This task adds columns to the light curve table. We are mainly interested in 'corr_rate', which is the corrected count rate, and corr_err, which is the corrected errors. To plot the light curve, I use *xray/xtiming/ltcplot*:

```
PACKAGE = xtiming
   TASK = ltcplot

table   =               ltcrv  Input Table File
(column =           corr_rate) Plot data column name
(plot_ty=               histo) Plot type - histo|bar
(ecolumn=            corr_err) Error column name
(plot_ti=       Light Curve :) Plot Title
(x_title=      Time (seconds)) Plot Title
(y_title= Count Rate (cts/sec)) Plot Title
(hdcopy =                 yes) Output to Printer?
(gclose =                  no) Close Graph window after plot?
outplt  =                toto  Plot output file
(num_plo=                   1) number of plots
(x_units=             seconds) X axis units - seconds | col name
(x_tics =                  4.) num x labels
(y_tics =                  4.) num y labels
(label_s=                0.75) labels format
(tlabel_=                 0.5) tic labels format
(x_range=                auto) x_range - auto|input limits
(y_range=                auto) y_range - auto|input limits
(clobber=                  no) Overwrite existing plot?
(cursor =                    ) Graphics cursor input
(mode   =                  ql)
```

V. GET THE EFFECTIVE EXPOSURE TIME

Run *euv/euvtools/effexp*:

```
PACKAGE = euvtools
  TASK = effexp

interval=             ds.qp  QPOE file or ST table specifying valid intervals
(stime  =             stime) Column in input table for start times
(etime  =             etime) Column in input table for end times
(ltcurv =         ltcrv_ltc) Light curve table
(t0     =            INDEF) Start time of integration
(t1     =            INDEF) End time of integration
(mask   =                 ) Pixel list with region mask
(primbsc=      primbsch.tab) Primbsch table
(ptimeco=         TableTime) Column in primbsch table for times
(usedead=               yes) Include deadtime correction?
(useprim=               yes) Include primbsch correction?
(quadran=             INDEF) Quadrant to use for all times
(detpos =          detposds) Detector position table
(dtimeco=         TableTime) Column in detpos table for times
(dxcol  =               UDX) Column in detpos table for x position
(dycol  =               UDY) Column in detpos table for y position
(usevign=               yes) Include vignetting correction?
(undisto=                no) Undistort before vignetting correction?
(refdata= reference/detector) Name of ST table for reference data
(detecto=                ds) Detector name [sca|scb|scc|ds]
(filtert=               lex) Type of filter [lex|alc|dag|tin]
(sublen =               0.1) Subinterval length
(exposur=             INDEF) Effective exposure
(rawexpo=             INDEF) Raw exposure
(verbose=                no) Print exposure on STDOUT?
(mode   =                al)
```

IV. CREATE A LIGHT CURVE

Run *xray/xtiming/ltcurve* with the two files you just made for input:

```
PACKAGE = xtiming
   TASK = ltcurv

source_f=          source  Input Source Timing File
backgrou=      background  Input Background Timing File
ltcurv_f=        algol_ds  root name for output file [root_ltc.tab]
bins    =               0  Number of Bins (0 for bin length)
bin_leng=           5000.  Bin Length in Secs
(display=               1) Display Level
(clobber=              no) clobber
(get_gin=             yes) get good intvs from qpoe hdr
(bk_norm=              1.) Bkgd normalization factor
(mode   =              ql)
```

ltcurve outputs an ST table containing counts and errors. For timing information to be useful, though, effects such as primbsching and the dead spot need to be taken into account. To do this, use *euv/survey/effexp*.

