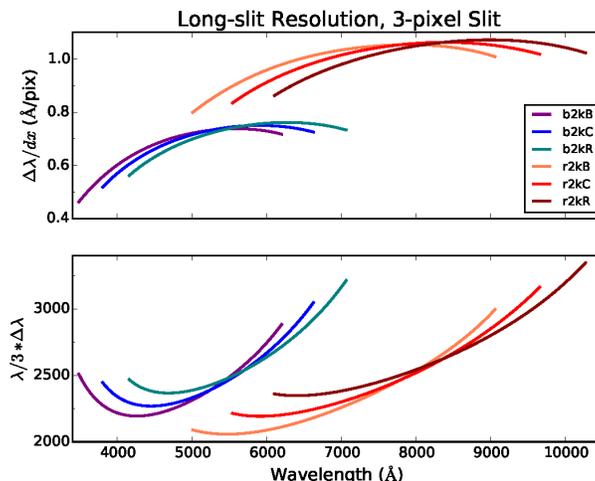


Fig. 1.5: The dispersion, or change in wavelength with position (*upper*) and spectral resolution (*lower*) are shown for the 3px longslit with the b2k and r2k VPH grisms, in all field locations. Click image to enlarge.

### 1.1.5 Throughput

The system throughput, shown below, is relatively good from atmospheric cut-off to longward of 9000 Ang. The shape of the curves are dominated by the grism efficiency function.

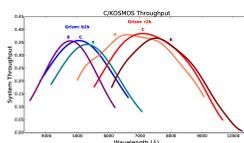


Fig. 1.6: System throughput for each grism+longslit combination (see labels); throughput includes the optics, detector sensitivity, grating efficiency, and atmospheric transmission through one airmass. Click image to enlarge.

## 1.2 Getting Started

There are some preliminary items to attend to, and reference materials to gather prior to calibrating and processing your C/KOSMOS data.

### 1.2.1 Retrieve Data

Retrieve the science and calibration data from the [NOAO Science Archive Portal](#) for each calendar night of interest. If you are a PI or Co-I of an observing program and you wish to retrieve your *proprietary* data, you will need to login before searching for you data. If you are a general archive user, a login is not necessary to retrieve non-proprietary

data. It is probably best to retrieve **all** data for the observing date (and possibly, the whole *observing block*) when your targets of interest were observed.

**Caution:** Note that applicable ancillary data (arc lamp, flat-field or, standard star exposures) may have been obtained on another night within the *observing block* when your target(s) of interest were observed. You may need these exposures to calibrate the science data.

## Archive Searches

Information about specific targets that were observed in a spectroscopic (long-slit or MOS) mode are *not recorded* in Archive metadata. This makes searching for spectra of scientific targets of interest more challenging, and in general requires that you download data in order to figure out if spectra exist of specific targets. That borders on impractical, but it is possible to customize your [NSA Portal](#) query to gain some insight into what spectra exist by including the `object` field in the results. The results may be more readable by omitting fields from the default query that are not helpful for this purpose (e.g., `surveyid`, `telescope`, `instrument`, `prodtype`, `seeing`, `depth`).

**As an example, suppose you are interested in spectra of the Orion nebula (M42, NGC1976).**

1. Enter “Orion” in the “object name” field of the **Simple Query Form** and click **Resolve**.
2. Then select a large (600 arcmin) search box and click **Search**.
3. Now click the **Advanced Query Form** tab to customize the query, and enter the following in the dialog box:

**SELECT** reference, dtpropid, start\_date, release\_date, date\_obs, dtpi, **object**, ra, dec, filter, exposu

**Click Search to execute the revised query. On the results page you can further refine the search:**

4. Select *Observation Type* from the **Categorize by:** pull-down menu as shown below.
5. Click the **Object** tab.
6. Filter the search by restricting “Observing mode” to “*sos\_slit*”

0 rows selected	Access	Proposal ID	Release date	Observing date	UT	PI	RA	Observing mode	Observation type	Observing mode	Processing	Original filename	Archive
<input type="checkbox"/>	Retrieve	2013B-2001	2014-01-09	2014-01-10	2014-01-11 07:01:48	Jay Elias	83.490708	-0.456056 OG530 KOG530	1200	object	sos_slit	Raw	/home/data/Night1.06786.fits k4k_140111_1
<input type="checkbox"/>	Retrieve	2013B-2001	2014-01-09	2014-01-10	2014-01-11 07:22:37	Jay Elias	83.490708	-0.456056 OG530 KOG530	1200	object	sos_slit	Raw	/home/data/Night1.06787.fits k4k_140111_1
<input type="checkbox"/>	Retrieve	2013B-2001	2014-01-10	2014-01-11	2014-01-12 06:04:06	Jay Elias	84.264583	-0.750361 OG530 KOG530	1200	object	sos_slit	Raw	/home/data/Night2.07020.fits k4k_140112_1
<input type="checkbox"/>	Retrieve	2013B-2001	2014-01-10	2014-01-11	2014-01-12 06:24:55	Jay Elias	84.264583	-0.750361 OG530 KOG530	1200	object	sos_slit	Raw	/home/data/Night2.07021.fits k4k_140112_1

Fig. 1.7: Results of the above search for spectra obtained with the Orion nebula (NGC1976). Note that the search can be restricted to spectroscopic exposures on fields of potential interest.

Clicking the “Retrieve” link in any row will download the associated exposure, for evaluation. If the data are of interest, refine the Portal search to include all observations for that observing night (or *observing block*), retrieve them in bulk, and reduce the data of interest.

## Slit-mask Definition Files

If your targets were observed with a slit-mask (*MOS* mode) it is helpful to also obtain the slit-mask definition (*KMS*) files. These files (with the extension `.kms` or `.cms`) contain the aperture dimensions and locations within the acquisition field, and distinguish between apertures used for field acquisition vs. target spectra. If you were the observer, you generated these files in preparation for your observing run. If you are an Archive user, you may wish to download this gzipped tar of all slit-mask files.

**Note:** Neither the locations nor the sky positions of the target slits are recorded anywhere in the raw data headers. This information must be obtained from the *KMS* files, which are not currently stored in the *NSA*. Target locations may alternatively be measured from the world coordinates of the slits in the final acquisition image.

Download an sqlite3 `slit-mask` database to browse the essential information in all slit-mask definition files (as of 2015-Sep-17).

## Data File Structure

The raw data from the observing environment are written as tile-compressed *MEF* files (i.e., FITS files with extensions). The primary header-data unit (PHDU) contains keywords (but no pixel data) that apply to all image extensions; the two extensions (HDUs) each contain a header and compressed raw pixel data for one of the amplifier sections of the full CCD array. (Technically, these extensions in a tile-compressed FITS file take the form of binary tables, but are logically images.) Note that not all applications (including IRAF tasks) can access compressed images directly: in such cases the files must be uncompressed with a utility such as `funpack` (which is included in the IRAF `fitsutil` external package).

The raw data also includes a 50 pixel overscan regions for each amplifier, as illustrated below. The width of the overscan region is the same size regardless of the CCD binning or the *RoI*.

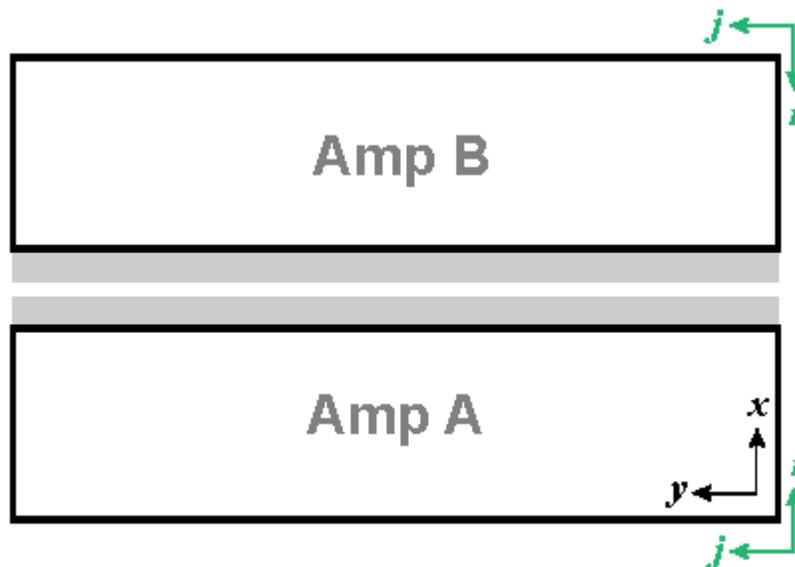


Fig. 1.8: Schematic of C/KOSMOS pixel array just after readout. Virtual overscan regions are indicated as shaded boxes (*grey*), but are expanded for clarity. Amplifier locations are near the origins of the green amplifier coordinates ( $i, j$ ), prior to transformation to detector (CCD) coordinates (*black coordinate axes*).

## 1.2.2 Review Reference Materials

It is handy to have the following documents available when reducing your data:

- [\[IM\]](#) C/KOSMOS Instrument Manual
- [\[MD\]](#) Mask Design Software for C/KOSMOS (if you are reducing *MOS* spectra)
- [\[DHB\]](#) NOAO Data Handbook (chapters 1 and 5)
- The [C/KOSMOS web site](#)

Other material in the *Useful Resources* chapter may also be relevant.

## 1.2.3 Getting Organized

Many observing runs (on clear nights, at least) generate hundreds of exposures. Some of them (such as *focus* or *test* exposures) are not useful or relevant for data reduction, and were likely obtained to configure the instrument for observing.

