

# STIS Data Analysis

## In This Chapter...

Data Reduction Applications / 23-1  
STIS-Specific Reduction and Analysis Tasks / 23-2  
Working with Two Dimensional Extracted Spectra / 23-3  
TIME-TAG Data / 23-5  
Target Acquisition Data / 23-7

In this chapter we discuss the data reduction applications available to work with STIS data, and describe specific analyses you may want to apply to your spectral, timetag, and acquisition data.

---

## 23.1 Data Reduction Applications

Most of the *new* software tools for operating on STIS FITS files are contained in two STSDAS packages:

- **toolbox.imgtools.mstools**: Contains image manipulation tasks created especially for STIS and NICMOS image set (*imset*) data and which exploit the error and data quality arrays in their operations (e.g., **msarith**, **mscombine**, **msstatistics**, **mssplit**, and **msjoin**). These tasks are described in “Working with STIS and NICMOS Imsets” on page 3-12.
- **hst\_calib.stis**: Contains the STIS specific tasks **basic2d**, **calstis**, **ocrreject**, **wavecal**, **x1d**, and **x2d**.

In addition to the above packages, most basic image manipulation software (e.g., **display**, **daophot**, **imexamine**, **contour**, etc.) and spectral analysis software (e.g., **splot**, **tables**, **specfit**, **igi**, etc.) available in IRAF/STSDAS can be used on STIS data, either directly through the IRAF FITS interface or by converting data to another IRAF format. Chapter 3 includes information about how to display STIS images and extracted spectra as well as how and when to convert data formats and a description of spectral analysis tasks. Table 23.1 lists some of the more useful IRAF/STSDAS applications for working with STIS data.

**Table 23.1:** Spectral Analysis Tasks

Task	Input Formats	Purpose
<b>echplot</b>	3-D tables	Plots multiple STIS echelle spectral orders.
<b>nfit1d</b>	2-D & 3-D tables, images	General 1-D feature fitting; part of the STSDAS <b>fitting</b> package.
<b>igi</b>	2-D & 3-D tables, images	General presentation graphics; supports world coordinates.
<b>sgraph</b>	2-D & 3-D tables, images	General 1-D plotting; supports world coordinates.
<b>specfit</b>	1-D images, ASCII tables	General 1-D spectrum modeling package.
<b>splot</b>	<i>multispec</i> images	General 1-D spectral analysis.

Some of the tasks are intended for browsing data or producing publication-quality plots: the **igi** and **sgraph** tasks were described in Chapter 3. The **echplot** task is useful for browsing STIS extracted spectra. You can plot single spectral orders, overplot multiple orders on a single plot, or plot up to 4 orders in separate panels on the same page. For example, we overplot the orders contained in rows 2 through 4 and 6 on one page:

```
c1> echplot "stis_x1d.fits[1][r:row=(2:4,6)]" output.igi \
>>> plot_style=m
```

Note that the `plot_style` parameter governs whether the spectral orders are plotted one per page, overplotted, or plotted one per panel, for parameter values of “s”, “m”, or “p” respectively. The brightness unit is calibrated FLUX by default, though other quantities (e.g., NET counts) can be specified using the `flux_col` parameter.

## 23.2 STIS-Specific Reduction and Analysis Tasks

In Chapter 21 we discussed the components of the STIS pipeline that can be run as individual tasks. Observers may find that they wish to perform parts of the pipeline reduction again on their data using these tasks. Typical examples will be to re-perform cosmic ray rejection, altering the input parameters or using data from separate datasets, or to perform one-dimensional spectral extraction on long-slit data or to modify the input parameters (e.g., aperture extraction box, or background region) when doing one-dimensional spectral extraction for echelle data. For completeness we list the tasks again below. To run these tasks you need to retrieve the calibration reference files they require from the Archive and set the `oref` parameter appropriately (see “Mechanics of Full Recalibration” on page 21-31). An example of using **x1d** with a user specified extraction box (11 pixels high) and a user specified center (`line=500`) is given below:

```
c1> x1d o3s41301o_flt.fits 3s41301o_x1d.fits center=500 \
>>> size=11
```

Table 23.2 lists some of the stand-alone tasks for working with STIS data. Consult the on-line help for more information about all these tasks. We expect additional options to become available within the stand-alone (i.e., outside the pipeline) versions of these tasks as we increase the number of parameters which can be changed through **epar** and as we add new functions, such as background subtraction correction for echelle modes and optimal extraction for echelle and first order spectra to **x1d**. Stay tuned to the STANs and WWW.