III. TIME-SORT THE QPOE FILE

The QPOE file must be time-sorted to be used by the light curve task. Run *xray/xtiming/timsort* using the following format to define your source and background apertures:

```
PACKAGE = xtiming
   TASK = timsort

image   =            ds.qp  input qpoe file name
soutfile=           source  Output time-ordered file name
sregion =    c 1024 1024 50  source region descriptor
boutfile=       background  Output time-ordered bkgd file name
bregion = a 1024 1024 70 100  background region descriptor
(exp    =            NONE) exposure mask
(ethresh=            100.) min. percent of exp time for inclusion
(esize  =          region) QPOE event size
(clobber=              no) OK to delete existing output file?
(display=               1) 0=no disp, 1=header
(ssize  =         1000000) max number of events per sort
(qpi    =             yes) prompt for qp internals?
(psize  =            1024) page size for qpoe file
(bsize  =            2048) bucket length of qpoe file
(debug  =               0) qpoe debug level
(mode   =              ql)
```

The format for the source region is ‘type x-center y-center radius’ and for the background regions ‘type x-center y-center radius 1 radius 2’. So ‘c 1024 1024 50’ describes a ‘c’ircle centered at (1024,1024), with a radius of 50 pixels. ‘a 1024 1024 70 100’ describes an ‘a’nnulus centered at (1024,1024) with an inner radius of 70 pixels and an outer radius of 100 pixels. See ‘help xray.regions’ for other possibilities.

Put this filter, combined with any other filter(s) you want, into the qpoe file using *euv/euvtools/tfilter*:

```
PACKAGE = euvtools
  TASK = tfilter

input   =      ds_gt,dspot_gt  Input table name
(output =              ds.qp) Output file
(name   =           total_gt) Name of output macro
(start  =              stime) Name of starting column
(end    =              etime) Name of ending column
(cmdline=                 no) Command line output flag
(format =            %12.2f) Interval format control
(mode   =                 al)
```

II. MAKE SURE THE QPOE IS PROS COMPATIBLE

You should have the following header cards in your QPOE file:

XS-EVENT = '**{d:time,s:x,s:y,r:ra,r:dec,s:dx,s:dy}**'
DEFFILT = 'time=(total_gt)'

If these header cards aren't there (the last one probably isn't) or if they are different, use *xray/ximages/qphedit* to change them. Use all lower case letters when using qphedit.

```
PACKAGE = ximages
  TASK = qphedit

qpoe    =              ds.qp  qpoe(s) to be edited
fields  =            deffilt  fields to be edited
value   =     time=(total_gt)  value expression
(add    =                yes) add rather than edit fields
(delete =                 no) delete rather than edit fields
(verify =                yes) verify each edit operation
(show   =                yes) print record of each edit operation
(update =                yes) enable updating of the image header
(mode   =                 ql)
```

MAKING A DEEP SURVEY LIGHT CURVE

I. ELIMINATE DEAD SPOT INTERVALS

You may need to create a time filter that will eliminate all time intervals when the source was in the dead spot. If you don't already have a file called *detpos.tab*, create one using the task *euv/euvtools/detpos*:

```
PACKAGE = euvtools
   TASK = detpos

input   =   observation/table1  Name of input aspect ST table
output  =               detpos  Prefix for output aspect ST tables
(refdata=   reference/detector) Name of ST table for reference data
(detecto=                   ds) Detector list [sca|scb|scc|ds]
(ra     =             12.48159) Source RA in decimal hours
(dec    =             2.052389) Source DEC in decimal degrees
(mode   =                   al)
```

The resulting ST table gives the detector position of the source. The next step is to create a time filter that will filter out any times when the source was in the dead spot using *euv/euvtools/masktimes*:

```
PACKAGE = euvtools
   TASK = masktimes

mask    =            dsdead.pl  pixel list file containing region mask
(detpos =         detposds.tab) Detector position table
output  =             dspot_gt  Output ST table of time intervals
(zerotim=                   no) Return intervals in zero mask regions?
(maskext=                    0) Mask value to assume beyond mask boundaries
(dtimeco=            TableTime) Column in detpos table for times
(dxcol  =                   DX) Column in detpos table for x position
(dycol  =                   DY) Column in detpos table for y position
(sublen =                  0.1) Subinterval length
(verbose=                  yes) Print processing info?
(mode   =                   al)
```