### Types of Observations

The following types of observations are typically obtained, depending upon the observing program:

Table 1.3: Types of Observations

Type	Frequency	Description
<i>Focus</i>	10 or more per night	Part of a sequence of exposures of a bright star or a flat-field lamp used to obtain the best focus of the telescope or the spectrograph. Not used for calibration.
Bias (zero)	10 or more per night and per read-out region	Sequence of zero-second exposures used to characterize the bias structure. Combined to form the Bias Residual Mastercal
Flat-field	several nightly per filter	Sequence of exposures with the dome flat-field apparatus. They are combined and normalized to apply the pixel-level sensitivity correction. May also include flats obtained of the twilight sky.
Comparison Arc	one or more per night per slit/grating combination	Used to derive geometric rectification and wavelength calibration; adjusted per target for <i>MOS</i> observations.
Image	one or more per filter per target field	<b>Science image</b> obtained with <code>ObsMode = imaging</code> . May also be used for acquisition.
<i>Acquisition image</i>	one or more per target field	Short-duration image obtained through the slit-mask ( <code>ObsMode = acq</code> ). Used to determine offsets from targets to slits; not used in data reductions other than to verify that targets were acquired in the intended aperture(s).
Long-slit spectrum	one or more per target position	<b>Science spectrum</b> obtained with <code>ObsMode = sos_slit</code> .
Slit-mask spectrum	one or more per target position	<b>Science spectra</b> obtained with <code>ObsMode = mos_apmask</code> ; one spectrum per slit including field stars

## Generate an Observing Log

Data files from even a single night of observing can number in the hundreds. Add to that the files generated during the course of data reduction: intermediate files, MasterCal reference files, combined exposures, etc. and the task of keeping track of files in your workflow rapidly becomes a challenge. There is no uniquely correct way to do this, but there are a couple of common approaches:

1. Separate files of various types into subdirectories and/or make ASCII lists of files with common attributes. This is a typical approach for preparing to run IRAF file processing tasks (see *In a Nutshell: Processing with IRAF*).
2. Create an `sqlite3` database of files and their attributes. This may be useful even if you choose IRAF for your image processing, since you can view exposure attributes with the `sqlite browser`.

The python script `obslog.py` will generate an observing log of the exposures, in the form of an `sqlite3` database. The `-v` option will write the metadata for each file to STDOUT (which you can re-direct to an ASCII file). The keywords listed below are harvested from the data headers. Many of the names are obscure, so they are re-mapped to somewhat more intuitive field names in the database. Fields may be added (or deleted: *not recommended*) by changing the `KW_MAP` definition at the top of the script.

Table 1.4: Header Metadata Stored in Observing Log

Key-word	DB Field	Description
	<code>use_me</code>	Flag; indicates file usage (1) or exclusion (0). Not present in the ASCII version of the log.
<code>FILE</code>	<code>File</code>	Name of the data file
<code>DTCALDATE</code>	<code>CalDate</code>	Calendar date of observation (with a noon MST zero-point)
<code>OBJECT</code>	<code>Object</code>	Observer name for content of exposure
<code>OBSTYPE</code>	<code>ObsType</code>	Type of observation ( <code>flat focus comp object zero</code> )
<code>OBSMODE</code>	<code>ObsMode</code>	Mode of observation ( <code>acq imaging mos_apmask sos_slit</code> ). <b>Note:</b> field value for MOS mode corrected with script. <b>See warning below.</b>
<code>NOCROIPROI</code>	<code>ROI</code>	Region of interest for CCD read-out ( <code>FullFrame 2kx2k 1kx1k 320x4k</code> )
<code>CCDSUM</code>	<code>CcdBin</code>	Pixel binning during read-out ( <code>1 1 2 1 2 2</code> )
<code>KSFILCM</code>	<code>Filter</code>	Name of filter used
<code>KSDPOS</code>	<code>Dispers</code>	Name of disperser used ( <code>b2k r2k Open</code> )
<code>KSSPOS</code>	<code>Slit</code>	Name for slit(mask) used
<code>GCCROTAR</code>	<code>Rotator</code>	Position angle ( $\rho$ ) of slit on sky (N through E). At $\rho = 90$ (KOSMOS default) N is “up” and E is to the “left” in a default <code>ds9</code> display; at $\rho = 270$ (COSMOS default) N is “down” and E is to the “right.”
<code>DATE-OBS</code>	<code>DateObs</code>	Date-time of observation (ISO-8601 string)
<code>EXPTIME</code>	<code>_exp</code>	Exposure duration (s)
<code>AIRMASS</code>	<code>Airmas</code>	Atmospheric column through which exposure was obtained

**Caution:** The `OBSTYPE` keyword value is populated by the script that was run in the observing environment to acquire the data. These scripts are selected by the observer, who may or may not have used the appropriate one for some exposures. Thus, some observations with, e.g., `OBSTYPE=object` may in fact be flat-fields or acquisition exposures. Carefully examine the observing log, especially the value of the `OBJECT` keyword, and the raw data to identify and correct discrepancies.

Note that the `OBSMODE` keyword value in raw exposures is incorrect for MOS mode observations (it is `sos_slit` when it should be `mos_apmask`); the value is corrected in the output if the `KSSPOS` keyword value does not match the signature for a facility long-slit when the instrument is configured for spectroscopy.

Now would be a good time to browse the observing log and remove (or mark as “excluded”) any files that should **not** be processed, such as test or acquisition exposures. Exposures with `OBSTYPE=focus` are by default excluded from the log.

Continue the data reduction with *Processing Science Data*.

## 1.3 Processing Science Data

As described in the *Instrument Overview*, C/KOSMOS has three distinct modalities:

- imaging,
- long-slit spectroscopy, and
- slit-mask spectroscopy

Therefore the path through data reduction is somewhat different for each mode, though there is some overlap. The different modalities become most apparent for users who reduce data with IRAF, as the packages (and the legacy reduction manuals) are tuned to simpler, mono-mode instruments.

### 1.3.1 Prerequisites

Processing of individual targets can proceed once the following steps have been concluded:

- Data have been acquired from the Archive (see *Getting Started*)
- Software tools have been installed (see *Useful Resources*)
- An observing log has been built and reviewed for problem data (see *Getting Started*)
- Reference material has been retrieved and reviewed (see *Useful Resources*)

### 1.3.2 General Considerations for Reductions

The following chapters describe the attributes of C/KOSMOS data, some considerations for reductions and processing in general, and various instrument and data foibles to be aware of.

#### Creating Master Reference Files

**Master reference files** are derived from calibration observations, and are used to remove the various components of the instrument signature from the data. Calibration exposures may combined or characterized to create a calibration, so that it may be applied when science data are processed. This chapter summarizes how to create the reference files that will be needed for your data reduction.

#### Contents

- *Creating Master Reference Files*
  - *Bias Residual*
  - *Darks*
  - *Flat-Field*
  - *Wavelength Calibration*
  - *Photometric Calibration*

## Bias Residual

The Bias Residual is a persistent, spatially variable component of the read-out process that is not removed during overscan subtraction. The structure in this residual gives some indication of the stability of the read-out electronics. Generally, this structure will be different for full-frame read-outs than for subsets of the array, such as those used for imaging mode, or Region-of-Interest spectral read-outs. The C/KOSMOS detector electronics introduce a moire pattern that is **not** stable over time (or even repeatable in consecutive bias exposures), as shown in the Figure below.

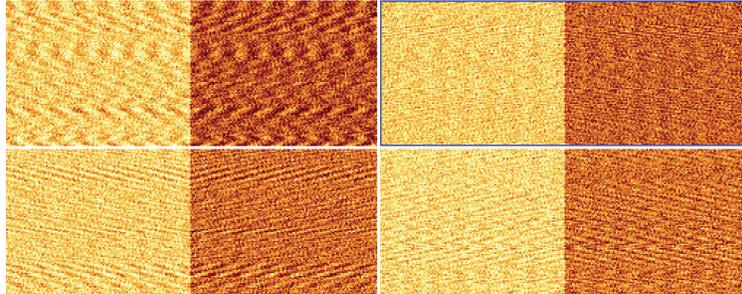


Fig. 1.9: Small, central portions of full-frame residual bias images from four consecutive nights (2015 Apr 16–19 from upper left to lower right), rendered in false color with a linear intensity stretch over a range of 30 ADU. Images were created from an average of about 10 consecutive bias exposures. Note the change in the moire pattern from night to night. Click image to enlarge.

There is, however, a slowly varying spatial curvature along Axis1 that is fairly consistent. This pattern is particularly prominent for the 320x4k RoI, but is present at some level for all RoIs. The curvature is much more stable than the moire pattern, which makes it suitable for characterization in the Master Bias Residual, to be subtracted from all science images with the same RoI.

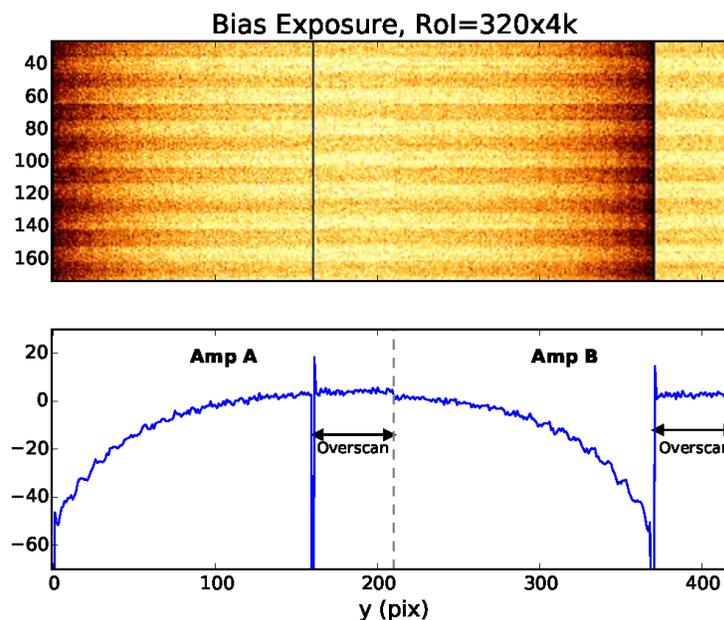


Fig. 1.10: Center portion of a Bias exposure for a 320x4k RoI (*upper panel*), and a average of the profile along the slit (*lower panel*). The mean of the overscan region for each Amp has been subtracted.