**Table 23.2:** STIS-Specific Tasks

Task	Description
<b>basic2d</b>	Perform basic 2-D calibration.
<b>inttag</b>	Integrate TIMETAG event list to form an image.
<b>ocrreject</b>	Combine images, rejecting cosmic rays.
<b>wavecal</b>	Process wavecal images, determine spectral and spatial shifts.
<b>x1d</b>	Extract 1-D spectrum.
<b>x2d</b>	Rectify first-order spectral image.

## 23.3 Working with Two Dimensional Extracted Spectra

### Sensitivity Units and Conversions

Your two dimensional extracted spectrum (`_sx2`, or `_x2d` file) has units of  $\text{erg cm}^{-2} \text{sec}^{-1} \text{\AA}^{-1} \text{arcsec}^{-2}$ . The conversion from counts to surface brightness ( $I_\lambda$ ) is calculated by the pipeline as:

$$I_\lambda = \text{counts} * \text{gain} * h * c / (\text{exptime} * t_\lambda * A_{\text{HST}} * d * m_s * W)$$

where

- gain is the conversion from counts to electrons for the CCD, and is unity for MAMA observations.
- $h$  is Planck's constant.
- $c$  is the speed of light.
- exptime is the exposure time.
- $t_\lambda$  is the wavelength-dependent integrated system throughput divided by the physical area of the HST mirror, which is given in the PHOTTAB reference file table.
- $A_{\text{HST}}$  is the physical area of the HST mirror.

- $d$  is the dispersion in Å/pixel, derived from the DISPTAB reference table.
- $m_s$  is the plate scale in arcseconds/pixel in the cross-dispersion direction, which is the product of the CD2\_2 header keyword value and (3600 arcsec/degree).
- $W$  is the slit width in arcseconds.

The flux from a fully extended continuum source as transmitted by the science slit, over an arbitrary number of pixels in the spatial direction ( $N_{pix_s}$ ) and taken over an arbitrary number of pixels in the wavelength direction ( $N_{pix_\lambda}$ , where to keep from degrading the spectral purity  $N_{pix_\lambda} * m_\lambda$  must be less than  $W$ ) is given by:

$$\mu(I_\lambda) \times W \times N_{pix_s} \times m_s$$

where:

- $\mu(I_\lambda)$  is the average value of the surface brightness,  $I_\lambda$ , over the  $N_{pix_s} * N_{pix_\lambda}$  pixels in the rectangle being integrated.

For an emission line source, the situation is somewhat different. There, the line surface brightness,  $I_{line}$ , from an emission line feature, in  $\text{ergs sec}^{-1} \text{cm}^{-2} \text{arcsec}^{-2}$  is given as:

$$I_{line} = \frac{\text{counts} \times h \times c}{(\text{exptime} \times t_\lambda \times A_{HST} \times m_s \times m_\lambda)} x = I_\lambda \times (W \times d / m_\lambda)$$

where:

- $I_\lambda$  is as given above (i.e., the values in the `_x2d.fits` or `_sx2.fits` image).
- $m_\lambda$  is the plate scale in the dispersion direction as given in the CD1\_1 header keyword.
- All other parameters are as above.

Having performed this conversion, one can take an average or sum of the value of  $I_{line}$  over the extent of a feature to get the mean,  $\mu(I_{line})$ , or to get the total flux from the line  $F_{line} = \mu(I_{line}) * N_{pix_\lambda} * m_\lambda * N_{pix_s} * m_s$  over a  $N_{pix_\lambda} * m_\lambda$  by  $N_{pix_s} * m_s$  region on the sky, where  $N_{pix_\lambda} * m_\lambda$  equals the width of the emission feature in the spectral direction.

The factor  $W*d/m_\lambda$  which converts between diffuse continuum source surface brightness and diffuse emission line surface brightness is given in the CONT2EML header keyword and is simply the slit width expressed in Angstroms. The surface area of a pixel in arcseconds ( $m_s * m_\lambda$ ) is given in the header keyword OMEGAPIX.

