SOI Notes

S. Points, 12 Oct 2011.

In the following document GNU/Linux and IRAF commands are preceded by "linux-prompt>" and "IRAFPackage>", respectively.
You may want to download soar.cl, soinoao.tar, and soimsu.tar.gz first.

1) Start IRAF on your local machine

linux-prompt> cl - ecl
ecl> soar
soar> soinoao
soinoao> mscred
mscred>

Get filter information from the FITS header. Please remember that SOI can hold up to eight (8) filters in two (2) filter wheels at any given time. This can be done with IRAF using the TABLES keypar package, but can be cumbersome. An example of an IRAF script that can do this can be found at get_soi_filter_keyword.cl.
This task is more easily done by using the WCSTools *gethead* program. If the WCSTools are installed on your local machine, you can determine the SOI filter information by entering the following on a command line

linux-prompt> *gethead* FILPOS FILTERS FILTER1 FILTER2 *fits

This returns the following information:

dflat.001.fits 1 5 s0000_s0002 s0000 Open s0002 B Bessell
...
dflat.011.fits 1 4 s0000_s0003 s0000 Open s0003 V Bessell
...
dflat.021.fits 1 3 s0000_s0004 s0000 Open s0004 R Bessell
...
dflat.031.fits 1 2 s0000_s0005 s0000 Open s0005 I Bessell
...
sflat.001.fits 2 1 s0001_s0000 s0001 U Bessell s0000 Open

The filter position is used to record the filter name in the images headers and is required to handle image subsets. After the filter information is determined.

From the above example with "*gethead*", we can see that:

U_Bessell is in filpos 2 1
B_Bessell is in filpos 1 5
V_Bessell is in filpos 1 4
R_Bessell is in filpos 1 3
I_Bessell is in filpos 1 2

Note: Position 1 in each filter wheel is normally designated as "Open".

2) Use IRAF to create the necessary subsets

```
mscred> hsel *.fits[1] $I 'filpos == "2 1"' > listU
mscred> hsel *.fits[1] $I 'filpos == "1 5"' > listB
mscred> hsel *.fits[1] $I 'filpos == "1 4"' > listV
mscred> hsel *.fits[1] $I 'filpos == "1 3"' > listR
mscred> hsel *.fits[1] $I 'filpos == "1 2"' > listI
```

Trim the last eight (8) characters of the filenames, i.e., ".fits[1]". This can be done easily using the following GNU/Linux commands:

```
linux-prompt> cp listU listU2
linux-prompt> vi listU2
```

After you can edit the list file, perform a global search and replace of the .fits[1] characters using the following:

```
:1,$ s/.fits\[1\]//g
```

Repeat this for other filters.

If you are not familiar with vi, you can trim the last 8 characters of the filename using the following IRAF code on the command line:

```
mscred> int size_str
    list="listU"
    while (fscan (list,s1) !=EOF) {
    size_str=strlen(s1)
    print substr(s1,1,size_str-8) >> listU2
    }
```

Repeat this for other filters.

Update the "FILTER" parameter in the image headers within IRAF

```
mscred> soiupfilter @listU2 filter_=yes filter=U
mscred> soiupfilter @listB2 filter_=yes filter=B
```

```
mscred> soiupfilter @listV2 filter_=yes filter=V
mscred> soiupfilter @listR2 filter_=yes filter=R
mscred> soiupfilter @listI2 filter_=yes filter=I
```

Check the IMTYPE keyword and the Filter subsets

```
mscred> ccdlist *fits[1]
```

This should give something like the following:

```
dflat.001.fits[1][563,2048][ushort][FLAT][1][B]:Dome flat
...
dflat.011.fits[1][563,2048][ushort][FLAT][1][V]:Dome flat
...
dflat.021.fits[1][563,2048][ushort][FLAT][1][R]:Dome flat
...
dflat.031.fits[1][563,2048][ushort][FLAT][1][I]:Dome flat
...
sflat.001.fits[1][563,2048][ushort][SKYFLAT][1][U]:sflat
...
obj.006.fits[1][563,2048][ushort][OBJECT][1][U]:T Phe
...
obj.007.fits[1][563,2048][ushort][OBJECT][1][B]:T Phe
...
obj.008.fits[1][563,2048][ushort][OBJECT][1][V]:T Phe
...
obj.009.fits[1][563,2048][ushort][OBJECT][1][I]:T Phe
```

3) Check the BIASSEC, DATASEC, and TRIMSEC parameters of the FITS files.
We have received reports that these may be off by a pixel. Thus, it is always a good idea to inspect the data by hand before moving on. The latest and greatest values for these parameters that we have are...