There is also a mild curvature in the dispersion direction, which is apparent in the average of a bias residual along

rows (Axis 1). This pattern is best characterized with a function (*spline3* of order 15 or so). This low-order fit in the dispersion direction is co-added to the average along the slit (see above) to form the final **Master Bias Residual** reference image.

---

**Note:**

**The instability of the read-out electronics results in bands, morier patterns, and other features in bias images, most of which are**

1. Perform the overscan correction on each image
1. Determine the mean for each row of the overscan region
2. Subtract from the Amp array a low-order (6-7) fit to this mean column
3. Trim the overscan regions from each Amp section, and concatenate them to form the full image
2. Subtract the matching Master Bias Residual from the image

**The Master Bias Residual files are created in the following way:**

1. Perform overscan correction on several bias exposures, with the following properties in common:
  1. taken on the same night, preferably in sequence
  2. the same *RoI*
  3. the same CCD binning
2. Combine the corrected bias exposures using a clipped mean
3. Form the average of the combined array along both the dispersion ( $x$ -) direction, and also the slit ( $y$ -) direction
4. Create the Master Bias Residual by co-adding the (normalized) slit-averaged array from the fit to the dispersion-averaged array

It is not useful (but probably not harmful) to co-add more than about 10 sequential bias exposures to create a **Master Bias Residual**.

---

## Darks

Because of the low dark-current of the C/KOSMOS sensors, and because of the high and temporally variable read noise (see previous section), dark exposures (finite-length exposures with the shutter closed) are seldom obtained and are not used for data reduction.

## Flat-Field

Flat-field exposures of the dome white-spot or of the twilight sky, possibly including a filter, may in most cases be combined to create a Master Flat-field. Separate flats must be created for each choice of:

- read-out Region of Interest (*RoI*)
- CCD binning factor
- filter, for imaging mode, or
- combination of disperser and aperture for spectroscopic modes
- choice of illumination: dome-flat or twilight sky

The process for creating Flat-field MasterCals is somewhat different for each observing mode: imaging, long-slit, or *MOS*. For the discussion below, it is assumed that the flat-field exposures have been bias-corrected using a Bias Residual MasterCal that matches the *RoI* and CCD binning.

**Imaging Flats** Creating Master Flat-field reference files for imaging mode is the simplest of all.

---

**Note:**

**Process imaging flat-field exposures in a common filter:**

1. Combine the flat images, with outlier rejection, scaling by the mode of the images to account for variations in the flat-field source.
  2. Normalize:
    - (a) Divide by the median or clipped mean of the combined image.
    - (b) Refine the normalization by excluding pixels that deviate from the mean by 20% or so.
    - (c) Condition the flat by setting a floor and ceiling on the pixel values. Corrections to pixel-level response of more than a few tens of percent are not likely to result in high photometric accuracy.
- 

It is best to have no fewer than 5 well exposed flat-field exposures to keep noise in the Master Flat-field from dominating the uncertainties in well exposed portions of science images.

**Spectroscopic Flats** It is best to combine 10 or more well exposed flat-field exposures (if available) to keep noise in the flat-field from dominating the uncertainties. This is largely because of the difficulty of obtaining enough counts in the blue end of the spectrum without saturating the red end.

**Long-Slit Flats** Flat-field exposures with one of the facility long slits may be combined directly, provided they were obtained using a common disperser and slit (including slit location).

---

**Note:**

**Processing long-slit spectroscopic flat-field exposures is straightforward:**

1. Combine the flat images, with outlier rejection, scaling by the mode of the images.
  2. Characterize the shape in the dispersion direction (i.e., the color term) by fitting a polynomial or spline of order 10 or more to an average over the slit (*x*-direction). It is safest to perform the fit interactively. Narrow spectral features must be rejected from the fit.
  3. Divide each column of the combined flat by the polynomial.
  4. Normalize the color-corrected flat-field image.
  5. Condition the flat by setting a floor and ceiling on the pixel values. Corrections to pixel-level response of more than a few tens of percent are not likely to result in high photometric accuracy.
- 

The normalized spectral flats will show a variety of features, including charge traps, dust on the dewar window, and narrow features of the slit throughput, as shown in the figure below.

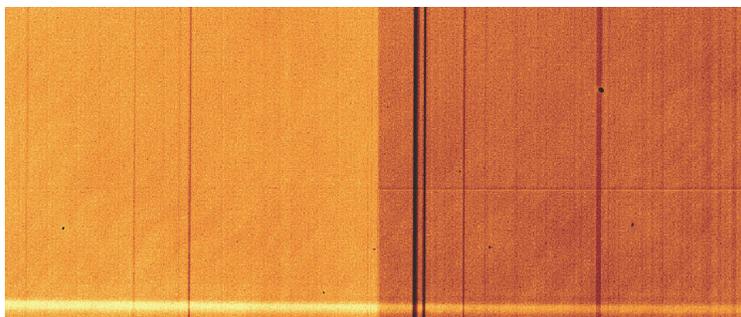


Fig. 1.11: Small central portion of a normalized spectral flat with grating `b2k` and slit `4pxB`. The spatial direction lies along the horizontal axis, and wavelength increases upward along the vertical axis. Note the slit transmission features (dark vertical curves), and the horizontal faux emission from zero-order light (see *Anomalous Spectral Features*).

**Caution:**

If you have C/KOSMOS data obtained during or shortly after the commissioning period, you may notice one or more anomalies.

1. Form an average along the dispersion direction.
2. Interpolate over the emission feature(s).
3. Fit the 1-D average with a polynomial, with an order of at least 10.
4. Divide the image by the polynomial (this involves broadcasting the 1-D function to the dimensions of the average flat).

**To restore the slit illumination function to the interpolated region:**

5. Form an average slit profile of a nearby region (about 50 rows).
6. Multiply the interpolated region by this slit profile.

**Sky Flats** Flat-field exposures of the twilight sky are likely to show sky emission and absorption features at various points in the spectrum, particularly in the extreme blue and red. While these features make them inapplicable for removing variations in pixel-to-pixel sensitivity, they are very helpful when averaged along the dispersion to yield good approximation to the slit illumination function.

**MOS Flats** Flat-fields for Multi-object spectral (*MOS*) mode are much more complicated. Each slit produces its own wavelength dependent flat-field, owing to changes in pixel response with wavelength, and to the fringe pattern. Creating a normalized flat-field for early-epoch spectra compromised by *Anomalous Spectral Features* is especially involved.

MOS flats may be combined, but must be extracted in a similar way to each target spectrum, then normalized, before they can be applied to source spectra.

[More text to be added.]

### Wavelength Calibration

Bias-corrected exposures of the He-Ne-Ar lamp are used to determine the dispersion solution for a given disperser and slit position. An atlas of such a comparison arc is shown below.

The labelled emission lines will likely suffice for initiating the dispersion solution with, e.g., the *identify* task in IRAF. Additional line identifications may be found in the *[IM]*.

**Note:** You should use the high-dispersion line list to determine the dispersion solution, which for IRAF users is found in `linelists$henearhires.dat`. It should be possible to include more than 100 lines in the dispersion solution (with an

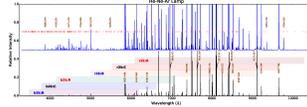


Fig. 1.12: High resolution He-Ne-Ar spectrum at full scale (*black*) and at 10x (*blue*, offset vertically for clarity). Identifiable lines are marked (*red ticks*), and some of the brighter or more isolated lines are labelled. Approximate wavelength coverage for combinations of the VPH and the facility longslits is shown (*colored bars*). Click image to enlarge.

RMS less than a few tenths of a pixel) for most grisms and slit positions. The fit will require a polynomial of order 6 or 7 to yield an acceptable solution.

**Dispersion Solution** The grism dispersers used in C/KOSMOS introduce significant nonlinearity to the dispersion relation, which can generally be well characterized with a Legendre or Chebychev polynomial of order 6 or 7. If a non-linear dispersion solution is written directly into the FITS header (as it is by the IRAF task `dispcor` when linearization is turned off), it will consist of a number of terms including the coefficients of the fitted polynomial. The coefficients are described in the paper [The IRAF/NOAO Spectral World Coordinate Systems](#) (1991, F. Valdes). The following excerpt describes how to compute wavelengths from the nonlinear function of choice.

There are three coordinates of relevance: the pixel coordinate  $p$  of the spectrum array; the normalized coordinates  $n$  over the domain of the fitting function, in the interval  $[-1, 1]$ ; and the world coordinates  $w$  at each pixel. The transformation from pixel to normalized coordinates  $n$  is:

$$n = \frac{2p - (p_{max} + p_{min})}{(p_{max} - p_{min})}$$

Note that in practice the range of pixels will extend somewhat beyond the domain over which the fitting function was defined. For a single function type (the usual case unless comparison arcs taken immediately before and after a science exposure are used to refine the wavelength zero-point), the transformation from pixel coordinates  $p$  and world coordinates  $w$  is:

$$w = \frac{\Delta\lambda + \Lambda(p)}{(1 + z)}$$

where  $z$  is the Doppler factor. The dispersion function  $\Lambda(p)$  at pixel  $p$  can be evaluated over the function coefficients  $c_i$ :

$$\Lambda(p) = \sum_{i=1}^{order} c_i x_i$$

where  $x_1 = 1$ ;  $x_2 = n$ . The non-linear terms for order  $i > 2$  may be computed recursively; for Chebyshev polynomials we have:

$$x_i = 2nx_{i-1} - x_{i-2}$$

or for Legendre polynomials:

$$x_i = \frac{(2i - 3)nx_{i-1} - (i - 2)x_{i-2}}{(i - 1)}$$

IRAF spectroscopic tasks have a built-in capability to read dispersion solutions with the above form. For python users the following snippet of code may be used to construct a wavelength array from Legendre or Chebyshev function parameters and coefficients, using the functions in `poly.py`:

```

import numpy as np
import poly as pl

# Extract function, parameters, and coefficients from WAT2_00x keywords.
# An example from the center of an arc comparison exposure using grism b2k
# and facility longslit 3pxC:
pMin, pMax = [90.65229797363281, 4054.766357421875]
c = np.array([5160.180854771875, 1399.010545377342, 64.60055185877235, -24.74632014374652, 0.13134655])
nPix = 4096

# Generate an array of world coordinates (in Angstroms, the declared WCS unit).
n = pl.getNormCoords(pMin, pMax, nPix)
w = pl.evDispersion(pl.evLegendre, c, n)

```

## Photometric Calibration

Whether the observing mode is spectroscopic or imaging, you will need a catalog of standards to calibrate the final data product.