Finally, the DIFF2PT keyword in the data header gives the conversion factor to flux units in  $\text{erg cm}^{-2} \text{sec}^{-1} \text{\AA}^{-1}$  for a point source. The DIFF2PT keyword is calculated as:

$$\text{DIFF2PT} = W * m_s / \overline{A}_\lambda,$$

where  $W$  and  $m_s$  are as above and  $\overline{A}_\lambda$  is the wavelength averaged point source aperture throughput for the science aperture, determined from the APTHRUTAB reference table. That is if you integrate the `_x2d` or `_sx2` file over the full wings for a point source (which can be quite extensive) and multiply by the DIFF2PT parameter you will recover the flux of the point source. Of course, point source observers are better advised to use **x1d** to extract a one-dimensional aperture extracted spectrum from their long-slit first-order data, which will then use the wavelength dependent aperture throughput and the defined point source extraction aperture and calibration, as well as perform background subtraction.

In general we note that the cross dispersion profiles can be quite extended (particularly in the far-UV and in the near-infrared); fluxes derived for extended sources from the `_x2d.fits` files as above assume that the sources are extended on scales which contain the bulk of the cross dispersion flux from a point source. As we make further progress analyzing the cross dispersion profiles and their effect on the accuracies of point source and diffuse source fluxes we will provide updates on the WWW and through the STANs.

See also Chapter 6 of the *STIS Instrument Handbook* for a more detailed discussion of units and conversions for different source types.

### 23.3.1 Wavelength and Spatial Information

Two-dimensionally extracted spectra have been wavelength calibrated and rectified to a linear wavelength scale. Tasks such as **splot** can work directly on the `_sx2.fits` and `_x2d.fits` files and can read the wavelength header information which is stored in the standard FITS CD matrix keywords. Alternatively you can use these keywords directly to determine the wavelength or distance along the slit at any pixel as:

- $\lambda(x) = \text{CRVAL1} + (x - \text{CRPIX1}) * \text{CD1\_1}$
- $s(y) = (y - \text{CRPIX2}) * \text{CD2\_2}$ .

where  $\lambda(x)$  is the wavelength at any given  $x$  pixel, and  $s(y)$  is the distance along the slit from slit center for any given  $y$  pixel, in units of degrees.

---

## 23.4 TIME-TAG Data

As described in “Tabular Storage of STIS Data” on page 20-5, raw time-tagged data are stored in an event table which contains separate columns for the  $x$  and  $y$  pixel coordinates and the arrival time of each detected photon event. There will also be an associated table containing “good time intervals.” In general these data can be analyzed in one of two ways:

- Convert the data to QPOE format and use the tasks in the **xray** and **xtiming** IRAF packages to work with the time series data directly.

- Generate a series of ACCUMulated images from user specified time intervals of data and then process the image(s) with **calstis**.

### Using Existing PROS Time Series Analysis Software

The PROS **xray** and the **xtiming** packages in IRAF—originally developed to analyze Roentgen Satellite (ROSAT) data—can be used to do time series analysis of the STIS TIME-TAG data. To use this package, the `*_tag.fits` files need to be converted to PROS QPOE format (extension `*.qpp`), using the task **fits2qp**. A few particularly useful tasks within the PROS software package are listed in Table 23.3.

**Table 23.3:** Useful PROS Tasks

Task	Purpose
<b>period</b>	Find period from a dataset
<b>fldplot</b>	Plot the light curve with the period folded into it
<b>chiplot</b>	Chi-square plot for various periods
<b>lcurv</b>	Plot simple light curve from a dataset

Note that the times given in the first column of the raw time-tag data (which is the time relative to the start of the exposure) are not corrected for the time delay due to the motion of the earth or the spacecraft. The change in time delay due to the motion of the spacecraft can be a maximum of about 30 millisc in one orbit. The absolute time delay relative to the time-frame at the earth barycenter can itself be much larger. To correct for these effects, the ephemeris of the earth and the spacecraft are necessary. A separate task is being developed at present, similar to the tasks for the High Speed Photometer (HSP), to correct for these effects and to convert the times to the barycenter of the earth. This task can be used by observers, or can be processed at STScI as a special request, for those science uses of TIME-TAG where it becomes an important effect.

### Using **inttag** to Produce ACCUM Images of Specified Time Slices

The **inttag** task can be used to integrate STIS TIMETAG data into an image or set of images. The default behavior for **inttag** is to accumulate all events from the table, writing the results as one image set in the output FITS file. The user has the option, however, to specify explicitly a starting time, time interval, and number of intervals over which to integrate, in which case a collection of image sets will be written to the output file, simulating a REPEATOBS ACCUM observation. Breaking the data into multiple, short exposures can be useful not only for variable targets but also to improve the flatfielding when the Doppler shift is significant.