For Pre 14 Nov 2008 Data (SOI Leach II Controller, 2 2kx4k CCDs):

AMP	BIASSEC	TRIMSEC	DATASEC
1	[541:562,1:2048]	[29:540,1:2048]	[29:540,1:2048]
2	[1:23,1:2048]	[24:535,1:2048]	[24:535,1:2048]
3	[541:562,1:2048]	[29:540,1:2048]	[29:540,1:2048]
4	[1:23,1:2048]	[24:535,1:2048]	[24:535,1:2048]

For 14 Nov 2008 -- DD MMM 2010 Data (SOI Leach III Controller, 2 2kx4k CCDs):

AMP	BIASSEC	TRIMSEC	DATASEC

1	[541:568,1:2048]	[29:540,1:2048]	[29:540,1:2048]
2	[1:28,1:2048]	[29:540,1:2048]	[29:540,1:2048]
3	[541:568,1:2048]	[29:540,1:2048]	[29:540,1:2048]
4	[1:28,1:2048]	[29:540,1:2048]	[29:540,1:2048]

For Post DD MMM 2010 Data (SOI Leach III Controller, 1 4kx4k CCD):

AMP	BIASSEC	TRIMSEC	DATASEC
1	TBD	TBD	TBD
2	TBD	TBD	TBD
3	TBD	TBD	TBD
4	TBD	TBD	TBD

If the BIASSEC, TRIMSEC, and DATASEC keywords are not correct, you will need to change them. It is easy to do this within IRAF using the mscred.ccdhedit task. Be sure to use the appropriate values depending upon the date of your observations.

4) Combine the bias frames

```
mscred> epar zerocombine
                        I R A F
              Image Reduction and Analysis Facility
PACKAGE = mscred
  TASK = zerocombine

input    =                zero* List of zero level images to combine
(output =                Zero) Output zero level name
(combine=            median) Type of combine operation
(reject =          avsigclip) Type of rejection
(ccdtype=              ZERO) CCD image type to combine
(process=               yes) Process images before combining?
(delete =                no) Delete input images after combining?
(scale  =              none) Image scaling
(statsec=                  ) Image section for computing statistics
(nlow   =                 0) minmax: Number of low pixels to reject
(nhigh  =                 1) minmax: Number of high pixels to reject
(nkeep  =                 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip  =               yes) Use median in sigma clipping algorithms?
(lsigma =                3.) Lower sigma clipping factor
(hsigma =                3.) Upper sigma clipping factor
(rdnoise=               4.4) ccdclip: CCD readout noise (electrons)
(gain   =                 2) ccdclip: CCD gain (electrons/DN)
```

```
(snoise =                4.4) ccdclip: Sensitivity noise (fraction)
(pclip  =               -0.5) pclip: Percentile clipping parameter
(blank  =                 0.) Value if there are no pixels
(mode   =                  q)

mscred> epar ccdproc
                          I R A F
               Image Reduction and Analysis Facility
PACKAGE = mscred
  TASK = ccdproc

images  =          zero*fits  List of Mosaic CCD images to process
(output =                   ) List of output processed images
(bpmasks=                   ) List of output bad pixel masks
(ccdtype=              ZERO) CCD image type to process
(noproc =                 no) List processing steps only?
(xtalkco=                 no) Apply crosstalk correction?
(fixpix =                 no) Apply bad pixel mask correction?
(oversca=                yes) Apply overscan strip correction?
(trim   =                yes) Trim the image?
(zerocor=                 no) Apply zero level correction?
(darkcor=                 no) Apply dark count correction?
(flatcor=                 no) Apply flat field correction?
(sflatco=                 no) Apply sky flat field correction?
(split  =                 no) Use split images during processing?
(merge  =                 no) Merge amplifiers from same CCD?
(xtalkfi=                   ) Crosstalk file
(fixfile=                   ) List of bad pixel masks
(saturat=              45000) Saturated pixel threshold
(sgrow  =                  0) Saturated pixel grow radius
(bleed  =             INDEF) Bleed pixel threshold
(btrail =                 10) Bleed trail minimum length
(bgrow  =                  0) Bleed pixel grow radius
(biassec=              image) Overscan strip image section
(trimsec=              image) Trim data section
(zero   =          Zero.fits) List of zero level calibration images
(dark   =                   ) List of dark count calibration images
(flat   =                   ) List of flat field images
(sflat  =                   ) List of secondary flat field images
(minrepl=                 1.) Minimum flat field value
(interac=                 no) Fit overscan interactively?
(functio=          legendre) Fitting function
```