Skip to *In a Nutshell: Processing with IRAF* or *In a Nutshell: Processing with Python*.

## World Coordinate Systems

### Approximate World Coordinate Calibration

It is useful to have at least an approximate World Coordinate Solution (*WCS*) specified in the header of your science images; this solution can be refined later in target processing. Unfortunately, the WCS-related keywords stored in the headers of raw images by the data taking system, and labelled “Placeholder WCS” in the header keyword `WCSASTRM`, are not complete nor are extant keyword values accurate. The python script `wcsTask.py` will fix the problem for bias-corrected images and spectra; the needed corrections are described in the subsections below.

**Imaging WCS** For exposures taken with C/KOSMOS in imaging mode, the WCS may be of scientific interest even when imaging was not the focus of the original observing program. For observing programs that used custom MOS slits, the WCS in acquisition images is helpful for associating slit locations with specific targets (or regions within extended astronomical objects).

**Setting the WCS Description** An approximate solution was derived by the above script from the telescope alignment during each exposure, and the instrument rotator angle (taken from `GCCROTAT` in the header). While several of the keyword values are correct as inserted by the data taking system, others must be updated. The following table lists the keywords that are inserted or updated with `wcsTask.py` to specify a complete FITS WCS in the image extension header.

Table 1.5: C/KOSMOS Image WCS Keyword Changes

Keyword	Update	Value	Meaning
RADECSYS	Deleted	–	Deprecated keyword
WCSASTRM	Deleted	–	Not used for WCS
RADESYS	Added	FK5	Celestial coordinate reference frame
WCSAXES	Added	2	Number of axes in WCS description
CTYPE1	Updated	RA---TAN	Coordinate type for axis1
CTYPE2	Updated	DEC--TAN	Coordinate type for axis2
CUNIT1	Updated	deg	Coordinate units for axis1
CUNIT2	Updated	deg	Coordinate units for axis2
LTV1	Updated	0	CCD to image offset: axis1
LTV2	Updated	0	CCD to image offset: axis2
CDi_j	Updated	(see below)	Derivative of World Coordinate values <i>i</i> w.r.t pixel array <i>j</i> at the reference location

The CD matrix is given by the following:

$$\begin{pmatrix} \text{CD1}_1 & \text{CD1}_2 \\ \text{CD2}_1 & \text{CD2}_2 \end{pmatrix} = \begin{pmatrix} \sigma \cos(90 + \rho) & -\sigma \sin(90 + \rho) \\ -\sigma \sin(90 + \rho) & -\sigma \cos(90 + \rho) \end{pmatrix}$$

where:

$$\begin{aligned} \rho &= \text{GCCROTAT} \\ \sigma &= (\text{platescale})/3600 \end{aligned}$$

**Refining the Reference Coordinates** The world coordinates at the reference pixel are taken from the commanded telescope pointing, which may be off by up to a few arcmin. The WCS zero-point can be adjusted by correcting the CRVALi keywords with offsets determined from stars in the field. Often this correction can be determined using the SAOImage DS9 image display tool. The process is:

- Process the image through bias- and flat-fielded correction, and run `wcsTask` to initialize the WCS
- Display the image in DS9
- From the pull-down menu, select “WCS → degrees” for the coordinate display
- Select **Analysis** → Catalogs → Optical → USNO UCAC3
- Compare the pattern of star locations with those of the catalog, as shown below

**Highly Accurate WCS** If your science objectives require a highly accurate WCS you must determine a full WCS solution with community software, such as the IRAF `mscred.msfinder.msctpeak` task (see the [tutorial](#)). An astrometric catalog will be needed for this calibration; magnitudes in the same bandpass will be needed for the photometric calibration. Although the process to fit a full WCS solution is involved, it is possible to characterize optical distortions into the WCS (using the TNX projection); RMS uncertainties of 200mas should be achievable.

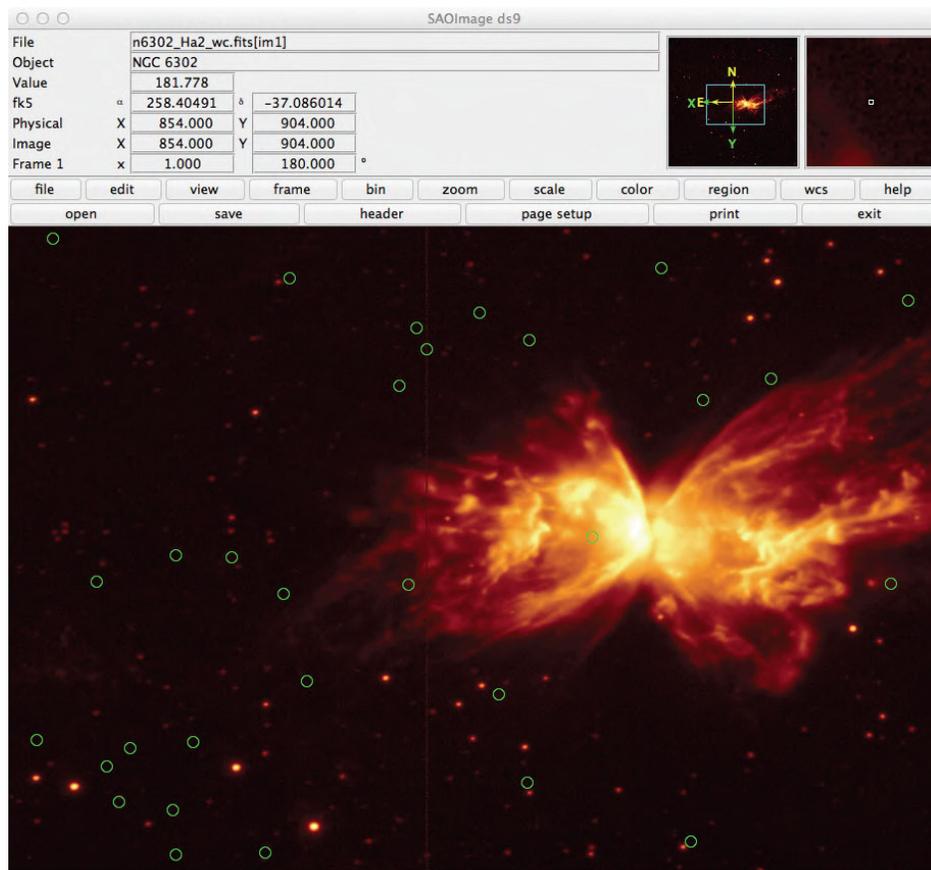


Fig. 1.13: Image of NGC 6302, with positions of catalog stars plotted (*green circles*). Note the “c” shaped pattern of catalog stars (bottom, left) appears to match that of stars in the image (bottom, center).

- Select any star from the catalog (sorting by RA or Dec may help) and:
  - record the coordinates from both the catalog star and the image display cursor at the position of that star in the image
  - compute the difference (i.e., the offset values in degrees) in each coordinate
- Update the CRVAL1 and CRVAL2 keyword values with these offsets

**Spectroscopic WCS** An approximate WCS may be useful even for MOS mode. The default WCS consists of a linear dispersion relation (which is only a crude linear approximation) and a linear spatial displacement from the center of the slit. As with imaging, a number of keywords need to be added or updated in the image extension:

Table 1.6: C/KOSMOS Spectral WCS Keywords

Keyword	Value	Meaning
WCSAXES	2	Number of axes in WCS description
CTYPE1	LINEAR	Coordinate type for axis1
CTYPE2	AWAV-A2W	Coordinate type for axis2
CRVAL1	0.	Coordinate value at reference pixel for axis2
CUNIT1	arcsec	Coordinate units for axis1
CUNIT2	Angstrom	Coordinate units for axis2
CDi_j	(see below)	Derivative of World Coordinate values $i$ w.r.t pixel array $j$

The central wavelength and average dispersion for each facility longslit is given in the table below; the spectral coordinates are accurate to about 10 Ang near the center of the image. The first-order dispersion term, applicable at the center of the image, is also given. The dispersion is well fit with a Legendre polynomial with an order of 6 or 7. The fit residual should be only a few tenths of an Ang with 100 or more identified features using the `henearhres` line list.

Table 1.7: C/KOSMOS Spectral WCS Values

Slit	Gen. Wave	d_Wave/dx
b2kB	4715.	0.701
b2k	5110.	0.723
b2kR	5620.	0.740
r2kB	6930.	1.030
r2k	7510.	1.043
r2kR	8100.	1.053

### Refined World Coordinate Calibration

**Refined Imaging WCS** Solving for the WCS in a image requires using community software. There are a few possibilities, some of which also require local access to an astrometric catalog. The options include:

- The [IRAF MSCRED package](#) (requires catalog)
- The [astrometry.net](#) software (downloading & installing software may be necessary)
- Using [Aladin](#) to determine WCS, and transferring the keywords to the header manually
- Using the [USNO image and catalog service](#)

**Refined Spectroscopic WCS** TBD

**Advanced Longslit WCS** It is possible to create a linear WCS that will describe the RA and Dec along the slit, as well as wavelengths in the dispersion direction, by introducing a degenerate third image axis, as described by Calabretta & Greisen (2002, *A&A*, 395, 1077; Sect. 7.4.3).