To generate each time-filtered image, **inttag** compares the arrival time for each event to both the user-specified interval and the set of *good time intervals*. (Good time intervals, or *GTIs*, are intervals during which STIS is known to be taking valid data.) If the event qualifies, the raw *x* and *y* coordinates will be mapped to the appropriate output pixel location (to account for binning, subarray location,

and Doppler correction for echelle spectroscopy), and the detected photon count of the output pixel will be incremented.

The **inttag** task requires two arguments: the names of the input and output FITS files, in that order. Optional parameters include writing the output image(s) in high-res format (the default is low-res). To specify times interval(s) different from the entire duration of the exposure, specify the starting time in seconds since the beginning of the exposure (**sttime**, the default is INDEF, meaning to start at the beginning of the first GTI), the integration time in seconds (**increment**, the default is INDEF, meaning to end at the last GTI), and the number of intervals (**rcount**, the default is one). If no other value is specified, one ACCUM image set will be created containing all the events. A simple example would be to create a single image set, including only the first 1000 seconds:

```
cl> inttag stis_tag.fits kilo_raw.fits sttime=0 increm=1000
```

A somewhat more complex example is to create 20 high-res imsets, each 100 seconds in duration, starting at the first time in the GTI table. Note that while 20 imsets were specified, fewer may actually be written if any of the 100-second intervals is not contained within any GTI, or if the last time in the GTI table is less than 2000 (i.e. 100 sec \* 20 intervals); **inttag** will report how many imsets were created.

```
cl> inttag stis_tag.fits multi_raw.fits sttime=0 \  
>>> increm=100 rcount=20 high+
```

Once the images have been created, it is straightforward to process them with **calstis** and analyze the output images or spectra, as appropriate (see “Recalibration of STIS Data” on page 21-30).