(order =   2) Number of polynomial terms or spline pieces
(sample =   *) Sample points to fit
(naverag=   1) Number of sample points to combine
(niterat=   4) Number of rejection iterations
(low_rej=   3.) Low sigma rejection factor
(high_re=   3.) High sigma rejection factor
(grow   =   0.) Rejection growing radius
(fd   =   )
(fd2   =   )
(mode   =   q)

After you have checked all of the parameters, you can run zerocombine. It is generally useful to fit the first few overscan regions interactively until you are satisfied with the parameters.

5) Create your master flat fields in each filter

mscred> epar flatcomb
                I R A F
        Image Reduction and Analysis Facility
PACKAGE = mscred
  TASK = flatcombine

input   =   dflat*fits List of flat field images to combine
(output =   Dflat_) Output flat field root name
(combine=   median) Type of combine operation
(reject =   avsigclip) Type of rejection
(ccdtype=   ) CCD image type to combine
(process=   yes) Process images before combining?
(subsets=   yes) Combine images by subset parameter?
(delete =   no) Delete input images after combining?
(scale =   mode) Image scaling
(statsec=   ) Image section for computing statistics
(nlow   =   1) minmax: Number of low pixels to reject
(nhigh =   1) minmax: Number of high pixels to reject
(nhigh =   1) minmax: Number of high pixels to reject
(nkeep =   1) Minimum to keep (pos) or maximum to reject (neg)
(mclip =   yes) Use median in sigma clipping algorithms?
(lsigma =   3.) Lower sigma clipping factor
(hsigma =   3.) Upper sigma clipping factor
(rdnoise=   4.4) ccdclip: CCD readout noise (electrons)
(gain   =   2) ccdclip: CCD gain (electrons/DN)

```
(snoise =                0.) ccdclip: Sensitivity noise (fraction)
(pclip  =             -0.5) pclip: Percentile clipping parameter
(blank  =                1.) Value if there are no pixels
(mode   =                 q)

mscred> epar ccdproc
                          I R A F
                Image Reduction and Analysis Facility
PACKAGE = mscred
  TASK = ccdproc

images  =         dflat*fits  List of Mosaic CCD images to process
(output =                  ) List of output processed images
(bpmasks=                  ) List of output bad pixel masks
(ccdtype=              FLAT) CCD image type to process
(noproc =                no) List processing steps only?
(xtalkco=                no) Apply crosstalk correction?
(fixpix =                no) Apply bad pixel mask correction?
(oversca=               yes) Apply overscan strip correction?
(trim   =               yes) Trim the image?
(zerocor=               yes) Apply zero level correction?
(darkcor=                no) Apply dark count correction?
(flatcor=                no) Apply flat field correction?
(sflatco=                no) Apply sky flat field correction?
(split  =                no) Use split images during processing?
(merge  =                no) Merge amplifiers from same CCD?
(xtalkfi=                 ) Crosstalk file
(fixfile=                 ) List of bad pixel masks
(saturat=             45000) Saturated pixel threshold
(sgrow  =                 0) Saturated pixel grow radius
(bleed  =            INDEF) Bleed pixel threshold
(btrail =                10) Bleed trail minimum length
(bgrow  =                 0) Bleed pixel grow radius
(biassec=             image) Overscan strip image section
(trimsec=             image) Trim data section
(zero   =         Zero.fits) List of zero level calibration images
(dark   =                  ) List of dark count calibration images
(flat   =                  ) List of flat field images
(sflat  =                  ) List of secondary flat field images
(minrepl=                1.) Minimum flat field value
(interac=                no) Fit overscan interactively?
(functio=         legendre) Fitting function
```

(order = 2) Number of polynomial terms or spline pieces
(sample = *) Sample points to fit
(naverag= 1) Number of sample points to combine
(niterat= 4) Number of rejection iterations
(low_rej= 3.) Low sigma rejection factor
(high_re= 3.) High sigma rejection factor
(grow = 0.) Rejection growing radius
(fd =)
(fd2 =)
(mode = q)

6) Process your science frames for overscan, trim-correction, zero-subtraction, and flat field corrections.

In general you will want to make a list of your data for each filter. Then you will want to apply a WCS to your data and create a mosaicked image. See the Mosaic Data reduction guide, by Frank Valdes.