**MOS-mode WCS** TBD

### Anomalous Spectral Features

In early-epoch C/KOSMOS data (during the commissioning period) anomalous “emission features” may be apparent in flats, comparison arcs, and science data. The features arise because the grating holder allows some zeroth-order (i.e., non-dispersed) light to pass through the VPH grating, as illustrated below.

**Warning:** The narrow pseudo-emission features, if present, must be **removed** from the Master Flat-field in order to keep from imprinting these features on science data during calibration.

A temporary hardware fix to attenuate the undispersed light has been implemented while parts are fabricated for a permanent solution. A pair of closely spaced absorption features is visible in the r2k spectra of the flat-field lamp, but their amplitudes are only about 5% and small enough to ignore.

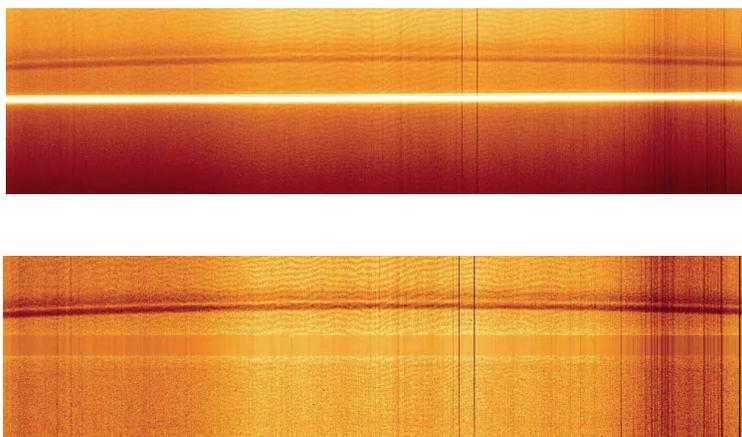


Fig. 1.14: Portion of a flat-field image near the center, taken with the COSMOS r2k grism. The illuminated longslit (above, from unobstructed zero-order light) shows up as a faux emission feature in the spectrogram; a genuine but weak absorption doublet can be seen just above it. Below, the emission has been interpolated over and the flat has been normalized by an average color term. Note that the slit illumination function was preserved (from an average of a nearby region), although the pixel-to-pixel sensitivity variation is lost in the corrected region.

The anomalous emission features in Multi-object spectral mode are much more complicated, as illustrated below.

The illuminated regions in this flat-field image differ in intensity because the MOS slits are of different size: typically the slits used for the acquisition of field stars are larger than those used for targets. The illuminated slit appears as a faux emission feature just to the left of each slit spectrum (and overlap it if the slit is wide enough) because the disperser introduces some field distortion which does not affect the zero-order light. The “emission” features are displaced vertically in proportion to the displacements in the dispersion direction of the slits in the mask.

Return to [Processing Science Data](#)

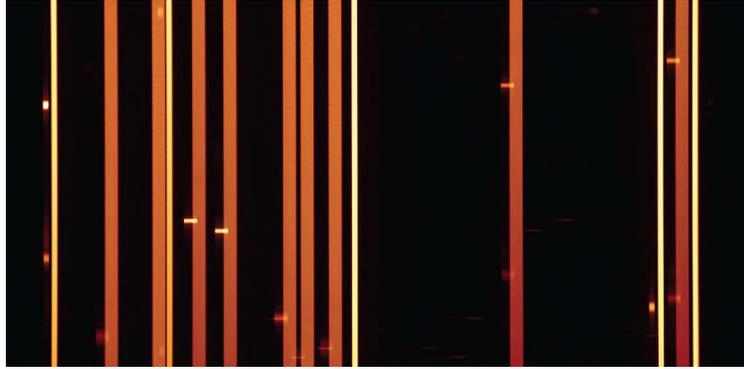


Fig. 1.15: Portion of a MOS flat-field image near the center, taken with the COSMOS r2k grism, with a square-root intensity stretch. Spectra from each slit run vertically, in the y-direction.

### 1.3.3 Reduction Work-flow

The following chapters provide a step-by-step workflow for processing, including invocation of specific tasks or tools needed to progress through data reduction. The nature of workflow (and indeed, even the nature of the output data products) is different for IRAF users, vs. Python users.

#### In a Nutshell: Processing with IRAF

IRAF (version 2.16.1) offers a large number of tools for data reduction and calibration. The recipes below provide a viable **but not unique** path for processing your C/KOSMOS science data. This tutorial assumes you have retrieved all the science and supporting calibration exposures of interest from the [NOAO Science Archive](#). See *Getting Started* for details.

---

**Note:** In the code blocks below the commands are intended to be executed within the IRAF command language (**cl**), unless otherwise noted. The traditional prompt (e.g., “cl>”) has been omitted to make it easier to cut-and-paste commands into your own IRAF session. Terse comments are prepended with a pound sign (“#”, or hashtag).

---

#### Contents

- *In a Nutshell: Processing with IRAF*
  - *Processing Scripts*
  - *File Selection*
    - \* *Uncompress Files*
    - \* *Observing Log*
    - \* *Simple FITS Format*
  - *Basic Exposure Processing*
    - \* *Bias Correction*
    - \* *Flat-field Correction*
  - *Image Processing*
    - \* *WCS Calibration*
    - \* *Photometric Calibration*
  - *Spectral Processing*
    - \* *Long-Slit Spectra*
    - \* *MOS Spectra*
    - \* *Common Processing for Long-Slit and MOS*

## Processing Scripts

A small set of IRAF scripts is available to perform portions of the data reduction process. Download the `cosmos` package and unpack the tar in a convenient directory, such as your start-up IRAF directory. You may load the package explicitly from within an IRAF session, or insert the following commands into your `loginuser.cl` startup file:

```
# Load the cosmos package:
set cosmosdir = /path/to/cosmos/package/
task $cosmos = cosmosdir$cosmos.cl
cosmos
```

These scripts should be regarded as illustrative, and will be used along with existing IRAF tools in the data reduction recipes below. You can access the help files for these tasks, e.g.:

```
help cosmosdir$doc/mkBias.hlp fi+
```

## File Selection

**Uncompress Files** The first order of business is to uncompress all of the tile-compressed FITS files you retrieved from the archive, because IRAF tasks cannot read them directly. In the directory that contains the data to be processed:

```
# Uncompress the files using the fitsutil.funpack task.
# Remove the compressed files when finished.
funpack *.fits.fz keep-
```

**Observing Log** It is helpful to build an observing log from information in the file headers. This may be accomplished with the python script `obslog.py`, or with the IRAF task `imutil.hselect`:

```
# Include the header keyword names in the "fields" string below.
string keywords = "$I dtcaldat object obstype obsmode ccdsum nocroipt ksfilcmd ksdpos gcrotat date-

# Write more meaningful field names to the log:
string fieldNames = "# File CalDate Object ObsType ObsMode RoI CcdBin Filter Disperser Slit Rotator I

# Execute the task in program mode and redirect output to the log file:
print (fieldNames, > "obsLog.txt")

# Build the log, excluding "focus" exposures.
# Note the use of the first HDU header, which inherits from the PHU of raw files.
hselect ("*.fits[1]", keywords, "obstype!='focus'", >> "obsLog.txt")
```

Modify the above strings to make lists of files for creating, e.g., Master Reference files, or flat-fields.

**Simple FITS Format** It is worth noting that IRAF `cl` scripts can read image extensions, but are unable to create them *ab initio*. While the `fitsutil` external package can create and edit multi-extension FITS (MEF) files, it is very cumbersome to use these tasks for routine `cl` processing.

**Note:** The IRAF examples in this chapter will create and use SIF (simple image format) files, where the pixel array is stored in the PHDU, with no image extensions. Thus, data quality masks and other concomitant arrays, if created, must reside in separate files.

## Basic Exposure Processing

Basic image processing is often accomplished in IRAF with the `imred.ccdred.ccdproc` task, or with tasks in the `imred.ccdred` package. However, the modalities for C/KOSMOS data require complex selection criteria that are not well addressed with `ccdproc`. Still, creating custom scripts is a viable approach for processing.

**Bias Correction** Overscan correction is best done by subtracting from the Amp image a low-order fit to the mean along the slit (axis-1) of the overscan region, and then correcting the residual bias pattern with a 2-D **Master Bias** (see *Bias Residual* for details). These steps can be performed on an entire list of files with the `ckProc.cl` script which, as noted, outputs SIF files (i.e., simple FITS files with no image extensions). This task is first used to perform the overscan correction on a series of bias exposures. The `mkBias.cl` script combines these corrected input files (with 3-sigma clipping) to create a **Master Bias-Residual** file.

```
# Bias exposures must match the RoI, binning, and observation date:
string biasSelect = "obstype=='zero' && nocroipt=='2kx2k' && ccdsum=='1 1' && dtcaldat=='2014-04-13'"

# Perform overscan correction on bias exposures
hselect ("*_zri.fits[1]", "$I", biasSelect, > "biasfiles.txt")
ckProc ("@biasfiles.txt")

# Create a list of files, then create a Master Bias-residual.
hselect ("*_zri_oc.fits", "$I", biasSelect, > "biascomb.txt")
mkBias ("@biascomb.txt", "bias2k.fits")
```

**Flat-field Correction** Constructing a **Master Flat-field** is largely a matter of combining flat-field exposures, with appropriate scaling, outlier rejection, normalization, and conditioning. This can be accomplished for a list of flat-field exposures with the `mkFlat.cl` script. Note that the Master Flats must match the *CCD binning, RoI, and filter*; and in addition for spectroscopic mode the *disperser* and *slit* of the images to be processed. Flats are also commonly grouped by *calendar date*.

```
# Perform bias correction on matching raw flat-field files.
string flatSelect = "obstype=='flat' && obsmode=='imaging' && nocroipt=='2kx2k' && ccdsum=='1 1' && dtcaldat=='2014-04-13'"
hselect ("*_fri.fits[1]", "$I", flatSelect, > "flats_Ha.txt")
ckProc @flats_Ha.txt biasref=bias2k.fits

# Create Master Flat-field reference file.
# Set hard limits for min and max flat-field correction.
hselect ("*_fri_bc.fits", "$I", flatSelect, > "flatscomb_Ha.txt")
mkFlat @flatscomb_Ha.txt flat2k_Ha.fits hilim=1.2 lolim=0.2

# Apply flat-field correction to a list of matching files:
ckProc @img_Ha_files.txt biasref=bias2k.fits flatref=flat2k_Ha.fits
```

*Additional normalization is necessary for spectroscopic flats to remove the color term in the dispersion direction. See [Sensitivity Calibration](#).*

**Caution:** Flat-field exposures obtained with a custom slit-mask require special handling, and cannot be normalized in the way described below. See XXX for details.