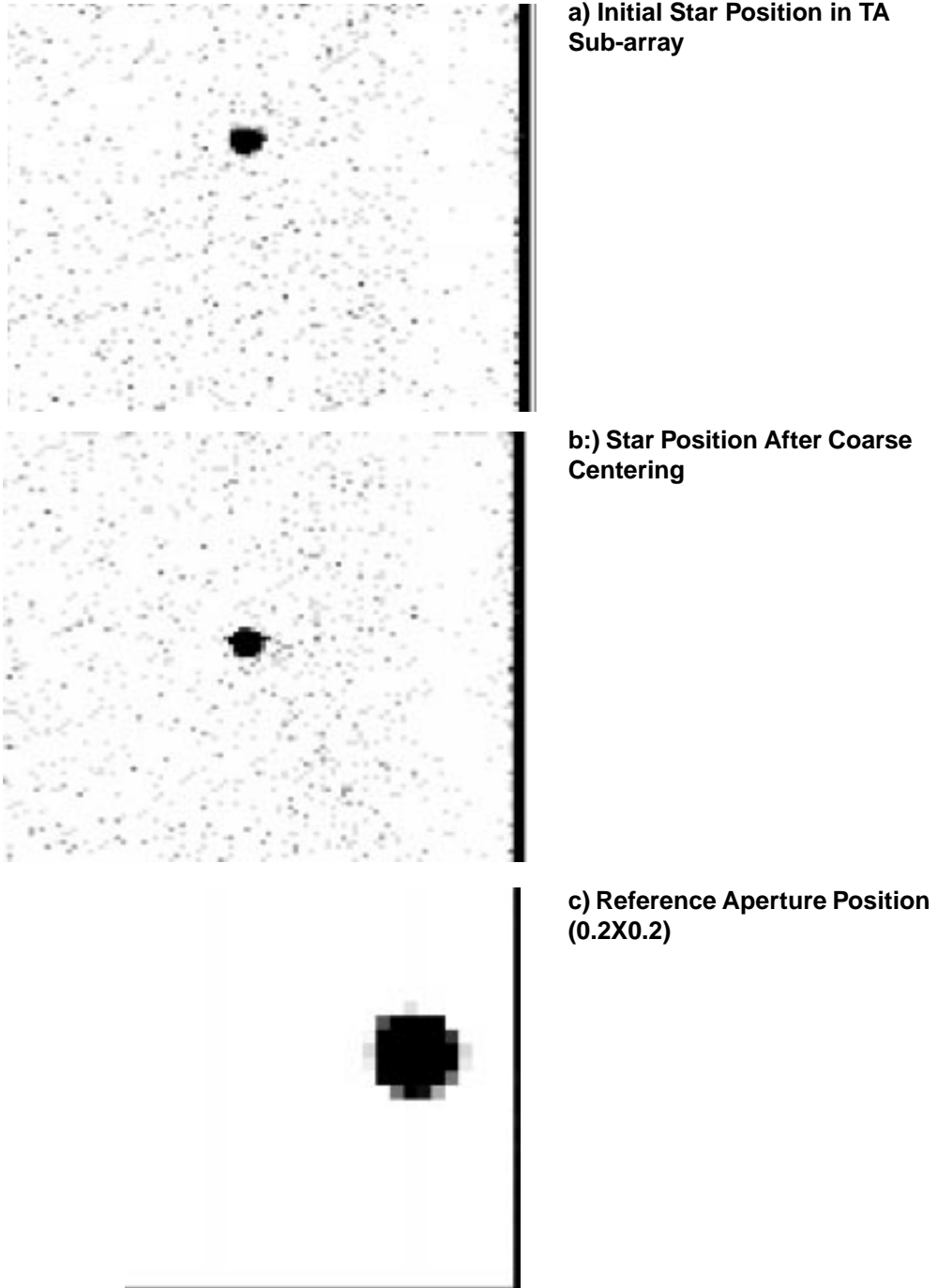
## 23.5 Target Acquisition Data

### Target Acquisition Basics

There are two types of STIS target acquisition: ACQ and ACQ/PEAK; for more details on target acquisition, see *STIS ISR 97-03*. For ACQ observations, there are three parts to the target acquisition data that you receive. The first is an image of the target in the target acquisition sub-array (100 x 100 pixel for POINT sources, user-defined for DIFFUSE sources) based on the initial pointing (see Figure 23.1a); `image_raw.fits[sci,1]`). The software then determines the position of the target with a flux-weighted pointing algorithm and calculates the slew needed to place the target at a reference point in the target acquisition sub-array; for DIFFUSE sources, an option to perform a geometric centroiding is available. An image of the target at this corrected position is then obtained (see Figure 23.1b) `image_raw.fits[sci,2]`—this is the coarse centering. To perform the fine centering (i.e., to place the object precisely in a slit), an image (32 x 32) of the reference 0.2x0.2 aperture is obtained (see Figure 23.1c) `image_raw.fits[sci,3]`), and the location of the aperture determined. A fine slew is then performed to center the target in the reference aperture, which should be accurate to 0.5 pixels (0.025 arcseconds). A final slew to center the

target in the science aperture is performed at the start of the follow-on science observation.