For convenience, here is a selection for a spectroscopic flat with disperser `r2k` and slit `4pxR`:

```
# Select spectroscopic flats.
string flatSelect = "obstype=='flat' && ksdpos=='r2k' && ksspos=='4pxR' && obsmode=='sos_slit' && nocroipt=='2kx2k' && ccdsum=='1 1' && dtcaldat=='2014-04-13'"
```

## Image Processing

Starting with a list of bias- and flat-field corrected image exposures, it remains is to establish a world coordinate system and the photometric calibration coefficients for each of the relevant passbands.

```
# Perform basic processing on selected science exposures (H-alpha filter):
string ha_select = "obstype=='object' && obsmode=='imaging' && nocroipt=='2kx2k' && ccdsum=='1 1' &&
hselect ("*_ori.fits[1]", "$I", ha_select, > "img_Ha.txt")
ckProc @imgbc_Ha.txt biasref=bias2k.fits flatref=flat2k_Ha.fits
```

If there is more than one exposure of the same field, you may wish to stack them. This is straightforward with `imcombine` if the exposures have the exact same telescope alignment, but if they are spatially dithered then it is vital to first perform an accurate WCS calibration.

**WCS Calibration** Establishing a world coordinate description for images consists of fixing and adding keywords (using e.g., the `imutil.hedit` task) to the image headers. It also means establishing the transformation matrix for the celestial coordinates, which depends upon the instrument rotator angle (see *Imaging WCS*). Use the `wcsfix` task on flat-fielded exposures obtained in imaging mode (regardless of filter).

```
string img_select = "obstype=='object' && obsmode=='imaging' && nocroipt=='2kx2k' && ccdsum=='1 1' &&
hselect ("*_ori_fc.fits", "$I", img_select, > "imgbc_Ha.txt")
wcsfix @imgbc_Ha.txt update+
```

Now the images have a complete WCS description in the header, with values for the reference coordinate and rotation angle as obtained in the observing environment. These values may not be sufficiently accurate: the RA/Dec may be off by as much as a few arcmin and the rotation may be off by a fraction of a degree. A correction to the reference point can be determined using the DS9 image display server, as described in *Refining the Reference Coordinates*. Add these offsets to the `CRVAL1` and `CRVAL2` keyword values using `hedit`:

```
# Example: selected star's coordinates in myImage.fits:
# DS9:      258.42616,   -37.10972
# Catalog: 258.4609115, -37.1147084
hselect myImage.fits CRVAL* yes
258.422125      -37.0957222

# Compute difference and add to CRVALi:
x = 258.422125 + (258.4609115 - 258.42616)
y = -37.0957222 - ( 37.1147084 - 37.10972)
hedit ("myImage.fits", fields="CRVAL1", value=x, update+)
hedit ("myImage.fits", fields="CRVAL2", value=y, update+)
```

To refine the WCS still further, and to characterize distortions in the image (using the `TNX` projection), you can use the `IRAFmscfinder.msctpeak` task; a [tutorial](#) is available. This task requires a reasonably good approximate WCS, so the above steps are likely to be necessary.

**Caution:** If you use the image display server `SAOImage DS9`, you may need to set an environment variable **before starting the cl** in order for `IRAF` to communicate with it.

```
setenv IMTDEV inet:5137      # (t)csh users
export IMTDEV="inet:5137"   # bash users
```

**Photometric Calibration** Photometric calibration is beyond the scope of this *Cookbook*, and there no standard way of recording the calibration within the images. However the mechanism in its simplest form is straightforward: Use a photometry program such as `SExtractor` to measure instrumental magnitudes in each passband for detected stars, and

determine the photometric zero-point for each image. Complications arise for very crowded fields (where stellar PSFs overlap), and PSF shapes that vary over the *FoV*.

## Spectral Processing

### Long-Slit Spectra

**Flat-Field Normalization** *Basic Exposure Processing* for spectral-mode exposures proceeds identically to that for images through the combination of flat-field exposures. The next step for long-slit spectra is to remove the color term from the flats using the `twodspec.longslit.response` task. If there are anomalous spectral features, these must be excluded from the fit to the color term. A relatively large order will be needed, as well as some clipping of deviant points

```
response flat_b2k4pxB.fits flat_b2k4pxB.fits flatNorm_b2k4pxB.fits interact+ function=spline3 order=
```

The shape of a dome flat-field averaged in the dispersion direction (i.e., the slit function) includes the slowly varying effect of dome illumination which is not identical to sky illumination. This pattern should be removed if possible to ensure photometric uniformity everywhere along the slit. This illumination pattern can be characterized with sky flats, if they exist (and they normally don't).

Note that the high-frequency structure in the slit function likely reflects actual variation in slit throughput.

**Combining Science Exposures** Very often observers will obtain multiple exposures of the same science target(s) with identical telescope alignment, in order to reject cosmic rays. If multiple exposures for the same field are identical in all respects except exposure time (and are **not spatially dithered**), they may be processed through flat-field correction and combined using `imcombine`.

**Trace, Aperture Definition, and 1-D Extraction** Further calibration processing (wavelength and sensitivity calibrations) is performed on 1-D spectra. The `longslit.apextract.apall` task is designed to extract 1-D spectra from the 2-D spectrograms. (See the tutorial.)

**MOS Spectra** A key difference between MOS processing and that for long-slit is that the flat-field must be extracted and applied to each slit spectrum (and background region) individually, since the wavelength scales are different for each slit. After 1-D extraction the processing is identical to that for long-slit mode. [Remaining material TBD]

### Common Processing for Long-Slit and MOS

**Wavelength Calibration** Establishing the spectral WCS is a two-step process: first extract a 1-D spectra from a comparison arc exposure using the extraction parameters as were used for each target, and establish the dispersion solution using `onedspec.identify`, then apply that solution to the target spectra using `onedspec.dispcor`. Reference dispersion solutions have been created, so in may be helpful to download and use the `identify_r2k.par` pset from one solution (for r2k/3pxC) and edit it for the disperser and slit of your choice. To use the pset,

```
epar identify
# ...in the eparam editor, read the pset from disk:
:r identify_r2k.par
```

For the initial central wavelength and first-order dispersion for the facility longslits, see *Spectroscopic WCS*. Note that when deriving the dispersion relation you should use: a Legendre or Chebyshev polynomial of order 6 or 7, and the linelist `henearhres.dat` to get an accurate solution. See *Wavelength Calibration* for details.

**Sensitivity Calibration** The sensitivity calibration is derived using spectra of observed standard stars. The IRAF task `onedspec.standard` integrates the spectra of the standard stars over the calibration bandpasses and associates that tabulation with the reference magnitudes for the standard stars. The IRAF task `onedspec.sensfunc` actually applies the correction for atmospheric dispersion to the integrated spectra and computes the sensitivity function (i.e., the ratio of instrumental mags to standard mags) for the set of standards and input spectra from `standard`.

[Remaining material TBD]

## In a Nutshell: Processing with Python

### Packages and Tools

Python and some common supporting packages provide a viable path for C/KOSMOS data reduction. A collection of tools will be used in the tutorial below. The tools are known to be compatible with Python v2.7.x (but not Python 3.x). The *required packages* are:

- `numpy` – numerical python with N-dimensional arrays
- `astropy` – Astronomy core package for python
- `scipy` – Python open-source software for math, science, and engineering

All of these packages (and many more) are included in the [Anaconda python distribution](#), which is highly recommended. You will probably also want to install the [DB browser for SQLite](#). All of the host-level tools use the python [argparse library](#), so basic help for these command-line tasks may be viewed in the standard way:

```
python <taskname>.py -h  
  
# or if the execute privilege is set:  
<taskname>.py -h
```

The recipes below provide a viable **but not unique** path for processing your C/KOSMOS science data.

**Note:** In the code blocks below the commands are intended to be executed from the Unix command line, unless otherwise noted. The traditional prompt (e.g., “% ”) has been omitted to make it easier to cut-and-paste commands into your session. Terse comments are prepended with a pound sign (“#”, or hashtag).

**Caution:** The python tools are not yet fully developed. They can be used to perform basic image processing, but not for extracting 1-D spectra and calibrating them.

**Contents**

- *In a Nutshell: Processing with Python*
  - *Packages and Tools*
  - *Observing Log*
  - *Basic Exposure Processing*
    - \* *Bias Correction*
    - \* *Flat-field Correction*
  - *Image Processing*
    - \* *WCS Calibration*
    - \* *Photometric Calibration*
  - *Spectral Processing*
    - \* *Long-Slit Spectra*
    - \* *MOS Spectra*
    - \* *Common Processing for Long-Slit and MOS*

**Observing Log**

An observing log, constructed as an SQLite database from information in the file headers, is used by the python tasks in this tutorial to manage data processing. Build the observing log with the python script `obslog.py`:

```
python obslog.py -d obsLog.sqlite3
# Create an ASCII version too, by redirecting STDOUT to a file
python obslog.py -d obsLog.sqlite3 -v > obsLog.txt
```

Most tools described below use SQL with this database to select appropriate files on which to operate. You may view the contents directly and flexibly with the [DB browser for SQLite](#).

**Basic Exposure Processing**

Basic image processing is accomplished primarily with the `kosmos.py` library; the command-line tasks for the most part parse the user input and call the library to process data.

**Bias Correction** Overscan correction is best done by subtracting from the Amp images a low-order fit to the mean along the slit (axis-1) of the overscan region, and then correcting the residual bias pattern with a 2-D Master Bias (see *Bias Residual* for details). The `mkBias.py` task performs overscan correction on all bias exposures that match the selection criteria, combines them using a median or mean, and then constructs a **Master Bias Residual** file from the cross-product of the average slit and the fitted dispersion vectors.

Note that the default 1x1 CCD binning is assumed, and verbose output is turned on.

```
# Create the Master Bias Residual reference file:
python mkBias.py obsLog.sqlite3 bias2k_2015Apr09.fits MEAN -r 2kx2k -c 2014-04-09 -v
```

The `biasTask.py` task performs overscan and (optionally) bias correction on selected images. This task is mostly useful for evaluating the bias correction independently of other processing.

```
# Perform bias correction on selected exposures:
python biasTask.py obsLog.sqlite3 -r 2kx2k -f *.fits -v
```

**Flat-field Correction** Constructing a **Master Flat-field** is largely a matter of combining flat-field exposures, with appropriate scaling, outlier rejection, normalization, and conditioning. This can be accomplished for a list of flat-field exposures with the `mkFlat.py` task. Note that constituent exposures must match the *CCD binning*, *RoI*, and *filter*; and in addition for spectroscopic mode the *disperser* and *slit* of the images to be processed. Flats are usually also grouped by *calendar date*.

```
# Create a flat-field MasterCal file for H-alpha imaging. Defaults assumed for:
# slit, disperser == 'Open'
# CCD binning == '1 1'
# obsmode == 'imaging'
python mkFlat.py obsLog.sqlite3 bias2k.fits flat2k_Ha.fits -r 2kx2k -f Halpha -c 2014-04-09 -v

# Perform bias and flat-field corrections on images:
python flatTask.py bias2k.fits flat2k_Ha.fits <images>_ori.fits.fz -c 2014-04-09 -v
```

Additional normalization is necessary for spectroscopic flats to remove the color term in the dispersion direction. See *Sensitivity Calibration*.

**Caution:** Flat-field exposures in *MOS* mode require special handling. While they can ordinarily be combined, they cannot be normalized in the way described below. See *Flat-Field* for details.

## Image Processing

Starting with a list of bias- and flat-field corrected image exposures, it remains to establish a world coordinate system and the photometric calibration coefficients for each of the relevant passbands.

If there is more than one exposure of the same field, you may wish to stack them. This is straightforward with the `imcombine` task if the exposures have the exact same telescope alignment, but if they are spatially dithered then it is essential to first perform an accurate WCS calibration.

**WCS Calibration** Establishing a world coordinate description for images consists of fixing and adding keywords (using e.g., the `imutil.hedit` task) to the image headers. It also means establishing the transformation matrix for the celestial coordinates, which depends upon the instrument rotator angle (see *Imaging WCS*). Use the `wcsfix` task on flat-fielded exposures obtained in imaging mode.

```
% python wcsfix.py <image>_fc.fits.fz
```

Now the images have a complete WCS description in the header, with values for the reference coordinate and rotation angle as obtained in the observing environment. These values may not be sufficiently accurate: the RA/Dec may be off by as much as a few arcmin and the rotation may be off by a fraction of a degree. A correction to the reference point can be determined using the DS9 image display server, as described in *Refining the Reference Coordinates*. Add these offsets to the `CRVAL1` and `CRVAL2` keyword values using `hedit`:

```
# Example: selected star's coordinates in myImage.fits:
# DS9:      258.42616,   -37.10972
# Catalog:  258.4609115, -37.1147084
cl> hselect myImage.fits CRVAL* yes
258.422125      -37.0957222

# Compute difference and add to CRVALi:
wcsTask.py <image>_fc.fits.fz -d <delDec> -r <delRa> -u
```

To refine the WCS still further, and to characterize distortions in the image (using the `TNX` projection), you can use the IRAF `mscfinder.msctpeak` task; a [tutorial](#) is available. This task requires a reasonably good approximate WCS,

so the above steps are likely to be necessary. It is also possible, though perhaps more difficult, to use other software such as [astrometry.net](http://astrometry.net) to determine the WCS.

**Photometric Calibration** Photometric calibration is beyond the scope of this *Cookbook*, and there no standard way of recording the calibration within the images. However the mechanism in its simplest form is straightforward: Use a photometry program such as [SExtractor](#) to measure instrumental magnitudes in each passband for detected stars, and determine the photometric zero-point for each image. Complications arise for very crowded fields (where stellar PSFs overlap), and PSF shapes that vary over the *FoV*.

### Spectral Processing

#### Long-Slit Spectra

**Flat-Field Normalization** *Basic Exposure Processing* for spectral-mode exposures proceeds identically to that for images through the combination of flat-field exposures. The next step for long-slit spectra is to remove the color term from the flats using the `normFlat.py` task. If there are anomalous spectral features, these must be excluded from the fit to the color term. A relatively large order will be needed, as well as some clipping of deviant points

The shape of a dome flat-field averaged in the dispersion direction (i.e., the slit function) includes the slowly varying effect of dome illumination which is not identical to sky illumination. This pattern should be removed if possible to ensure photometric uniformity everywhere along the slit. This illumination pattern can be characterized with sky flats, if they exist (and they normally don't).

Note that the high-frequency structure in the slit function likely reflects actual variation in slit throughput.

**Combining Science Exposures** Very often observers will obtain multiple exposures of the same science target(s) with identical telescope alignment, in order to reject cosmic rays. If multiple exposures for the same field are identical in all respects (except exposure time), they may be processed through flat-field correction and combined (with artifact rejection).

**Warning:** Python tools for all subsequent steps have yet to be developed.

**Trace, Aperture Definition, and 1-D Extraction** Further calibration processing (wavelength and sensitivity calibrations) is performed on 1-D spectra.

**MOS Spectra** A key difference between MOS processing and that for long-slit is that the flat-field must be extracted and applied to each slit spectrum (and background region) individually, since the wavelength scales are different for each slit. After 1-D extraction the processing is identical to that for long-slit mode. [Remaining material TBD]

#### Common Processing for Long-Slit and MOS

**Wavelength Calibration** [Remaining material TBD]

**Sensitivity Calibration** [Remaining material TBD]

**Caution:** IRAF (and the associated *cosmos* package) provide a complete set of tools for reducing all modes of C/KOSMOS data. The python tools only provide complete reductions for “imaging” mode, although many of the tasks are applicable to spectroscopic modes. A more complete python package may be developed in the future.

## 1.4 Useful Resources

Resources for your data reduction needs are collected here for easy reference. Information about the C/KOSMOS instrument design, operation, and observing may be found on the [C/KOSMOS Home Page](#). For help with C/KOSMOS data reduction contact [kosmos\(at\)noao\(dot\)edu](mailto:kosmos(at)noao(dot)edu).

### 1.4.1 Archive Portal

Users of C/KOSMOS should retrieve their data from the [NOAO Science Archive Portal](#). No account is necessary to retrieve data, except for recent data covered by a proprietary period during which only the PI or designated Co-Is of the observing program have access. Calibration data are not proprietary. For help with the NSA Portal, contact the Portal HelpDesk: [sdmhelp\(at\)noao\(dot\)edu](mailto:sdmhelp(at)noao(dot)edu).

### 1.4.2 Standards and Catalogs

- [He-Ne-Ar reference spectrum](#) (atlas, in FITS format) and [ASCII line list](#).
- [Spectrophotometric standard stars](#) from ESO.
- [DSS ESO Online Digitized Sky Survey](#)
- [USNO Image and Catalog Archive Server](#)

### 1.4.3 Software Tools

#### IRAF Tools

IRAF (version 2.16.1) has a variety of packages for the reduction and calibration of images and longslit spectra. For help with IRAF software, post a message to [iraf.net](mailto:iraf.net)

Table 1.8: IRAF Data Processing References

Resource	Description
<a href="#">WCS Tutorial</a>	<i>Creating a Mosaic World Coordinate System</i> 2000 by F. Valdes (Tucson: NOAO/IRAF). Although the material is intended for images from the NOAO MOSAIC Cameras it is quite general and can be applied to most images. <b>Note:</b> many of the links on this page are stale.
<a href="#">Slit Spectral Reductions</a>	<i>User's Guide to Reducing Slit Spectra with IRAF</i> [MVB]
<a href="#">LS Spectral Extraction</a>	Tutorial for extracting longslit spectra using the <i>doslit</i> task.

A small package of custom processing scripts, `cosmos`, is available for processing your data. See: *In a Nutshell: Processing with IRAF* for details. These scripts are meant to be illustrative of the process; users may wish to customize them for their specific scientific needs.

#### Python Tools

The python tools used within this *Cookbook* (see *In a Nutshell: Processing with Python*) are available as the `kosmos` package. These tools are sufficient for reduction of exposures obtained in **imaging mode**, but do not presently provide for **spectroscopic mode** reductions.