**Figure 23.1:** Three Stages of an ACQ Observation

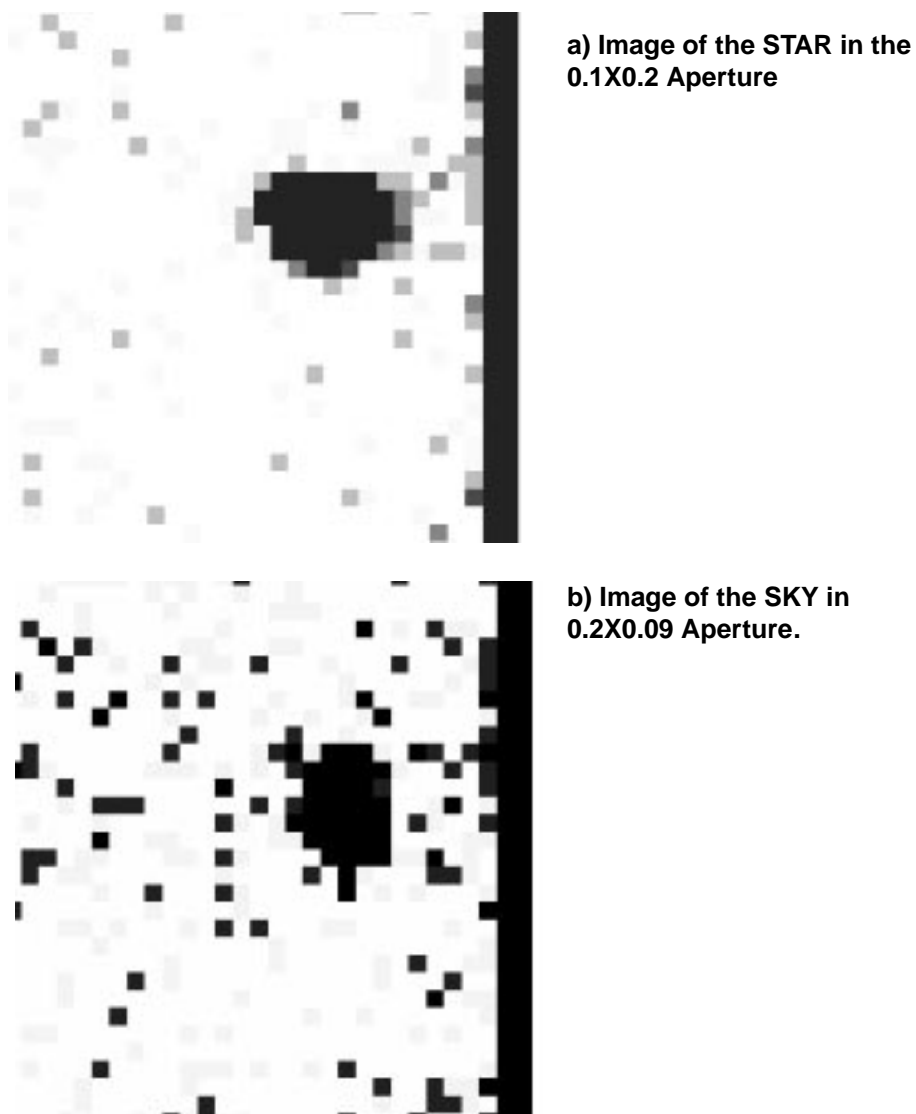


If a narrow slit is used for the science, an ACQ/PEAK acquisition may have been performed. The slit is scanned across the object with a pattern determined by

the aperture selected. The telescope is then slewed to center the star in the aperture, and a confirmation image (a 32 x 32 grid) is obtained; the accuracy of the ACQ/PEAK is 5% of the slit width. Note that the last extension in the file (`image_raw.fits[4]`) contains the values in the individual steps of the ACQ/PEAK (use **listpix** to view these values).

When examining the confirmation image, note that the slit will be illuminated by the sky even if no star is present (see Figure 23.2; `image_raw.fits[sci,1]`). To confirm the presence of a star, use the **imexamine** task and make certain that the FWHM is small. For the images in Figure 23.2, the measured values are given in Table 23.4.

**Figure 23.2:** ACQ/PEAK Confirmation Images



**Table 23.4:** Measured Brightness for STIS ACQ Image

Type	Enclosed	Gaussian	Direct
Star	1.59	1.57	1.49
Sky	15.22	142.84	4.81

### ACQ Data

An examination of the target acquisition data (either from the raw data or the paper products) will allow you to detect gross errors in the centering of your target. A comparison of the initial ([sci,1]) and post-coarse slew ([sci,2]) images should show the object moving close to the center of the acquisition sub-array. You can also verify that the fluxes in both images, which are found in pixel 722 of the `_spt` file, are consistent by performing the following steps in IRAF:

```
cl> listpix image_spt.fits[1] | grep 722
cl> listpix image_spt.fits[2] | grep 722
```

The first value will be the target flux in the maximum checkbox (3 x 3 for POINT sources, user-defined for DIFFUSE sources) in the initial image, while the second is the maximum checkbox in the post-coarse slew image. If the fluxes are not consistent, or if the object did not move closer to the center of the array, there is likely a problem with your acquisition.

### ACQ/Peak Data

To verify that the ACQ/PEAK worked, examine the flux values at each stage of the peakup (listed in the paper products or in the data file). Comparison of the maximum flux value during the peakup with the flux in the post-ACQ/PEAK confirmation image should show that the flux in the confirmation image was greater than or equal to the maximum flux in the peakup grid. If this is not the case, then there is likely a problem with your peakup acquisition.

Note that you will need to perform two corrections to the data prior to your comparison. When calculating the flux in the confirmation image, you will need to subtract 32768 if the minimum flux value is greater than or equal to 32768. The flux values in the peakup scan also need to be adjusted to add back the minimum flux value that was subtracted prior to processing of the peakup data. This value can be found in pixel 712 of the `_spt` image; to display the value on the screen, do the following in IRAF:

```
cl> listpix image_spt.fits[1] | grep 712
```