Table 1.9: Python Processing Tools

File	Description
<code>biasTask.py</code>	Apply bias correction to raw files.
<code>flatTask.py</code>	Apply bias- and flat-field corrections to raw files.
<code>mkBias.py</code>	Create Master Bias Residual reference file.
<code>mkFlat.py</code>	Create Master Flat-field reference file.
<code>kosmos.py</code>	Core library of processing tools.
<code>normFlat.py</code>	Normalize a spectral Master Flat-field reference file and remove anomalous “emission” feature from zero-order light.
<code>obslog.py</code>	Creates a log of observations based on FITS file metadata. Output is an sqlite database and optionally an ASCII representation.
<code>wcsTask.py</code>	Fixes and updates WCS-related keywords in science exposures. Keyword values are approximate and must be refined in subsequent calibration steps.

These tools have dependencies on some common Python libraries, listed in the Table below:

Table 1.10: Python Package Dependencies

File	Description
<code>numpy</code>	Numerical operations on arrays
<code>astropy</code>	General astronomical utilities including FITS i/o
<code>matplotlib</code>	2-D python plotting library.
<code>scipy</code>	General scientific and mathematical utilities
<code>sqlite</code>	Database creation and access tools

These libraries are included by default in the [Anaconda distribution of python](#), which is highly recommended.

### Third-Party Software

Various third-party software tools may be useful for the data reduction process, depending upon the scientific goals. While most astronomers will have many or most of these tools, they are listed here for convenience.

Table 1.11: Third-Party Tools

Tool	Description
<code>Aladin</code>	Interactive sky atlas; may be used to refine WCS solution in images. The desktop tool can access many on-line astronomical catalogs as well as local (private) catalogs.
<code>astrometry.net</code>	WCS calibration software. Users may download the software or upload images to a web service.
<code>ds9</code>	SAOImage DS9 image display tool
<code>SQLite Browser</code>	Browser for sqlite3 database files

#### 1.4.4 Acknowledgements

Scientific publications that make use of data obtained from NOAO facilities, and/or that made use of IRAF for data reduction, should include the appropriate acknowledgement described on the [NOAO Publications Acknowledgements](#) page.

Scientific publications that use data retrieved from the [NOAO Science Archive](#), or services provided by the Archive Portal, should a statement of acknowledgement. See [Acknowledging the NOAO Science Archive](#) for details.

If you made use of [Astropy](#) software for your data reduction, please include an appropriate acknowledgement in your publication, as described on the [Acknowledging or Citing Astropy](#) page.

You may also wish to cite the C/KOSMOS instrument description paper by [MES].

Citations to this *Cookbook* should read:

Shaw, R. A. 2015, *C/KOSMOS Data Reduction Cookbook* (Version 1.0; Tucson, AZ: National Optical Astronomy Observatory)

## 1.4.5 Literature References

## 1.5 Glossary

**ADC** An *atmospheric dispersion corrector* generally consists of a pair of prisms that are used in the optical system of a telescope or instrument to compensate for the dispersive effect of Earth’s atmosphere.

**FoV** The *field of view*, or spatial extent of sky the from the optical system of the telescope+instrument that actually falls on the detector.

**KMS** Slit-mask definition files, with the extension `.kms` (or sometimes, `.cms`). These files are created by the slit-mask design software (see [MD]) to specify the manufacture of custom slit masks. The contents include the coordinates in (RA, Dec) and in (x,y; in mm), as well as the slit dimensions for both the targets and the acquisition stars.

**MasterCal** A *Master Calibration* file, which may be an image that captures a particular instrumental signature, or a table consisting of a calibration or reference information. *MasterCals* are often built by combining calibration exposures in a particular way, or by recording coefficients of a function that characterizes, e.g., the dispersion solution. They may also consist of a catalog of reference information, such as astrometric or photometric standards.

**MEF** *Multi-extension FITS* format files contain a primary header, and one or more FITS extensions each of which contains a header and data (HDU). For C/KOSMOS raw data there are two extensions of type `IMAGE`, one for the output of each amplifier on the CCD.

**MOS** *Multi-object spectroscopy* is the capability of obtaining multiple spectra of different targets (or different regions within an extended object) in the same exposure. This may be achieved by rotating a facility longslit or using a custom slitmask with multiple apertures corresponding to the target(s) of interest.

**Observing Block** A contiguous run of full or partial nights of observing time, awarded to a single observing program. An observing program may consist of multiple observing blocks, and may span more than one observing semester.

**Overscan** Refers to the portion of the amplifier read-out of the serial register after all science pixels have been accumulated. These *overscan* “virtual” pixels are appended to the science array, and are used to characterize the bias level and stability of the readout electronics.

**RoI, Region of Interest** Refers to the portion of the CCD that is read out after an exposure. The *RoI* is usually `2kx2k` for imaging mode, and either `FullFrame` or `320x4k` for the spectroscopic mode depending upon the spatial extent of the target. Note that the *RoI* for science exposures must be matched to the calibration exposures (bias, flat, etc.).

**VPH Grism, Volume Phase Holographic Grism** A transmissive dispersing element composed of a volume phase holographic (*VPH*) grating bonded between a pair of prisms. For C/KOSMOS the prisms are designed to yield a zero-deviation optical path in spectroscopic mode.

**WCS, World Coordinate System** A mapping from image pixel coordinates to physical coordinates. For direct images the mapping is to the equatorial (RA, Dec) system; for extracted spectra the mapping is to the dispersion axis, usually in Angstroms, and position along the slit.

## 1.6 About This Guide

This *Cookbook* was created to help users reduce and calibrate spectra obtained with the C/KOSMOS spectrographs. Note the version number in the top banner of your browser. Users may find the description of C/KOSMOS data products the *Getting Started* chapter and in the [NOAO Data Handbook](#) (Chapter 5) very helpful.

---

**Note:** Content for this document is still being developed.

---

### 1.6.1 Typographical Conventions

Technical documentation is often a struggle to read. This document features some typographical conventions to help the reader understand the content, and to help distinguish explanatory text from dialog with the computer. Limitations of reStructured text prohibit elaborate textual markup, however.

#### Technical Terminology

The descriptions in this Cookbook include many technical terms. The first time a technical term, such as *WCS*, is used on a page it is linked with a definition in the *Glossary*.

#### Software Packages

Names of software packages, including scripts that were built for this *Cookbook*, appear in `typewriter` font, and downloadable via a link. Names of third-party packages are often linked to the web site from which they can be downloaded. See the chapter on *Useful Resources* for information about the software packages that are required to make use of the processing scripts.

---

**Note:** Scripts developed for this *Cookbook* are meant to be illustrative and useful. However they do not include extensive error checking, and are not likely to be robust against unexpected input. They may serve as a guide for users to develop their own personal processing pipelines.

---

#### Code Blocks and Literals

When describing a process for using software, the text that should be typed by the user appears in a code block:

```
echo 'Hello, world!'
```

with `fixed-space` font and (often) syntax highlighting that is appropriate for the context. Text describing a literal command-line, names of arguments, directory and file names, etc. are also set in `fixed-space` font.

### 1.6.2 Colophon

This Cookbook was written using [Sphinx](#), which uses reStructuredText as the markup language (see [Sphinx Documentation](#)). If the source or configuration files (found under the `/source` subdirectory) are altered, the document HTML files should be rebuilt with:

```
sphinx-build -b html source build
```

which will update the `.html` files in the `/build` subdirectory. Then make a tar of the `/build` directory contents, copy it to the deployment directory, and unpack.

A PDF document may be generated (e.g., in the subdirectory `tex`) by invoking the LaTeX build option:

```
sphinx-build -b latex source tex
cd tex
pdflatex CKOSMOSCookbook
```

The quality of the rendering is not very good, however, and the LaTeX processing does not complete without errors. Improving it would at least require fixing the relevant LaTeX style files.

## Authors

Version 1.0 of this *Cookbook* was written by Dick Shaw, with major contributions from Sean Points.

## 1.7 Background Material

The following documents are important reading for understanding the C/KOSMOS instruments and the data they produce.

- **C/KOSMOS Instrument Manual:** [\[IM\]](#) This manual describes the attributes of the instrument, how to configure it for observations, and how observing programs are carried out.
- **C/KOSMOS Instrument Description paper:** [\[MES\]](#) This paper presents the design, construction, and deployment of the C/KOSMOS instruments.
- **C/KOSMOS Mask Design Manual:** [\[MD\]](#) This manual describes how to specify the attributes of custom slit masks for *MOS* spectroscopy of sources in a field.
- **NOAO Data Handbook:** [\[DHB\]](#) Chapter 1 describes common elements of NOAO data products and how to access them with the [NSA Portal](#). Chapter 5 summarizes the C/KOSMOS instrument attributes, and describes the raw data products in detail.



## INDEX AND SEARCH

- genindex
- search



## BIBLIOGRAPHY

- [AP] Robitaille, T. P., et al. 2013, *Astropy: A Community Python Package for Astronomy* *A&A*, 558, 33
- [DHB] Shaw, R. A. ed. 2015, *NOAO Data Handbook* (Version 2.2; Tucson, AZ: NOAO)
- [FITS] Pence, W.D., Chiappetti, L., Page, C.G., Shaw, R.A., & Stobie, E. 2010, *Definition of the Flexible Image Transport System (FITS), Version 3.0*, *A&A*, 524, 42
- [IM] Points, S, Elias, J., Martini, P., & Beers, T. 2014, *C/KOSMOS Instrument Manual* (Version 1.7; Tucson, AZ: NOAO)
- [MD] Points, S, Martini, P., Elias, J., & Briceno, C. 2014, *Mask Design Software for C/KOSMOS* (Tucson, AZ: NOAO)
- [MES] Martini, P., et al. 2014, *KOSMOS and COSMOS: New Facility Instruments for the 4-meter Telescopes*, *SPIE*, 9147, oZ
- [MVB] Massey, P., Valdes, F., & Barnes, J. 1992, *User's Guide to Reducing Slit Spectra with IRAF*, (Tucson: NOAO/IRAF)